Bayesian Inference for Irreducible Diffusion Processes Using the Pseudo-Marginal Approach

Osnat Stramer *
Department of Statistics and Actuarial Science
University of Iowa, Iowa City, IA

Matthew Bognar †
Department of Statistics and Actuarial Science
University of Iowa, Iowa City, IA

January 25, 2011

Abstract

In this article we examine two relatively new MCMC methods which allow for Bayesian inference in diffusion models. First, the Monte Carlo within Metropolis (MCWM) algorithm (O'Neil, Balding, Becker, Serola and Mollison, 2000) uses an importance sampling approximation for the likelihood and yields a Markov chain. Our simulation study shows that there exists a limiting stationary distribution that can be made arbitrarily "close" to the posterior distribution (MCWM is *not* a standard Metropolis-Hastings algorithm, however). The second method, described in Beaumont (2003) and generalized in Andrieu and Roberts (2009), introduces auxiliary variables and utilizes a standard Metropolis-Hastings algorithm on the enlarged space; this method preserves the original posterior distribution. When applied to diffusion models, this *pseudo-marginal* (PM) approach can be viewed as a generalization of the popular data augmentation schemes that sample jointly from the missing paths and the parameters of the diffusion volatility. The efficacy of the PM approach is demonstrated in a simulation study of the Cox-Ingersoll-Ross (CIR) and Heston models, and is applied to two well known datasets. Comparisons are made with the MCWM algorithm and the Golightly and Wilkinson (2008) approach.

Key Words: Diffusion process, Euler discretization, Markov chain Monte Carlo (MCMC), Pseudo-Marginal (PM) Algorithm, Grouped Independence Metropolis-Hastings (GIMH), Monte Carlo within Metropolis (MCWM)

*osnat-stramer@uiowa.edu
†matthew-bognar@uiowa.edu

1 Introduction

A diffusion process is described as a solution to the stochastic differential equation (SDE)

$$\mathrm{d}X_t = \mu(X_t, \theta)\,\mathrm{d}t + \sigma(X_t, \theta)\,\mathrm{d}W_t, \quad 0 \le t \le \mathcal{T}, \tag{1}$$

where X_t takes values in $\Re^d$, μ and $\nu = \sigma\sigma^T$ are drift and covariance coefficients of dimension d and $d \times d$ respectively, θ is the parameter vector, and W_t is a d-dimensional Brownian motion. As in Milstein, Schoenmakers and Spokoiny (2004), we assume the drift μ and covariance ν are bounded and are infinitely differentiable with continuous and bounded derivatives of all order, and $\sigma(\cdot)$ is invertible with bounded inverse. This implies existence and uniqueness of (1), and smoothness of the transition density. For ease of notation we assume that X is time homogeneous.

We wish to perform Bayesian inference for the parameters of a continuous-time Markov process X which is observed (possibly with noise) at discrete time points $t_i = i\Delta$ $(i = 0, \ldots, n)$ yielding observations $\mathbf{x} = (x_0, \ldots, x_n)$. We denote the transition (or conditional) density of $X_{t+\Delta} = y$ given $X_t = x$ by $p(\Delta, y|x, \theta)$. By the Markov property, if all components of X at time t_i $(i = 0, \ldots, n)$ are observed without noise, the likelihood function is

$$L(\mathbf{x}|\theta) = \prod_{i=0}^{n-1} p(\Delta, x_{i+1}|x_i, \theta)$$

and the posterior distribution is given by

$$\pi(\theta|\mathbf{x}) \propto \pi(\theta) L(\mathbf{x}|\theta),$$

where $\pi(\theta)$ is the prior distribution on θ.

It is well known that if the data are recorded at discrete times, parametric inference for diffusions using the likelihood of the data is difficult. This is primarily because the corresponding likelihood function is not available in closed form. See Sørensen (2004) for a review of inferential methods for diffusions. We focus on Bayesian inference in this paper.

Many methods have been proposed and studied for diffusions which can be transformed to have unit diffusion coefficient (the so-called *reducible* diffusions). Methods that rely on introducing missing (latent) data are in Elerian, Chib and Shephard (2001); Roberts and Stramer (2001); Eraker (2001). It has been well documented that naïve data augmentation techniques lead to problems of high dependency between the covariance parameters and the diffusion paths (see Elerian et al., 2001; Roberts and Stramer, 2001). The dependency problems can largely be solved by an appropriate re-parametrization for reducible diffusions (see Roberts and Stramer, 2001; Kalogeropoulos, 2007). A different approach for reducible diffusions, built upon exact simulation, has been developed in Beskos, Papaspiliopoulos, Roberts and Fearnhead (2006); Beskos, Papaspiliopoulos and Roberts

(2009).

Inference for irreducible diffusions is much harder. This class of models, which includes most interesting multi-dimensional diffusion models, is not thoroughly covered or understood in the literature. One approach for Bayesian inference is to use a standard Metropolis-Hastings (MH) algorithm with an approximation for the likelihood. One such method, described in Stramer, Bognar and Schneider (2010), is to use the analytical closed-form (CF) likelihood approximations of Aït-Sahalia (2002, 2008) to approximate the likelihood. Their method also addresses the problem that the CF likelihood approximation does not integrate to 1 when far in the tails. This method requires that the time interval Δ is "small".

Methods that rely on introducing missing (latent) data are challenging for irreducible diffusions because there is not an obvious re-parametrization to break down the dependency between the covariance parameters and the diffusion paths. We explore the possibility of augmenting without re-parametrization techniques using the closed-form (CF) analytical log-likelihood approximations derived in Aït-Sahalia (2002, 2008). Methods that rely on re-parametrization techniques are defined in Kalogeropoulos, Roberts and Dellaportas (2010); Golightly and Wilkinson (2008). The Kalogeropoulos et al. (2010) approach, defined through time change transformations, is very efficient, although strong conditions on the covariance coefficient are required. The Golightly and Wilkinson (2008) (GW) approach follows Chib, Pitt and Shephard (2006) and provides another possible transformation to overcome the dependency structure. While the promising GW approach can be applied to a large class of diffusions, it is not yet rigorously justified in the literature.

The need for additional efficient algorithms for irreducible diffusions is apparent. In this article we apply two general Bayesian algorithms to irreducible diffusions. One technique, defined in O'Neil et al. (2000), is the Monte Carlo within Metropolis (MCWM) algorithm. Because MCWM replaces the likelihood with a simulation-based approximation, MCWM is *not* a standard MH algorithm and therefore all of the well known properties of MH samplers do not apply. MCWM is discussed and studied in Beaumont (2003) and Andrieu and Roberts (2009). We discuss the application of the MCWM algorithm to diffusion models in Section 3.

Another algorithm, introduced in Andrieu and Roberts (2009), is called the *pseudo-marginal* (PM) approach. The PM algorithm is a generalization of the *Grouped Independence Metropolis-Hastings* (GIMH) algorithm introduced by Beaumont (2003). In short, suppose we wish to sample from the density function $p(\theta)$, but this is not possible because $p(\theta)$ is intractable. Suppose, however, that we *can* sample from the joint distribution $(\theta, \mathbf{u}_1, \ldots, \mathbf{u}_N)$ where $\mathbf{u}_1, \ldots, \mathbf{u}_N$ are N independent auxiliary variables. Using the samples from $(\theta, \mathbf{u}_1, \ldots, \mathbf{u}_N)$, we can simply marginalize to obtain samples from $p(\theta)$. In this article, we apply this general approach to diffusion models (see Section 4); in particular we discuss different updating schemes of the parameters. The PM approach is a generalization of *jointly* updating the parameters and missing data in a Metropolis-Hastings (MH) algorithm (Golightly and Wilkinson, 2006). It overcomes the problem of low acceptance rate of the

latter.

The remainder of this paper is organized as follows. Section 2 discusses data augmentation techniques Sections 3, 4, and 5 detail the MCWM and PM algorithms, while Section 6 applies these techniques to general stochastic volatility models. Section 7 provides a detailed simulation study for the Cox Ingersoll Ross (CIR) and Heston models. Section 8 uses these aforementioned models to analyze two real-world datasets, and the competing Golightly and Wilkinson (2008) algorithm is briefly compared to the PM and MCWM approaches. Section 9 contains concluding remarks.

2 Data Augmentation

One common approach in the literature for Bayesian estimation of diffusion models, studied independently by Jones (1999), Eraker (2001), and Elerian et al. (2001), is to consider estimating diffusion models on the basis of discrete measurements as a classic missing-data problem. The idea is to introduce augmented data points between every two consecutive (observed) data points so that the likelihood can be well approximated. The time-step interval $[0, \Delta]$ is partitioned into M sub-intervals via grid points $0 = \tau_0 < \tau_1 < \ldots < \tau_M = \Delta$ (each sub-interval has length $h = \Delta/M$) such that the resulting partition is sufficiently fine for some discrete approximation of the diffusion X to be sufficiently accurate. The unobserved data points of the process X are treated as missing data. The resulting posterior distribution is

$$\pi^{\text{miss}}_{(M)}(\theta, \mathbf{u}|\mathbf{x}) \propto \pi(\theta) \prod_{i=0}^{n-1} \prod_{m=0}^{M-1} p_{(a)}(h, u_{i,m+1}|u_{i,m}, \theta), \tag{2}$$

with $u_{i,0} = x_i$, $u_{i,M} = x_{i+1}$, $\mathbf{u} = (\mathbf{u}_0, \ldots, \mathbf{u}_{n-1})$, where $\mathbf{u}_i = (u_{i,1}, \ldots, u_{i,M-1})$ is a discrete-time skeleton of (1) between x_i and x_{i+1}, and $p_{(a)}(h, u_{i,m+1}|u_{i,m}, \theta)$ is some approximation of the transition density $p(h, u_{i,m+1}|u_{i,m}, \theta)$. One such approximation is the Euler approximation:

$$p_{(a)}(h, u_{i,m+1}|u_{i,m}, \theta) = \phi\left(u_{i,m+1};\ u_{i,m} + h\mu(u_{i,m}, \theta),\ h\nu(u_{i,m}, \theta)\right)$$

where $\phi(\cdot; \mu, \nu)$ denotes the normal density with mean μ and covariance ν.

The resulting algorithm proceeds by alternating between simulation of θ conditional on the augmented data $\mathbf{u}$, and simulation of the missing data blocks conditional on θ. Updating the augmented data $\mathbf{u}_i$ between x_i and x_{i+1}, $i = 0, \ldots, n-1$, requires generating the intermediate points according to their conditional distribution given the end points. A series of Metropolis within Gibbs steps is commonly used. Various sampling strategies for the missing data have been proposed; see Golightly and Wilkinson (2008) for a review. A commonly used proposal for a missing data block $\mathbf{u}_i$ is the so called *Modified Brownian Bridge* (MBB) sampler defined in Durham and

Gallant (2002) as

$$U_{i,m+1} = u_{i,m} + \frac{x_{i+1} - u_{i,m}}{M-m} + \sqrt{h\,\frac{M-m-1}{M-m}}\;\sigma(u_{i,m},\theta)Z_{m+1} \quad m = 0,\ldots,M-2 \qquad (3)$$

where $U_{i,m} = u_{i,m}$, $u_{i,0} = x_i$, $u_{i,M} = x_{i+1}$, and $\{Z_1,\ldots,Z_{M-1}\}$ are i.i.d. standard multivariate normal variables. The MBB sampler has the desirable property that the conditional mean of $U_{i,m+1}|U_{i,m} = u_{i,m}$ is a linear interpolation, over the time interval $[mh, Mh = \Delta]$ of $u_{i,m}$ and the final state x_{i+1} $(= u_{i,M})$ at time Δ. It also has the advantage that the conditional covariance is a linear interpolation of the covariance at time mh and the covariance (zero) at time $Mh = \Delta$.

