

# Accelerating pseudo-marginal Metropolis-Hastings by correlating auxiliary variables

Johan Dahlin, Fredrik Lindsten, Joel Kronander and Thomas B. Schön\*

November 18, 2015

## Abstract

Pseudo-marginal Metropolis-Hastings (pmMH) is a powerful method for Bayesian inference in models where the posterior distribution is analytical intractable or computationally costly to evaluate directly. It operates by introducing additional auxiliary variables into the model and form an extended target distribution, which then can be evaluated point-wise. In many cases, the standard Metropolis-Hastings is then applied to sample from the extended target and the sought posterior can be obtained by marginalisation. However, in some implementations this approach suffers from poor mixing as the auxiliary variables are sampled from an independent proposal. We propose a modification to the pmMH algorithm in which a Crank-Nicolson (CN) proposal is used instead. This results in that we introduce a positive correlation in the auxiliary variables. We investigate how to tune the CN proposal and its impact on the mixing of the resulting pmMH sampler. The conclusion is that the proposed modification can have a beneficial effect on both the mixing of the Markov chain and the computational cost for each iteration of the pmMH algorithm.

This work was conducted independently from the recent arXiv pre-print by Deligiannidis et al. [2015].

**Keywords:** Bayesian inference, pseudo-marginal algorithms, auxiliary variables, Crank-Nicolson, particle filtering, importance sampling

**JEL codes:** C11, C32, C58.

---

\*E-mail to corresponding author: [johan.dahlin@liu.se](mailto:johan.dahlin@liu.se). JD is with the Department of Electrical Engineering, Linköping University, Sweden. FL is with the Engineering, University of Cambridge, United Kingdom. JK is with the Department of Science and Technology, Linköping University, Sweden. TS is with the Department of Information Technology, Uppsala University, Sweden.

# 1 Introduction

We are interested in approximating some probability distribution  $\bar{\pi}(\theta, u)$  using simulation methods based on Markov chain Monte Carlo (MCMC; Robert and Casella, 2004). In Bayesian inference,  $\bar{\pi}(\theta, u)$  could represent the posterior distribution of the parameters  $\theta \in \Theta$  and some auxiliary variables  $u \in \mathcal{U}$ , which e.g. can be latent states in the model or missing data. Often, we are interested in the marginal *posterior distribution of the parameters* given by

$$\pi_{\theta}(\theta) = \frac{p(\theta)p(y|\theta)}{\int_{\Theta} p(\theta)p(y|\theta)d\theta} = \frac{p(\theta)p_{\theta}(y)}{p(y)}, \quad (1)$$

where  $p_{\theta}(y) \triangleq p(y|\theta)$  denotes the *likelihood function* and  $p(\theta)$  denotes the *parameter prior distribution*. The denominator  $p(y)$  is usually referred to as the *marginal likelihood* or the model evidence.

In principle, we can apply MCMC methods to sample directly from (1) using e.g. the Metropolis-Hastings (MH; Metropolis et al., 1953, Hastings, 1970) algorithm. However, in practice  $\pi_{\theta}(\theta)$  can be analytically intractable or too costly from a computational perspective to evaluate point-wise. By introducing the auxiliary variables  $u$ , we can sometimes mitigate these problems and obtain an algorithm which can be of practical use.

This approach is discussed by Andrieu and Roberts [2009], where the pseudo-marginal MH (pmMH) algorithm is introduced to sample from  $\bar{\pi}(\theta, u)$  using a standard MH sampler. This approach can be seen as an *exact approximation* of the ideal algorithm, where  $\pi_{\theta}(\theta)$  can be recovered by marginalisation. A concrete example is the particle MH (PMH; Andrieu et al., 2010), where the auxiliary variables are all the random variables generated in a run of a sequential Monte Carlo (SMC; Del Moral et al., 2006) algorithm. In this setting, the SMC algorithm is used to provide an unbiased estimate of the likelihood function. This is useful for inference in latent variables models such as state space models (SSMs; Flury and Shephard, 2011, Pitt et al., 2012), mixture models [Fearnhead and Meligkotsidou, 2015] and generalised linear mixed models [Tran et al., 2014].

A typical problem in pmMH is that the resulting Markov chain can be highly autocorrelated, which corresponds to a sticky chain and poor exploration of the posterior. This is the result of that occasionally we obtain an estimate of the posterior which is much larger than its true value. This problem can typically be mitigated by increasing the number of auxiliary variables  $N_u$ , which increase the accuracy of the estimate of the target. However, this increases the computational cost for each iteration of the algorithm, which can limit practical use of the pmMH algorithm for many interesting but challenging models. The trade-off between increasing  $N_u$  and the number of iterations  $K$  in the pmMH algorithm is studied by Pitt et al. [2012], Sherlock et al. [2015] and Doucet et al. [2015]. They conclude that  $N_u$  should be selected such that the standard deviation in the log-target estimate is roughly between 1 and 2.

The main contribution of this paper is to introduce a positive correlation in the auxiliary variables  $u$  between two consecutive iterations of the pmMH algorithm. Intuitively, this allows us to decrease  $N_u$  and therefore the computational cost while keeping the mixing of the Markov chain constant. The correlation is introduced by changing the proposal for  $u$  from the standard independent proposal to a random walk proposal in the auxiliary space  $\mathcal{U}$ .

The idea for this originates in the field of computer graphics, where a similar approach is used for rendering images by sampling valid light paths connecting the light sources in the scene to the camera using MCMC methods. In Kelemen et al. [2002], the authors propose the primary sample path space Metropolis light transport algorithm that operates directly on a set of uniform variates used in a second step to construct a light ray through the scene, using sequential importance sampling. Recently, Hachisuka et al. [2014] extended the original algorithm to include multiple importance sampling (MIS; Owen and Zhou, 2000, Veach and Guibas, 1997, Kronander and Schön, 2014) to improve the efficiency further.

The idea of introducing correlation in  $u$  between iterations has previously been suggested (independently) in the original particle MCMC (PMCMC) paper by Andrieu et al. [2010] and in the connected discussions by Lee and Holmes [2010]. Furthermore, a similar idea of introducing correlation in  $u$  is proposed by Andrieu et al. [2012]. The main difference in our contribution is the use of the Crank-Nicolson (CN; Beskos et al., 2008, Cotter et al., 2013, Hairer et al., 2014) proposal to update  $u$  at every iteration of the algorithm. This in contrast with the standard pmMH proposal for  $u$ , which samples new random variables independently at each iteration of the algorithm. The CN proposal is a natural choice as it is known to scale independently with the dimension of the space. Furthermore, we provide the reader with both a theoretical and numerical analysis of how the mixing and the computational cost is affected and propose an adaptive algorithm based on the theoretical results. Deligiannidis et al. [2015] also proposes (independently) to make use of the CN proposal in this setting.

We consider two different numerical examples to illustrate the properties of the proposed modifications to the pmMH algorithm. In the first example, we infer the parameters in IID data from the Gaussian distribution to analyse how to tune the CN proposal and show how it improves upon the idea proposed by Lee and Holmes [2010]. In the second example, we conduct inference of the log-volatility using a stochastic volatility (SV) model with leverage on real-world stock index data.

We continue this paper by introducing the proposed modifications in Section 2 and analyse its theoretical properties, in a simplified setting, in Section 3. We end the paper by some numerical illustrations of the proposed method in Section 4 from which we draw some general conclusions in Section 5.

## 2 Introducing correlation into the auxiliary variables

We would like to sample from the parameter posterior  $\pi_\theta(\theta)$  in (1). However, this is not possible using a direct implementation of the MH algorithm as we cannot evaluate  $\pi_\theta(\theta)$  point-wise. Instead, we introduce the auxiliary variables  $u = (u_1, \dots, u_{N_u})$  as  $N_u$  independent standard Gaussian random variables, i.e.  $u_i \sim \mathcal{N}(0, 1)$ . We make use of the setup used by Andrieu and Roberts [2009] and Doucet et al. [2015]. Furthermore, we make no notational distinction between a random variable and its realisation for brevity.

