Journal of Computational Physics 228 (2009) 4312–4331

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Particle filtering with path sampling and an application to a bimodal ocean current model

Jonathan Weare

Courant Institute, New York University, 251 Mercer Street, New York, NY 10012, USA

ARTICLE INFO

Article history:
Received 31 March 2008
Received in revised form 9 February 2009
Accepted 10 February 2009
Available online 6 March 2009

PACS:
65C05
86A10

Keywords:
Particle filter
Path sampling
Parallel marginalization
Hybrid Monte Carlo
Kuroshio
Weather prediction

ABSTRACT

This paper introduces a recursive particle filtering algorithm designed to filter high dimensional systems with complicated non-linear and non-Gaussian effects. The method incorporates a parallel marginalization (PMMC) step in conjunction with the hybrid Monte Carlo (HMC) scheme to improve samples generated by standard particle filters. Parallel marginalization is an efficient Markov chain Monte Carlo (MCMC) strategy that uses lower dimensional approximate marginal distributions of the target distribution to accelerate equilibration. As a validation the algorithm is tested on a 2516 dimensional, bimodal, stochastic model motivated by the Kuroshio current that runs along the Japanese coast. The results of this test indicate that the method is an attractive alternative for problems that require the generality of a particle filter but have been inaccessible due to the limitations of standard particle filtering strategies.

1. Introduction

The reconstruction of unknown quantities from noisy observations is a recurrent theme in many fields. Examples include weather prediction and forcasting, robot tracking, stochastic volatility estimation, image analysis, and many more (see [1]). These problems motivate the need for efficient estimation procedures. As the observations arrive sequentially, any efficient algorithm will necessarily be recursive in the sense that the estimate given the value of a newly arrived observation relies on information calculated for the previous observation.

In the simplest case of Gaussian evolution and Gaussian observations a recursive solution to the problem is given by the Kalman filter (see [2]). While this algorithm has been modified and extended to handle more general problems (see [3]), it remains unsuitable for many problems with significantly non-Gaussian features (see [4]). A very general recursive technique, particle filtering, was first suggested in [5,6]. This algorithm is extremely widely applicable, but as discussed below, can be very inefficient. There have been many attempts to improve the efficiency of the basic particle filter (see [1]). In this paper I introduce a recursive particle filtering algorithm designed to filter high dimensional systems with complicated non-linear and non-Gaussian effects. The method essentially combines a particle filter with conditional path

E-mail address: weare@cims.nyu.edu

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.02.033

sampling of the underlying stochastic process (see [7,8]). The conditional path sampling is accelerated by a combination of the hybrid Monte Carlo method (HMC) and the parallel marginalization (PMMC) method recently introduced in [9].

As a validation the algorithm is tested on a 2516 dimensional, bimodal, stochastic model motivated by the bimodal behavior of the Kuroshio current that runs along the Japanese coast. This current exhibits transformations between a small meander during which the current remains close to the coast of Japan, and a large meander during which the current bulges away from the coast (see Fig. 1). These states typically persist for 5–10 years, while the transitions between meanders occurs in only a few months.

The bi-modality of this current was first studied by Yoshida in 1959 (see [12]). Since then there have been many attempts to model this behavior. One such model was suggested by Chao in [13]. In Chao's model the large and small meander states are basins of attraction of the system forced by the Kyushu wedge and Izu ridge (see Fig. 3). Both meanders coexist only for certain inflow volume conditions. Chao demonstrates that it is possible to observe the transition between meanders by deterministically varying the inflow condition.

In the present study, the model of Chao is modified to include an additive space-time white noise. The resulting model should not be taken seriously as a geophysical system. Here I am more focused on testing parallel marginalization and the conditional path sampling approach than on the geophysical implications of the model. Thus the important feature of the model is that it indeed exhibits rare transitions between the two metastable meanders (see Fig. 4).

Before the model and results are presented I discuss particle filters in general and the method that will be applied here. After discussing the non-linear filtering problem and a modification of the particle filter the bimodal ocean current model that will serve as a test for the algorithm is introduced. Following a description of the model numerical results are presented along with some concluding remarks. Some details of the modified particle filtering scheme including a description of the hybrid Monte Carlo and parallel marginalization algorithms are given in an appendix.