The problem with this approach is that there exists a perfect correlation between the augmented data points and the covariance ν as $h \to 0$. This was noted in a simulation study in Elerian et al. (2001) and was justified theoretically in Roberts and Stramer (2001). The reason for this is the property of diffusions that relates ν with the quadratic variation of the process,

$$\lim_{h\to 0}\sum_{m=0}^{M-1}(X_{(m+1)h} - X_{mh})(X_{(m+1)h} - X_{mh})^T = \int_0^\Delta \nu(X_s,\theta)\,ds \quad \text{a.s.}$$

This translates into reducibility when $h \to 0$. Therefore, while data augmentation schemes can be satisfactory for small M, they can break down as M increases. The problem may be solved if we apply a transformation so that the algorithm based on the transformed diffusion is no longer reducible as $h \to 0$.

As a side project of this article, we extensively experimented with augmentation using the closed-form (CF) analytical log-likelihood approximations derived in Aït-Sahalia (2002, 2008). The posterior distribution is

$$\pi^{\text{miss}}_{(M)}(\theta,\mathbf{u}|\mathbf{x}) \propto \pi(\theta)\prod_{i=0}^{n-1}\prod_{m=0}^{M-1} p^{(K)}_{CF}(h,u_{i,m+1}|u_{i,m},\theta),$$

where $p^{(K)}_{CF}(h,u_{i,m+1}|u_{i,m},\theta)$ denotes the K^{th} order closed-form approximation of sub-density $p(h,u_{i,m+1}|u_{i,m},\theta)$. The main idea behind data augmentation is to choose M sufficiently big so that we can accurately approximate the transition density over time intervals of length $h = \Delta/M$. Choosing a "better" approximation (than the Euler approximation) for the transition density may allow for fewer augmented points and thus reduce the dependency between the diffusion function and augmented data.

We therefore employed the closed-form approximation with the MBB sampler as the proposal distribution for the missing data. Our simulation study showed that, in general, the transition density can be accurately estimated with a larger time step h using the CF approximation than can be used with the Euler approximation. However, the CF approximation is a local approximation

and the error increases as θ moves away from the MLE (it can explode to infinity in the tails of the posterior). Although the error may be small in absolute terms, this error tends to propagate as the amount of data augmentation increases (i.e. as the number of CF sub-densities increases). In a simulation study for the CIR model defined in (14), we found that the sampler would frequently become stuck when in the tails of the posterior distribution; less data augmentation (smaller M) tended to minimize the probability of the sampler becoming stuck since there was less propagation of error. If accurate estimates of the transition density can be obtained using the CF approximation with small M (say $M \leq 5$), then this scheme may be practical. In general, however, using the CF approximation for the sub-densities simply can not be robustly applied to a wide range of models.

Another way to overcome the dependency structure is to update all parameters and missing data simultaneously. However, this will typically result in a very low acceptance rate due to the high dimensionality of the update. In fact, this is a special case of the PM algorithm described in Section 4.

3 Monte Carlo within Metropolis (MCWM)

We use the following notation throughout the paper:

- $\boldsymbol{u} = (\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{n-1})$ denotes the entire collection of samples, where
- $\boldsymbol{u}_i = (\boldsymbol{u}_{i,1}, \ldots, \boldsymbol{u}_{i,N})$ denotes the collection of samples within the ith block, $i = 0, \ldots, n-1$, and
- $\boldsymbol{u}_{i,k} = (u_{i,k,1}, \ldots, u_{i,k,M-1})$ denotes the kth sample (path) within the ith block where $u_{i,k,0} = x_i$, $u_{i,k,M} = x_{i+1}$, $k = 1, \ldots, N$, and $i = 0, \ldots, n-1$.

Monte Carlo within Metropolis (MCWM) is based on importance sampling estimators for the transition density. The transition density $p(\Delta, x_{i+1}|x_i, \theta)$ is approximated by

$$p_{(M)}(\Delta, x_{i+1}|x_i, \theta) = \int \prod_{m=0}^{M-1} \phi\left(u_{m+1};\ u_m + h\mu(u_m, \theta),\ h\nu(u_m, \theta)\right) du_1, \ldots, u_{M-1} \tag{4}$$

where $h = \Delta/M$, $u_0 = x_i$, $u_M = x_{i+1}$, and M is "big" enough so that the resulting partition is sufficiently fine for the Euler approximation to be sufficiently accurate.

The integral in (4) is evaluated in Durham and Gallant (2002) using importance sampling:

$$p_{(M)}(\Delta, x_{i+1}|x_i, \theta) = E_q[R_M(\mathbf{U})] \tag{5}$$

where $\mathbf{U} = (U_1, \ldots, U_{M-1})$,

$$R_M(\mathbf{U}) = \frac{\prod_{m=0}^{M-1} \phi\left(U_{i,m+1};\, U_{i,m} + h\mu(U_{i,m}, \theta),\, h\nu(U_{i,m}, \theta)\right)}{q(\mathbf{U})}$$

$q(\cdot)$ is a density function on $\Re^{d\times(M-1)}$ referred to as the importance sampler (or importance sampling density), and E_q is the expectation with respect to density q. Thus, $\mathbf{U}$ is generated according to q and is weighted by $R_M(\mathbf{U})$. The posterior distribution is therefore estimated as

$$\pi_{(M)}(\theta|\mathbf{x}) \propto \pi(\theta) \prod_{i=0}^{n-1} p_{(M)}(\Delta, x_{i+1}|x_i, \theta). \tag{6}$$

The expectation in (5) cannot be evaluated, but it can be estimated by drawing N independent paths using the importance sampler $q(\cdot)$, evaluating the ratio R_M for each path, and determining the (sample) mean of R_M. We denote this estimator by $p_{(M,N)}(\Delta, y|x, \theta)$ where

$$p_{(M,N)}(\Delta, x_{i+1}|x_i, \theta) = \frac{1}{N} \sum_{k=1}^{N} R_M(\mathbf{U}_k) \tag{7}$$

and $\mathbf{U}_k = (U_{k,1}, \ldots, U_{k,M-1})$ is a random sample from q, $k = 1, \ldots, N$. Note that $p_{(M,N)}(\Delta, x_{i+1}|x_i, \theta) \overset{a.s.}{\to} p_{(M)}(\Delta, x_{i+1}|x_i, \theta)$ as $N \to \infty$.

The posterior distribution is thus stochastically (vs analytically) approximated as

$$\pi_{(M,N)}(\theta|\mathbf{x}) \propto \pi(\theta) \prod_{i=0}^{n-1} p_{(M,N)}(\Delta, x_{i+1}|x_i, \theta)$$

where $p_{(M,N)}$ is defined in (7). Note that nN MBB samplers are needed to evaluate $\pi_{(M,N)}(\theta|\mathbf{x})$ for each θ. Therefore, there is no target distribution and standard MH algorithms cannot be automatically used.

Yet, technically the MCWM algorithm follows the same steps as a standard MH algorithm with "target distribution" $\pi_{(M,N)}(\theta|\mathbf{x})$. To avoid confusion, we describe one iteration of the MCWM algorithm.

Algorithm 1. *MCWM Algorithm*

1. *Given the current state of the chain θ, for each block i, $i = 0, \ldots, n-1$, do the following: Draw $\boldsymbol{u}_{i,k}$, a random sample (path) between x_i and x_{i+1} from the MBB sampler* (3)*, for $k = 1, \ldots, N$. This MBB sampler is independent of the MBB samplers from previous iterations. Calculate the importance sampling based approximation $p_{(M,N)}(\Delta, x_{i+1}|x_i, \theta)$ of $p_{(M)}(\Delta, x_{i+1}|x_i, \theta)$ as described in* (7)*.*

2. *Propose a new value θ^* from some proposal density $q(\theta, \cdot)$. Given θ^*, repeat step 1 to obtain the*

importance samples $\boldsymbol{u}^*_{i,1}, \ldots, \boldsymbol{u}^*_{i,N}$ *in each block* i, $i = 0, \ldots, n-1$ *(again, this is independent of the MBB samplers from previous iterations). Using these new importance samples* $\mathbf{u}^*$*, compute* $p_{(M,N)}(\Delta, x_{i+1}|x_i, \theta^*)$.

3. *Accept* θ^* *with probability*

$$\alpha_{(M,N)}(\theta, \theta^*) = \min \left[\frac{\pi(\theta^*) \prod_{i=0}^{n-1} p_{(M,N)}(\Delta, x_{i+1}|x_i, \theta^*)}{\pi(\theta) \prod_{i=0}^{n-1} p_{(M,N)}(\Delta, x_{i+1}|x_i, \theta)} \frac{q(\theta^*, \theta)}{q(\theta, \theta^*)}, \ 1 \right]$$

As in Andrieu and Roberts (2009), we note that due to the fact that the MBB's are *independent at each iteration*, it can be easily checked that the MCWM algorithm generates a Markov chain. However, $\pi_{(M)}(\theta|\mathbf{x})$, defined in (6), is *not* the invariant distribution for the chain. Since MCWM is not a standard MH algorithm, the existence of an invariant distribution for each N needs to be assumed (we denote it by $\tilde{\pi}_{(M,N)}$ for each $N \in \mathbb{N}^+$). This is not obvious, however. Convergence of $\tilde{\pi}_{(M,N)}$ to $\pi_{(M)}$ needs to be explored.

4 The Pseudo-Marginal Approach

Following the pseudo-marginal algorithm introduced in Andrieu and Roberts (2009), we define a target density on $\Theta \times \Re^{nN(M-1)}$ as follows:

$$\pi^{\text{New Target}}_{(M,N)}(\theta, \mathbf{u}|\mathbf{x}) \propto \pi(\theta) \prod_{i=0}^{n-1} p_{(M,N)}(\Delta, x_{i+1}|x_i, \theta) \prod_{i=0}^{n-1} \prod_{k=1}^{N} q(u_{i,k,1}, \ldots, u_{i,k,M-1}|\theta) \tag{8}$$

where $q(\cdot|\theta)$ is the MBB sampler defined in (3). Note that (2) is a special case of (8) with $N = 1$. Also, in contrast to $\pi_{(M)}(\theta|\mathbf{x})$ defined in (6), $\pi^{\text{New Target}}_{(M,N)}(\theta, \mathbf{u}|\mathbf{x})$ can be explicitly evaluated (up to a constant of proportionality). It is easy to check that $\pi^{\text{New Target}}_{(M,N)}(\theta, \mathbf{u}|\mathbf{x})$ is a probability density function on $\Re^{nN(M-1)}$ with marginal distribution $\pi_{(M)}(\theta|\mathbf{x})$ for all $N \in \mathbb{N}$.