Let the *potential function*  $\Phi_\theta(u) : \Theta \times \mathbb{R}^{N_u} \rightarrow \mathbb{R}$  be defined such that

$$\mathbb{E} \left[ \exp \left( - \Phi_\theta(u) \right) \right] = c \pi_\theta(\theta), \quad \forall \theta \in \Theta,$$

and some constant  $c > 0$ . Hence, we can define the *extended target distribution* as

$$\bar{\pi}(\theta, u) = \frac{\exp \left( - \Phi_\theta(u) \right)}{c} \mathcal{N}(u; \mathbf{0}, \mathbf{I}_{N_u}), \quad (2)$$

where  $\mathcal{N}(u; \mathbf{0}, \mathbf{I}_{N_u})$  denotes a  $N_u$ -dimensional multivariate standard Gaussian distribution. Note that the parameter posterior is the marginal of  $\bar{\pi}(\theta, u)$  as

$$\int_{\mathbb{R}^{N_u}} \bar{\pi}(\theta, u) \, du = \frac{1}{c} \mathbb{E} \left[ \exp \left( - \Phi_\theta(u) \right) \right] = \pi_\theta(\theta),$$

as required in the exact approximation scheme used in the pmMH algorithm. Hence, we conclude that this is a valid *extended target* for sampling from the parameter posterior  $\pi_\theta(\theta)$ . A concrete example of a suitable potential function follows from the definition in (1). In this case, we have

$$\begin{aligned} \Phi_\theta(u) &= - \left( \log \hat{p}_\theta(y; u) + \log p(\theta) \right), \\ \exp \left( - \Phi_\theta(u) \right) &= \hat{p}_\theta(y; u) p(\theta), \\ \mathbb{E} \left[ \exp \left( - \Phi_\theta(u) \right) \right] &= p_\theta(y) p(\theta) = \underbrace{p(y)}_{=c} \pi_\theta(\theta), \end{aligned}$$

where the constant  $c$  corresponds to the marginal likelihood and  $\hat{p}_\theta(y; u)$  denotes the non-negative and unbiased estimator of the likelihood constructed by the auxiliary variables. (The typical application of the pmMH algorithm makes use of importance sampling or particle filtering to construct such an estimator.)

From (2), we have that the target is high-dimensional in  $u$  whenever  $N_u$  is large, and that the extended target correspond to a change of measure from the Gaussian prior. In this setting, we know that the CN proposal is a suitable choice as it is dimension independent [Cotter et al., 2013]. An intuition for this can be given by realising that the CN proposal is a discretisation of the Ornstein-Uhlenbeck stochastic

differential equation, see e.g. Kloeden and Platen [1992]. The main difference between using the CN proposal and a standard random walk proposal is its autoregressive nature, which results in the mean-reverting behaviour. As a consequence, the CN proposal has the standard Gaussian distribution as its limiting distribution, which suits us well in this setting. Therefore, we assume a proposal distribution for  $u$  and  $\theta$  with the form

$$\begin{aligned} q_{\theta,u}(\{\theta', u'\}|\{\theta, u\}) &= q_{\theta}(\theta'|\{\theta, u\})q_u(u'|u) \\ &= \mathcal{N}(\theta'; \mu(\theta, u), \Sigma(\theta, u)) \mathcal{N}(u'; \sqrt{1 - \sigma_u^2}u, \sigma_u^2 \mathbf{I}_{N_u}). \end{aligned} \quad (3)$$

This corresponds to using a standard Gaussian random walk for  $\theta$ , where  $\mu(\theta, u)$  and  $\Sigma(\theta, u)$  denote a mean and covariance function possibly depending on the auxiliary variables in the previous step, respectively. Note that by introducing  $u$  into  $q_{\theta}$ , we can make use of gradient and Hessian information as proposed by Dahlin et al. [2015a] to improve the performance of the algorithm. The CN proposal for  $u$  is parametrised by the step length  $\sigma_u \in (0, 1)$ , which is determined by the user. We return to discussing how to tune  $\sigma_u$  in Section 3. Note that (3) is in contrast with the standard independent pmMH proposal, which we recover for the choice  $\sigma_u = 1$ .

The corresponding pmMH algorithm for sampling from (2) follows directly from the choice of proposal in (3). During iteration  $k$  of the algorithm, we first sample from (3) to obtain the *candidate parameter*  $\{\theta', u'\}$  given  $\{\theta_{k-1}, u_{k-1}\}$ . We then compute the acceptance probability by

$$\alpha(\{\theta', u'\}, \{\theta_{k-1}, u_{k-1}\}) = 1 \wedge \exp\left(\Phi_{\theta_{k-1}}(u_{k-1}) - \Phi_{\theta'}(u')\right) \frac{q_{\theta}(\theta_{k-1}|\{\theta', u'\})}{q_{\theta}(\theta'|\{\theta_{k-1}, u_{k-1}\})}, \quad (4)$$

where we make use of the notation  $a \wedge b \triangleq \min\{a, b\}$ . This follows directly from the properties of the CN proposal [Cotter et al., 2013]. In the last step, we accept the candidate parameter, i.e. set  $\{\theta_k, u_k\} \leftarrow \{\theta', u'\}$ , with the probability given by (4). Otherwise, we reject the candidate parameter and set  $\{\theta_k, u_k\} \leftarrow \{\theta_{k-1}, u_{k-1}\}$ .

The resulting pmMH is presented in Algorithm 1. In Line 4, we need to evaluate the potential function in  $\{\theta', u'\}$ . In this paper, we make use of importance sampling and particle filtering in Section 4 to construct  $\Phi_{\theta}(u)$ . Note that Algorithm 1 only differs from a standard pmMH algorithm in Lines 3 and 4, where we add a CN proposal for  $u$  and evaluate the potential function  $\Phi_{\theta}(u)$ . Hence, implementing the new algorithm only requires minor changes to the existing code.

In many models, we are required to make use of non-Gaussian random variables to generate samples from the proposal distributions in e.g. importance sampling and particle filtering algorithms. Furthermore, we require uniform random variables for the resampling step in the particle filter, see e.g. Doucet and Johansen [2011]. These variables can be obtained by using an inverse cumulative distribution function

---

**Algorithm 1** Pseudo-marginal Metropolis-Hastings (pmMH)

---

INPUTS:  $K > 0$  (no. MCMC steps),  $\theta_0$  (initial parameters),  $\Phi_\theta(u)$  (potential) and  $q_{\theta,u}$  (proposal).  
OUTPUT:  $\{\{\theta_1, u_1\} \dots, \{\theta_K, u_K\}\}$  (approximate samples from  $\bar{\pi}(\theta, u)$ ).

---

```
1: Generate  $u_0 \sim p(u_0)$  and compute  $\Phi_{\theta_0}(u_0)$ .
2: for  $k = 1$  to  $K$  do
3:   Sample  $\{\theta', u'\}$  using the proposal in (3).
4:   Compute  $\exp(-\Phi_{\theta'}(u'))$  and  $\alpha(\{\theta', u'\}, \{\theta_{k-1}, u_{k-1}\})$  given by (4).
5:   Sample  $\omega_k$  uniformly over  $[0, 1]$ .
6:   if  $\omega_k \leq \alpha(\{\theta', u'\}, \{\theta_{k-1}, u_{k-1}\})$  then
7:     Accept  $\{\theta', u'\}$ , i.e.  $\{\theta_k, u_k\} \leftarrow \{\theta', u'\}$ .
8:   else
9:     Reject  $\{\theta', u'\}$ , i.e.  $\{\theta_k, u_k\} \leftarrow \{\theta_{k-1}, u_{k-1}\}$ .
10:  end if
11: end for
```

---

(CDF) transformation also known as a quantile transformation. Firstly, we compute a uniform random variable by  $u^* = \Phi(u)$ , where  $\Phi(u)$  denotes the CDF of the standard Gaussian distribution. Secondly, we compute  $\bar{u}$  by a quantile transform of  $u^*$  using the inverse CDF for the distribution that we would like to simulate a random variable from. There exists other approaches e.g. accept-reject sampling that also can be useful for simulating random variables when the CDF cannot be inverted in closed-form. See e.g. Ross [2012] or Robert and Casella [2004] for more information.

### 3 Theoretical analysis

The aim of the theoretical analysis in this section is to give some guidance of how to tune  $\sigma_u$  to obtain a reasonable performance in the pmMH algorithm. More specifically, we are interested in determining the value of  $\sigma_u$  that maximises the mixing and to determine how this translates into a suitable acceptance rate in the algorithm. We begin by setting up the model for the analysis in Section 3.1, which depends on the multivariate random variable  $u$ . We then reformulate the model to replace  $u$  with a univariate random variable  $z$ . This enables use to make use of an analysis of a discretised model [Gelman et al., 1996, Yang and Rodríguez, 2013] in Section 3.2 to tune the CN proposal to optimise the mixing in the Markov chain. In Section 3.3, we report the results of this analysis, which we return to In Section 4 where we compare with the optimal tuning of  $\sigma_u$  in some practical scenarios.