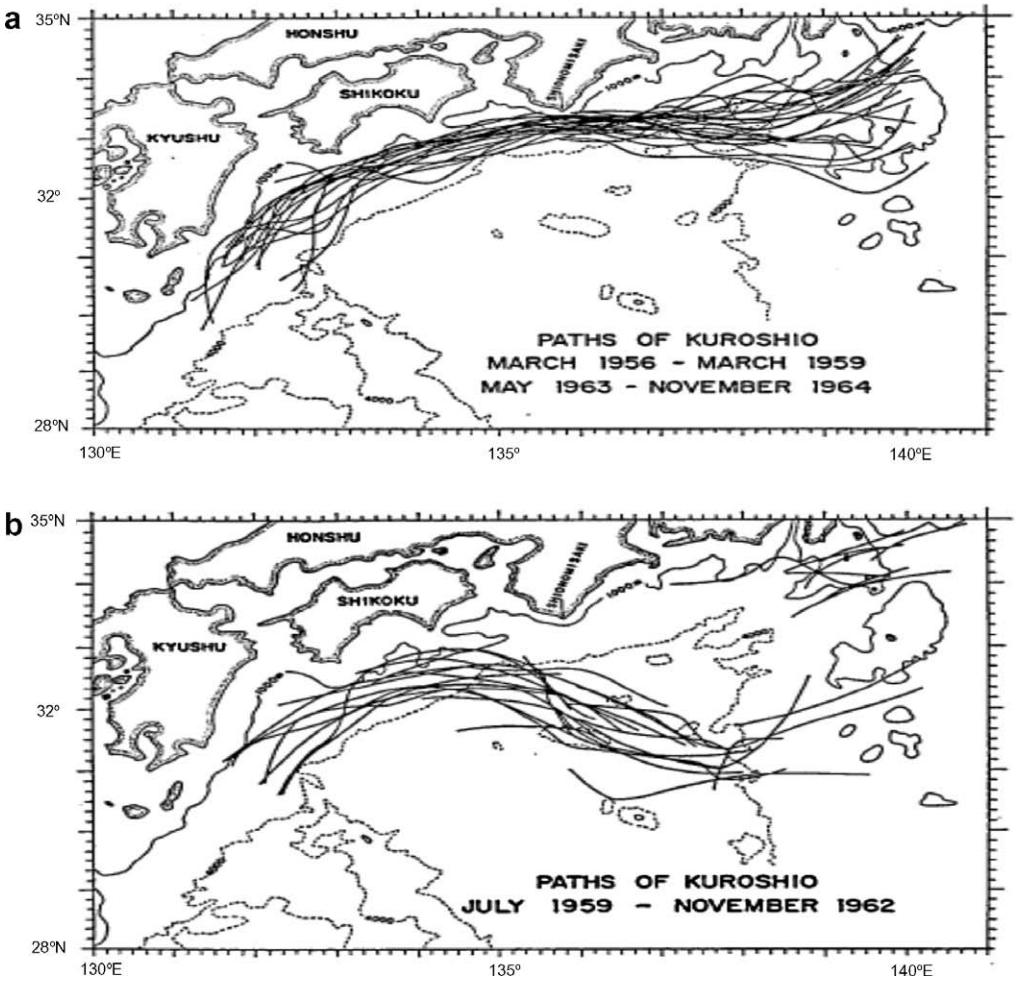

Fig. 1. (a) Paths in the small meander state. (b) Paths in the large meander state. ((a) and (b) reproduced from [10]. Originally adapted from [11]).