As often is the case, simulating a chain $\{(\theta^t, \mathbf{u}^t)\}_{t=0}^{\infty}$ with stationary density $\pi^{\text{New Target}}_{(M,N)}(\theta, \mathbf{u}|\mathbf{x})$ can be done in many different ways using MCMC algorithms. Assume that (1) can be written as

$$\mathrm{d}X_t = \mu(X_t, \theta_1)\,\mathrm{d}t + \sigma(X_t, \theta_2)\,\mathrm{d}W_t, \quad 0 \le t \le \mathcal{T}.$$

To overcome the dependency structure between θ_2 and $\mathbf{u}$, we propose alternating between updating $(\theta_2, \mathbf{u})$ and θ_1. We term this the *Pseudo-Marginal* (PM) algorithm as this is a special case of the general PM algorithm. Thus, under some regularity conditions that guarantee irreducibility and aperiodicity, $\pi_{(M)}(\theta|\mathbf{x})$ is the marginal ergodic density for the PM algorithms. This is true for all $N \in \mathbb{N}$. The acceptance rate of the PM algorithm as a function of N is rigorously analyzed in Andrieu and Roberts (2009). We expect the acceptance rate to increase in N. For our applications

this is supported by noting that for large N,

$$\prod_{i=0}^{n-1} p_{(M,N)}(\Delta, x_{i+1}|x_i, \theta) \approx \prod_{i=0}^{n-1} p_{(M)}(\Delta, x_{i+1}|x_i, \theta) \tag{9}$$

which implies that $\prod_{i=0}^{n-1} p_{(M,N)}(\Delta, x_{i+1}|x_i, \theta)$ is almost independent of the latent component $\mathbf{u}$:

$$\begin{aligned}
\pi_{(M,N)}^{\text{New Target}}(\mathbf{u}|\theta, \mathbf{x}) &\propto \pi(\theta) \prod_{i=0}^{n-1} p_{(M,N)}(\Delta, x_{i+1}|x_i, \theta) \prod_{i=0}^{n-1} \prod_{k=1}^{N} q(u_{i,k,1}, \ldots, u_{i,k,M-1}|\theta) \\
&\approx \pi(\theta) \prod_{i=0}^{n-1} p_{(M)}(\Delta, x_{i+1}|x_i, \theta) \prod_{i=0}^{n-1} \prod_{k=1}^{N} q(u_{i,k,1}, \ldots, u_{i,k,M-1}|\theta) \\
&\propto \prod_{i=0}^{n-1} \prod_{k=1}^{N} q(u_{i,k,1}, \ldots, u_{i,k,M-1}|\theta).
\end{aligned}$$

Therefore, a small N may lead to low acceptance rates due to the discrepancy between the proposed distribution of the latent component $\prod_{i=0}^{n-1} \prod_{k=1}^{N} q(u_{i,k,1}, \ldots, u_{i,k,M-1}|\theta)$ and the true conditional distribution of the latent component $\pi_{(M,N)}^{\text{New Target}}(\mathbf{u}|\theta)$, while a "big" N will eliminate this problem. We now describe one iteration of the PM algorithm.

Algorithm 2. *PM Algorithm*

1. *For ease of notation, let* $(\theta_1^t, \theta_2^t, \mathbf{u}^t) = (\theta_1, \theta_2, \mathbf{u})$. *Propose a new value* $(\theta_2^*, \mathbf{u}^*)$ *for* $(\theta_2^{t+1}, \mathbf{u}^{t+1})$ *from the proposal density*

$$q_2\left((\theta_2, \mathbf{u}), (\theta_2^*, \mathbf{u}^*)\right) = \tilde{q}_2(\theta_2, \theta_2^*) \prod_{i=0}^{n-1} \prod_{k=1}^{N} q(u_{i,k,1}^*, \ldots, u_{i,k,M-1}^*|\theta_2^*)$$

   where $\tilde{q}_2(\theta_2, \cdot)$ *is some proposal density and* $q(\cdot|\theta_2^*)$ *is the MBB sampler* (3) *with covariance function* $\nu(\cdot, \theta_1, \theta_2^*)$. *Note that unlike the MCWM algorithm, we do not generate a "fresh" set of* $\mathbf{u}$ *values;* $\mathbf{u}$ *is simply "dragged" from the previous iteration. Only* $\mathbf{u}^*$ *needs to be generated.*

2. *Accept* $(\theta_2^*, \mathbf{u}^*)$ *(i.e. set* $\theta_2^{t+1} = \theta_2^*$ *and* $\mathbf{u}^{t+1} = \mathbf{u}^*$*) with probability* $\alpha_{(M,N)}((\theta_2, \mathbf{u}), (\theta_2^*, \mathbf{u}^*)) =$

$\min\left[r_{(M,N)}((\theta_2,\mathbf{u}),(\theta_2^*,\mathbf{u}^*)),\ 1\right]$ *where*

$$
\begin{aligned}
& r_{(M,N)}((\theta_2,\mathbf{u}),(\theta_2^*,\mathbf{u}^*)) \\
&= \frac{\pi_{(M,N)}^{\text{New Target}}(\theta_1,\theta_2^*,\mathbf{u}^*|\mathbf{x})}{\pi_{(M,N)}^{\text{New Target}}(\theta_1,\theta_2,\mathbf{u}|\mathbf{x})}\,\frac{\prod_{i=0}^{n-1}\prod_{k=1}^{N} q(u_{i,k,1},\ldots,u_{i,k,M-1}|\theta_2)}{\prod_{i=0}^{n-1}\prod_{k=1}^{N} q(u^*_{i,k,1},\ldots,u^*_{i,k,M-1}|\theta_2^*)}\,\frac{\tilde{q}_2(\theta_2^*,\theta_2)}{\tilde{q}_2(\theta_2,\theta_2^*)} \\
&= \frac{\pi(\theta_1,\theta_2^*)\prod_{i=0}^{n-1} p_{(M,N)}(\Delta,x_{i+1}|x_i,\theta_1,\theta_2^*)\prod_{i=0}^{n-1}\prod_{k=1}^{N} q(u_{i,k,1},\ldots,u_{i,k,M-1}|\theta_2^*)}{\pi(\theta_1,\theta_2)\prod_{i=0}^{n-1} p_{(M,N)}(\Delta,x_{i+1}|x_i,\theta_1,\theta_2)\prod_{i=0}^{n-1}\prod_{k=1}^{N} q(u_{i,k,1},\ldots,u_{i,k,M-1}|\theta_2)} \\
&\quad \times\frac{\prod_{i=0}^{n-1}\prod_{k=1}^{N} q(u_{i,k,1},\ldots,u_{i,k,M-1}|\theta_2)}{\prod_{i=0}^{n-1}\prod_{k=1}^{N} q(u^*_{i,k,1},\ldots,u^*_{i,k,M-1}|\theta_2^*)}\,\frac{\tilde{q}_2(\theta_2^*,\theta_2)}{\tilde{q}_2(\theta_2,\theta_2^*)} \\
&= \frac{\pi(\theta_1,\theta_2^*)\prod_{i=0}^{n-1} p_{(M,N)}(\Delta,x_{i+1}|x_i,\theta_1,\theta_2^*)}{\pi(\theta_1,\theta_2)\prod_{i=0}^{n-1} p_{(M,N)}(\Delta,x_{i+1}|x_i,\theta_1,\theta_2)}\,\frac{\tilde{q}_2(\theta_2^*,\theta_2)}{\tilde{q}_2(\theta_2,\theta_2^*)},
\end{aligned}
$$

and $p_{(M,N)}(\Delta,x_{i+1}|x_i,\theta_1,\theta_2^*)$, $p_{(M,N)}(\Delta,x_{i+1}|x_i,\theta_1,\theta_2)$ *are defined in* (7) *with* $\mathbf{u}^*$ *and* $\mathbf{u}$ *respectively. Otherwise set* $\theta_2^{t+1}=\theta_2$ *and* $\mathbf{u}^{t+1}=\mathbf{u}$.

3. *Propose a new value* θ_1^* *for* θ_1^{t+1} *from some proposal density* $\tilde{q}_1(\theta_1,\theta_1^*)$.

4. *Accept* θ_1^* *(i.e. set* $\theta_1^{t+1}=\theta_1^*$ *) with probability* $\alpha_{(M,N)}(\theta_1,\theta_1^*)=\min\left[r_{(M,N)}(\theta_1,\theta_1^*),\ 1\right]$ *where*

$$
\begin{aligned}
& r_{(M,N)}(\theta_1,\theta_1^*) \\
&= \frac{\pi_{(M,N)}^{\text{New Target}}(\theta_1^*,\theta_2^{t+1},\mathbf{u}^{t+1}|\mathbf{x})}{\pi_{(M,N)}^{\text{New Target}}(\theta_1,\theta_2^{t+1},\mathbf{u}^{t+1}|\mathbf{x})}\,\frac{\tilde{q}_1(\theta_1^*,\theta_1)}{\tilde{q}_1(\theta_1,\theta_1^*)} \\
&= \frac{\pi(\theta_1^*,\theta_2^{t+1})\prod_{i=0}^{n-1} p_{(M,N)}(\Delta,x_{i+1}|x_i,\theta_1^*,\theta_2^{t+1})}{\pi(\theta_1,\theta_2^{t+1})\prod_{i=0}^{n-1} p_{(M,N)}(\Delta,x_{i+1}|x_i,\theta_1,\theta_2^{t+1})}\,\frac{\tilde{q}_1(\theta_1^*,\theta_1)}{\tilde{q}_1(\theta_1,\theta_1^*)}
\end{aligned}
$$

It is tempting to split up the latent process into blocks $\mathbf{u_i}$ and alternate between updating $\theta=(\theta_1,\theta_2)$ and $\mathbf{u_i}$, $i=0,\ldots,n-1$. However, using this blocking strategy will lead to inefficient algorithms. For $N=1$ this is exactly the naïve data augmentation algorithm discussed in Section 2 with the MBB used as the proposal density for the missing data. As was mentioned before, this algorithm suffers from high dependency between θ_2 and the missing data $\mathbf{u}$. Increasing N does not help either as it decreases the acceptance rate of θ. This is because, as N increases, $p_{(M,N)}(\Delta,x_{i+1}|x_i,\theta)\approx p_{(M)}(\Delta,x_{i+1}|x_i,\theta)$ and therefore the acceptance rate of the auxiliary variables $\mathbf{u}_0,\ldots,\mathbf{u}_{n-1}$ will be very high. However, acceptance rate for $\theta_2|\mathbf{u}$ will be very small when N is big; if $\mathbf{u}_{i,k}$ are the MBB proposals generated with parameter θ, then $\pi_{(M,N)}^{\text{New Target}}(\theta^*,\mathbf{u}|\mathbf{x})$ will be significantly smaller than $\pi_{(M,N)}^{\text{New Target}}(\theta,\mathbf{u}|\mathbf{x})$. The ineffectiveness of this blocking technique is demonstrated at the end of Section 7.1.

5 Choosing M and N

The transition density $p(\Delta, y|x, \theta)$ is approximated by the transition density of the Euler approximation $p_{(M)}(\Delta, y|x, \theta)$ with time step $h = \Delta/M$. From Bally and Talay (1996), the error due to discretization is of order $1/M$ and can be reduced by choosing a small time step h. Choice of M will be important since it must be sufficiently large for the likelihood to be accurately approximated. Convergence of the marginal posterior densities may be used as an overall diagnostic that M is sufficiently big. The idea is to choose M equal to the value M_0 such that the estimated marginal posterior densities are approximately the same for $M \geq M_0$.

We next consider the question of choosing the number of importance samples N. The PM and MCWM algorithms have differing optimal values of N. The choice of N for a special case of the pseudo-marginal (PM) algorithm is introduced in Andrieu, Berthelsen, Doucet and Roberts (2010) and is applied to our PM algorithms. The speed of convergence (and rapidity of mixing after convergence) depends heavily on N. Following Pasarica and Gelman (2010) and Andrieu et al. (2010), N can be optimized to maximize the *expected squared jump distance* (ESJD) defined as

$$\text{ESJD} = \sum_{t=0}^{T-1} \alpha_{(M,N)} \left[(\theta^t, \mathbf{u}^t), (\theta^*, \mathbf{u}^*)\right] \|\theta^* - \theta^t\|^2$$

where T is the number of Monte-Carlo iterations, θ^* and $\mathbf{u}^*$ are the proposals for θ^{t+1} and $\mathbf{u}^{t+1}$ respectively. As expected, our simulation study (see Section 7) shows that ESJD increases in N. In other words, when N is relatively small, the ESJD is low, which suggests slow mixing or convergence rate. Thus, more iterations are needed to obtain any given degree of accuracy in posterior inferences. On the other hand, the algorithm has a shorter execution time when N is small. Increasing N causes the execution time to increase, but the sampler will mix more quickly. From an efficiency standpoint, appropriate tuning of N is required to optimize this MCMC efficiency trade-off. Similar to Andrieu et al. (2010), we seek an N that maximizes ESJD/N. Other ways of balancing mixing rate with computational cost are possible and require more study. An adaptive approach for updating N is in Andrieu et al. (2010) and can be applied to our PM algorithms. It is not pursued here.