#### 3.1 Setting up the model

To accomplish the aforementioned aim we will study the algorithm in a simplified setting. Firstly, since we are primarily interested in the effect of the correlation in the  $u$ -variables, we will start out by making the simplifying assumption that we fix  $\theta$  in the extended target  $\bar{\pi}(\theta, u)$ . Hence, we would like to analyse

sampling from

$$\bar{\pi}(u) = \frac{\exp(-\Phi(u))}{c} \mathcal{N}(u; 0, \mathbf{I}_{N_u}), \quad (5)$$

using the CN proposal  $q_u(u'|u)$  in (3). We believe that this is a reasonable proxy for the complete model (2) since in many cases only local moves are made in the  $\theta$  variable. Following Pitt et al. [2012] and Doucet et al. [2015] we also assume that  $\Phi(u)$  (which depend on  $N_u$ ) follows a central limit theorem (CLT) and that it therefore can be accurately approximated as being Gaussian distributed:

$$-\Phi(u) \sim \mathcal{N}\left(-\frac{\sigma_\Phi^2}{2}, \sigma_\Phi^2\right), \quad \text{when } u \sim \mathcal{N}(u; 0, \mathbf{I}_{N_u}). \quad (6)$$

for some  $\sigma_\Phi > 0$ . This assumption follows from the properties of the log-likelihood estimator based on (sequential) importance sampling; see Pitt et al., 2012, Doucet et al., 2015 for further details. This case is of interested to us as we make use of this type of estimator in Section 4 for computing an estimate of the parameter posterior distribution.

We can readily check that assumption (6) implies that

$$-\Phi(u) \sim \mathcal{N}\left(\frac{\sigma_\Phi^2}{2}, \sigma_\Phi^2\right), \quad \text{when } u \sim \bar{\pi}(u), \quad (7)$$

where  $\bar{\pi}$  is the  $N_u$ -dimensional distribution defined in (5). Consequently, if we define the random variable

$$z \triangleq \frac{\sigma_\Phi}{2} - \frac{1}{\sigma_\Phi} \Phi(u)$$

it follows that the law of  $z$  under  $\bar{\pi}$  is given by

$$\tilde{\pi}(z) = \mathcal{N}(z; \sigma_\Phi, 1).$$

Note that  $z$  is a one-dimensional variable, which means that we have reduced the high-dimensional target (5) to one of its one-dimensional marginals. We will study the properties of the CN proposal in this one-dimensional marginal space.

Indeed, assume that  $u \sim \bar{\pi}(u)$  is distributed according to the target (5) (i.e., it can be viewed as the state of the Markov chain at stationarity) and that  $u'|u \sim q_u(u'|u)$  is simulated from the CN proposal in (3). Define  $z$  and  $z'$  as above, as transformations of  $u$  and  $u'$ , respectively. From a bivariate CLT we can then obtain a bivariate Gaussian approximation for the vector  $(z, z')^\top$ , suggesting that the CN proposal for  $u$  corresponds to the one-dimensional proposal:

$$\tilde{q}(z'|z) = \mathcal{N}\left(z'; \sqrt{1 - \sigma_z^2}, \sigma_z^2\right), \quad (8)$$

for  $z$ , for some  $\sigma_z > 0$ . Note that  $\sigma_z$  in general differs from  $\sigma_u$  and that it depends on the nonlinear transformation  $\Phi(u)$ . However, it is clear that  $\sigma_u = 0 \Rightarrow \sigma_z = 0$  and  $\sigma_u = 1 \Rightarrow \sigma_z = 1$  and, intuitively,  $\sigma_z$  is a monotone function of  $\sigma_u$ . We investigate the dependence between these two variables numerically in Section 4.1.

Note that the resulting acceptance probability, for the reduced one-dimensional MH algorithm with target  $\tilde{\pi}(z)$  and proposal  $\tilde{q}(z'|z)$ , is given by:

$$\tilde{\alpha}(z, z') = 1 \wedge \exp(\sigma_\Phi z' - \sigma_\Phi z). \quad (9)$$

### 3.2 Analysis by discretisation of the state space

We follow Gelman et al. [1996] and Yang and Rodríguez [2013] and analyse the mixing of the Markov chain using a state space discretisation approach. We know that the expectation of any integrable *test function*  $\varphi : \mathcal{Z} \rightarrow \mathbb{R}$  under the target distribution

$$\tilde{\pi}[\varphi] = \mathbb{E}_{\tilde{\pi}}[\varphi(z)] = \int_{\mathcal{Z}} \varphi(z) \tilde{\pi}(z) dz,$$

can be approximated by the ergodic theorem [Meyn and Tweedie, 2009, Tierney, 1994] as

$$\widehat{\tilde{\pi}}[\varphi] = \frac{1}{K} \sum_{k=1}^K \varphi(z_k),$$

using the samples  $\{z_1, z_2, \dots, z_K\} \triangleq z_{1:K}$  obtained from the pmMH algorithm. Under the assumption of geometric ergodicity, we have that the error of the estimate obeys a CLT [Meyn and Tweedie, 2009, Tierney, 1994] given by

$$\sqrt{K} \left( \widehat{\tilde{\pi}}[\varphi] - \tilde{\pi}[\varphi] \right) \xrightarrow{d} \mathcal{N}(0, \nu),$$

where  $\nu$  denotes the asymptotic variance,

$$\nu = \mathbb{V}_{\tilde{\pi}}[\varphi] \cdot \underbrace{\left[ 1 + 2 \sum_{\tau=1}^{\infty} \widehat{\rho}_\tau(z) \right]}_{\triangleq \text{IACT}(z)}, \quad (10)$$



where  $\mathbb{V}_{\tilde{\pi}}[\varphi]$  and  $\text{IACT}(z)$  denote the variance of  $\varphi(z)$  over  $\tilde{\pi}(z)$  and the integrated autocorrelation time (IACT), respectively. Here,  $\rho_\tau(z) = \text{corr}(\varphi(z_k), \varphi(z_{k+\tau}))$  denotes the lag- $\tau$  autocorrelation of  $\varphi$ .

To optimise the mixing, we would like to determine  $\sigma_z$  such that  $1/\nu$  is as large as possible. To estimate the asymptotic variance, we partition the interval  $(z_{\min}, z_{\max})$  into  $L$  bins of equal length  $\Delta = (z_{\max} - z_{\min})/L$ . Hence, the center point of bin  $l$  is given by  $z_l = z_{\min} + (l - 0.5)\Delta$ . We can then define the transition matrix  $P = \{p_{lm}\}$  following Yang and Rodríguez [2013] by

$$p_{lm} = \begin{cases} \tilde{q}(z_m|z_l)\tilde{\alpha}(z_l, z_m)\Delta, & \text{for } l, m \in \{1, \dots, K\}, l \neq m \\ 1 - \sum_{k \neq l} p_{lk}, & \text{for } l = m \end{cases}, \quad (11)$$

where the proposal (8) and acceptance probability (9) enters into  $p_{lm}$ . Hence, we can estimate the *acceptance rate* (the probability of a jump) by

$$P_{\text{jump}} = \sum_{l=1}^L \pi_l(1 - p_{ll}), \quad (12)$$

where the *stationary probability* is calculated as

$$\pi_l = \mathcal{N}(z_l; \sigma_\Phi, 1).$$

Furthermore from the asymptotic results in Peskun [1973] and Kemeny and Snell [1976], we can estimate  $\nu$  by

$$\hat{\nu}(f, \pi, P) = \varphi^\top (2BZ - B - BA)\varphi, \quad (13)$$

where  $\varphi = [\varphi(z_1), \varphi(z_2), \dots, \varphi(z_L)]$  and  $B = \text{diag}(\pi_1, \pi_2, \dots, \pi_L)$ . Here, we introduce the *fundamental matrix* by  $Z = [I - (P - A)]^{-1}$  and the *limiting matrix* by  $A = \{a_{lm}\}$  with  $a_{lm} = \pi_m$ .