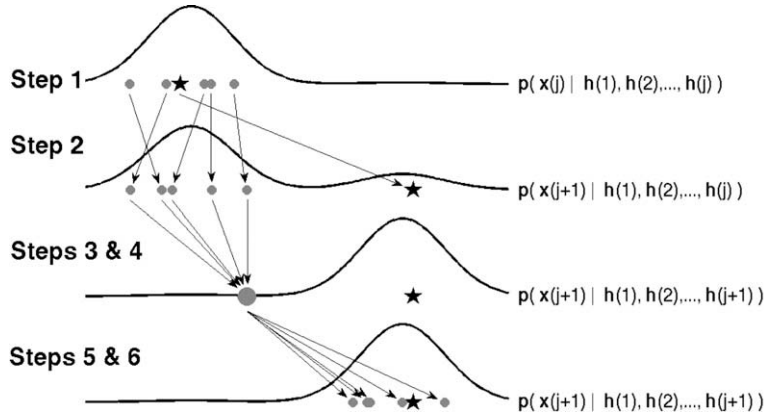

Fig. 2. Diagram of the steps in Algorithm 2. In each step the position of the hidden signal is represented by black star. In Step 1 the five samples are represented by small grey dots which are distributed according to the posterior distribution at time j, $\mathbf{p}(\mathbf{x}(j)|\mathbf{h}(1),\mathbf{h}(2),\ldots,\mathbf{h}(j))$ (represented by the top thick black curve). In Step 2 the samples are evolved forward according to p^j, the Markov transition density for the system. The samples produced by Step 2 are also represented by small grey dots which are distributed according to the predictive distribution at time $j+1$, $\mathbf{p}(\mathbf{x}(j+1)|\mathbf{h}(1),\mathbf{h}(2),\ldots,\mathbf{h}(j))$ (represented by the top thick black curve). In Steps 3 and 4 the samples are weighted and resampled producing (in this idealized picture) five samples with the same position which is approximately distributed according to the posterior distribution at time $j+1$, $\mathbf{p}(\mathbf{x}(j+1)|\mathbf{h}(1),\mathbf{h}(2),\ldots,\mathbf{h}(j+1))$ (represented by the top thick black curve). The five samples are all represented the large grey dot. Finally in Steps 5 and 6 the samples are "corrected" by the MCMC step. They will again be approximately distributed according to the posterior distribution at time $j+1$, $\mathbf{p}(\mathbf{x}(j+1)|\mathbf{h}(1),\mathbf{h}(2),\ldots,\mathbf{h}(j+1))$ (represented by the top thick black curve).

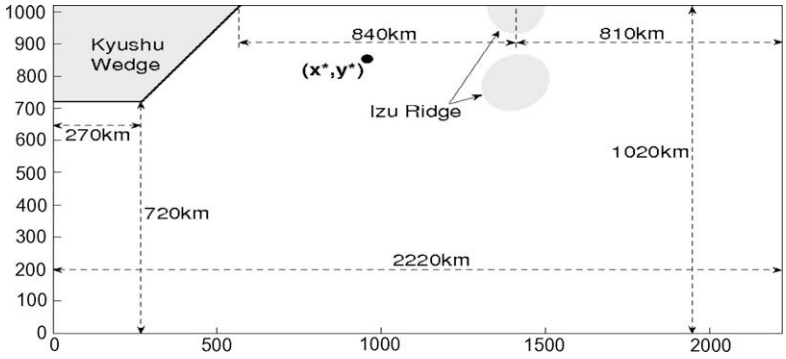

Fig. 3. Model geometry.

2. Non-linear filtering methods

The algorithm applied in Section 6 is a modification of a standard particle filter. I therefore begin with a brief general description of the filtering problem and particle filters. Consider some Markov process $\mathbf{x}(j) \in \mathbb{R}^{d_\mathbf{x}}$ governed by the transition density

$$p^j(\mathbf{x}(j+1)|\mathbf{x}(j)).$$

In many situations the process $\mathbf{x}$ models the behavior of some physical process which can only be partially observed (the weather for example). Suppose that one takes noisy observations of the form

$$\mathbf{h}(j) = G(\mathbf{x}(j), \boldsymbol{\chi}(j)) \tag{1}$$

for some function G where the random variables $\{\boldsymbol{\chi}(j)\}$ are independent and identically distributed. The process $\mathbf{x}$ should be considered "hidden" and revealed only through the observations $\mathbf{h}$. The goal of any filtering technique is to accurately reconstruct $\mathbf{x}$. Ideally one would like to be able to calculate modes and moments of the conditional distribution of the hidden signal $\mathbf{x}$ given the observations $\mathbf{h}$.