The MCWM algorithm is different. One can get good convergence rates and mixing behavior regardless of N (the reader will witness this in the simulation study in Section 7; specifically in Figures 3 and 6). However, MCWM may yield an inaccurate estimate of $\pi_{(M)}(\theta|\mathbf{x})$ if N isn't large enough. This can be seen in Figure 3 where the estimated marginal posterior density of σ, $\pi_{(M)}(\sigma|\mathbf{x})$, is poor when $N = 1, 2, 5, 10$ (better estimates are obtained when $N = 20$), and in Figure 6 where the estimated marginal posterior densities of σ and ρ remain unsatisfactory even when $N = 20$. The performance of the MCWM algorithm depends on how well the importance sampling estimator approximates the transition density. A qualitative and asymptotic result is in

Stramer and Yan (2007) where $N = M^2$. The heuristic reason is that the error due to the bias of the Euler approximation is $O(1/M)$ and the Monte Carlo error is $O(1/\sqrt{N})$. Thus to match the two different sources of error, we need $N = M^2$. We suggest letting $N = N_0$ where N_0 is the minimum value where the marginal posterior densities remain relatively unchanged for $N > N_0$.

6 Application: Stochastic Volatility Models

The MCWM and PM algorithms can be applied to stochastic volatility (SV) models of the form $[Y_t, V_t]$ where Y_t is the log-price of a stock or the short term interest rate with volatility $\sigma_Y(\cdot)$ which is a function of a latent diffusion V. We assume that $[Y_t, V_t]$ follows,

$$\begin{aligned}
\mathrm{d}Y_t &= \mu_Y(Y_t, V_t, \theta)\,\mathrm{d}t + \rho\sigma_Y(Y_t, V_t, \theta)\,\mathrm{d}W_t + \sqrt{1-\rho^2}\;\sigma_Y(Y_t, V_t, \theta)\,\mathrm{d}B_t \\
\mathrm{d}V_t &= \mu_V(Y_t, V_t, \theta)\,\mathrm{d}t + \sigma_V(Y_t, V_t, \theta)\,\mathrm{d}W_t,
\end{aligned}$$

where B and W are two independent standard Brownian motions, and the instantaneous correlation between $\mathrm{d}Y_t$ and $\mathrm{d}V_t$ is controlled by ρ. We assume the process Y is observed (possibly with noise) at discrete time points $t_i = i\Delta$ $(i = 0, \ldots, n)$ yielding observations $\mathbf{y} = (y_0, \ldots, y_n)$.

The Heston model, proposed in Heston (1993), and its variants are commonly used SV models where the instantaneous variance process V is defined by the CIR model in (14). The Heston model $X_t = [Y_t, V_t]^T$ follows:

$$\mathrm{d}Y_t = (\mu - 0.5V_t)\,\mathrm{d}t + \rho\sqrt{V_t}\,\mathrm{d}W_t + \sqrt{1-\rho^2}\sqrt{V_t}\,\mathrm{d}B_t \tag{10}$$

$$\mathrm{d}V_t = \beta(\alpha - V_t)\,\mathrm{d}t + \sigma\sqrt{V_t}\,\mathrm{d}W_t \tag{11}$$

Option prices being traded assets, we need to endow the time series model (10)-(11) with risk premia for arbitrage-free pricing under the auxiliary pricing measure $\mathbb{Q}$. To keep the simulation study simple we make the assumption of risk premia such that $W_t = W_t^{\mathbb{Q}}$ and $dB_t = dB_t^{\mathbb{Q}} + \frac{r-\mu}{\sqrt{(1-\rho^2)V_t}}dt$ and no adjustments in the variance drift are necessary ($\alpha = \alpha^{\mathbb{Q}}$ and $\beta = \beta^{\mathbb{Q}}$).

Instantaneous stochastic variance is latent, even though a time series of *implied* variance is often available (for example the VIX implied volatility index published by the CBOE). To account for the stochastic nature and mean reversion of index variance, we use the fact that for short-maturity at-the-money options the Black-Scholes formula is approximately linear in volatility. Affinity of the variance $\mathbb{Q}$-drift (which is the same as the drift in (11) because we assume zero risk premia) together with Fubini's theorem enables us to write:

$$\frac{1}{\xi}\,\mathbb{E}_t^{\mathbb{Q}}\left[\int_t^{t+\xi} V_s\,ds\right] = A(\alpha, \beta, \xi) + B(\beta, \xi)\,V_t, \quad \xi > 0 \tag{12}$$

where

$$B(\beta,\xi)=\frac{1-e^{-\xi\beta}}{\xi\beta},\quad A(\alpha,\beta,\xi)=\alpha(1-B(\beta,\xi)).$$

We take average expected variance as a proxy for implied variance IV_t,

$$IV_t\approx\frac{1}{\xi}\,\mathbb{E}_t^{\mathbb{Q}}\left[\int_t^{t+\xi}V_s\,ds\right], \tag{13}$$

and choose $\xi = 22/252$ as in Jones (2003). This approximation has been used in Aït-Sahalia and Kimmel (2007), Johannes, Polson and Stroud (2009), Chernov, Gallant, Ghysels and Tauchen (2003), Eraker (2004), and Jones (2003).

The likelihood function for Heston's model is not known in closed form[1], hence the need for the PM and MCWM algorithms. To apply the PM and MCWM algorithms, denote the transition density of $(Y_{t_{i+1}}, IV_{t_{i+1}}) = (y_{i+1}, iv_{i+1})$ given $(Y_{t_i}, IV_{t_i}) = (y_i, iv_i)$ by $p_{(Y,IV)}(\Delta,(y_{i+1},iv_{i+1})|(y_i,iv_i),\theta)$ and that of $(Y_{t_{i+1}}, V_{t_{i+1}}) = (y_{i+1}, v_{i+1})$ given $(Y_{t_i}, V_{t_i}) = (y_i, v_i)$ by $p_{(Y,V)}(\Delta,(y_{i+1},v_{i+1})|(y_i,v_i),\theta)$. Using simple transformation techniques, from (12) and (13) we have

$$p_{(Y,IV)}(\Delta,(y_{i+1},iv_{i+1})|(y_i,iv_i),\theta)=\frac{p_{(Y,V)}(\Delta,(y_{i+1},v_{i+1})|(y_i,v_i),\theta)}{B(\beta,\xi)}$$

where $v_i=\frac{iv_i-A(\alpha,\beta,\xi)}{B(\beta,\xi)}$. For ease of notation we omit (Y,V) from $p_{(Y,V)}$. The likelihood function is

$$L(\mathbf{y},\mathbf{iv}|\theta)=\prod_{i=0}^{n-1}\frac{p\left(\Delta,(y_{i+1},v_{i+1})|(y_i,v_i),\theta\right)}{B(\beta,\xi)}.$$

The transition density $p\left(\Delta,(y_{i+1},v_{i+1})|(y_i,v_i),\theta\right)$ is not available, but can be estimated as in (7) where $\mathbf{U}_{ik}=(U_{i,k,1},\ldots,U_{i,k,M-1})$, $k=1,\ldots,N$, is a random sample from the two dimensional MBB defined in (3) with starting point $(y_i,v_i)^T$ and end point $(y_{i+1},v_{i+1})^T$. We denote this estimator by $p_{(M,N)}\left(\Delta,(y_{i+1},v_{i+1})|(y_i,v_i),\theta\right)$. Similarly to (8), we can now define

$$\begin{aligned}\pi_{(M,N)}^{\text{New Target}}(\theta,\mathbf{u}|\mathbf{y},\mathbf{iv}) \;\propto\; & \pi(\theta)\prod_{i=0}^{n-1}\frac{p_{(M,N)}\left(\Delta,(y_{i+1},v_{i+1})|(y_i,v_i),\theta\right)}{B(\beta,\xi)}\\ & \times\prod_{i=0}^{n-1}\prod_{k=1}^{N}q(u_{i,k,1},\ldots,u_{i,k,M-1}|\theta)\end{aligned}$$

where $\mathbf{v}=(v_0,\ldots,v_n)$, $v_i=V_{t_i}$, and $q(\cdot|\theta)$ is the MBB sampler defined in (3). The MCWM and PM algorithms now proceed similarly to the algorithms in Sections 3 and 4.

[1]See Lamoureux and Paseka (2005) for an expression of the density of the Heston model using a Fourier inversion of the characteristic function. This reduces the dimensionality of the required integration to a one dimensional integral, the remaining integral is over a modified Bessel function of non integer order.

7 Simulation Study

7.1 CIR Model

The CIR model (Cox, Ingersoll and Ross, 1985) is characterized by the SDE

$$dX_t = \beta(\alpha - X_t)\,\mathrm{d}t + \sigma\sqrt{X_t}\; dW_t \tag{14}$$

where α is the mean reverting level, β is the speed of the process, and σ is the volatility parameter. Since this model has a known transition density, which is a scaled non-central chi-squared distribution, and is frequently used in applications, it provides a convenient means of evaluating the effectiveness of the PM and MCWM algorithms. We compare Bayesian analyses using the exact (non-central chi-square) CIR transition density (and likelihood) in a standard Metropolis-Hastings (MH) sampler, the PM algorithm, and the MCWM algorithm.

For the simulation study, we generate two data sets from the true CIR transition density with $n = 500,\ \alpha = 0.07,\ \beta = 0.15,$ and $\sigma = 0.07$. The commonly analyzed monthly FedFunds dataset (see Section 8) yields parameter estimates close to $\alpha = 0.07,\ \beta = 0.15,$ and $\sigma = 0.07$. We used $\Delta = 1$ (yearly) and $\Delta = 1/12$ (monthly), thus our simulated datasets mimic yearly as well as monthly real-world data. For more discussion of the FedFunds dataset, see Section 8.

We apply the random scan Gibbs sampler which randomly selects a component(s) of the parameter vector θ to update within each iteration. Our sampler randomly selected either a joint $(\alpha, \beta)-$move or a $\sigma-$move; the joint $(\alpha, \beta)-$move was chosen with probability 2/3 and a $\sigma-$move was chosen with probability 1/3. Uniform random walk proposals were used throughout, although more optimal proposals could certainly be envisioned. The prior is defined in (15). All simulation studies focus on σ since convergence and mixing behavior for σ is the most problematic. All algorithms were run for 500,000 iterations including a 100,000 iteration burn-in period (this eliminates the effect of the initial starting point).

We wrote all computer code in the $C{+}{+}$ language. The C-based non-central chi-square functions used by the R software package (`http://www.r-project.org`) were called from within $C{+}{+}$ for the exact–likelihood analyses. It should be noted that the importance samples in block i are generated *independently* from the importance samples in block j, $i \neq j$. This lack of dependency can be exploited if multiple processors are available. Each processor can be given the task of generating the importance samples for some subset of the blocks, decreasing the execution time (a similar strategy can be used when evaluating the likelihood). This *parallelization* was performed using the OpenMP software package (`http://www.openmp.org`). One difficulty, however, is generating the random variates across multiple processors; specifically, seeding the individual processors can be problematic if one is not careful. Recently, much work has been done in this area. One option is to use the SPRNG package (`http://sprng.cs.fsu.edu`), another less complex option is to use the

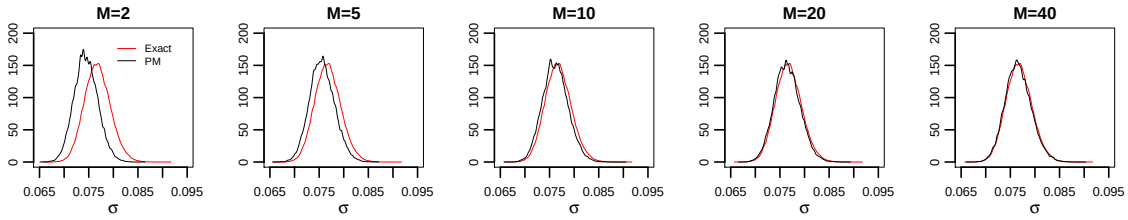

Figure 1: Yearly data: Estimated marginal posterior distribution of σ (in black) for PM with $M = 2, 5, 10, 20, 40$ and $N = 20$. Exact sampler is depicted in red.

