Particle Metropolis-Hastings using gradient and Hessian information

Johan Dahlin, Fredrik Lindsten and Thomas Schön

Particle Metropolis-Hastings using gradient and Hessian information[*]

Johan Dahlin, Fredrik Lindsten and Thomas B. Schön[†]

September 24, 2014

Abstract

Particle Metropolis-Hastings (PMH) allows for Bayesian parameter inference in nonlinear state space models by combining Markov chain Monte Carlo (MCMC) and particle filtering. The latter is used to estimate the intractable likelihood. In its original formulation, PMH makes use of a marginal MCMC proposal for the parameters, typically a Gaussian random walk. However, this can lead to a poor exploration of the parameter space and an inefficient use of the generated particles.

We propose a number of alternative versions of PMH that incorporate gradient and Hessian information about the posterior into the proposal. This information is more or less obtained as a byproduct of the likelihood estimation. Indeed, we show how to estimate the required information using a fixed-lag particle smoother, with a computational cost growing linearly in the number of particles. We conclude that the proposed methods can: (i) decrease the length of the burn-in phase, (ii) increase the mixing of the Markov chain at the stationary phase, and (iii) make the proposal distribution scale invariant which simplifies tuning.

[*]The final publication is available at Springer via:
`http://dx.doi.org/10.1007/s11222-014-9510-0`

[†]This work was supported by: Learning of complex dynamical systems (Contract number: 637-2014-466) and Probabilistic modeling of dynamical systems (Contract number: 621-2013-5524) and CADICS, a Linnaeus Center, all funded by the Swedish Research Council. JD is with the Division of Automatic Control, Linköping University, Linköping, Sweden. E-mail: `johan.dahlin@liu.se`. FL is with the Department of Engineering, University of Cambridge, Cambridge, United Kingdom. E-mail: `fredrik.lindsten@eng.cam.ac.uk`. TS is with Division of Systems and Control, Uppsala University, Uppsala, Sweden. E-mail: `thomas.schon@it.uu.se`.

1 Introduction

We are interested in Bayesian parameter inference in nonlinear state space models (SSM) of the form

$$x_t|x_{t-1} \sim f_\theta(x_t|x_{t-1}), \quad y_t|x_t \sim g_\theta(y_t|x_t), \tag{1}$$

where the latent states and the measurements are denoted by $\mathbf{x} = x_{0:T} \triangleq \{x_t\}_{t=0}^T$ and $\mathbf{y} = y_{1:T} \triangleq \{y_t\}_{t=1}^T$, respectively. Here, $f_\theta(\cdot)$ and $g_\theta(\cdot)$ denote the transition and observation kernels, respectively, parametrised by the unknown static parameter vector $\theta \in \Theta \subset \mathbb{R}^d$. The initial state is distributed according to some distribution $\mu(x_0)$ which, for notational simplicity, is assumed to be independent of θ.

The aim of Bayesian parameter inference (in SSMs) is to compute the *parameter posterior distribution*

$$p(\theta|\mathbf{y}) = \frac{p_\theta(\mathbf{y})p(\theta)}{p(\mathbf{y})}, \tag{2}$$

where $p(\theta)$ denotes the prior of θ and $p_\theta(\mathbf{y})$ denotes the likelihood, which for an SSM can be expressed as

$$p_\theta(\mathbf{y}) = p_\theta(y_1) \prod_{t=2}^{T} p_\theta(y_t|y_{1:t-1}). \tag{3}$$

The one-step ahead predictor $p_\theta(y_t|y_{1:t-1})$, and thus also the likelihood function, is in general not analytically tractable. However, unbiased estimators of the likelihood can be constructed using sequential Monte Carlo (SMC) [Doucet and Johansen, 2011, Del Moral, 2004] and these can be used as *plug-in estimators*. This is especially useful in the Metropolis-Hastings (MH) algorithm that can be used for estimating the parameter posterior in (2).

This combination of MH and SMC is known as the particle Metropolis-Hastings (PMH) algorithm [Andrieu et al., 2010]. The MH acceptance probability depends on the intractable likelihood, which in PMH is estimated using SMC (see Section 2). Despite the apparent approximation, this results in an algorithm that targets the correct posterior distribution [Andrieu et al., 2010]. The original PMH algorithm makes use of a marginal proposal for θ, i.e. only the current parameter is used when proposing a new parameter. The theoretical properties of the marginal PMH algorithm have been analysed in Andrieu and Vihola [2012], Pitt et al. [2012], Doucet et al. [2012] and it has been applied for a number of interesting applications in, e.g., economics, social network analysis and ecology [Flury and Shephard, 2011, Everitt, 2012, Golightly and Wilkinson, 2011].

In this paper, we show that information such as the gradient and the Hessian about the posterior can be included in the construction of the PMH proposal. This idea is first suggested by Doucet et al. [2011] in the discussions following Girolami and Calderhead [2011]. In two previous proceedings, we have applied and extended this idea with gradient information [Dahlin et al., 2013] and also using Hessian information [Dahlin et al., 2014]. The present article builds upon and extends this preliminary work. A PMH method using gradient information similar to Dahlin et al. [2013] has recently been proposed by Nemeth and Fearnhead [2014].

In the context of MH sampling, it has been recognised that the gradient and Hessian can be used to construct efficient proposal distributions. In the Metropolis adjusted Langevin algorithm (MALA) [Roberts and Stramer, 2003], a drift term is added to the proposal in the direction of the gradient, which intuitively guides the Markov chain to regions of high posterior probability. In the manifold MALA (mMALA) [Girolami and Calderhead, 2011], the Hessian (or some other appropriate metric tensor) is also included to scale the proposal to take the curvature of the log-posterior into account. Drawing parallels with the optimisation literature, mMALA shares some properties with Newton-type optimisation algorithms (where MALA is more similar to a steepest ascent method). In particular, scaling the proposal with the Hessian can considerably simplify the tedious tuning of the method since it removes the need for running costly pilot runs, which are commonly used to tune the covariance matrices of the random walk MH and the MALA.

In our problem, i.e. for inference in a nonlinear SSM (1), the gradient and Hessian cannot be computed analytically. However, in analogue with the intractable likelihood, these quantities can be estimated using SMC algorithms, see e.g. Poyiadjis et al. [2011], Doucet et al. [2013]. This provides us with the tools necessary to construct PMH algorithms in the flavour of the MALA and the mMALA, resulting in the two methods proposed in this paper, PMH1 and PMH2, respectively. In particular, we make use of a fixed-lag (FL) particle smoother [Kitagawa and Sato, 2001] to estimate the gradient and Hessian. The motivation for this is that this smoother only makes use of the weighted particles computed by the particle filter. Consequently, we obtain this information as a *byproduct* of the likelihood computation in the PMH algorithm. This results in only a small computational overhead for the proposed methods when compared to the marginal method.

Finally, we provide numerical experiments to illustrate the benefits of using the gradient and Hessian and the accuracy of the FL smoother. We demonstrate some interesting properties of the proposed algorithms, in particular that they enjoy (i) a shorter burn-in compared with the marginal algorithm, (ii) a better mixing of the Markov chain in the stationary phase, and (iii) a simplified tuning of the step length(s), especially when the target distribution is non-isotropic.

2 Particle Metropolis-Hastings

In this section, we review the PMH algorithm and show how the random variables used to compute the likelihood estimator can be incorporated in the proposal construction. We also outline the idea of how this can be used to construct the proposed PMH1 and PMH2 algorithms.

2.1 MH sampling with unbiased likelihoods

The MH algorithm (see, e.g. Robert and Casella [2004]) is a member of the MCMC family for sampling from a target distribution $\pi(\theta)$ by simulating a carefully constructed Markov chain on Θ. The chain is constructed in such a way that it admits the target as its unique stationary distribution.

The algorithm consists of two steps: (i) a new parameter θ'' is sampled from a proposal distribution $q(\theta''|\theta')$ given the current state θ' and (ii) the current parameter is changed to θ'' with probability $\alpha(\theta', \theta'')$, otherwise the chain remains at the current state. The acceptance probability is given by

$$\alpha(\theta', \theta'') = 1 \wedge \frac{\pi(\theta'')}{\pi(\theta')} \frac{q(\theta'|\theta'')}{q(\theta''|\theta')}, \tag{4}$$

where we use the notation $a \wedge b \triangleq \min\{a, b\}$.

In this paper, we have the parameter posterior distribution (2) as the target distribution, i.e. $\pi(\theta) = p(\theta|\mathbf{y})$. This implies that the acceptance probability (4) will depend explicitly on the intractable likelihood $p_\theta(\mathbf{y})$, preventing direct application of the MH algorithm to this problem. However, this difficulty can be circumvented by using a *pseudo-marginal* approach [Beaumont, 2003, Andrieu and Roberts, 2009].

Assume that there exists an unbiased, non-negative estimator of the likelihood $\widehat{p}_\theta(\mathbf{y}|u)$. We introduce explicitly the random variable $u \in \mathsf{U}$ used to construct this estimator, and we let $m_\theta(u)$ denote the probability density of u on U.

The pseudo-marginal method is then a standard MH algorithm operating in a non-standard extended space $\Theta \times \mathsf{U}$, with the *extended target*

$$\pi(\theta, u|\mathbf{y}) = \frac{\widehat{p}_\theta(\mathbf{y}|u) m_\theta(u) p(\theta)}{p(\mathbf{y})} = \frac{\widehat{p}_\theta(\mathbf{y}|u) m_\theta(u) p(\theta|\mathbf{y})}{p_\theta(\mathbf{y})},$$

and proposal distribution $m_{\theta''}(u'')q(\theta''|\theta')$.