Throughout this paper the symbol $\mathbf{p}$ will be used to represent the joint density of all of the $\mathbf{h}$ and $\mathbf{x}$ random variables. Thus

$$\mathbf{p}(\mathbf{x}(j)|\{\mathbf{h}(l)\}_1^j)$$

is the conditional density of $\mathbf{x}(j)$ given the observations $\mathbf{h}(1),\mathbf{h}(2),\ldots,\mathbf{h}(j)$. This density is called the posterior distribution. One might also be interested in predicting the state of $\mathbf{x}$ (say at time k) given only observations in the past (say at times

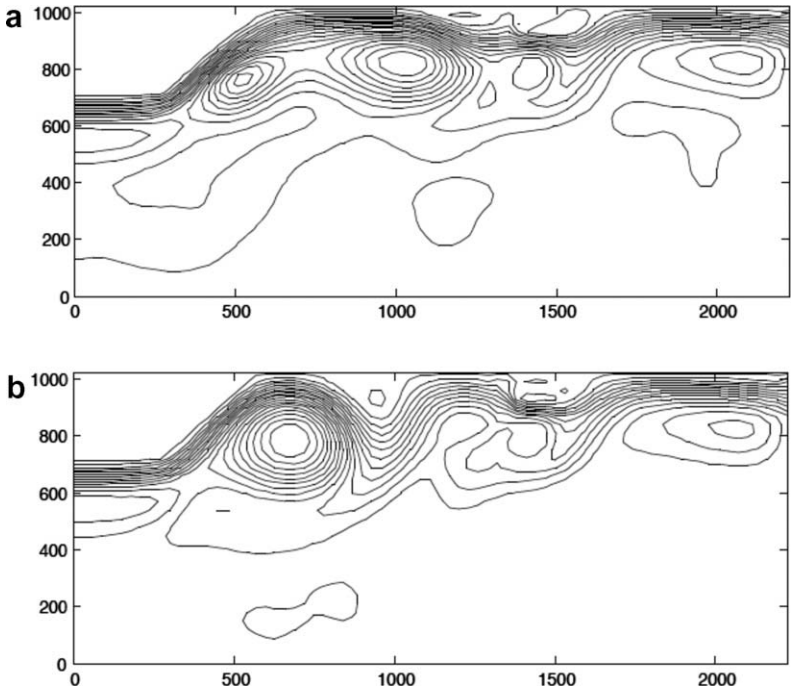

Fig. 4. (a) Small meander state of the approximate model, (22). (b) Large meander state of the approximate model, (22). The domain in both figures is as depicted in Fig. 3.

$1 < \cdots < j < k$), or in refining an estimate of $\mathbf{x}(k)$ given past, current, and future observations ($j > k$). These are the prediction and smoothing problems, respectively and their relevant conditional distributions are given by,

$$\mathbf{p}(\mathbf{x}(k)|\{\mathbf{h}(l)\}_1^j),$$

where $j < k$ or $j > k$. Here, for simplicity, the focus is on the case $j = k$, i.e. the filtering problem.

Henceforth it is assumed that the variables $\mathbf{h}(j) = G(\mathbf{x}(j), \boldsymbol{\chi}(j))$ admit a density proportional to some function $g(\mathbf{h}(j), \mathbf{x}(j))$, i.e.

$$\mathbf{p}(\mathbf{h}(j)|\mathbf{x}(j)) \propto g(\mathbf{h}(j), \mathbf{x}(j)). \tag{2}$$

The function g is often easy to evaluate and gives the likelihood of a particular value of the observation $\mathbf{h}(j)$ given a value of the state variables $\mathbf{x}(j)$.

2.1. Particle filtering

In Section 5, a Markov process $\mathbf{x}$ is introduced which models the behavior of the Kuroshio current which runs along the eastern coast of Japan. The goal will be to calculate averages of the current state of $\mathbf{x}$ given current and past observations ($\{\mathbf{h}\}$). For example at the time of observation $j+1$ one may wish to calculate the posterior average of some objective function f, i.e. one may wish to calculate the conditional expectation,

$$\mathbf{E}\left[f(\mathbf{x}(j+1))|\{\mathbf{h}(l)\}_1^{j+1}\right] = \int f(\mathbf{x}(j+1))\,\mathbf{p}\left(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1}\right)\mathbf{dx}(j+1). \tag{3}$$

This subsection contains a description of the standard particle filter approximation of (3). The reader familiar with particle filters may wish to skip this subsection and refer back to it for notation as needed.