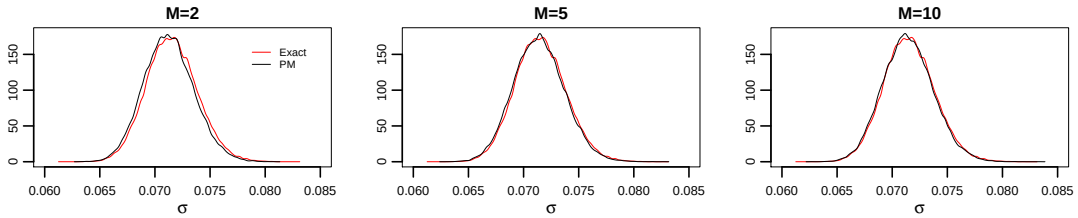

Figure 2: Monthly data: Estimated marginal posterior distribution of σ for PM (black lines) with $M = 2, 5, 10$ and $N = 20$; the exact sampler is depicted in red.

cryptography-based PURG package (`http://bill.cochranpages.com`). Although PURG lacks built-in generators for the common statistical distributions, we simply transformed its uniform random variates into normal random variates (which are extensively used in the MBB) using the Box–Muller method.

First, we chose the number of sub-intervals M by comparing the behavior of the PM algorithm to the exact algorithm with different values of M. The estimated marginal posterior densities $\pi(\sigma|\mathbf{x})$ using the PM algorithm and the exact algorithm are shown in Figure 1 for yearly data and in Figure 2 for monthly data. Clearly, the discretization with $M = 20$ ($M = 5$) can be considered to be sufficiently fine for yearly (monthly) data; we use $M = 20$ ($M = 5$) for the remainder of the analysis of the yearly (monthly) dataset.

We next consider the question of choosing the number of importance samples N for the PM and MCWM algorithms. For yearly data, the left panel of Figure 3 displays the estimated marginal posterior distribution of σ for the PM and MCWM algorithms with $M = 20$ and $N = 1, 2, 5, 10, 20$ (the exact algorithm is also shown). For the monthly data, Figure 4 depicts the estimated marginal posterior densities for PM and MCWM with $M = 5$ and $N = 1, 2, 5$. Clearly, increasing N allows the MCWM algorithm to more accurately approximate the posterior distribution $\pi_{(M)}(\theta|\mathbf{x})$ regardless of the time step Δ. Although MCWM does not have $\pi_{(M)}(\theta|\mathbf{x})$ as its limiting distribution for any finite N, the limiting distribution is quite close to $\pi_{(M)}(\theta|\mathbf{x})$ for sufficiently large N. Note that increasing M will require a subsequent increase in N; one would need $N > 20$ for the yearly data (where $M = 20$), while $N = 5$ or 10 would probably be sufficient for the monthly data (where $M = 5$).

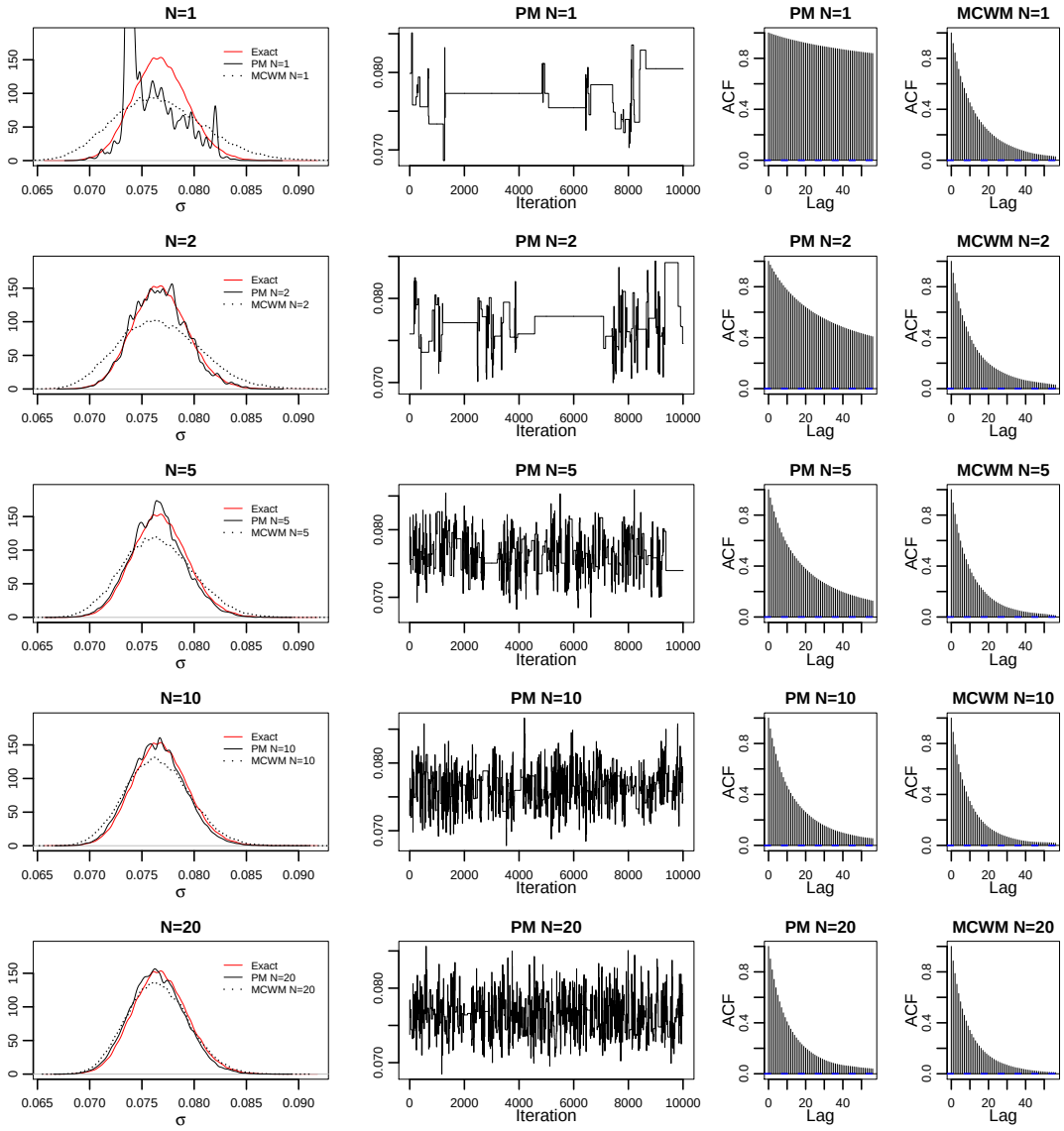

Figure 3: Yearly data: Left panels depict the estimated marginal posterior distribution of σ for PM (solid black line) and MCWM (dotted black line) with $M = 20$ and $N = 1, 2, 5, 10, 20$; the exact sampler is depicted in red. Second column depicts trace plots of the first 10,000 post-burn-in iterations of the PM sampler; right two panels depict the ACF (ACF plots are based upon post-burn-in sampler output only) for the PM and MCWM samplers.

Trace plots for σ using a portion of the yearly PM output is shown in the second column of Figure 3. Trace plots for MCWM are not shown as rapidity of mixing is largely unaffected by N. Autocorrelation function (ACF) plots for the PM and MCWM samplers are depicted in the right two columns. For the PM algorithm, it can be seen that N dramatically influences the rapidity of mixing. Small N deflates the acceptance probability (and mixing rate), increases autocorrelation, and will thus require the samplers to be run longer to obtain any given degree of accuracy in the posterior estimates. Increasing N dramatically improves mixing behavior and usefully decreases

		Exact	$N = 1$	$N = 2$	$N = 5$	$N = 10$	$N = 20$
PM	Yearly	*0.375*	0.020	0.072	0.167	0.233	0.280
	Monthly	*0.355*	0.263	0.300	0.328	0.338	0.344
MCWM	Yearly	*0.375*	0.403	0.392	0.377	0.372	0.370
	Monthly	*0.355*	0.352	0.349	0.349	0.352	0.354

Table 1: Acceptance rates for σ using the PM and MCWM algorithms. Yearly is calculated with $M = 20$ and monthly with $M = 5$. Acceptance rates for the exact algorithm are also included.

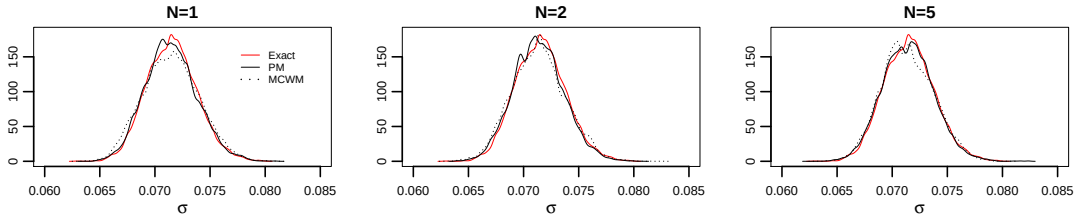

Figure 4: Monthly data: Estimated marginal posterior density plots for PM (solid black line) and MCWM (dotted black line) with $M = 5$ and $N = 1, 2, 5$. The exact sampler is also displayed (red line).

the ACF. We found this to be especially true for yearly data; mixing behavior for the monthly data improved as N increased, but not nearly as dramatically (graph not shown). The MCWM samplers mixed rapidly for both the yearly and monthly data. Although the MCWM chain mixes rapidly when N is small, it clearly does *not* converge to the desired limiting distribution.

Acceptance rates for σ when $N = 1, 2, 5, 10, 20$ are shown in Table 1. For the PM algorithm, the acceptance rate for σ is strictly increasing in N, the increase being most dramatic for the yearly data. The acceptance rates for the MCWM algorithm don't appreciably change with N, however. Table 2 displays the estimated ESJD for the PM algorithm. The ESJD is strictly increasing in N, not surprisingly the increase is most pronounced for the yearly data. Based upon the ESJD/N metric in Table 2, it appears that $N = 1$ yields the most efficient algorithm for monthly data, while $N = 2$ is most efficient for yearly data.

In our simulation studies (and in the FedFunds analysis in Section 8), the PM algorithm was approximately 3 times faster than MCWM for any given combination of M and N. The speed

		Exact	$N = 1$	$N = 2$	$N = 5$	$N = 10$	$N = 20$
ESJD $\times 10^{-6}$	Yearly	*4.534*	0.340	1.045	2.389	3.090	3.559
	Monthly	*4.264*	3.518	3.884	4.162	4.238	4.250
ESJD/$N \times 10^{-6}$	Yearly	*4.534*	0.340	0.523	0.478	0.309	0.178
	Monthly	*4.264*	3.518	1.942	0.832	0.424	0.213

Table 2: Expected squared jump distance (ESJD) for σ using the PM algorithm with $M = 20$ for yearly data and $M = 5$ for monthly data. ESJD for the exact algorithm is also included.