### 3.3 Tuning the CN proposal for the univariate case

We make use of a modified version of the C-code provided by Yang and Rodríguez [2013] to implement the discretisation approach. We set  $z_{\min} = -4$ ,  $z_{\max} = \sigma_\Phi + 4$  and  $L = 1,000$ . Here, we use  $\varphi(z) = z$  to optimise the mixing of the posterior mean. We vary  $\sigma_\Phi$  in the target and  $\sigma_z$  in the proposal in  $\{0, 0.25, \dots, 3.5\}$  and  $\{0.05, 0.075, \dots, 1\}$ , respectively. For each  $\sigma_\Phi$ , we find the  $\sigma_z$  that maximises  $P_{\text{jump}}$  computed by (12) and  $1/\hat{\nu}(f, \pi, P)$  calculated by (13). In Figure 1, we present the resulting  $\sigma_z$  as a function of  $\sigma_\Phi$ . We note that for  $\sigma_\Theta \in [1.0, 1.8]$  as recommended by e.g. Doucet et al. [2015], we should have  $\sigma_z \in [0.95, 0.7]$  to obtain the optimal mixing and this corresponds to an acceptance rate of around 75% to 85% in the Markov chain for  $u$ .

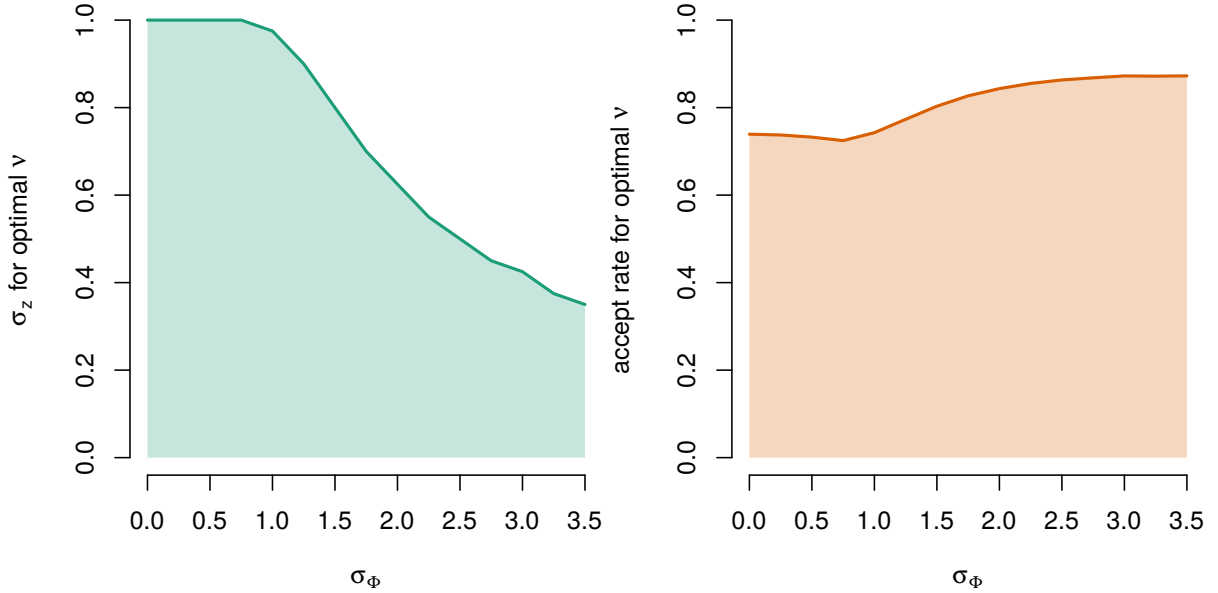


Figure 1: Optimal  $\sigma_u$  (left) and acceptance rate (right) for minimising the  $1/\hat{v}(f, \pi, P)$ .

Note that these results hold for the case when  $u$  can be described by the univariate Gaussian random variable  $z$ . We return to numerically investigate how to optimal value of  $\sigma_z$  in the CN proposal for the univariate random variable  $z$  relates to the corresponding value  $\sigma_u$  used in the CN proposal for the multivariate random variable  $u$  in Section 4.

## 4 Numerical illustrations

In this section, we investigate the numerical properties of the proposed alterations to the pmMH algorithm. We are especially interested in how the mixing is affected by making use of the CN proposal instead of an independent proposal for  $u$ . For this end with compare the mixing in two different models: (i) a Gaussian IID model with synthetic data and (ii) a nonlinear state space model with real-world data. The implementation details are presented in Appendix A, where we also describe how to estimate the value of the target distribution (log-likelihood) for each model.

We quantify the mixing by estimating the IACT using the empirical autocorrelation function. The estimate is computed by

$$\hat{\text{IF}}(\theta_{K_b:K}) = 1 + 2 \sum_{\tau=1}^{100} \hat{\rho}_\tau(\theta_{K_b:K}), \quad (14)$$

where  $\hat{\rho}_\tau(\theta_{K_b:K})$  denotes the empirical lag- $\tau$  autocorrelation of  $\theta_{K_b:K}$ , and  $K_b$  denotes the *burn-in* time.

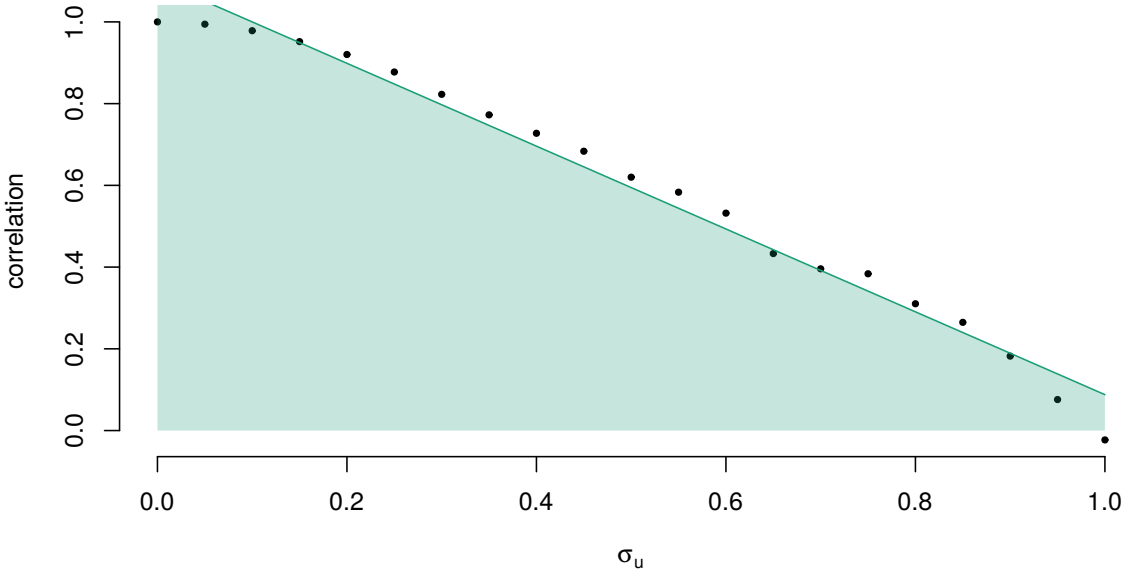


Figure 2: The estimated correlation (dots) in  $\widehat{p}_\theta(y; u)$  for the Gaussian IID model (15) as a function of  $\sigma_u$  and a linear regression (green). The linear approximation provides a reasonable approximation for much of the central part of the interval.

#### 4.1 Gaussian IID model

Consider the independent and identically (IID) distributed Gaussian model given by

$$x_0 \sim \delta_0, \quad x_t \sim \mathcal{N}(x_t; \mu, \sigma_v^2), \quad y_t | x_t \sim \mathcal{N}(y_t; x_t, \sigma_e^2), \quad (15)$$

with parameters  $\theta = \{\mu, \sigma_v, \sigma_e\}$  and where  $\delta_{x'}$  denotes the Dirac measure placed at  $x = x'$ . We generate a realisation with  $T = 10$  observations using the parameters  $\theta^* = \{0.5, 0.3, 0.1\}$ .