Since the likelihood estimator is unbiased, $\mathbb{E}_{u|\theta}[\widehat{p}_\theta(\mathbf{y}|u)] = p_\theta(\mathbf{y})$, it follows that the extended target admits $p(\theta|\mathbf{y})$ as a marginal. Hence, by simulating from the extended target $\pi(\theta, u|\mathbf{y})$ we obtain samples from the original target distribution $p(\theta|\mathbf{y})$ as a byproduct.

If the likelihood is estimated by using SMC (see Section 3) we obtain the PMH algorithm. The random variable u then corresponds to all the weighted particles generated by the SMC algorithm. However, these random variables carry useful information, not only about the likelihood, but also about the geometry of the posterior distribution. We suggest to incorporate this information into the proposal construction. With (θ', u') being the current state of the Markov chain we simulate $\theta'' \sim q(\cdot|\theta', u')$ and $u'' \sim m_{\theta''}(\cdot)$, using some proposal q (see Section 2.2).

It follows that the (standard) MH acceptance probability for the extended target is given by

$$\begin{aligned}\alpha(\theta'', u'', \theta', u') &= 1 \wedge \frac{\widehat{p}_{\theta''}(\mathbf{y}|u'')m_{\theta''}(u'')p(\theta'')}{\widehat{p}_{\theta'}(\mathbf{y}|u')m_{\theta'}(u')p(\theta')} \frac{m_{\theta'}(u')q(\theta'|\theta'', u'')}{m_{\theta''}(u'')q(\theta''|\theta', u')} \\ &= 1 \wedge \frac{\widehat{p}_{\theta''}(\mathbf{y}|u'')p(\theta'')}{\widehat{p}_{\theta'}(\mathbf{y}|u')p(\theta')} \frac{q(\theta'|\theta'', u'')}{q(\theta''|\theta', u')}. \end{aligned} \tag{5}$$

Note that $q(\theta''|\theta', u')$ may depend on the auxiliary variable u' in a (formally) arbitrary way. In particular, in Section 3 we propose a construction making use of *biased* estimates of the gradient and Hessian of the log-posterior. Nevertheless, expression (5) still defines a correct MH acceptance probability for the extended target, ensuring the validity of our approach. Note also that the aforementioned proposal construction opens up for a wide range of adapted proposals, possibly different from the ones considered in this work.

2.2 Constructing PMH1 and PMH2

We now turn to the construction of a proposal that makes use of the gradient and Hessian of the log-posterior. Following Robert and Casella [2004], we do this by a Laplace approximation of the parameter posterior around the current state θ'. Hence, consider a second order Taylor expansion of $\log p(\theta''|\mathbf{y})$ at θ':

$$\begin{aligned}\log p(\theta''|\mathbf{y}) \approx \log p(\theta'|\mathbf{y}) &+ (\theta'' - \theta')^{\top} \Big[\nabla \log p(\theta|\mathbf{y})\Big]_{\theta=\theta'} \\ &+ \frac{1}{2}(\theta'' - \theta')^{\top} \Big[\nabla^2 \log p(\theta|\mathbf{y})\Big]_{\theta=\theta'} (\theta'' - \theta').\end{aligned}$$

Taking the exponential of both sides and completing the square, we obtain

$$p(\theta''|\mathbf{y}) \approx \mathsf{N}\Big(\theta''; \theta' + \mathsf{I}_T^{-1}(\theta')\mathsf{S}_T(\theta'), \mathsf{I}_T^{-1}(\theta')\Big),$$

where we have introduced $\mathsf{S}_T(\theta') = \nabla \log p(\theta|\mathbf{y})|_{\theta=\theta'}$ and $\mathsf{I}_T(\theta') = -\nabla^2 \log p(\theta|\mathbf{y})|_{\theta=\theta'}$, for the gradient and the negative Hessian of the log-posterior, respectively. Here, we assume for now that the negative Hessian is positive definite; see Section 3.5 for further discussion on this matter.

As pointed out above, these quantities cannot be computed in closed form, but they can be estimated from the random variable u' (see Section 3). This suggests three different versions of the PMH algorithm, each resulting from a specific choice of the proposal:

$$q(\theta''|\theta', u') = \begin{cases} \mathsf{N}\left(\theta', \Gamma\right), & \text{[PMH0]} \\ \mathsf{N}\left(\theta' + \frac{1}{2}\Gamma\widehat{\mathsf{S}}_T(\theta'|u'), \Gamma\right), & \text{[PMH1]} \\ \mathsf{N}\left(\theta' + \widehat{\mathsf{G}}(\theta'|u'), \widehat{\mathsf{H}}(\theta'|u')\right). & \text{[PMH2]} \end{cases} \tag{6}$$

Here, we use the notation $\widehat{\mathsf{G}}(\theta|u) = \frac{1}{2}\Gamma\,\widehat{\mathsf{I}}_T^{-1}(\theta|u)\,\widehat{\mathsf{S}}_T(\theta|u)$ and $\widehat{\mathsf{H}}(\theta|u) = \Gamma\,\widehat{\mathsf{I}}_T^{-1}(\theta|u)$ for the natural gradient and scaled inverse Hessian, respectively. Furthermore,

Algorithm 1 Second order particle Metropolis-Hastings

INPUTS: Algorithm 2. $M > 0$ (no. MCMC steps), θ_0 (initial parameters), γ (step length).
OUTPUT: $\theta = \{\theta_1, \ldots, \theta_M\}$ (samples from the posterior).

```
1:  Run Algorithm 2 to obtain $\widehat{p}_{\theta_0}(\mathbf{y})$, $\widehat{\mathsf{S}}_T(\theta_0)$ and $\widehat{\mathsf{I}}_T(\theta_0)$.
2:  for $k = 1$ to $M$ do
3:      Sample $\theta' \sim q(\theta' | \theta_{k-1}, u_{k-1})$ by (6), $\widehat{\mathsf{S}}_T(\theta_{k-1})$ and $\widehat{\mathsf{I}}_T(\theta_{k-1})$.
4:      Run Algorithm 2 to obtain $\widehat{p}_{\theta'}(\mathbf{y})$, $\widehat{\mathsf{S}}_T(\theta')$ and $\widehat{\mathsf{I}}_T(\theta')$.
5:      Sample $\omega_k$ uniformly over $[0, 1]$.
6:      if $\omega_k < \alpha(\theta', u', \theta_{k-1}, u_{k-1})$ given by (5) then
7:          $\theta_k \leftarrow \theta'$. {Accept the parameter}
8:          $\{\widehat{p}_{\theta_k}(\mathbf{y}), \widehat{\mathsf{S}}_T(\theta_k), \widehat{\mathsf{I}}_T(\theta_k)\} \leftarrow \{\widehat{p}_{\theta'}(\mathbf{y}), \widehat{\mathsf{S}}_T(\theta'), \widehat{\mathsf{I}}_T(\theta')\}$.
9:      else
10:         $\theta_k \leftarrow \theta_{k-1}$. {Reject the parameter}
11:         $\{\widehat{p}_{\theta_k}(\mathbf{y}), \widehat{\mathsf{S}}_T(\theta_k), \widehat{\mathsf{I}}_T(\theta_k)\} \leftarrow \{\widehat{p}_{\theta_{k-1}}(\mathbf{y}), \widehat{\mathsf{S}}_T(\theta_{k-1}), \widehat{\mathsf{I}}_T(\theta_{k-1})\}$.
12:     end if
13: end for
```

Γ denotes a scaling matrix that controls the step lengths of the proposal. For PMH0 and PMH1, Γ can be chosen as the inverse of an estimate of the posterior covariance matrix. However, computing this estimate typically requires costly and tedious trial runs. For PMH2, the curvature of the problem is captured by the Hessian matrix, i.e. a single step length can by used which can significantly simplify the tuning. It is also possible to choose different step lengths for the drift term and for the covariance matrix of the proposal.

The final PMH2 algorithm is presented in Algorithm 1. It makes use of Algorithm 2, described in Section 3, to estimate the quantities needed for computing the proposal and the acceptance probability. Clearly, PMH0 and PMH1 are special cases obtained by using the corresponding proposal from (6) in the algorithm. Note that, while the algorithm make explicit reference to the auxiliary variable u, it only depends on this variable through the estimates $\widehat{p}_{\theta'}(\mathbf{y})$, $\widehat{\mathsf{S}}_T(\theta')$ and $\widehat{\mathsf{I}}_T(\theta')$.

2.3 Properties of the PMH1 and PMH2 proposals

In the sequel, we use a single step size $\Gamma = \gamma^2 I_d$ for all the parameters in the (standard) proposal. This is done to illustrate the advantage of adding the Hessian information, which rescales the step lengths according to the local curvature. Hence, it allows for taking larger steps when the curvature is small and vice verse.

This property of PMH2 makes the algorithm scale-free in the same manner as a Newton algorithm in optimisation [Nocedal and Wright, 2006, Chapter 3]. That is, the proposal is invariant to affine transformations of the parameters.

Note that, since the local information is used, this is different from scaling the proposal in PMH0 with the posterior covariance matrix estimated from a pilot run, as this only takes the geometry at the mode of the posterior into account.