	$N=1$	$N=2$	$N=5$	$N=10$	$N=20$
σ Acc. Rate	0.194	0.103	0.070	0.056	0.031
$\mathbf{u}_i$ Acc. Rate	0.047	0.122	0.243	0.490	0.986

Table 3: Acceptance rates for σ and blocks of missing $\mathbf{u}_i$ (fixing $\alpha = 0.07$ and $\beta = 0.15$) using a blocking strategy. Simulated yearly data, calculated with $M = 20$ and $N = 1, 2, 5, 10, 20$.

differential depends upon several factors, but it mainly depends on the probability of choosing a $\sigma-$move. In our simulation studies, we chose a $\sigma-$move with probability 1/3. *Increasing* this move probability will increase the number of importance samples that need to be generated and thus *slow down* the PM algorithm. This added computational burden will yield more rapid mixing for σ since we attempt to update σ more often. Decreasing the $\sigma-$move probability increases the speed of the algorithm while decreasing the rapidity of mixing of the σ parameter. Because the MCWM algorithm generates two fresh sets of importance samples within every iteration (i.e., for both $(\alpha, \beta)-$moves and $\sigma-$moves), it is largely unaffected by the move probabilities.

We now demonstrate the inefficiencies of the blocking strategy described in Section 4. Using the yearly simulated dataset, the acceptance rates for the volatility σ and the blocks $\mathbf{u}_i$ are shown in Table 3 (we used $M = 20$ and $N = 1, 2, 5, 10, 20$). As N increases, it is clear that the acceptance rate for σ *decreases*, while the acceptance rate for the blocks $\mathbf{u}_i$ increase.

7.2 Heston Model: Weekly Data ($\Delta = 1/52$)

We compare Bayesian analyses using the PM and MCWM algorithms. For the simulation study, we generate one dataset from the Heston model defined in (10)-(11) with $\alpha = 0.1$, $\beta = 3$, $\mu = 0.05$, $\rho = -0.8$, and $\sigma = 0.25$ using an Euler discretization of the process. We use 100 sub-intervals per sampling interval; 99 out of every 100 observations are then discarded, leaving only observation at a weekly frequency. We generate 1000 observations, but discard the first 500 (see Aït-Sahalia and Kimmel, 2007).

As in Kalogeropoulos et al. (2010), we assume a flat prior for all parameters, restricting $\alpha > 0$, $\beta > 0$, $\sigma > 0$, and $\rho \in (-1, 1)$. Our simulation study focuses on σ and ρ since convergence and mixing behavior for the covariance coefficient is the most problematic. The systematic scan Gibbs algorithm used the following proposals: $\alpha^* \sim N(\alpha, 0.1^2)$, $\beta^* \sim N(\beta, 1.0^2)$, $\sigma^* \sim N(\sigma, 0.1^2)$, $\mu^* \sim N(\mu, 0.447^2)$, and $\rho^* \sim N(\rho, 0.1^2)$ (the parameters were updated in this order as well). All algorithms were run for 110,000 iterations including a 10,000 iteration burn-in period (for the systematic scan algorithm, one iteration consists of updating all 5 parameters).

We first consider the question of choosing M. We performed a simulation study using the PM algorithm with $M = 5, 10, 20, 30$ and $N = 20$ (graphic is not shown). Unlike the CIR simulation study, an exact sampler is not available, and thus we have no "reference" for comparison like we

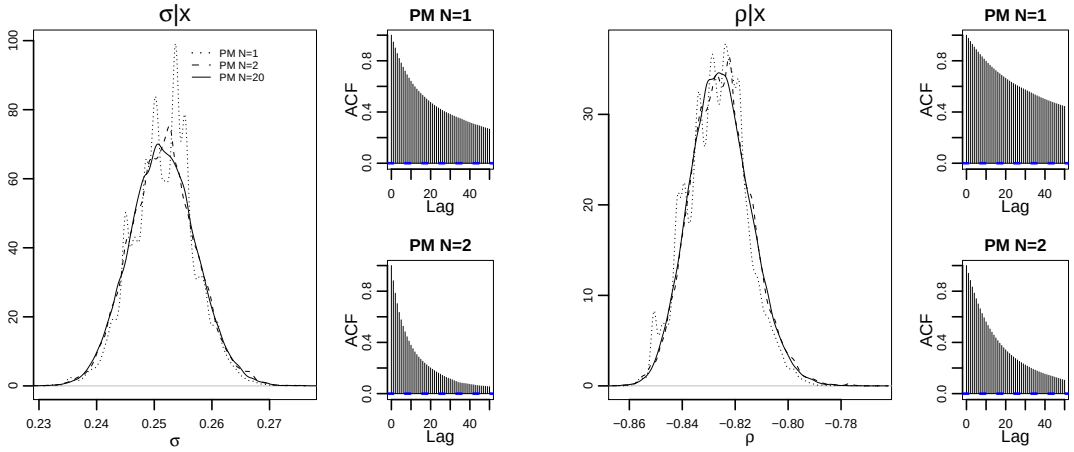

Figure 5: Weekly simulated Heston data: Estimated marginal posterior distribution of σ and ρ for PM with with $M = 20$, $N = 1, 2, 20$. ACF is also depicted.

used in Figures 1 and 2 for determining an adequate value for M. What we can do, however, is determine how large M must be for the marginal posterior distributions to "stabilize". We found that the estimated marginal posterior distributions remained virtually unchanged after increasing M from 20 to 30, thus we consider the discretization with $M = 20$ to be sufficiently fine. We use $M = 20$ in the remainder of this section.

We next consider the question of choosing the number of importance samples N. In Figure 5 we compare the PM algorithm with $M = 20$ and $N = 1, 2$ to the PM algorithm with $M = 20$ and $N = 20$. The estimated marginal posterior densities and the autocorrelation function (ACF) indicate that the convergence rate when $N = 1$ is extremely slow, although using $N = 2$ offers some improvement. Figure 6 displays the estimated marginal posterior density estimates and ACF for the PM and MCWM algorithms with $M = 20$ and $N = 5, 10, 20$. The performance of the PM algorithm is dramatically improved compared to when $N = 1, 2$. In fact, $N = 5$ with 100,000 (post burn-in) iterations can yield sufficiently accurate posterior estimates for the PM algorithm. Clearly, the MCWM algorithm will yield inferior posterior estimates even when $N = 20$, thus a larger N is needed to provide acceptable results.

8 Real Data

8.1 CIR Model: FedFunds Dataset

We now test the PM and MCWM algorithms with the FedFunds rate data observed monthly from January 1963 to December 1998 ($n = 432$) (see Figure 7). As in Di Pietro (2001) and Aït-Sahalia (1999), we chose the CIR model for the FedFunds rate for illustrative reasons; more complex models, like regime-switching models or SDE models with time-varying parameters, are potentially better models for the FedFunds data (see Di Pietro, 2001).

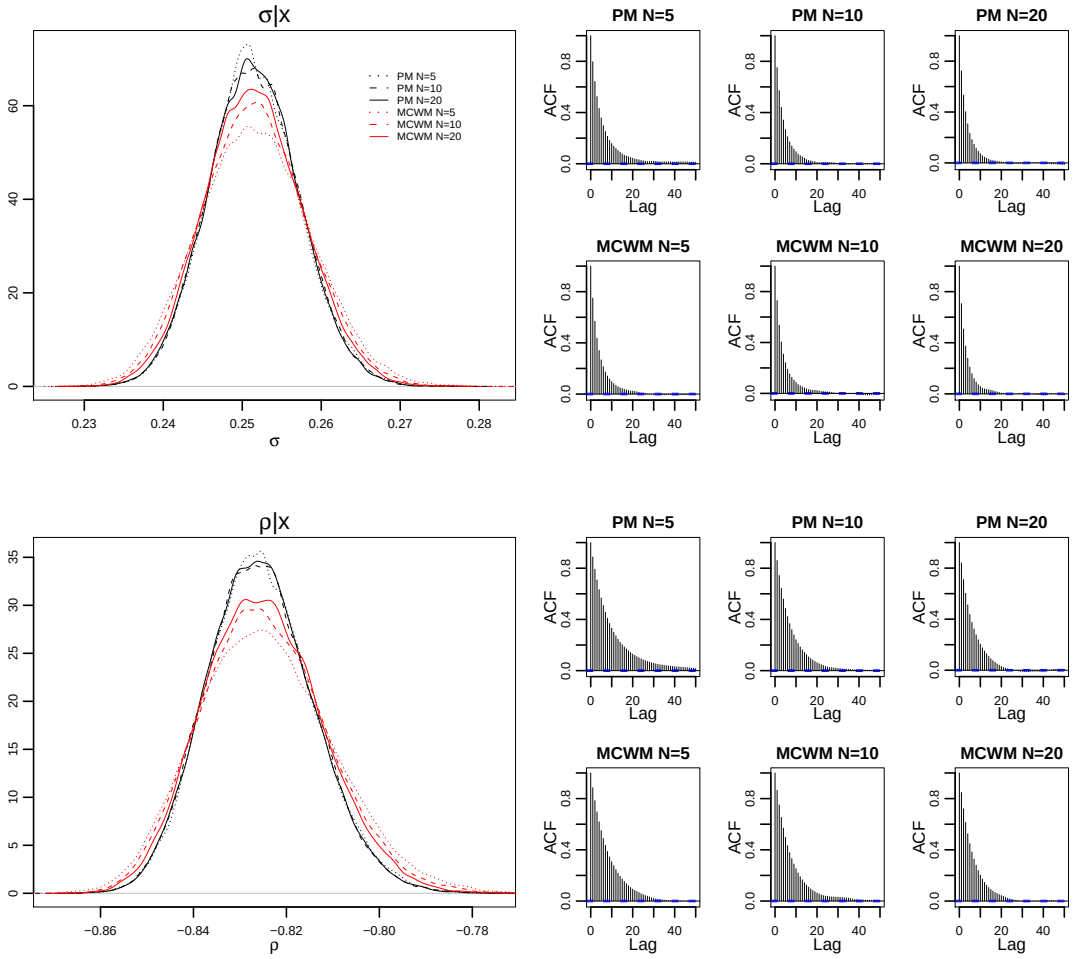

Figure 6: Weekly simulated Heston data: Left panels depict the estimated marginal posterior distribution of σ and ρ; right panels depict the ACF. Our plots are based on PM (black) and MCWM (red) with $M = 20$, $N = 5, 10, 20$.

Our prior specification is similar to Di Pietro (2001). The parameter α is the mean reverting level. We note that interest rates are non-negative, and the FedFunds rate peaked at just over 22.36% (or 0.2236) in 1981; worldwide, however, there is no clear upper bound for interest rates (in Zaire, the interest rate topped 10,000% in 1994). The prior on α should be dictated by the economy being modeled. For the FedFunds rate, we place a $Unif(0, 1)$ prior on α. We assume that the process exhibits mean *reversion* (commonly exhibited by interest rates), and thus constrain $\beta > 0$ (if $\beta < 0$, then the process runs *away* from the mean α). Since σ is the scale parameter of the Brownian motion, we specify the prior on σ in the usual way, $\sigma^{-1} I_{(0,\infty)}(\sigma)$, where I denotes the indicator function. Thus, the joint prior is

$$\pi(\theta) = \pi(\alpha, \beta, \sigma) = I_{(0,1)}(\alpha)\ I_{(0,\infty)}(\beta)\ \sigma^{-1} I_{(0,\infty)}(\sigma) \tag{15}$$

Random walk proposals were used: the joint proposal (α^*, β^*) in the $(\alpha, \beta)-$move was generated

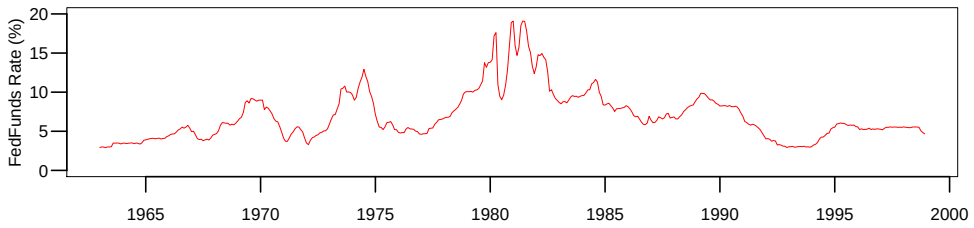

Figure 7: Figure depicts the monthly FedFunds Rate (in percent) from January 1963 to December 1998.

by choosing $\alpha^* \sim Unif(\alpha - 0.05, \alpha + 0.05)$ and $\beta^* \sim Unif(\beta - 0.125, \beta + 0.125)$, the $\sigma-$move used $\sigma^* \sim Unif(\sigma - 0.01, \sigma + 0.01)$.