We begin by investigating how the correlation of the log-likelihood estimated using importance sampling (see Appendix A.1) depends on  $\sigma_u$ . In Figure 2, we present the correlation between two consecutive estimates of the log-likelihood (keeping  $\mu$  fixed) when  $\sigma_u \in \{0, 0.05, \dots, 1.0\}$ . We note that the correlation is almost linear when  $\sigma_u > 0.2$  and can be described as  $\text{corr}(\widehat{p}_\theta(y; u), \widehat{p}_\theta(y; u')) = 1 - \sigma_u$ . We also estimate the standard deviation in the log-likelihood and obtain 3.5. Hence, we conclude that this implies from Figure 1 that the optimal correlation in the log-likelihood estimator is approximately 0.5.

We proceed by fixing  $\sigma_v$  and  $\sigma_e$  to their true values and would like to infer the parameter posterior of  $\mu$  given the data using Algorithm 1. Here, the potential function corresponds to the log-likelihood estimator for the importance sampler as discussed in Appendix A.1. The aim is to compute the IACT over a grid of  $\sigma_v$  and compare with the optimal theoretical results from Section 3. We would also like to compare the proposed changes to pmMH with the suggestions discussed by Lee and Holmes [2010]. For this end, we consider a generalisation of the CN proposal (3) in which we introduce so-called *global moves*. The mixture of local and global moves is a popular approach in e.g. computer graphics [Kelemen

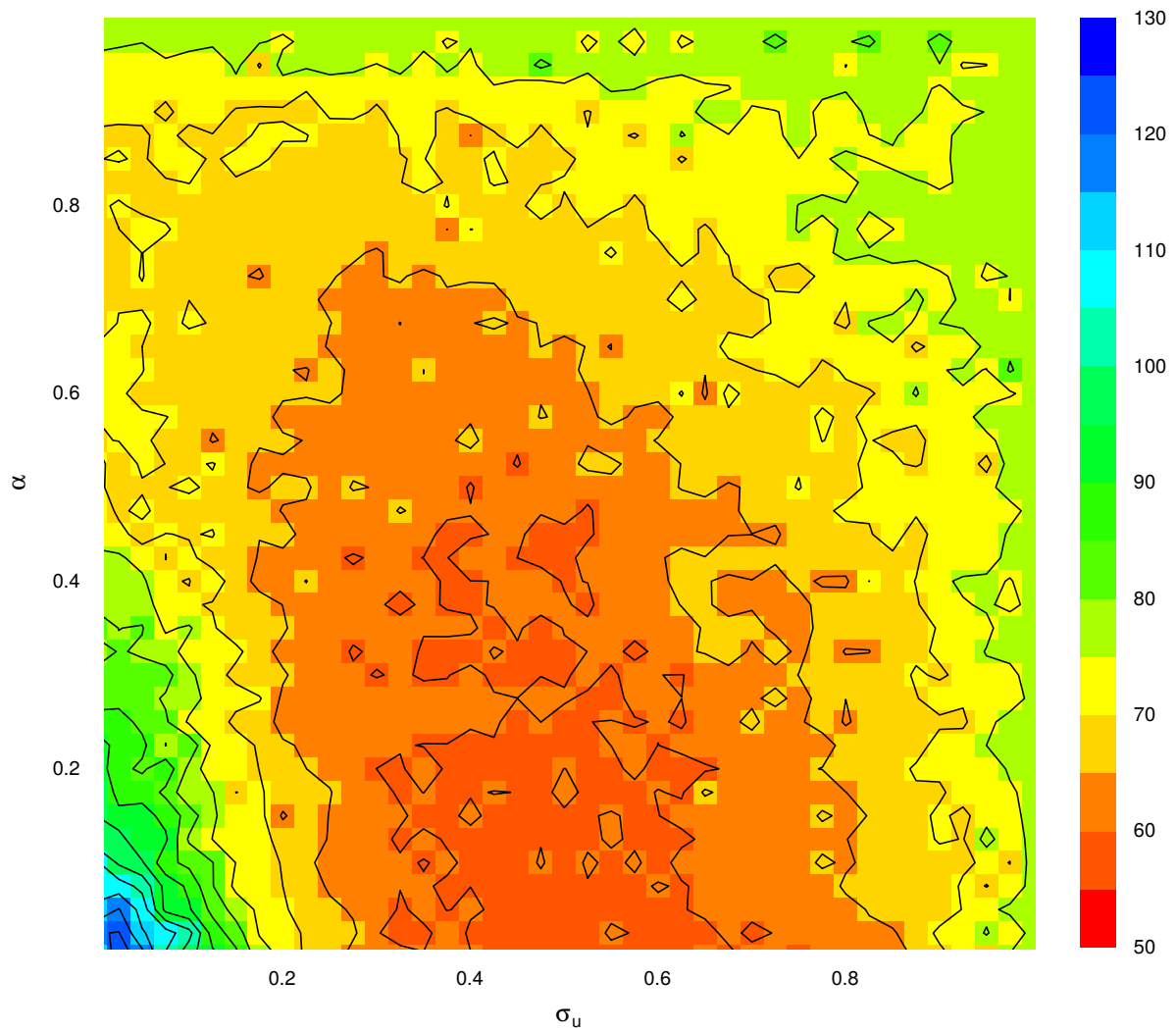


Figure 3: A heatmap of the IACT for the Gaussian IID model (15) when varying  $\alpha$  and  $\sigma_u$  in (16). The results presented are the median from 32 independent Monte Carlo runs.

et al., 2002] to promote exploration of the entire target space. In our setup, we can introduce such a mixture into  $q_u(u'|u_{k-1})$  and obtain the mixture proposal given by

$$\bar{q}_{\theta,u}(\{\theta', u\}|\{\theta, u\}) = q_\theta(\theta'|\{\theta, u\}) \left[ \alpha \mathcal{N}(u'; 0, \mathbf{I}_{N_u}) + (1 - \alpha) q_u(u'|u_{k-1}) \right] \quad (16)$$

where we make use of the original proposals for  $\theta$  and  $u$  from (3). Here,  $\alpha \in [0, 1]$  denotes the probability of a global move for  $u$ , where we recover the proposal in (3) by  $\alpha = 0$  and the proposal discussed by Lee and Holmes [2010] by  $\sigma_u = 0$  and  $\alpha \neq 0$ . We recover the standard pmMH proposal for  $u$  when  $\alpha = 1$  and/or  $\sigma_u = 1$ .

In Figure 3, we present a heatmap of the median IACT using (16) when varying  $\alpha, \sigma_u \in \{0, 0.025, \dots, 1.0\}$ . The smallest IACT values are obtained by using  $\sigma_u \in [0.4, 0.6]$  with a small (or zero) value of  $\alpha$ . This range of  $\sigma_u$  corresponds well with the results in Figure 1 with  $\sigma_\Phi = 3.5$ , which results in the optimal  $\sigma_z = 0.4$ . Furthermore, we note that using  $\sigma_u = 0$  results in worse performance than if  $\sigma_u > 0$  independently of the value of  $\alpha$ . Hence, we conclude that there is some benefit of using our proposed modification compared with the approach discussed by Lee and Holmes [2010] in this specific example. That is, using local moves seem to be more beneficial than global moves but there could be some merit to consider a mixture proposal in any case. However, we set  $\alpha = 0$  in the remainder of this section to simplify the tuning and make the conclusions more precise regarding the choice of  $\sigma_u$ .

## 4.2 Stochastic volatility model with leverage

Consider the problem of modelling the volatility in the daily closing prices of the NASDAQ OMXS30 index, i.e. a weighted average of the 30 most traded stocks at the Stockholm stock exchange. We extract the  $T = 747$  log-returns using Quandl<sup>1</sup> for the period January 2, 2011 and January 2, 2014. To model the underlying volatility, we make use of a stochastic volatility model with leverage given by

$$x_0 \sim \mathcal{N}\left(x_0; \mu, \frac{\sigma_v^2}{(1 - \phi^2)^2}\right), \quad \begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} \Big| x_t \sim \mathcal{N}\left(\begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix}; \begin{bmatrix} \mu + \phi(x_t - \mu) \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_v^2 & \rho \\ \rho & \exp(x_t) \end{bmatrix}\right), \quad (17)$$

with parameters  $\theta = \{\mu, \phi, \sigma_v, \rho\}$ . In this model, we would like to infer the parameter posterior of  $\theta$  given the data using Algorithm 1. Here, the potential function corresponds to the log-likelihood estimator using the particle filter as discussed in Appendix A.2. The aim is to again compute the IACT and compare with the optimal theoretical results from Section 3.