Some analyses of the statistical properties are available for PMH0 [Sherlock et al., 2013], MH using a random walk [Roberts et al., 1997] and MALA [Roberts and Rosenthal, 1998]. It is known from these analyses that adding the gradient into the proposal can increase the mixing of the Markov chain. Note that these results are obtained under somewhat strict assumptions. Also, we know from numerical experiments [Girolami and Calderhead, 2011] that there are further benefits of also taking the local curvature into account.

3 Estimation of the likelihood, gradient, and Hessian

In this section, we show how to estimate the likelihood together with the gradient and Hessian using SMC methods.

3.1 Auxiliary particle filter

An auxiliary particle filter (APF) [Pitt and Shephard, 1999] can be used to approximate the sequence of joint smoothing distributions (JSDs) $p_\theta(x_{1:t}|y_{1:t})$ for $t = 1$ to T. The APF makes use of a particle system consisting of N weighted particles $\{x_{1:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$ to approximate the JSD at time t by

$$\widehat{p}_\theta(\,\mathrm{d}x_{1:t}|y_{1:t}) \triangleq \sum_{i=1}^{N} \frac{w_t^{(i)}}{\sum_{k=1}^N w_t^{(k)}} \delta_{x_{1:t}^{(i)}}(\mathrm{d}x_{1:t}). \tag{7}$$

Here, $\delta_z(\mathrm{d}x_{1:t})$ denotes the Dirac measure placed at z. The particle system is propagated from $t-1$ to t by first sampling an *ancestor index* $a_t^{(i)}$, with

$$\mathbb{P}(a_t^{(i)} = j) = \nu_{t-1}^{(j)} \left[\sum_{k=1}^{N} \nu_{t-1}^{(k)} \right]^{-1}, \quad i, j = 1, \ldots, N, \tag{8}$$

where $\nu_{t-1}^{(i)}$ denotes the resampling weights. Given the ancestor index, a new particle is sampled according to

$$x_t^{(i)} \sim R_\theta\Big(x_t | x_{1:t-1}^{a_t^{(i)}}, y_t\Big), \quad i = 1, \ldots, N. \tag{9}$$

Finally, we append the obtained sample to the trajectory by $x_{1:t}^{(i)} = \{x_{1:t-1}^{a_t^{(i)}}, x_t^{(i)}\}$ and compute a new importance weight by

$$w_t^{(i)} \triangleq \frac{w_{t-1}^{a_t^{(i)}}}{\nu_{t-1}^{a_t^{(i)}}} \frac{g_\theta\Big(y_t \Big| x_t^{(i)}\Big) f_\theta\Big(x_t^{(i)} \Big| x_{t-1}^{a_t^{(i)}}\Big)}{R_\theta\Big(x_t^{(i)} \Big| x_{1:t-1}^{a_t^{(i)}}, y_t\Big)}, \quad i = 1, \ldots, N. \tag{10}$$

Hence, the empirical approximations of the smoothing distributions (7) can be computed sequentially for $t = 1$ to T by repeating (8)–(10).

Note that the random variables u appearing in the extended target of the PMH algorithm correspond to all the random variables generated by the APF, i.e. all the particles and ancestor indices,

$$u = \left(\left\{ x_t^{(i)}, a_t^{(i)} \right\}_{i=1}^{N}, t = 1, \ldots, T \right).$$

In this article, we make use of two important special cases of the APF: the bootstrap particle filter (bPF) [Gordon et al., 1993] and the fully adapted particle filter (faPF) [Pitt and Shephard, 1999]. For the bPF, we select the proposal kernel $R_\theta(x_t|x_{1:t-1}, y_t) = f_\theta(x_t|x_{t-1})$ and the auxiliary weights $\nu_t = w_t = g_\theta(y_t|x_t)$. The faPF is obtained by $R_\theta(x_t|x_{1:t-1}, y_t) = p_\theta(x_t|y_t, x_{t-1})$ and $\nu_t = p_\theta(y_{t+1}|x_t)$, resulting in the weights $w_t \equiv 1$. Note, that the faPF can only be used in models for which these quantities are available in closed-form.

3.2 Estimation of the likelihood

The likelihood for the SSM in (1) can be estimated using (3) by inserting estimated one-step predictors $p_\theta(y_t|y_{1:t-1})$ obtained from the APF. The resulting likelihood estimator is given by

$$\widehat{p}_\theta(\mathbf{y}|u) = \frac{1}{N^T} \sum_{i=1}^{N} w_T^{(i)} \left\{ \prod_{t=1}^{T-1} \sum_{i=1}^{N} \nu_t^{(i)} \right\}. \tag{11}$$

It is known that this likelihood estimator is unbiased for any number of particles, see e.g. [Pitt et al., 2012] and Proposition 7.4.1 in [Del Moral, 2004]. As discussed in Section 2.1, this is exactly the property that is needed in order to obtain $p(\theta|\mathbf{y})$ as the unique stationary distribution for the Markov chain generated by the PMH algorithm.

Consequently, PMH will target the correct distribution for any number of particles $N \geq 1$. However, the variance in the likelihood estimate is connected with the acceptance rate and the mixing of the Markov chain. Therefore it is important to determine the number of particles that balances a reasonable acceptance rate with a reasonable computational cost. This problem is studied for PMH0 in Pitt et al. [2012], Doucet et al. [2012].

3.3 Estimation of the gradient

As we shall see below, the gradient of the log-posterior can be estimated by solving a smoothing problem. The APF can be used directly to address this problem, since the particles $\{x_{1:T}^{(i)}, w_T^{(i)}\}_{i=1}^{N}$ provide an approximation of the JSD at time T according to (7) (see also Poyiadjis et al. [2011]). However, this method can give estimates with high variance due to *particle degeneracy*.

Instead, we make use of the FL smoother [Kitagawa and Sato, 2001] which has the same linear computational cost, but smaller problems with *particle degeneracy* than the APF. Alternative algorithms for estimating this information are also available [Del Moral et al., 2010, Poyiadjis et al., 2011].

The gradient of the parameter log-posterior is given by

$$\mathsf{S}_T(\theta) = \nabla \log p(\theta) + \nabla \log p_\theta(\mathbf{y}), \tag{12}$$

where it is assumed that the gradient of the log-prior $\nabla \log p(\theta)$ can be calculated explicitly. The gradient of the log-likelihood $\nabla \log p_\theta(\mathbf{y})$ can, using *Fisher's identity* [Cappé et al., 2005], be expressed as

$$\nabla \log p_\theta(\mathbf{y}) = \mathbb{E}_\theta \left[\nabla \log p_\theta(\mathbf{x}, \mathbf{y}) \Big| \mathbf{y} \right], \tag{13}$$

where for an SSM (1) we can write the gradient of the complete data log-likelihood as

$$\nabla \log p_\theta(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{T} \xi_\theta(x_t, x_{t-1}), \text{ where} \tag{14}$$
$$\xi_\theta(x_t, x_{t-1}) = \nabla \log f_\theta(x_t | x_{t-1}) + \nabla \log g_\theta(y_t | x_t).$$

Combining (14) with Fisher's identity (13) yields

$$\nabla \log p_\theta(\mathbf{y}) = \sum_{t=1}^{T} \int \xi_\theta(x_t, x_{t-1}) p_\theta(x_{t-1:t} | \mathbf{y}) \, \mathrm{d}x_{t-1:t},$$

which depends on the (intractable) two-step smoothing distribution $p_\theta(x_{t-1:t}|\mathbf{y})$. To approximate this quantity we use the FL smoother which relies on the assumption that there is a decaying influence of future observations $y_{t+\Delta:T}$ on the state x_t. This means that

$$p_\theta(x_{t-1:t}|\mathbf{y}) \approx p_\theta(x_{t-1:t}|y_{1:\kappa_t}),$$

holds for some large enough $\kappa_t = \min\{t + \Delta, T\}$. Here, Δ denotes a predetermined lag decided by the user, which depends on the forgetting properties of the model. By marginalisation of the empirical smoothing distribution $\widehat{p}_\theta(x_{1:\kappa_t}|y_{1:\kappa_t})$ over $x_{1:t-2}$ and $x_{t+1:\kappa_t}$, we obtain the approximation

$$\widehat{p}_\theta^{\Delta}(\mathrm{d}x_{t-1:t}|\mathbf{y}) \triangleq \sum_{i=1}^{N} w_{\kappa_t}^{(i)} \delta_{\tilde{x}_{\kappa_t, t-1:t}^{(i)}}(\mathrm{d}x_{t-1:t}). \tag{15}$$

Here, we use the notation $\tilde{x}_{\kappa_t,t}^{(i)}$ to denote the ancestor at time t of particle $x_{\kappa_t}^{(i)}$ and $\tilde{x}_{\kappa_t,t-1:t}^{(i)} = \{\tilde{x}_{\kappa_t,t-1}^{(i)}, \tilde{x}_{\kappa_t,t}^{(i)}\}$. Inserting (14)–(15) into (13) provides an estimator of (12),

$$\widehat{\mathsf{S}}_T(\theta|u) = \nabla \log p(\theta) + \sum_{t=1}^{T} \sum_{i=1}^{N} w_{\kappa_t}^{(i)} \xi_\theta\Big(\tilde{x}_{\kappa_t,t}^{(i)}, \tilde{x}_{\kappa_t,t-1}^{(i)}\Big), \tag{16}$$

which is used in the proposal distributions in (6).