To determine the optimal amount of discretization, we ran PM chains with $M = 2, 5, 10, 20$ and $N = 10$, and plotted the estimated marginal posterior densities $\pi(\alpha|\mathbf{x})$, $\pi(\beta|\mathbf{x})$, and $\pi(\sigma|\mathbf{x})$ (the graphic is not shown, but it is similar in flavor to Figure 1). For choosing M (and N below), all algorithms were run for 510,000 iterations including a 10,000 iteration burn-in period. The estimated marginal posterior densities are approximately the same for $M = 10$ and $M = 20$ (while $M = 2, 5$ noticeably differed); thus, we consider the discretization with $M = 20$ to be sufficiently fine (using $M = 10$ would probably suffice, however).

We next consider the question of choosing the number of importance samples N. Using $M = 20$, we ran PM chains with $N = 1, 2, 5, 10, 20$ and recorded the expected squared jump distance (ESJD) for σ. The results are in Table 4. As in the simulation study, the ESJD increases with the number of importance samples N, however, the algorithm is most efficient (according to the ESJD/N metric) with $N = 1$; this was also the case for the simulation study with monthly data.

In Figure 8 we show the estimated marginal posterior densities constructed using the output of PM and MCWM chains with $M = 20$ and $N = 5$ (the output from the exact sampler is depicted for comparison). All chains were run for 500,000 iterations (not including a 10,000 iteration burn-in period) on a Debian GNU Linux machine utilizing an Intel i7 2.8GHz quad-core processor. Both the PM and MCWM samplers yield excellent marginal posterior density estimates, though, when compared to the PM results, the MCWM sampler appears to yield slightly inferior estimates. The

	Exact	$N = 1$	$N = 2$	$N = 5$	$N = 10$	$N = 20$
ESJD $\times 10^{-6}$	*4.391*	3.349	3.913	4.133	4.259	4.420
ESJD$/N \times 10^{-6}$	*4.391*	3.349	1.957	0.827	0.426	0.221

Table 4: FedFunds data. Expected squared jump distance (ESJD) for σ using the PM algorithm with $M = 20$ and $N = 1, 2, 5, 10, 20$. ESJD for the exact algorithm is also included.

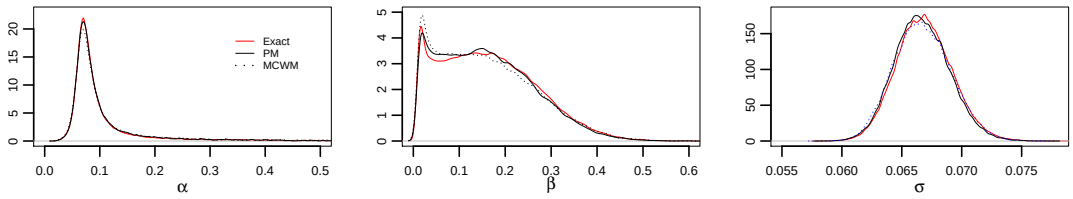

Figure 8: FedFunds data. Estimated marginal posterior distribution of α, β, and σ for PM (solid black lines) and MCWM (dotted black lines) with $M = 20$ and $N = 5$. The exact sampler is depicted in red.

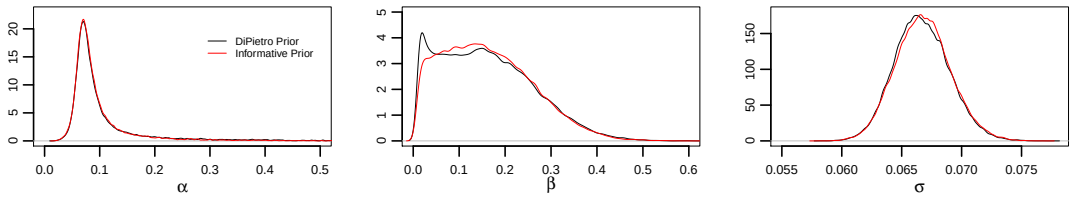

Figure 9: FedFunds data. Estimated marginal posterior distribution of α, β, and σ using the prior of Di Pietro (in black) and an informative prior (in red). PM sampler used $M = 20$ and $N = 5$.

PM sampler is much more efficient as the execution time for the PM sampler was approximately 50 minutes while the MCWM sampler took 153 minutes. The estimates from the MCWM algorithm could be improved if N was increased, albeit with a subsequent increase in execution time.

Finally, we performed a small sensitivity analysis. We compared Di Pietro's prior (15) with a truncated Gaussian prior

$$\pi(\theta) = \pi(\alpha, \beta, \sigma) \propto I_{(0,1)}(\alpha)\phi(\alpha; 0.1, 0.2^2)\ I_{(0,\infty)}(\beta)\phi(\beta; 0.1, 0.4^2)\ I_{(0,\infty)}(\sigma)\phi(\sigma; 0.1, 0.1^2)$$

Figure 9 indicates that all parameters are quite robust to the choice of prior, especially α and σ. The speed β seems a little more sensitive to the prior as the drift parameters α and β are typically the most difficult to estimate (especially β).

8.2 Heston Model: *S&P 500*, *VIX* Bivariate Dataset

The bivariate *S&P 500* and *VIX* implied volatility data recorded daily ($\Delta = 1/252$) from January 2, 1998 to December 31, 2003 is depicted in Figure 10. The *VIX* tends to rise as fear and uncertainty in the market increases. The *VIX*, quoted in terms of percentage points, approximates the expected movement in the *S&P 500* index over the next 30-day period on an annualized basis. If, for example, the VIX is at 40, then the expected *annualized* change is 40% over the next 30 days; in other words, we can expect the *S&P 500* to move up or down $40\%/\sqrt{12} = 11.5\%$ over the next 30-day period. Via normality, one can assume that there is a 68% likelihood that the *S&P 500* will move less than

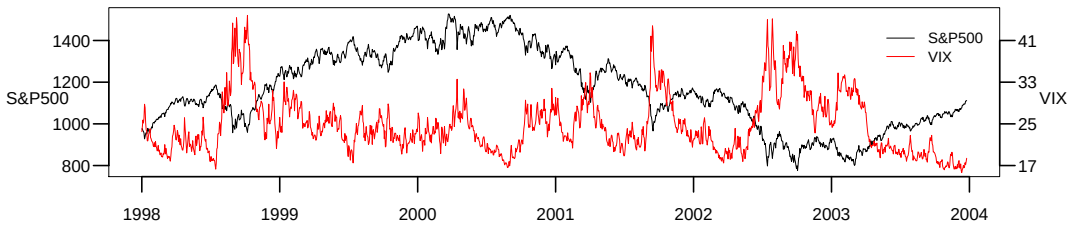

Figure 10: Figure depicts the *S&P 500* (in black) and *VIX* implied volatility (in red) observed daily from January 2, 1998 to December 31, 2003. The scale for the *S&P 500* is depicted on the left axis, the scale for the *VIX* is depicted on the right axis.

11.5% in the next 30 days (thus, there is a 32% chance that the *S&P 500* moves more than 11.5% in the next 30 days!)

We use Heston's model, defined in (10)-(11), for this dataset. We adopt relatively non-informative priors: $\alpha \sim N(0.1, 10^2)$ (truncated to $\mathcal{R}^+$), $\beta \sim N(2, 10^2)$ (truncated to $\mathcal{R}^+$), $\sigma \sim N(0.5, 10^2)$ (truncated to $\mathcal{R}^+$), $\mu \sim N(0.1, 10^2)$, $\rho \sim N(-0.5, 10^2)$ (truncated to $(-1, 1)$). The systematic scan Gibbs algorithm used the following proposals: $\alpha^* \sim N(\alpha, 0.1^2)$, $\beta^* \sim N(\beta, 1.414^2)$, $\sigma^* \sim N(\sigma, 0.1^2)$, $\mu^* \sim N(\mu, 0.447^2)$, and $\rho^* \sim N(\rho, 0.122^2)$ (the parameters were updated in this order as well). Both the PM and MCWM algorithms used $M = 10$ and $N = 5, 20$. Each algorithm was run for 100,000 iterations (one iteration consists of updating all 5 parameters) following a 10,000 iteration burn-in period.

For stochastic volatility models, the priors are much more important in analyzing drifts than volatilities, especially for high frequency data. This is because inferences about volatility parameters become arbitrarily accurate, at least in theory, as the sampling interval shrinks to zero. This is not true for the drift parameters, however. For stochastic volatility models, we are able to eliminate most posterior variance for (σ, ρ) by using daily data; any reasonably diffuse prior will have little, if any effect on the posterior. However, high frequency data provides very little information about the drift parameters. The drift estimate strongly depends on the length T of the available data. As noted by Aït-Sahalia and Kimmel (2007), "The volatility can be estimated to an arbitrary degree of precision by sampling frequently enough, but the drift estimate is independent of sampling frequency." Thus, inference for the drift is robust to the choice of the prior if we observe the data over a long enough time period. The simulation study in Section 7.2 had 500 observations of *weekly* data ($\Delta = 1/52$), and thus the drift parameters turned out to be reasonably robust to the choice of prior. For the *S&P 500 VIX* dataset, we observed $n = 1{,}508$ observations of *daily* data which, relatively speaking, contains much less information about the drift parameters.

Figure 11 displays the estimated marginal posterior distributions of α, β, σ, μ, and ρ and the autocorrelation function (ACF) for each sampler. The MCWM algorithm appears to inflate the