There are several possible methods by which one might hope to approximate the expectation in (3). Perhaps the most obvious approach is to simply compute the integral using some quadrature scheme. This approach suffers from two insurmountable difficulties. The first is that there is often no closed form expression for the density $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1})$. The second is that numerical quadrature becomes computationally impractical in more than a few dimensions. Another approach is to generate independent samples $\{\mathbf{x}^i(j+1)\}_1^N$ with respect to $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1})$ and compute the sample mean approximation

$$\mathbf{E}[f(\mathbf{x}(j+1))|\{\mathbf{h}(l)\}_1^{j+1}] \approx \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^i(j+1)).$$

Unfortunately in general there is no direct and efficient means of generating independent samples from $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1})$. A third option is to generate samples $\{\mathbf{x}^i(j+1)\}_1^N$ from some reference density $\mathbf{q}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1})$ which can be easily sampled and compute the weighted sample mean

$$\mathbf{E}[f(\mathbf{x}(j+1))|\{\mathbf{h}(l)\}_1^{j+1}] \approx \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^i(j+1))\frac{\mathbf{p}(\mathbf{x}^i(j+1)|\{\mathbf{h}(l)\}_1^{j+1})}{\mathbf{q}(\mathbf{x}^i(j+1)|\{\mathbf{h}(l)\}_1^{j+1})}$$

or the related estimate

$$\mathbf{E}[f(\mathbf{x}(j+1))|\{\mathbf{h}(l)\}_1^{j+1}] \approx \frac{\sum_{i=1}^{N} f(\mathbf{x}^i(j+1))\frac{\mathbf{p}(\mathbf{x}^i(j+1)|\{\mathbf{h}(l)\}_1^{j+1})}{\mathbf{q}(\mathbf{x}^i(j+1)|\{\mathbf{h}(l)\}_1^{j+1})}}{\sum_{i=1}^{N}\frac{\mathbf{p}}{(}\mathbf{x}^i(j+1)|\{\mathbf{h}(l)\}_1^{j+1})\mathbf{q}(\mathbf{x}^i(j+1)|\{\mathbf{h}(l)\}_1^{j+1})}, \tag{4}$$

where N has been replaced by the approximation

$$N \approx \sum_{i=1}^{N}\frac{\mathbf{p}(\mathbf{x}^i(j+1)|\{\mathbf{h}(l)\}_1^{j+1})}{\mathbf{q}}(\mathbf{x}^i(j+1)|\{\mathbf{h}(l)\}_1^{j+1}).$$

This basic procedure is called importance sampling and is at the heart of any particle filtering method.

Particle filtering is a recursive implementation of the importance sampling approach just described. It is based on the recursion

$$\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1}) \propto g(\mathbf{h}(j+1),\mathbf{x}(j+1))\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j}), \tag{5}$$

$$\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j}) = \int p^j(\mathbf{x}(j+1)|\mathbf{x}(j))\,\mathbf{p}(\mathbf{x}(j)|\{\mathbf{h}(l)\}_1^{j})d\mathbf{x}(j), \tag{6}$$

(see [1]). Notice that if one sets

$$\mathbf{q}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1}) = \mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j})$$

then from (5),

$$\frac{\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1})}{\mathbf{q}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1})} \propto g(\mathbf{h}(j),\mathbf{x}(j+1)).$$

This implies that the approximation in expression (4) becomes

$$\mathbf{E}[f(\mathbf{x}(j+1))|\{\mathbf{h}(l)\}_1^{j+1}] \approx \frac{\sum_{i=1}^{N} f(\mathbf{x}^i(j+1))g(\mathbf{h}(j+1),\mathbf{x}^i(j+1))}{\sum_{i=1}^{N} g(\mathbf{h}(j+1),\mathbf{x}^i(j+1))} \tag{7}$$

where the samples $\mathbf{x}^i(j+1)$ are drawn from the predictive distribution $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j})$.