In Figure 4, we present the median acceptance rate and IACT for the four parameters in the model when varying  $\sigma_u \in \{0.05, 0.10, \dots, 1.00\}$ . The maximum IACT (over the four variables) is minimised when  $\sigma_u = 0.55$ . However, the variation in the IACT is significant and a range between 0.4 and 0.8 seems

<sup>1</sup>The data is available for download from: <https://www.quandl.com/data/NASDAQOMX/OMXS30>.

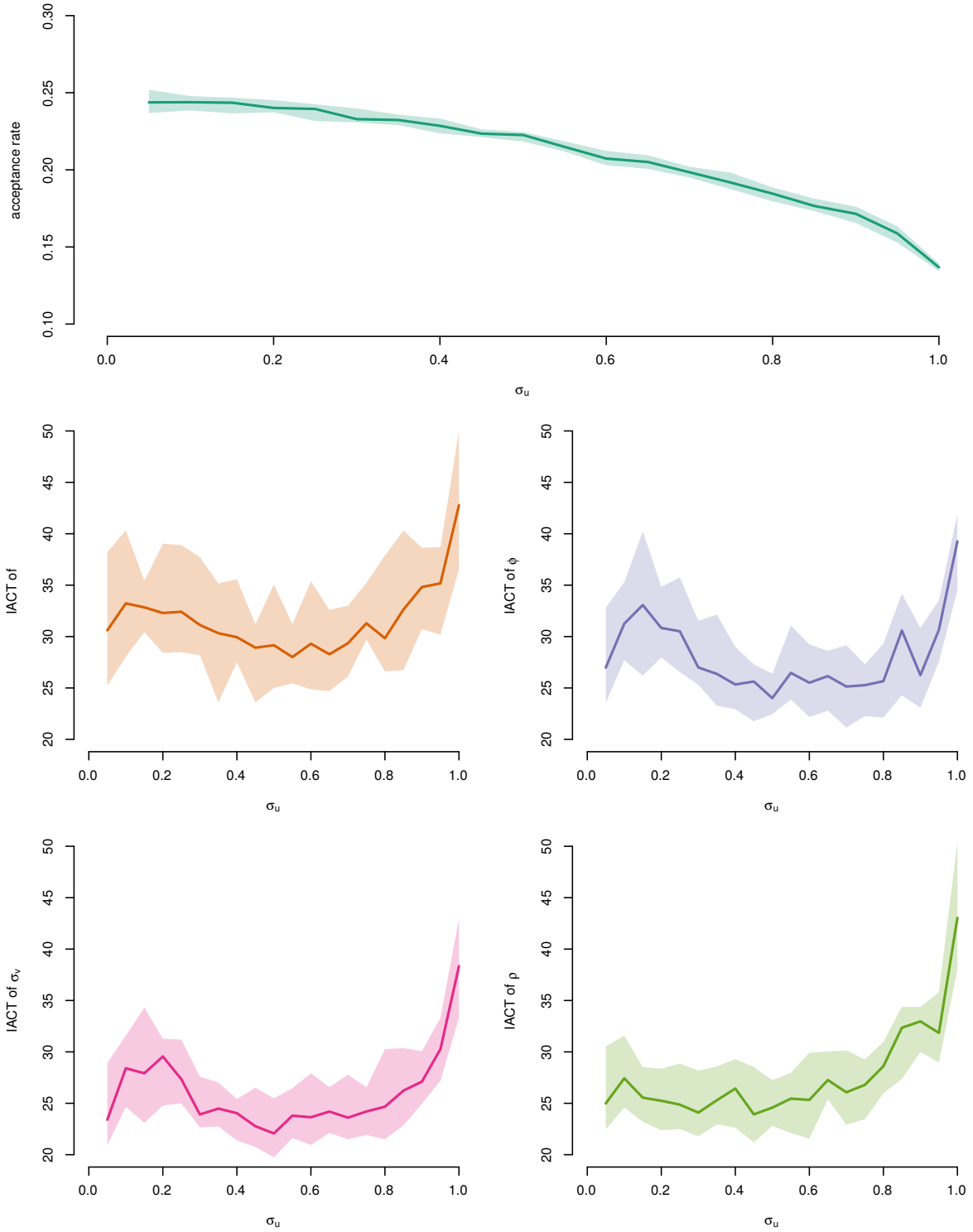


Figure 4: The acceptance probability (upper) and the resulting IACT (middle and lower) for  $\mu$  (orange),  $\phi$  (purple),  $\sigma_v$  (magenta) and  $\rho$  (light green) when varying  $\sigma_u$ . The results presented are the median (line) and first and third quantiles (shaded area) from 32 independent Monte Carlo runs.

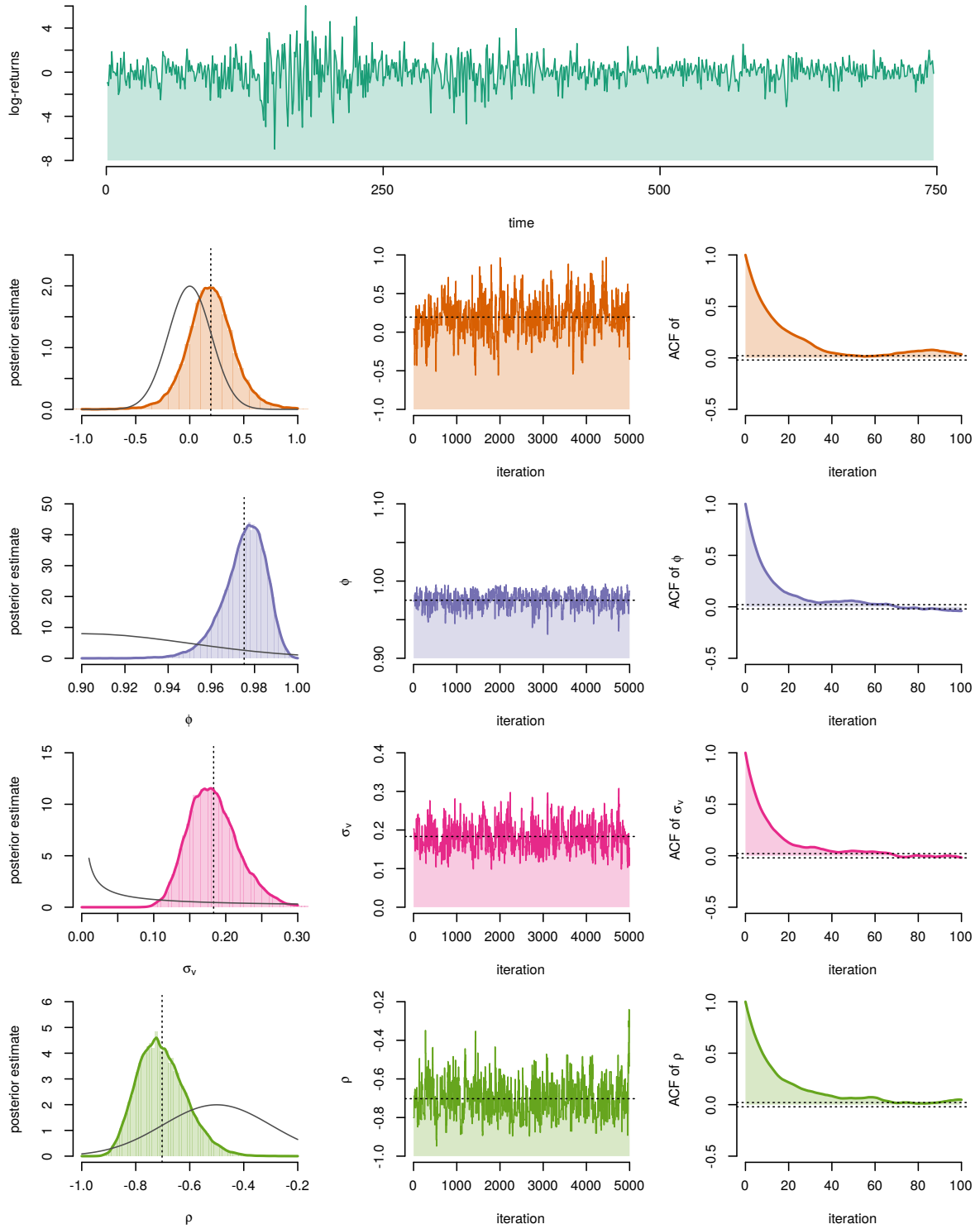


Figure 5: Upper: the closing log-returns for the NASDAQ OMXS30 index between January 2, 2011 and January 2, 2014. Lower: the parameter trace (left), autocorrelation function (center) and resulting posterior estimate (right) for the SV model (17) for  $\mu$  (orange),  $\phi$  (purple),  $\sigma_v$  (magenta) and  $\rho$  (light green). We make use of  $\sigma_u = 0.55$  in the CN proposal (3) and the histograms are computed using the output from 32 independent Monte Carlo runs.

as suitable choices for  $\sigma_u$ . The standard deviation of the log-likelihood estimator (around the estimated posterior mean) is 1.2, which by Section 3 would imply an optimal  $\sigma_z$  of around 0.9. This is clearly not an appropriate choice for this model. This can be due to that the Gaussian assumption for  $\Phi_\theta(u)$  is not fulfilled or that the standard deviation of the log-likelihood estimate varies with  $\theta$ .