3.4 Estimation of the Hessian

The negative Hessian of the parameter log-posterior can be written as

$$\mathsf{I}_T(\theta) = -\nabla^2 \log p(\theta) - \nabla^2 \log p_\theta(\mathbf{y}), \tag{17}$$

where it is assumed that the Hessian of the log-prior $\nabla^2 \log p(\theta)$ can be calculated analytically. The negative Hessian of the log-likelihood, also known as the *observed information matrix*, can using *Louis' identity* [Cappé et al., 2005] be expressed as

$$\begin{aligned} -\nabla^2 \log p_\theta(\mathbf{y}) = \nabla \log p_\theta(\mathbf{y})^2 - \mathbb{E}_\theta\left[\nabla^2 \log p_\theta(\mathbf{x},\mathbf{y})\middle|\mathbf{y}\right] \\ - \mathbb{E}_\theta\left[\nabla \log p_\theta(\mathbf{x},\mathbf{y})^2\middle|\mathbf{y}\right]. \end{aligned} \tag{18}$$

Here, we have introduced the notation $v^2 = vv^\top$ for a vector v. From this, we can construct an estimator of (17) using the estimator of the gradient in (16), of the form

$$\widehat{\mathsf{I}}_T(\theta|u) = -\nabla^2 \log p(\theta) + \widehat{\mathsf{S}}_T(\theta|u)^2 - \widehat{\mathsf{I}}_T^{(1)}(\theta|u) - \widehat{\mathsf{I}}_T^{(2)}(\theta|u), \tag{19}$$

where we introduce $\mathsf{I}_T^{(1)}(\theta) = \mathbb{E}_\theta\left[\nabla^2 \log p_\theta(\mathbf{x},\mathbf{y})|\mathbf{y}\right]$ and $\mathsf{I}_T^{(2)}(\theta) = \mathbb{E}_\theta\left[\nabla \log p_\theta(\mathbf{x},\mathbf{y})^2|\mathbf{y}\right]$. We obtain the estimator of the first term analogously to (16) as

$$\begin{aligned} \widehat{\mathsf{I}}_T^{(1)}(\theta|u) &= \sum_{t=1}^{T}\sum_{i=1}^{N} w_{\kappa_t}^{(i)} \zeta_\theta\left(\tilde{x}_{\kappa_t,t}^{(i)}, \tilde{x}_{\kappa_t,t-1}^{(i)}\right), \text{ where} \\ \zeta_\theta(x_t,x_{t-1}) &= \nabla^2 \log f_\theta(x_t|x_{t-1}) + \nabla^2 \log g_\theta(y_t|x_t). \end{aligned} \tag{20}$$

The estimator of the second term needs a bit more work and we start by rewriting the last term in (18) as

$$\begin{aligned} &\sum_{t=1}^{T}\sum_{s=1}^{T} \mathbb{E}_\theta\left[\xi_\theta(x_t,x_{t-1})\xi_\theta(x_s,x_{s-1})^\top\middle|\mathbf{y}\right] \\ &= \sum_{t=1}^{T}\Bigg\{\mathbb{E}_\theta\left[\xi_\theta(x_t,x_{t-1})^2\middle|\mathbf{y}\right] \\ &+ \sum_{s=1}^{t-1}\mathbb{E}_\theta\left[\big(\xi_\theta(x_t,x_{t-1}),\xi_\theta(x_s,x_{s-1})\big)^\dagger\middle|\mathbf{y}\right]\Bigg\}, \end{aligned} \tag{21}$$

where we have introduced the operator $(a,b)^\dagger = ab^\top + ba^\top$ for brevity. Consider the last term appearing in this expression, we can rewrite it as

$$\begin{aligned} &\sum_{s=1}^{t-1}\mathbb{E}_\theta\left[\xi_\theta(x_t,x_{t-1})\xi_\theta(x_s,x_{s-1})^\top\middle|\mathbf{y}\right] \\ &= \mathbb{E}_\theta\left[\xi_\theta(x_t,x_{t-1})\underbrace{\left\{\sum_{s=1}^{t-1}\mathbb{E}_\theta\left[\xi_\theta(x_s,x_{s-1})\middle|x_{t-1},y_{1:t-1}\right]\right\}^\top}_{\triangleq \alpha_\theta(x_{t-1})^\top}\middle|\mathbf{y}\right]. \end{aligned}$$

From this, we see that (21) can be written as an additive functional of the form

$$\sum_{t=1}^{T} \mathbb{E}_\theta \left[(\xi_\theta(x_t, x_{t-1}))^2 + \big((\xi_\theta(x_t, x_{t-1}), \alpha_\theta(x_{t-1}) \big)^\dagger \Big| \mathbf{y} \right],$$

which can be estimated using the FL smoother as before. However, for this we need to compute the quantities $\alpha_\theta(x_{t-1})$. One option is to make use of a type of fixed-lag approximation for $\alpha_\theta(x_{t-1})$, by assuming that x_s and x_t are conditionally independent given $y_{1:\kappa_t}$, whenever $|s-t| > \Delta$. This approach has previously been used by Doucet et al. [2013]. Alternatively, we can use a filter approximation according to

$$\widehat{\alpha}_\theta\left(x_t^{(i)}\right) = \widehat{\alpha}_\theta\left(x_{t-1}^{a_t^{(i)}}\right) + \xi_\theta\left(x_t^{(i)}, x_{t-1}^{a_t^{(i)}}\right), \tag{22}$$

for $i = 1, \ldots, N$. Note that this approach suffers from the same particle degeneracy as the APF. However, this only affects a small number of terms and in our experience this approximation works sufficiently well to give estimates with reasonable variance. The resulting estimator using (21) is

$$\widehat{\mathsf{I}}_T^{(2)}(\theta|u) = \sum_{t=1}^{T} \sum_{i=1}^{N} w_{\kappa_t}^{(i)} \eta_\theta\left(\tilde{x}_{\kappa_t,t}^{(i)}, \tilde{x}_{\kappa_t,t-1}^{(i)}\right), \text{ where} \tag{23}$$

$$\eta_\theta(x_t, x_{t-1}) = \xi_\theta(x_t, x_{t-1})^2 + \big(\xi_\theta(x_t, x_{t-1}), \widehat{\alpha}_\theta(x_{t-1})\big)^\dagger.$$

Hence, the Hessian can be estimated using (19) by inserting the estimators from (20), (22) and (23).

3.5 Regularisation of the estimate of the Hessian

The PMH2 proposal (6) relies on the assumption that the observed information matrix is positive definite (PD). The estimator given in (19) does not always satisfy this, especially when the Markov chain is located far from the posterior mode. Typically, the amount of information is limited in such regions and this results in that the curvature is difficult to estimate. To cope with this issue, one alternative is to regularize the Hessian by adding a diagonal matrix to shift the eigenvalues to be positive. The diagonal matrix can e.g. be selected such that

$$\Delta \widehat{I}_T = \max\left\{0, -2\lambda_{\min}\left(\widehat{I}_T\right)\right\} I_d, \tag{24}$$

where $\lambda_{\min}(\widehat{I}_T)$ denotes the smallest eigenvalue of $\widehat{I}_T(\theta|u)$. In this article, we make use of this method for handling non–PD estimates of the negative Hessian for the PMH2 algorithm. This heuristic is common for Newton-type optimisation algorithms [Nocedal and Wright, 2006, Chapter 3.4].

Note, that there are other solutions available for ensuring positive definiteness that only shifts the negative eigenvalues, see [Nocedal and Wright, 2006, Chapter 3]. We emphasise that this type of regularization keeps the Markov

Algorithm 2 Estimation of the likelihood, the gradient and the Hessian of the log-posterior

INPUTS: $\mathbf{y}$ (data), $R(\cdot)$ (propagation kernel), $\nu(\cdot)$ (weight function), $N > 0$ (no. particles), $0 < \Delta \leq T$ (lag).

OUTPUTS: $\widehat{p}_\theta(\mathbf{y})$ (est. of the likelihood), $\widehat{\mathsf{S}}_T(\theta)$ (est. of the gradient), $\widehat{\mathsf{I}}_T(\theta)$ (est. of the negative Hessian).

```
1: Initialise each particle $x_0^{(i)}$.
2: for $t = 1$ to $T$ do
3:     Resample and propagate each particle using (9).
4:     Calculate the weights for each particle using (10).
5: end for
6: Compute $\widehat{p}_\theta(\mathbf{y})$ by (11).
7: Compute $\widehat{\mathsf{S}}_T(\theta)$ and $\widehat{\mathsf{I}}_T(\theta)$ by (16) and (19), respectively.
8: if $\widehat{\mathsf{I}}_T(\theta) \leq 0$ then
9:     [standard] Regularize $\widehat{\mathsf{I}}_T(\theta)$ by adding $\Delta\widehat{\mathsf{I}}_T$ computed by (24)
10:    [hybrid] Replace $\widehat{\mathsf{I}}_T(\theta)$ by the inverse covariance matrix computed using
       the $L$ final samples of the Markov chain during the burn-in.
11: end if
```

chain invariant, i.e. still targets the correct posterior distribution (recall Section 2.1).