This discussion also indicates that when it is possible to generate samples from $\mathbf{x}^i(j+1)$ from the predictive distribution $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j})$ one can weight these samples by

$$W^i(j+1) = \frac{g(\mathbf{h}(j+1),\mathbf{x}^i(j+1))}{\sum_{i=1}^{N} g(\mathbf{h}(j+1),\mathbf{x}^i(j+1))}$$

and consider the weighted samples to be distributed according to the posterior distribution $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1})$.

Clearly this strategy can only be used when it is possible to generate samples from $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j})$. This issue can be addressed with the help of expression (6). Suppose that one has somehow already managed to generate samples $\{\mathbf{x}^i(j)\}_1^N$ from the posterior distribution at time j, $\mathbf{p}(\mathbf{x}(j)|\{\mathbf{h}(l)\}_1^{j})$. Then formula (6) implies that these samples can be used to generate samples $\{\mathbf{x}^i(j+1)\}_1^N$ from $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j})$ simply by evolving each $\mathbf{x}^i(j)$ according to the Markov transition probability $p^j(\mathbf{x}(j+1)|\mathbf{x}(j))$.

Thus if samples can be generated from the posterior distribution at time j, then by evolving these samples according to p^j one can generate samples which, after reweighting by $g(\mathbf{h}(j+1),\cdot)$, are approximately drawn from the posterior distribution at time $j+1$. To avoid waisted effort on samples with degenerate weights the weighted samples can be resampled (see Step 4 below). These steps are summarized by the following recursive algorithm first introduced in [5].

Algorithm 1 (*Particle filter 1*). One iteration of the standard particle filter algorithm is carried out as follows.

1. `Begin with` N `unweighted samples` $\mathbf{x}^i(j)$ `from` $\mathbf{p}(\mathbf{x}(j)|\{\mathbf{h}(l)\}_1^{j})$.
2. `Generate` N `samples` $\mathbf{x}^i(j+1)$ `from` $p^j(\mathbf{x}(j+1)|\mathbf{x}^i(j))$.
3. `Evaluate the weights`,

$$W^i(j+1) = \frac{g(\mathbf{h}(j+1),\mathbf{x}^i(j+1))}{\sum_1^N g(\mathbf{h}(j+1),\mathbf{x}^{\prime k}(j+1))}.$$

4. Generate N independent uniform random variables, $\{\Theta^i(j)\}_{i=1}^N$, in $(0,1)$. For $i = 1,\ldots,N$ let $\mathbf{x}^i(j+1) = \mathbf{x}'^k(j+1)$ where

$$\sum_{l=1}^{k-1} W^k(j+1) \leqslant \Theta^i(j) < \sum_{l=1}^{k} W^k(j+1).$$

5. Return to Step 1 with $j+1$ in place of j.

Notice that Step 4 in this algorithm will produce multiple copies of the samples $\mathbf{x}'^i(j+1)$ for which $g(\mathbf{h}(j+1),\mathbf{x}'^i(j+1))$ is relatively large, and will remove samples $\mathbf{x}'^i(j+1)$ for which $g(\mathbf{h}(j+1),\mathbf{x}'^i(j+1))$ is relatively small.

In Step 1 above the samples $\{\mathbf{x}'^i(j+1)\}$ are drawn from the predictive density $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^j)$ which does not incorporate information from the current $(j+1\text{st})$ observation. They are guesses of the current state of $\mathbf{x}$ given all of the past observations and as such, they may over or under represent regions of space with respect to the posterior density $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1})$. To resolve this in Steps 2–4 the samples are reweighted and resampled according to the likelihood g. If a sample occurs in a region deemed important by the newly arrived current observation, then the weight corresponding to the sample will be large, while if a sample occurs in a region deemed unimportant that sample will be assigned a small weight. Unfortunately in many cases far too many samples are generated in regions in which g is negligible. The next section describes how the samples can be "moved" into more important regions of space.

3. Particle filter with MCMC step

The particle filtering algorithm is a sequential importance sampling method and as such is subject to the limitations of any importance sampling algorithm. In particular, the speed of convergence of an importance sampling method is greatly affected by the degree to which the reference density (in this case $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^j)$) approximates the target density (in this case $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1})$). In problems that exhibit rare transitions between multiple metastable states this problem is particularly acute. Suppose that between two observations the hidden signal makes a transition from one metastable state to another. If at the time of the first observation, all particles are in the first metastable state, then at best only a few particles will make the transition to the new metastable state. Therefore, at the time of the next observation, when the new weights are calculated, almost all of the particles will receive negligible weight and will be discarded at the next resampling. In other words, the densities $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^j)$ and $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1})$ are not sufficiently close and as a result most of the random variables $W^i(j+1)$ will be negligible. This problem is common in importance sampling, but is compounded here because of the sequential structure of the particle filtering procedure.