From Figure 4, we conclude that using  $\sigma_u < 1$  results in a decrease of the IACT comparing with using an independent proposal for  $u$ . The improvement in the mixing is about 1.5 times. We present a specific case in Figure (5), where we fix  $\sigma_u = 0.55$ . We conclude that the chains are mixing well and the posterior estimates are reasonable with the estimated posterior mean  $\theta = \{0.19, 0.98, 0.18, -0.70\}$  with standard deviations  $\{0.22, 0.01, 0.04, 0.09\}$ .

## 5 Conclusions and future work

The numerical illustrations in Section 4 indicate that we can obtain improvements in the mixing of the pmMH algorithm by introducing correlation in  $u$ . In this paper, we consider using a CN proposal for introducing the correlation and it seems that selecting  $\sigma_u = 0.5$  offers good performance in the settings that we have considered. Hence, implementing the proposed alterations to the pmMH algorithm only requires changing a small number of lines of code and hopefully does not introduce any additional tuning parameters for the user. We are currently working on an adaptive method to further alleviate the work of tuning  $\sigma_u$  for the user. The main benefit of our proposed modifications of the pmMH algorithm is that we can decrease the number of particles  $N_u$  and obtain better mixing at the same time. Hence, this results in a significant decrease in the computational cost for Bayesian inference based on the pseudo-marginal framework.

Some important additional future work is to extend the theoretical analysis in Section 3 to a more realistic setting, where the influence of  $\theta$  also is taken into the account. Also it would be interesting to make use of Hilbert curve resampling proposed by Gerber and Chopin [2015] or some tree methods discussed by Lee [2008] in the particle filter. This would allow for inference in state space models with a multivariate state process.

It is also possible to make use of  $u$  to construct estimates of the gradient and Hessian of the log-posterior. This information can be included into the proposal for  $\theta$  as discussed by Dahlin et al. [2015a,b] to further improve the mixing in the Markov chain. The gradient information can also be used to reduce the variance in a post-processing step as discussed by Mira et al. [2013].

The proposed alterations of the pmMH algorithm can also be useful in models where the likelihood is intractable as discussed by e.g. Jasra [2015] and Dahlin et al. [2015b]. In this class of models, approximate Bayesian computations (ABC; Marin et al., 2012) can be used to approximate the log-likelihood using importance sampling and particle filtering. However, in practice these estimates suffer from a large



variance with results in bad mixing for the Markov chain. This approach can possible result in a large decrease in the computational cost for using the pmMH algorithm for inference in models with intractable likelihoods.

## **Acknowledgements**

This work was supported by: the Swedish Foundation for Strategic Research (SSF) through grant IIS11-0081, the project Probabilistic modeling of dynamical systems (Contract number: 621-2013-5524) funded by the Swedish Research Council and CADICS, a Linnaeus Center also funded by the Swedish Research Council. The simulations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at Linköping University, Sweden.

## References

- C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.
- C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.
- C. Andrieu, A. Doucet, and A. Lee. Discussion on constructing summary statistics for approximate Bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):451–452, 2012.
- A. Beskos, G. Roberts, A. Stuart, and J. Voss. MCMC methods for diffusion bridges. *Stochastics and Dynamics*, 8(03):319–350, 2008.
- S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White. MCMC methods for functions: modifying old algorithms to make them faster. *Statistical Science*, 28(3):424–446, 2013.
- J. Dahlin, F. Lindsten, and T. B. Schön. Particle Metropolis-Hastings using gradient and Hessian information. *Statistics and Computing*, 25(1):81–92, 2015a.
- J. Dahlin, F. Lindsten, and T. B. Schön. Quasi-Newton particle Metropolis-Hastings. In *Proceedings of the 17th IFAC Symposium on System Identification (SYSID)*, Beijing, China, October 2015b.
- J. Dahlin, M. Villani, and T. B. Schön. Efficient approximate Bayesian inference for models with intractable likelihoods. *Pre-print*, 2015c. arXiv:1506.06975v1.
- P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.
- G. Deligiannidis, A. Doucet, M. K. Pitt, and R. Kohn. The Correlated Pseudo-Marginal Method. *Pre-print*, 2015. arXiv:1511.04992v1.
- A. Doucet and A. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky, editors, *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011.
- A. Doucet, M. K. Pitt, G. Deligiannidis, and R. Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 102(2):295–313, 2015.
- P. Fearnhead and L. Meligkotsidou. Augmentation schemes for particle MCMC. *Statistics and Computing (in press)*, pages 1–14, 2015.

- T. Flury and N. Shephard. Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models. *Econometric Theory*, 27(5):933–956, 2011.
- A. Gelman, G. Roberts, and W. Gilks. Efficient Metropolis jumping rules. *Bayesian statistics*, 5:599–607, 1996.
- M. Gerber and N. Chopin. Sequential quasi Monte Carlo. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(3):509–579, 2015.
- T. Hachisuka, A. S. Kaplanyan, and C. Dachsbacher. Multiplexed Metropolis light transport. *ACM Transactions on Graphics (TOG)*, 33(4):100, 2014.
- M. Hairer, A. M. Stuart, and S. J. Vollmer. Spectral gaps for a Metropolis-Hastings algorithm in infinite dimensions. *The Annals of Applied Probability*, 24(6):2455–2490, 2014.
- W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- A. Jasra. Approximate Bayesian Computation for a Class of Time Series Models. *International Statistical Review*, 83(3):405–435, 2015.
- C. Kelemen, L. Szirmay-Kalos, G. Antal, and F. Csonka. A simple and robust mutation strategy for the Metropolis light transport algorithm. *Computer Graphics Forum*, 21(3):531–540, 2002.
- J. G. Kemeny and J. L. Snell. *Finite Markov chains*. Springer, New York, 1976.
- P. E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*, volume 23. Springer, 4 edition, 1992.
- J. Kronander and T. B. Schön. Robust auxiliary particle filters using multiple importance sampling. In *Proceedings of the 2014 IEEE Statistical Signal Processing Workshop (SSP)*, Gold Coast, Australia, July 2014.
- A. Lee. *Towards smooth particle filters for likelihood estimation with multivariate latent variables*. PhD thesis, University of British Columbia, 2008.
- A. Lee and C. Holmes. Discussion on Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):327–328, 2010.
- S. Malik and M. K. Pitt. Particle filters for continuous likelihood evaluation and maximisation. *Journal of Econometrics*, 165(2):190–209, 2011.
- J.-M. Marin, P. Pudlo, C. P. Robert, and R. J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Cambridge University Press, 2009.
- A. Mira, R. Solgi, and D. Imparato. Zero variance Markov chain Monte Carlo for Bayesian estimators. *Statistics and Computing*, 23(5):653–662, 2013.
- A. Owen and Y. Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.
- P. H. Peskun. Optimum Monte-carlo sampling using Markov chains. *Biometrika*, 60(3):607–612, 1973.
- M. K. Pitt, R. S. Silva, P. Giordani, and R. Kohn. On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2):134–151, 2012.
- C. P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer, 2 edition, 2004.
- S. M. Ross. *Simulation*. Academic Press, 5 edition, 2012.
- C. Sherlock, A. H. Thiery, G. O. Roberts, and J. S. Rosenthal. On the efficiency of pseudo-marginal random walk Metropolis algorithms. *The Annals of Statistics*, 43(1):238–275, 2015.
- L. Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1701–1728, 1994.
- M.-N. Tran, M. Scharth, M. K. Pitt, and R. Kohn. Importance Sampling Squared for Bayesian Inference in Latent Variable Models. *Pre-print*, 2014. arXiv:1309.3339v3.
- E. Veach and L. J. Guibas. Metropolis light transport. In *Proceedings of the 24th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 65–76, Los Angeles, CA, USA, August 1997.
- Z. Yang and Carlos E Rodríguez. Searching for efficient Markov chain Monte Carlo proposal kernels. *Proceedings of the National Academy of Sciences*, 110(48):19307–19312, 2013.