Another alternative is to replace the estimate of the negative Hessian with the inverse sample covariance matrix calculated using the trace of Markov chain when the estimate is not PD. This can be seen as a hybrid between the PMH2 algorithm and a *pre-conditioned PMH1 algorithm*. This resembles some other adaptive MH algorithms [Andrieu and Thoms, 2008] in which the same procedure is used to adapt the covariance matrix of a random walk proposal. For this, we can make use of the last L iterations of the MH algorithm after that the Markov chain has reached stationarity. During the burn-in phase, non–PD estimates can be handled using a regularization approach or by rejecting the proposed parameter. In this article, we refer to this method for handling non–PD estimates of the negative Hessian as the *hybrid PMH2 algorithm*, where we use the latter alternative during the burn-in phase. Note that this pre-conditioning can also be applied to the PMH0 and PMH1 algorithm, we return to this in Section 4.4.

3.6 Resulting SMC algorithm

In Algorithm 2, we present the complete procedure that combines the APF with the FL smoother to compute the estimates needed for the PMH2 proposal (6). Note that the two different methods to handle non–PD estimates of the negative Hessian matrix results in the *standard* and *hybrid* PMH2 algorithm, respectively.

We end this section by briefly discussing the statistical properties of the estimates of the gradient and Hessian obtained from the FL smoother. From Olsson et al. [2008], we know that the FL smoother gives biased estimates of the gradient and Hessian for any number of particles. Remember that this does not effect the invariance of the Markov chain (recall Section 2.1). The main advantage of the FL smoother over the APF (which gives a consistent estimate) is that the former enjoys a smaller variance than the APF, i.e. we obtain a favourable bias-variance trade-off for a certain choice of lag Δ. Note that a too small lag gives a large bias in the estimate and a too large lag gives a large variance in the estimate; we return to this choice in Section 4.

4 Numerical illustrations

In this section, we provide illustrations of the properties of the FL smoother and the different proposed algorithms. The source code in Python and the data used for some of the numerical illustrations are available for download at: `http://liu.johandahlin.com/`.

4.1 Estimation of the log-likelihood and the gradient

We begin by illustrating the use of the FL smoother for estimating the log-likelihood and the gradient. Here, we consider a linear Gaussian state space (LGSS) model given by

$$x_{t+1}|x_t \sim \mathsf{N}\Big(x_{t+1}; \phi x_t, \sigma_v^2\Big), \tag{25a}$$

$$y_t|x_t \sim \mathsf{N}\Big(y_t; x_t, \sigma_e^2\Big). \tag{25b}$$

We generate two data realisations of length $T = 100$ using parameters $\theta^{(1)} = \{\phi, \sigma_v^2, \sigma_e^2\} = \{0.5, 1.0, 0.1^2\}$ and $\theta^{(2)} = \{0.5, 1.0, 1.0\}$ with a known initial zero state. We use the lag $\Delta = 5$ and run the PFs with systematic resampling [Carpenter et al., 1999].

For this model, we can compute the true values of the log-likelihood and the gradient by running an RTS smoother [Rauch et al., 1965]. In Figure 1, we present boxplots of the L_1-errors in the estimated log-likelihood and the gradient of the log-posterior with respect to ϕ, evaluated at the true parameters. When $\sigma_e = 0.1$, we observe that the faPF has a large advantage over the bPF for all choices of N. When $\sigma_e = 1.0$, we get smaller difference in the error of the gradient estimates, but the log-likelihood estimates are still better for the faPF. Similar results are also obtained for the gradient with respect to σ_v.

In Figure 2, we present the error in the gradient estimates with respect to ϕ using a varying lag Δ and a varying number of particles N. The results are obtained by 1 000 Monte Carlo runs on a single data set generated from the previously discussed LGSS model with $T = 100$. We conclude again that faPF is preferable when available. The results are largely robust to the lag, as long

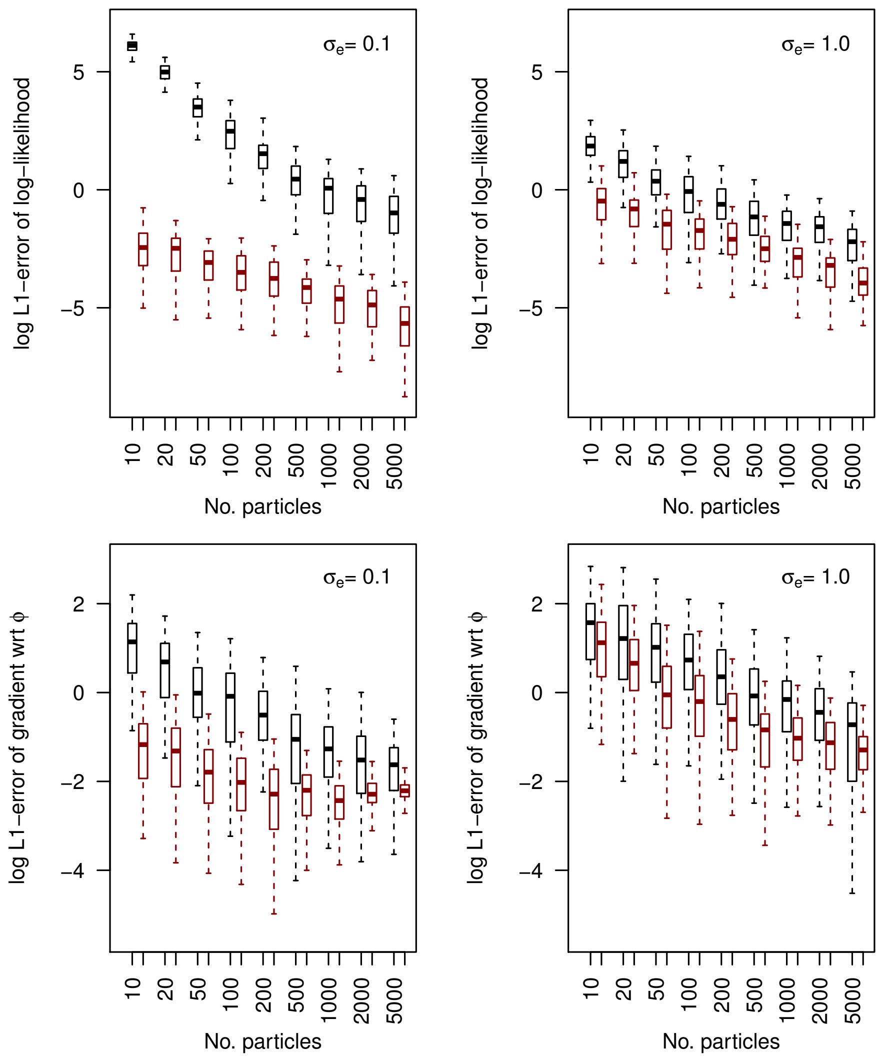


Figure 1: The log L_1-error in the log-likelihood estimates and the estimates of the gradient with respect to ϕ in the LGSS model with $\sigma_e = 0.1$ (left) and $\sigma_e = 1$ (right). The bPF (black) and faPF (red) are evaluated by 1 000 MC iterations using a fixed data set with $T = 100$.

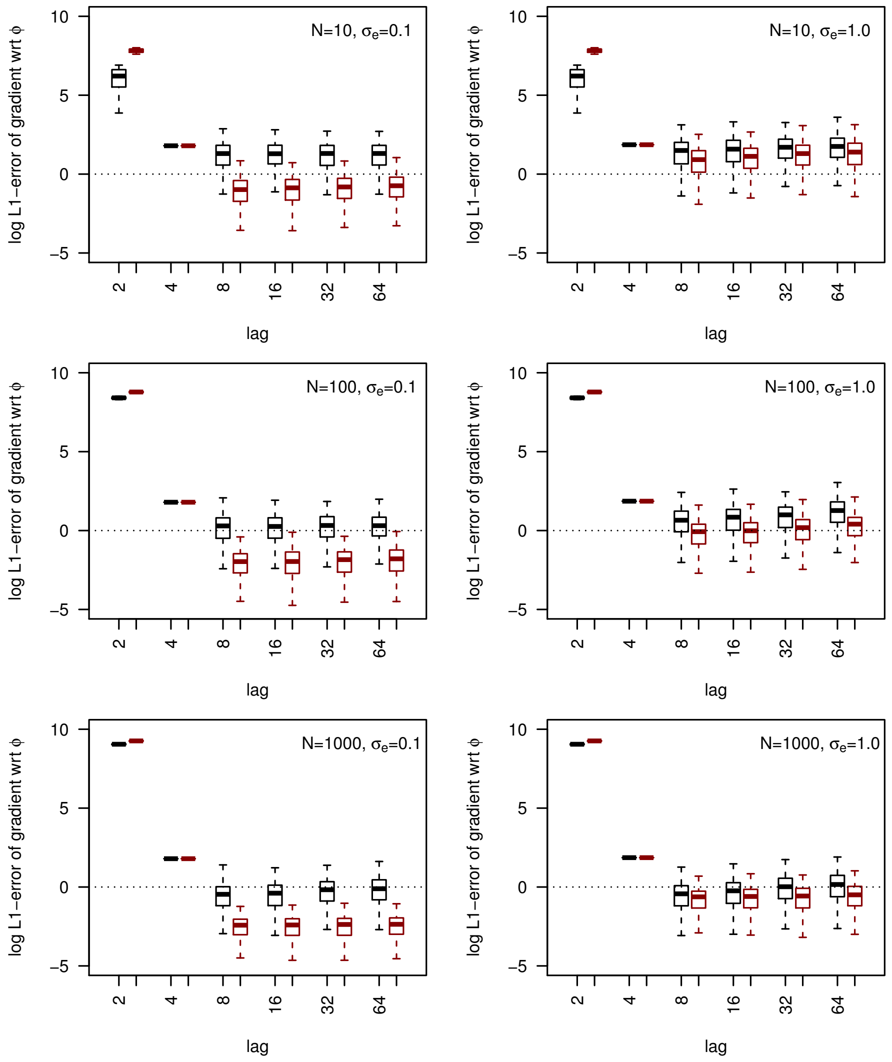


Figure 2: The log L_1-error in the estimates of the gradient with respect to ϕ in the LGSS model with $\sigma_e = 0.1$ (left) and $\sigma_e = 1$ (right). The bPF (black) and faPF (red) are evaluated by 1 000 Monte Carlo iterations using a fixed data set with $T = 100$.

as this is chosen large enough when using the faPF. A lag of about 12 seems to be a good choice for this model when $T = 100$ and when using the faPF with systematic resampling.