There are other methods which do not suffer from this deficiency. These methods usually approximate the predictive density $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^j)$ by a simpler density (for example, a Gaussian), $p^{ij}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^j)$, and analytically or numerically evaluate the mean and variance of the density,

$$p^{ij}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1}) \propto g(\mathbf{h}(j+1),\mathbf{x}(j+1))\,p^{ij}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^j),$$

which can then also be approximated by a Gaussian (see [3]). The advantage of such a method is that, unlike an empirical approximation to $\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^j)$, the approximation $p^{ij}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^j)$ usually is not compactly supported and is therefore positive in regions where $g(\mathbf{h}(j+1),\mathbf{x}(j+1))$ is significant. The disadvantage of these methods is that because of the approximation of the predictive distribution by a simple density, they behave poorly on problems where non-linear or non-Gaussian effects are important.

In this section, a Markov chain Monte Carlo step is appended to Algorithm 1 which will move samples away from statistically insignificant regions. This idea has been proposed by several authors (see for example [14]). To see how this might be effective consider a system with multiple metastable states. As already mentioned, if the current state and the next state of the hidden signal are in different metastable states then all or most of the samples generated in Step 2 of Algorithm 1 will be in statistically insignificant regions. If one is lucky and one or two samples end up near the hidden signal then it is likely that all other samples will be discarded in Step 4 and the resulting collection of samples will be very degenerate. In this case the MCMC step will move these samples independently and help to decrease this degeneracy. However if as is more likely, none of the samples are near the state of the hidden signal then after Step 4 the collection of samples will not only be degenerate but will also be far from the state of the hidden signal. In this case the MCMC step will move the collection of samples closer to the metastable state containing the hidden signal. This scenario is depicted by the (somewhat idealized) diagrammatic description in Fig. 2 of the steps in Algorithm 2. Of course there are many possible ways to achieve these objectives, but it is crucial that any procedure preserve the target measure,

$$\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1}).$$

As the discussion below reveals, Algorithm 2 has this property.

Before the MCMC step can be discussed Step 4 in Algorithm 1 requires a minor modification. The new Step 4 will resample the pairs $\{(\mathbf{x}^i(j),\mathbf{x}'^i(j+1))\}_1^N$ instead of simply the $\{\mathbf{x}'^i(j+1)\}_1^N$. The result will be multiple copies of the pairs $(\mathbf{x}^i(j),\mathbf{x}'^i(j+1))$ for which $g(\mathbf{h}(j+1),\mathbf{x}'^i(j+1))$ is relatively large, and fewer of the pairs $(\mathbf{x}^i(j),\mathbf{x}'^i(j+1))$ for which $g(\mathbf{h}(j+1),\mathbf{x}'^i(j+1))$ is relatively small.

4′ Generate N independent uniform random variables, $\{\Theta^i(j)\}_{i=1}^N$, in $(0,1)$. For $i=1,\ldots,N$ let $(\mathbf{x}^{*i}(j),\mathbf{x}^i(j+1)) = (\mathbf{x}^k(j),\mathbf{x}'^k(j+1))$ where

$$\sum_{l=1}^{k-1} W^k(j+1) \leqslant \Theta^i(j) < \sum_{l=1}^{k} W^k(j+1).$$

Consider the consequence of this modification. Suppose that one has samples $\{\mathbf{x}^i(j)\}_1^N$ drawn from the posterior distribution at time j, $\mathbf{p}(\mathbf{x}(j)|\{\mathbf{h}(l)\}_1^j)$.