## A Implementation details

In this appendix, we outline the implementation details of the numerical illustrations presented in Section 4. The implementations for estimating the log-likelihood are based on standard importance sampling and particle filtering. Extensive treatments of these methods are found in e.g. Doucet and Johansen [2011] and Robert and Casella [2004].

### A.1 Gaussian IID model

We make use of an importance sampler with  $N = 10$  to estimate the log-likelihood with the prior as the importance distribution. This results in that the log-likelihood can be estimated by

$$\log \hat{p}_\theta(y; u) = \sum_{t=1}^T \log \left[ \sum_{i=1}^N w_t^{(i)} \right] - T \log N, \quad (18)$$

where the weights  $w_t^{(i)}$  are generated using the procedure outlined in Algorithm 2.

For pmMH, we use  $K = 10,000$  iterations (discarding the first  $K_b = 1,000$  as burn-in) in Algorithm 1 and initialise in the true parameter  $\theta_0 = 0.5$ . The proposal for  $\theta$  is a standard Gaussian random walk proposal (3) with  $\mu(\theta, u) = \theta$  and  $\Sigma(\theta, u) = 0.10^2$ . Finally, we make use of the following prior

$$p(\mu) \sim \mathcal{TN}_{(0,1)}(\mu; 0, 1),$$

where  $\mathcal{TN}_{(-1,1)}(\mu; 0, 1)$  denotes a standard Gaussian distribution truncated to the interval  $(-1, 1)$ .

### A.2 Stochastic volatility model with leverage

For any SSM, we make use of a bootstrap particle filter (bPF; Doucet and Johansen, 2011) to estimate the log-likelihood. An SSM with latent states  $x_{0:T} = \{x_t\}_{t=0}^T$  and observations  $y_{1:T}$  is given by

$$x_0 \sim \mu_\theta(x_0), \quad x_{t+1}|x_t \sim f_\theta(x_{t+1}|x_t), \quad y_t|x_t \sim g_\theta(y_t|x_t), \quad (19)$$

where  $\theta \in \Theta \subseteq \mathbb{R}^p$  denotes the static unknown parameters. Here, we assume that it is possible to simulate from the distributions  $\mu_\theta(x_0)$  and  $f_\theta(x_{t+1}|x_t)$  and evaluate  $g_\theta(y_t|x_t)$  point-wise. A bPF to estimate  $\hat{p}_\theta(y; u)$  is presented in Algorithm 3.

Hence, the log-likelihood is computed using the same expression (18) as for importance sampling but the particles  $x_t^{(i)}$  are instead generated by sequential importance sampling with resampling. Note that we are required to sort the particles after the propagation step, see the discussion about smooth particle filter by Malik and Pitt [2011] for more details. We make use of the probability transform to generate the uniform random variables required for the systematic resampling step. Hence,  $u$  is a  $(N_u + 1) \cdot (T + 1)$ -

variate Gaussian random variable, where  $u_{2:N+1,t}$  is used directly in the propagation step and  $u_{1,t}$  is used in the resampling step after a transformation into a uniform random variable. Here, we make use of  $N_u = 50$  particles.

For pmMH, we use  $K = 10,000$  iterations (discarding the first  $K_b = 1,000$  as burn-in) in Algorithm 1 and initialise the parameters at  $\theta_0 = \{0.23, 0.98, 0.18, -0.72\}$  obtained using the approach discussed by Dahlin et al. [2015c]. The proposal for  $\theta$  is a standard Gaussian random walk proposal (3) with  $\mu(\theta, u) = \theta$ . Here, we make use of the rules of thumb by Sherlock et al. [2015] to select the covariance function. We extract an estimate of the Hessian using the method by Dahlin et al. [2015c] and set

$$\Sigma(\theta, u) = \frac{2.562^2}{p} \cdot 10^{-4} \cdot \begin{bmatrix} 384 & 3 & -5 & -16 \\ 3 & 1 & -3 & -2 \\ -5 & -3 & 12 & 3 \\ -16 & -2 & 3 & 65 \end{bmatrix}.$$

Finally, we use the following prior distributions

$$\begin{aligned} p(\mu) &\sim \mathcal{N}(\mu; 0, 2^2), & p(\phi) &\sim \mathcal{TN}_{(-1,1)}(\phi; 0.9, 0.05^2), \\ p(\sigma_v) &\sim \mathcal{G}(\sigma_v; 2.0, 0.05), & p(\rho) &\sim \mathcal{N}(\rho; -0.5, 0.2^2), \end{aligned}$$

where  $\mathcal{G}(a, b)$  denotes the Gamma distribution with mean  $a/b$ .

---

**Algorithm 2** Likelihood estimation using importance sampling with fixed random numbers

---

INPUTS:  $y$  (vector of  $T$  observations),  $N_u \in \mathbb{N}$  (no. samples) and  $u \in \mathcal{U}$  ( $T \cdot N_u$  standard Gaussian random variables).

OUTPUTS:  $\hat{p}_\theta(y; u)$  (est. of the likelihood).

NOTE: all operations are carried out over  $i = 1, \dots, N_u$ .

---

- 1: **for**  $t = 1$  to  $T$  **do**
- 2: Simulate from the proposal by

$$x_t^{(i)} = \mu + \sigma_v^2 u_t^{(i)},$$

using random variables  $u_{1:N_u, t}$ .

- 3: Calculate the weights by

$$w_t^{(i)} = \mathcal{N}(y_t; x_t^{(i)}, \sigma_e^2).$$

- 4: **end for**
  - 5: Estimate  $\hat{p}_\theta(y; u)$  by (18).
- 

---

**Algorithm 3** Likelihood estimation using particle filtering with fixed random numbers

---

INPUTS:  $y$  (vector of  $T$  observations), an SSM (19),  $N_u \in \mathbb{N}$  (no. particles) and  $u \in \mathcal{U}$  ( $(T+1) \cdot (N_u + 1)$  standard Gaussian random variables).

OUTPUTS:  $\hat{p}_\theta(y; u)$  (est. of the likelihood).

NOTE: all operations are carried out over  $i, j = 1, \dots, N_u$ .

---

- 1: Sample  $x_0^{(i)} \sim \mu_\theta(x_0)$  using  $u_{2:(N_u+1), 1}$ .
- 2: Set  $W_0^{(i)} = N_u^{-1}$  as initial weights.
- 3: **for**  $t = 1$  to  $T$  **do**
- 4: Apply the inverse CDF approach to transform  $u_{1, t+1}$  into a uniform random number  $\bar{u}_{1, t+1}$ .
- 5: Apply systematic resampling with  $\bar{u}_{1, t+1}$  to sample the ancestor index  $a_t^{(i)}$  from a multinomial distribution with

$$\mathbb{P}(a_t^{(i)} = j) = W_{t-1}^{(j)}.$$

- 6: Propagate the particles by sampling

$$x_t^{(i)} \sim f_\theta\left(x_t^{(i)} | x_{t-1}^{a_t^{(i)}}\right),$$

using random variables  $u_{2:(N_u+1), t+1}$ .

- 7: Extend the trajectory by  $x_{0:t}^{(i)} = \{x_{0:t-1}^{a_t^{(i)}}, x_t^{(i)}\}$ .
  - 8: Sort the particle trajectories according to the current state  $x_t^{(i)}$ .
  - 9: Calculate the particle weights by  $w_t^{(i)} = g_\theta(y_t | x_t^{(i)})$  which by normalisation (over  $i$ ) gives  $W_t^{(i)}$ .
  - 10: **end for**
  - 11: Estimate  $\hat{p}_\theta(y; u)$  by (18).
-