4.2 Burn-in and scale-invariance

Consider the problem of inferring $\{\theta_1, \theta_2\} = \{\phi, \sigma_v\}$ in the LGSS model (25). We simulate a single data set with parameters $\theta^{(1)}$ (as defined in the previous section) of length $T = 250$. We use an uniform parameter prior over $|\phi| < 1, \sigma_v > 0$ and initialise in $\theta_0 = \{0.1, 2\}$. We use faPF with systematic resampling, $N = 100$ and $\Delta = 12$. Here, we use the standard version of Algorithm 2 to adjust the estimate of the Hessian in the cases when it is not PD, resulting in the PMH2 algorithm.

We adjust the step lengths γ to give an acceptance rate during a pilot run of between 0.7 and 0.8 in the stationary phase. We obtain $\gamma = \{0.04, 0.065, 1.0\}$ for PMH$\{0, 1, 2\}$, respectively. Note that a single step length is used for each proposal to simplify the tuning. Of course, different step lengths can be used for each parameter, and we could also use different step lengths during the burn-in and the stationary phase of the algorithm using the approach discussed in Section 2.2. As previously mentioned, the PMH2 algorithm avoids this (potentially difficult and time-consuming) procedure, by taking the local geometric information into account.

In the left column of Figure 3, we present the first 50 iterations of the Markov chain from the three different algorithms. We note that the added information in the proposals of PMH1 and PMH2 aids the Markov chain in the burn-in phase. This results in that the Markov chains for the proposed algorithms reach the mode of the posterior quicker than the random walk used in PMH0.

To illustrate the scale invariance of the PMH2 algorithm, we reparametrise the LGSS model by $\{\theta_3, \theta_4\} = \{\phi, \sigma_v/10\}$. We keep the same settings as for the previous parametrisation and rerun the algorithms. From this run we obtain the middle column in Figure 3. We see clearly that the PHM1-algorithm does not perform well and gets stuck at the initial parameter value. The reason is that the second component of the gradient is increased by a factor 10 for the rescaled model. Since we still use the same step length, this will cause the PMH1 algorithm to overshoot the region of high posterior probability when proposing new values, and these will therefore never be accepted.

Finally, to improve the performance we recalibrate the three algorithms on the new parametrisation using the same procedure as before. We then obtain the new step lengths $\{0.005, 0.0075, 1.0\}$. The resulting Markov chains are presented in the right column of Figure 3. Despite the new step lengths, PMH0 and PMH1 continue to struggle. The reason is that the step lengths are limited by the small posterior variance in the θ_4-parameter, resulting in a very slow progression in the θ_3-direction. Again, for PMH2, the added Hessian information is used to rescale the proposal in each dimension resulting in a more efficient exploration of the posterior than for PMH0 and PMH1.

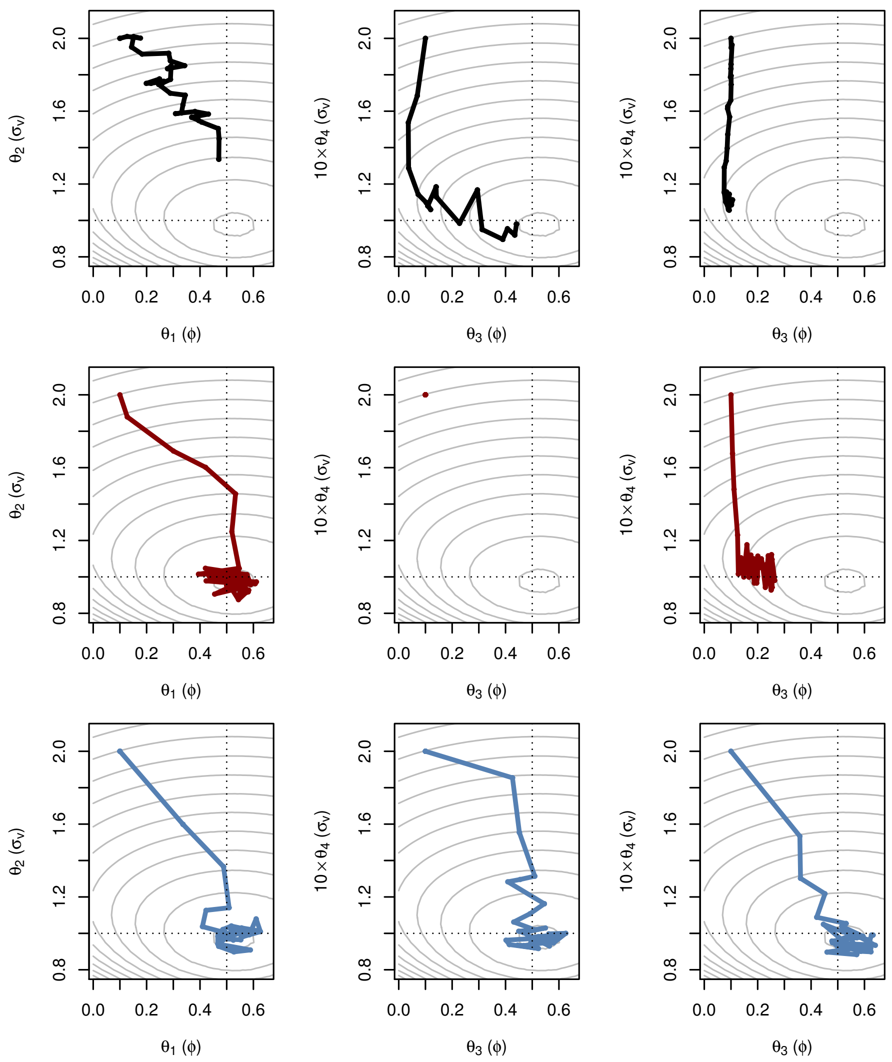


Figure 3: The trace plots of the first 50 steps using PMH0 (black), PMH1 (red) and PMH2 (blue). The dotted lines show the *true* parameters of the LGSS model. The gray contours show the log-posterior.

4.3 The mixing of the Markov chains at stationarity

We continue by investigating the mixing of the Markov chains at stationarity using an estimate of the integrated autocorrelation time (IACT) given by

$$\widehat{\mathsf{IACT}}(\theta_{1:M}) = 1 + 2\sum_{k=1}^{K} \widehat{\rho}_k(\theta_{1:M}), \tag{26}$$

where $\widehat{\rho}_k(\theta_{1:M})$ denotes the empirical autocorrelation at lag k of $\theta_{1:M}$ (after the burn-in has been discarded). A low value of the IACT indicates that we obtain many uncorrelated samples from the target distribution, implying that the chain is mixing well. Here, K is determined as the first index for which the empirical autocorrelation satisfies $|\widehat{\rho}_K(\theta_{1:M})| < 2/\sqrt{M}$, i.e. when the coefficient is statistically insignificant.

We return to the LGSS model in (25) with the original parameterisation $\{\theta_1, \theta_2\} = \{\phi, \sigma_v\}$ using the same settings as before. A total of 25 data sets are generated using the parameters $\theta^{(1)}$ and the algorithms are initialised at the true parameter values to avoid a long burn-in phase. The step sizes are determined using a series of pilot runs on the first generated dataset to minimise the total IACT for each algorithm. This is done to make a fair comparison between the different algorithms at their near *optimal* performance. The resulting step sizes are obtained as $\{0.08, 0.075, 1.50\}$.

Finally, we estimate the mixing in each of the 25 simulated data sets during $M = 30\,000$ MCMC iterations (discarding the first $10\,000$ iterations as burn-in). The results are presented in Table 1, where the median and interquartile range (IQR; the distance between the 25% and 75% quartiles) are presented for each PMH algorithm. Here, we present the results the standard version of Algorithm 2.

We see that the added information decreases the IACT about 2 times for PMH1 and PMH2 compared with PMH0. We conclude that the extra information brought by the gradient and the Hessian improves the mixing of the Markov chains in this model, which results in a more efficient exploration of the posterior. Note that, for this parametrisation of the LGSS model the posterior is quite isotropic (which can also be seen in the left column of Figure 3). Hence, the conditions are in fact rather favourable for PMH0 and PMH1.