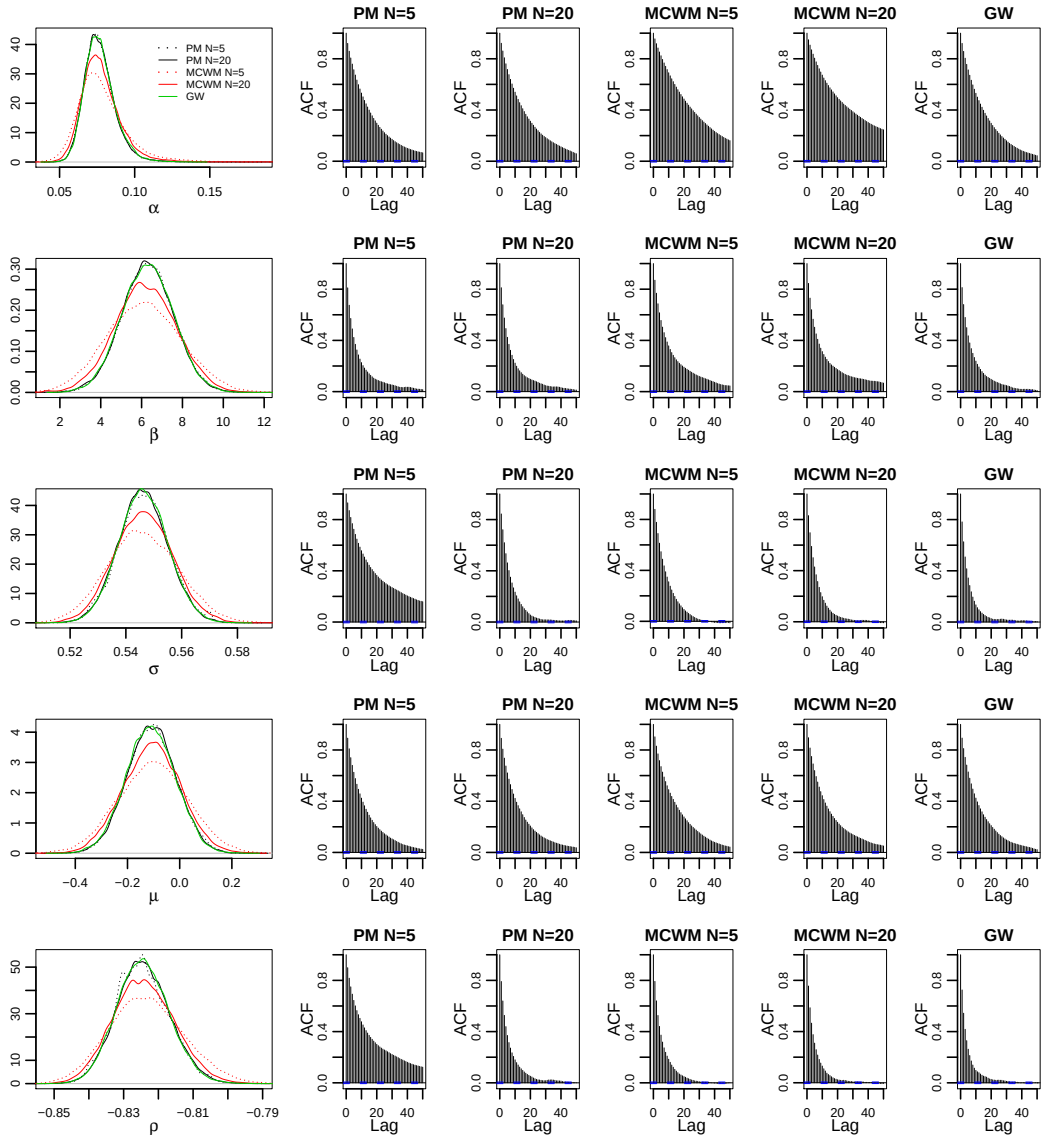

Figure 11: *S&P 500*, *VIX* analysis: Left panels depict the estimated marginal posterior distribution of $\alpha, \beta, \sigma, \mu,$ and ρ for the PM and MCWM algorithms using $M = 10$ and $N = 5, 20$ and for the GW algorithm with $M = 10$; right panels depict the corresponding ACF plots.

marginal posterior variance compared to the PM algorithm. Increasing N in the MCWM algorithm allows the marginal posterior distributions to be more closely approximated, but $N = 20$ still appears to be too small to obtain sufficiently accurate estimates. On the other hand, the estimated marginal posterior distributions are virtually identical for the PM algorithm when $N = 5, 20$. Unlike MCWM, the PM algorithm benefits from decreased autocorrelation for σ and ρ when N is increased from 5 to 20 (this comes with additional computational expense, however). Using $N > 20$ in the PM algorithm would probably yield little (if any) benefit in terms of autocorrelation, but would be more computationally costly.

Golightly-Wilkinson Algorithm

Golightly and Wilkinson (2008) introduced a transformation and an MCMC data augmentation algorithm which has been empirically shown to have good convergence rates when M is "big". This approach utilized the MBB samplers. We found the Golightly-Wilkinson (GW) approach to work quite well for the Heston model, although the updates are quite computationally demanding. For a given number of imputed data points, the PM approach with $N = 3$ is roughly the same speed, iteration to iteration, as GW.

We ran GW algorithm on the S&P500/VIX dataset using the same prior, proposals, and number of imputed ($M = 10$) as the PM analysis. Figure 11 shows that the estimated marginal posterior densities closely mirror those for PM. The ACF's for the GW algorithm are comparable to PM with $N = 20$. To obtain comparable Monte Carlo errors, it takes the PM algorithm (with $N = 20$) approximately 7 times as long to execute as GW. However, PM appears to be more amenable to efficient parallelization (although we haven't parallelized GW, our experience suggests that the speed increase will be less dramatic than parallelized PM). The time disadvantage of PM may be minimized given appropriate resources. In addition, we believe that PM will more easily scale to higher dimensional models; because GW involves the Jacobian, it will be computationally more intense for models with complicated covariance functions.

9 Discussion

We have proposed Monte Carlo within Metropolis (MCWM) and pseudo-marginal (PM) algorithms for simulating from the posterior distribution of the Euler-Mayumara approximation to diffusions. These algorithms can be applied to a broad class of multidimensional non-reducible diffusion processes which are observed (possibly with noise) only at discrete time points. These algorithms avoid the need for re-parametrization techniques that are not always available and/or justified; many are computationally intense.

Our simulation study focused on the Cox-Ingersoll-Ross (CIR) and Heston models. It demonstrated a strategy for choosing the number of sub-intervals M between observed data points, the effect of the number of importance samples N on the mixing behavior of the PM algorithm (and its lack of effect on mixing in for MCWM), the superiority of posterior inferences when using the PM algorithm (compared to MCWM), and the heightened computational efficiency of the PM algorithm. Our analysis of the FedFunds rate using the CIR model, and our analysis of the bivariate S&P 500/VIX dataset using Heston's model, demonstrated the efficacy of the PM and MCWM algorithms with real-world data, reiterated PM's efficiency advantage over MCWM, and demonstrated that, unlike the PM algorithm, the MCWM algorithm has a limiting distribution close to, but not exactly, the desired posterior distribution. A potential downside of the PM algorithm is

its dependence on the Markov property. The PM algorithm is not efficient when some components of the diffusion process are not observed and cannot be extracted. Finally, further work is required to investigate the relative merits of re-parametrization approaches (see Section 2) to the PM and MCWM schemes. Our analysis of the S&P 500/VIX dataset shows that Golightly-Wilkinson (GW) algorithm, a competitor of the PM algorithm, works very well for the Heston model. However, the efficacy of GW for more complex models needs to be explored. With two viable approaches, practitioners have the option to choose the algorithm with the best convergence and mixing properties for their model.

References

Y. Aït-Sahalia (1999). "Transition densities for interest rate and other nonlinear diffusions." *Journal of Finance* **54**, 1361–1395.

Y. Aït-Sahalia (2002). "Maximum likelihood estimation of discretely sampled diffusions: A closed-form approximation approach." *Econometrica* **70**, 223–262.

Y. Aït-Sahalia (2008). "Closed-form likelihood expansions for multivariate diffusions." *The Annals of Statistics* **36**, 906–937.

Y. Aït-Sahalia and R. Kimmel (2007). "Maximum likelihood estimation of stochastic volatility models." *Journal of Financial Economics* **83**, 413–452.

C. Andrieu, K. Berthelsen, A. Doucet and G. Roberts (2010). "Posterior sampling in the presence of unknown normalising constants: An adaptive pseudo-marginal approach." Technical report, University of Bristol.

C. Andrieu and G. Roberts (2009). "The pseudo-marginal approach for efficient Monte Carlo computations." *The Annals of Statistics* **37**, 697–725.

V. Bally and D. Talay (1996). "The law of the Euler scheme for stochastic differential equations. II: Convergence rate of the density (STMA V38 2092)." *Monte Carlo Methods and Applications* **2**, 93–128.

M. Beaumont (2003). "Estimation of population growth or decline in genetically monitored populations." *Genetics* **164**, 1139–1160.

A. Beskos, O. Papaspiliopoulos and G. O. Roberts (2009). "Monte Carlo maximum likelihood estimation for discretely observed diffusion processes." *Annals of Statistics* **37**, 223–245.

A. Beskos, O. Papaspiliopoulos, G. O. Roberts and P. Fearnhead (2006). "Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with

discussion)." *Journal of the Royal Statistical Society, Series B: Statistical Methodology* **68**, 333–382.

M. Chernov, A. R. Gallant, E. Ghysels and G. Tauchen (2003). "Alternative models for stock price dynamics." *Journal of Econometrics* **116**, 225–257.

S. Chib, M. K. Pitt and N. Shephard (2006). "Likelihood based inference for diffusion driven models." Technical report, Olin School of Business, Washington University, St Louis, MO.

J. C. Cox, J. E. Ingersoll and S. A. Ross (1985). "A theory of the term structure of interest rates." *Econometrica* **53**, 385–407.

M. Di Pietro (2001). "Bayesian inference for discretely sampled diffusion processes with financial applications." Ph.D. Thesis, Department of Statistics, Carnegie-Mellon University.

G. B. Durham and A. R. Gallant (2002). "Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes." *Journal of Business & Economic Statistics* **20**, 297–338.

O. Elerian, S. Chib and N. Shephard (2001). "Likelihood inference for discretely observed nonlinear diffusions." *Econometrica* **69**, 959–993.

B. Eraker (2001). "MCMC analysis of diffusion models with application to finance." *Journal of Business & Economic Statistics* **19**, 177–191.

B. Eraker (2004). "Do stock prices and volatility jump? Reconciling evidence from spot and option prices." *Journal of Finance* **59**, 1367–1404.

A. Golightly and D. J. Wilkinson (2006). "Bayesian sequential inference for nonlinear multivariate diffusions." *Statistics and Computing* **16**, 323–338.

A. Golightly and D. J. Wilkinson (2008). "Bayesian inference for nonlinear multivariate diffusion models observed with error." *Computational Statistics and Data Analysis* **52**, 1674–1693.

S. Heston (1993). "A closed-form solutions for options with stochastic volatility with applications to bonds and currency options." *Review of Financial Studies* **6**, 327–343.

M. S. Johannes, N. G. Polson and J. R. Stroud (2009). "Optimal filtering of jump diffusions: Extracting latent states from asset prices." *Review of Financial Studies* **22**, 2759–2799.

C. S. Jones (1999). "Bayesian estimation of continuous-time finance models." Unpublished paper, Simon School of Business. University of Rochester.

C. S. Jones (2003). "The dynamics of stochastic volatility: Evidence from underlying and options markets." *Journal of Econometrics* **116**, 181–224.

K. Kalogeropoulos (2007). "Likelihood-based inference for a class of multivariate diffusions with unobserved paths." *Journal of Statistical Planning and Inference* **137**, 3092–3102.

K. Kalogeropoulos, G. Roberts and P. Dellaportas (2010). "Inference for stochastic volatility models using time change transformations." *Annals of Statistics* **38**, 784–807.

C. G. Lamoureux and A. Paseka (2005). "Information in option prices and the underlying asset dynamics." Working paper, Eller School of Business, University of Arizona.

G. N. Milstein, J. G. Schoenmakers and V. Spokoiny (2004). "Transition density estimation for stochastic differential equations via forward-reverse representations." *Bernoulli* **10**, 281–312.

P. D. O'Neil, D. J. Balding, N. G. Becker, M. Serola and D. Mollison (2000). "Analyses of infectious disease data from household outbreaks by Markov chain Monte Carlo methods." *Applied Statistics* **49**, 517–542.

C. Pasarica and A. Gelman (2010). "Adaptively scaling the Metropolis algorithm using expected squared jumped distance." *Statistica Sinica* **20**, 343–364.

G. O. Roberts and O. Stramer (2001). "On inference for partially observed nonlinear diffusion models using the Metropolis-Hastings algorithm." *Biometrika* **88**, 603–621.

H. Sørensen (2004). "Parametric inference for diffusion processes observed at discrete points in time: a survey." *International Statistical Review* **72**, 337–354.

O. Stramer, M. Bognar and P. Schneider (2010). "Bayesian inference of discretely sampled Markov processes with closed-form likelihood expansions." *The Journal of Financial Econometrics* **8**, 450–480.

O. Stramer and J. Yan (2007). "Asymptotics of an efficient monte carlo estimation for the transition density of diffusion processes." *Methodology and Computing in Applied Probability* **9**, 483–496.