4.4 Parameter inference in a Poisson count model

In this section, we analyse the annual number of major earthquakes[1] (over 7 on the Richter scale) during the period from year 1900 to 2014. Following Langrock [2011], we model the data using

$$x_{t+1}|x_t \sim \mathsf{N}\Big(x_{t+1}; \phi x_t, \sigma^2\Big), \tag{27a}$$

$$y_t|x_t \sim \mathsf{P}\Big(y_t; \beta \exp(x_t)\Big), \tag{27b}$$

[1]The data is obtained from the Earthquake Data Base System of the U.S. Geological Survey, which can be accessed at `http://earthquake.usgs.gov/earthquakes/eqarchives/`.

		Acc. rate	IACT(ϕ)		IACT(σ_v)	
		Median	Median	IQR	Median	IQR
PMH0	bPF(500)	0.02	257	146	265	371
	bPF(1000)	0.06	83	129	79	118
	bPF(2000)	0.15	29	23	15	24
	faPF(50)	0.37	9	8	8	5
	faPF(100)	0.38	9	6	7	4
	faPF(200)	0.38	7	6	7	4
PMH1	bPF(500)	0.02	187	271	203	347
	bPF(1000)	0.10	64	85	49	72
	bPF(2000)	0.22	23	16	12	24
	faPF(50)	0.58	**3**	2	**3**	1
	faPF(100)	0.59	4	2	**3**	1
	faPF(200)	0.58	**3**	1	**3**	1
PMH2	bPF(500)	0.03	170	211	164	190
	bPF(1000)	0.10	59	73	65	80
	bPF(2000)	0.24	13	10	19	17
	faPF(50)	0.66	**3**	1	4	2
	faPF(100)	0.66	**3**	1	5	2
	faPF(200)	0.66	**3**	1	4	2

Table 1: Median and IQR for the acceptance rate and IACT using different SMC algorithms. The values are computed using 25 different data sets from the LGSS model.

with parameters $\theta = \{\phi, \sigma, \beta\}$ and uniform priors over $|\phi| < 1$, $\sigma > 0$ and $\beta > 0$. Here, $\mathsf{P}(\lambda)$ denotes a Poisson distribution with parameter λ.

We repeat the procedure from the previous subsection and obtain the step lengths $\{0.06, 0.006, 0.85\}$. Here, we use $M = 30\,000$ MCMC iterations (discarding the first 10 000 iterations as burn-in), the bPF with systematic resampling, $\Delta = 12$, $\theta_0 = \{0.5, 0.5, 18\}$ and $L = 2\,500$. In this model, the estimate of the negative Hessian is often non–PD (during about half of the iterations) and the choice of regularisation is therefore important. To explore the properties of the regularisation, we apply both the standard and hybrid version of the PMH2 algorithm discussed in Section 3.5. We compare these methods to standard and pre-conditioned versions of the the PMH0 and PMH1 algorithms, using the sample posterior covariance matrix calculated in the same manner as for the hybrid PMH2 algorithm.

In Table 2, we present the resulting acceptance rates and IACT values for each parameter and algorithm. We note the large decrease in IACT for β when using the Hessian information, where the hybrid PMH2 seems to perform better than standard version for this model. The improved mixing by using PMH2 is due to the scale invariance property, as the parameter β is at least an order of magnitude larger than ϕ and σ (c.f. Figure 3). Note that a reparameterisation or using separate step lengths for the parameters could possibly have helped in improving the mixing in β for the standard versions of PMH0 and PMH1.

Using the standard and hybrid version of PMH2, decreases the overall computational cost by a factor of about 100 for a specific number of effective samples. The poor performance of the pre-conditioned algorithms is probably due to that the sample posterior covariance matrix does not fully capture the geometry of the posterior distribution.

In Figure 4, we present the trace and posterior estimates for β using the standard versions of PMH0 and PMH1 as well as hybrid PMH2. The posterior estimates are obtained by pooling the 10 parallel Markov chains after the burn-ins have been discarded. We see that the traces behave rather differently with hybrid PMH2 exploring the space well compared with the other methods.

Using the parameter posterior estimate, we can compute point estimates for the parameters of the model. The posterior mean for hybrid PMH2 is obtained as $\{0.88, 0.15, 16.58\}$ with standard deviations $\{0.07, 0.03, 2\}$. The parameter estimate is comparable to the estimate $\{0.88, 0.15, 17.65\}$ obtained by a maximum likelihood-based method using the same data and model in Dahlin [2014, Example 4.9].

4.5 Robustness in the lag and step size

The PMH2 algorithm requires a number of parameters to be select by the user for each parameter inference problem. It is therefore interesting to discuss the robustness of the method with respect to these parameters. In the previous illustrations, we have seen that the number of particles N is an important factor in determining the mixing.

	Version	SMC alg.	**Acc. rate**	**IACT**(ϕ)		**IACT**(σ)		**IACT**(β)	
			Median	Median	IQR	Median	IQR	Median	IQR
PMH0	Standard	bPF(500)	0.26	497	712	16	3	2639	1163
	Standard	bPF(1000)	0.30	89	150	15	3	2680	438
	Pre-cond.	bPF(500)	0.43	35	17	16	1	107	105
	Pre-cond.	bPF(1000)	0.45	38	28	16	2	129	131
PMH1	Standard	bPF(500)	0.76	665	442	277	162	2651	364
	Standard	bPF(1000)	0.82	490	134	205	30	2875	1007
	Pre-cond.	bPF(500)	0.62	266	187	**9**	3	1728	1638
	Pre-cond.	bPF(1000)	0.70	98	209	**9**	3	1480	1732
PMH2	Standard	bPF(500)	0.24	91	17	53	14	222	37
	Standard	bPF(1000)	0.28	60	14	47	17	139	59
	Hybrid	bPF(500)	0.45	20	3	17	4	30	15
	Hybrid	bPF(1000)	0.49	**17**	4	18	3	**23**	5

Table 2: Median and IQR for the acceptance rate and IACT using different number of particles. The values are computed using 10 runs on the Earthquake count data model.

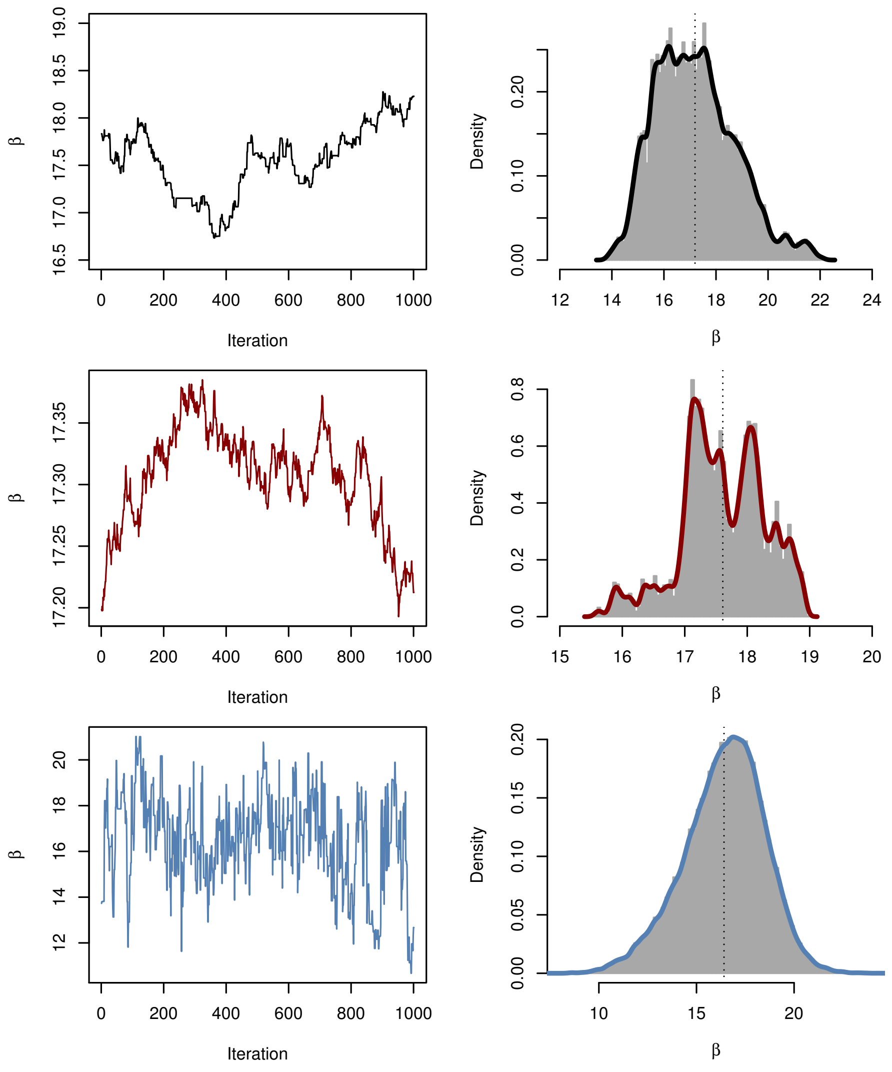
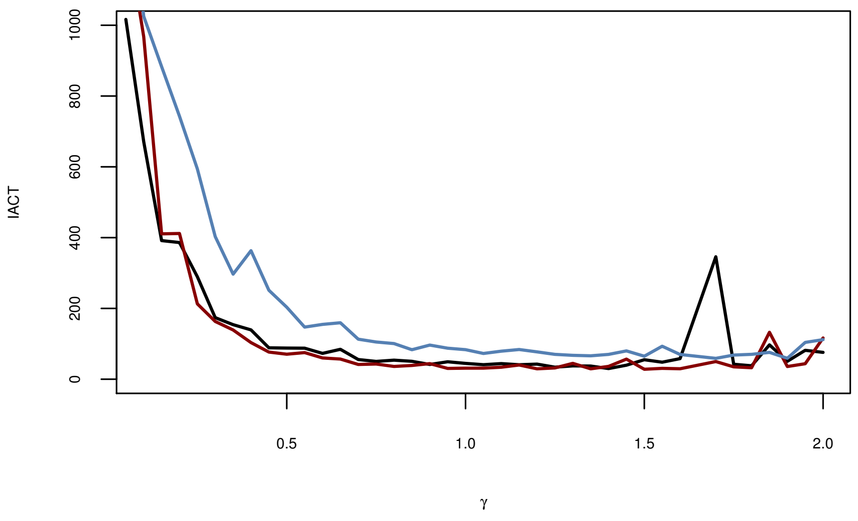


Figure 4: Part of the trace (left) and posterior estimates (right) for the β parameter in the earthquake count model using standard versions of PMH0 (black), PMH1 (red) and hybrid version of PMH2 (blue). Dotted lines indicate the posterior means.



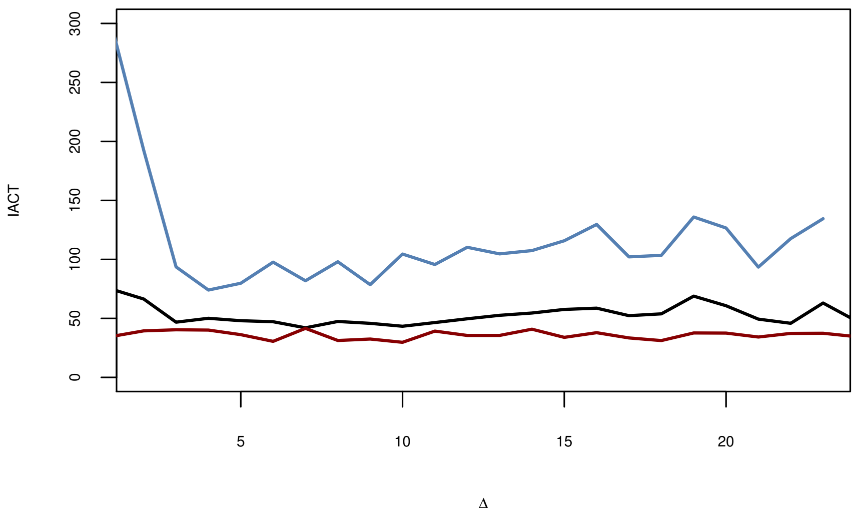


Figure 5: The IACT for ϕ (black), σ (red) and β (blue) for varying step sizes γ (upper) and lag Δ (lower). The values are computed as the median of 10 runs using standard PMH2 with the same data.

Two other important parameters are the step length γ and the lag in the FL-smoother Δ. To illustrate the impact of these quantities on the IACT, we return to the Earthquake model in (27) using the standard PMH2 algorithm with the same settings but with $M = 15\,000$ (discarding the first $5\,000$ iterations as burn-in) and $N = 1\,500$. In Figure 5, we present the IACT for the three parameters in the model when varying γ and Δ, keeping everything else fixed. The standard PMH2 algorithm seems to be rather robust to both the choice of Δ and γ after a certain threshold. Recall the discussion in Section 4.1 for the FL smoother. We conclude that a suitable standard choice for the step length could be $\gamma = 1$, which can be fine tuned if the performance is not good enough. This recommendation is also common in the literature concerning Newton-type algorithms.

5 Discussion and future work

Adding the gradient and Hessian information to the PMH proposal can have beneficial results including: (i) a shorter burn-in phase, (ii) a better mixing of the Markov chain, and (iii) scale-invariance of the proposal which simplifies tuning. The latter point is true in particular for PMH2, since this method takes the local curvature of the posterior into account, effectively making the method invariant to affine transformations.

It is common to distinguish between two phases of MCMC algorithms: the burn-in and stationary phases. We have seen empirically that the proposed methods can improve upon the original PMH0 during both of these phases but the *best* choices for the step lengths can differ between these two phases. Typically, a smaller step length is preferred during burn-in and a larger during stationarity (the opposite holds for PMH0). The reason for this is that during burn-in, the (natural) gradient information will heavily skew the proposal in a direction of increasing posterior probability. That is, the methods tend to be *aggressive* and propose large steps to make rapid progression toward regions of high posterior probability. While this is intuitively appealing, the problem is that we require the Markov chains to be reversible at all times. The reverse of these large steps can have very low probability which prevents them from being accepted.

One interesting direction for future work is therefore to pursue adaptive algorithms (see e.g. Andrieu and Thoms [2008], Peters et al. [2010], Pitt et al. [2012]), to automatically tune the step lengths during the different phases of the algorithms. Another interesting possibility is to relax the reversibility requirement during burn-in; see [Diaconis et al., 2000] for a related reference. This would cause the methods to behave like optimisation procedures during the initial phase, but transition into samplers during the second phase.

Finally, another very interesting direction for future work is to extend the proposed methods to develop a particle-version of the manifold Hamiltonian Monte Carlo (mHMC) algorithm [Duane et al., 1987, Neal, 2010, Girolami and Calderhead, 2011]. The reason for this is motivated by the large improvement in mixing seen by e.g. Neal [2010], Girolami and Calderhead [2011] for high dimensional problems in *vanilla* MH sampling.

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 and J. Thoms. A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373, 2008.

C. Andrieu and M. Vihola. Convergence properties of pseudo-marginal Markov chain Monte Carlo algorithms. arXiv.org, arXiv:1210.1484, October 2012.

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.

M. A. Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160, 2003.

O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models*. Springer, 2005.

J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings Radar, Sonar and Navigation*, 146(1):2–7, 1999.

J. Dahlin. *Sequential Monte Carlo for inference in nonlinear state space models*. Licentiate's thesis no. 1652, Linkping University, may 2014.

J. Dahlin, F. Lindsten, and T. B. Schön. Particle Metropolis Hastings using Langevin dynamics. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

J. Dahlin, F. Lindsten, and T. B. Schön. Second-order particle MCMC for Bayesian parameter inference. In *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, August 2014.

P. Del Moral. *Feynman-Kac Formulae - Genealogical and Interacting Particle Systems with Applications*. Probability and its Applications. Springer, 2004.

P. Del Moral, A. Doucet, and S. Singh. Forward smoothing using sequential Monte Carlo. *Pre-print*, 2010. arXiv:1012.5390v1.

P. Diaconis, S. Holmes, and R. Neal. Analysis of a nonreversible Markov chain sampler. *Annals of Applied Probability*, 10(3):685–1064, 2000.

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, P. Jacob, and A. M. Johansen. Discussion on Riemann manifold Langevin and Hamiltonian Monte Carlo methods. Journal of the Royal Statistical Society: Series B Statistical Methodology, 73(2), p 162, 2011.

A. Doucet, M. K. Pitt, G. Deligiannidis, and R. Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator, October 2012. arXiv:1210.1871v2.

A. Doucet, P. E. Jacob, and S. Rubenthaler. Derivative-Free Estimation of the Score Vector and Observed Information Matrix with Application to State-Space Models. *Pre-print*, 2013. arXiv:1304.5768v2.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

R. G. Everitt. Bayesian parameter estimation for latent Markov random fields and social networks. *Journal of Computational and Graphical Statistics*, 21 (4):940–960, 2012.

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.

M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):1–37, 2011.

A. Golightly and D. J. Wilkinson. Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface Focus*, 1(6):807–820, 2011.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings of Radar and Signal Processing*, 140(2):107–113, 1993.

G. Kitagawa and S. Sato. Monte Carlo smoothing and self-organising state-space model. In A. Doucet, N. de Fretias, and N. Gordon, editors, *Sequential Monte Carlo methods in practice*, pages 177–195. Springer, 2001.

R. Langrock. Some applications of nonlinear and non-Gaussian state–space modelling by means of hidden Markov models. *Journal of Applied Statistics*, 38(12):2955–2970, 2011.

R. M. Neal. MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, and X-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/ CRC Press, June 2010.

C. Nemeth and P. Fearnhead. Particle Metropolis adjusted Langevin algorithms for state-space models. *Pre-print*, 2014. arXiv:1402.0694v1.

J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2 edition, 2006.

J. Olsson, O. Cappé, R. Douc, and E. Moulines. Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14(1):155–179, 2008.

G. W. Peters, G. R. Hosack, and K. R. Hayes. Ecological non-linear state space model selection via adaptive particle Markov chain Monte Carlo. *Pre-print*, 2010. arXiv:1005.2238v1.

M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

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.

G. Poyiadjis, A. Doucet, and S. S. Singh. Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80, 2011.

H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, August 1965.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2 edition, 2004.

G. O. Roberts and J. S. Rosenthal. Optimal Scaling of Discrete Approximations to Langevin Diffusions. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 60(1):255–268, 1998.

G. O. Roberts and O. Stramer. Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and Computing in Applied Probability*, 4(4):337–357, 2003.

G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997.

C. Sherlock, A. H. Thiery, G. O. Roberts, and J. S. Rosenthal. On the efficency of pseudo-marginal random walk Metropolis algorithms. *Pre-print*, 2013. arXiv:1309.7209v1.