Given each $\mathbf{x}^i(j)$, the samples $\mathbf{x}'^i(j+1)$ are drawn from $p^j(\mathbf{x}(j+1)|\mathbf{x}^i(j))$. Thus the joint distribution of the pairs $\{(\mathbf{x}^i(j),\mathbf{x}'^i(j+1))\}_1^N$ is

$$p^j(\mathbf{x}(j+1)|\mathbf{x}(j))\,\mathbf{p}(\mathbf{x}(j)|\{\mathbf{h}(l)\}_1^j) = \mathbf{p}(\mathbf{x}(j),\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^j).$$

Thus when these samples are weighted by $W^i(j+1) \propto \mathbf{p}(\mathbf{h}(j+1)|\mathbf{x}'^i(j+1))$ they become approximately distributed according to the density

$$\frac{\mathbf{p}(\mathbf{h}(j+1)|\mathbf{x}(j+1))\mathbf{p}(\mathbf{x}(j),\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^j)}{\int \mathbf{p}(\mathbf{h}(j+1)|\mathbf{x}(j+1))\mathbf{p}(\mathbf{x}(j),\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^j)\,\mathbf{dx}(j)\,\mathbf{dx}(j+1)},$$

which, by Bayes' rule, is easily seen to be

$$\mathbf{p}(\mathbf{x}(j),\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1}). \tag{8}$$

In particular, the samples $\mathbf{x}^i(j+1)$ generated by the resampling in Step 4′ will be approximately drawn from the marginal distribution

$$\int \mathbf{p}(\mathbf{x}(j),\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1})\,\mathbf{dx}(j) = \mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1}),$$

which is the posterior distribution at time $j+1$, and the samples $\mathbf{x}^{*i}(j)$ are approximately drawn from the marginal distribution

$$\int \mathbf{p}(\mathbf{x}(j),\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1})\,\mathbf{dx}(j+1) = \mathbf{p}(\mathbf{x}(j)|\{\mathbf{h}(l)\}_1^{j+1}).$$

Now notice that the joint density in (8) can be factored as

$$\mathbf{p}(\mathbf{x}(j),\mathbf{x}(j+1)|\{\mathbf{h}(l)\}_1^{j+1}) = \mathbf{p}(\mathbf{x}(j+1)|\mathbf{x}(j),\mathbf{h}(j+1))\mathbf{p}(\mathbf{x}(j)|\{\mathbf{h}(l)\}_1^{j+1})$$

so that given $\mathbf{x}^{*i}(j)$, $\mathbf{x}^i(j+1)$ is an approximate sample from

$$\mathbf{p}(\mathbf{x}(j+1)|\mathbf{x}^{*i}(j),\mathbf{h}(j+1)).$$

Suppose that $P(\mathbf{y}(j+1)|\mathbf{x}(j+1))$ is a Markov chain transition kernel that preserves the density

$$\mathbf{p}(\mathbf{x}(j+1)|\mathbf{x}(j),\mathbf{h}(j+1)),$$

i.e..

$$\mathbf{p}(\mathbf{y}(j+1)|\mathbf{x}(j),\mathbf{h}(j+1)) = \int P(\mathbf{y}(j+1)|\mathbf{x}(j+1)) \times \mathbf{p}(\mathbf{x}(j+1)|\mathbf{x}(j),\mathbf{h}(j+1))\,d\mathbf{x}(j+1).$$

Then evolving the samples $\mathbf{x}^i(j+1)$ according to $P(\mathbf{y}(j+1)|\mathbf{x}(j+1))$ will yield new samples which are still approximately drawn from $\mathbf{p}(\mathbf{x}(j+1)|\mathbf{x}^{*i}(j),\mathbf{h}(j+1))$. Through this mechanism one can attempt to improve the samples generated by Algorithm 1. Indeed, if the Markov chain is Harris-recurrent and aperiodic (see [15]), then the resulting samples will asymptotically be drawn from $\mathbf{p}(\mathbf{x}(j+1)|\mathbf{x}^{*i}(j),\mathbf{h}(j+1))$. Thus if $\mathbf{Y}^{i,K}$ is the result of K iterations of P then as $K\to\infty$, the only error in the procedure will be due to the fact that $\mathbf{x}^{*i}(j)$ is not an exact sample from $\mathbf{p}(\mathbf{x}(j)|\{\mathbf{h}(l)\}_1^{j+1})$. In fact, in this case the convergence is monotonic in the total variation norm so that every step of the Markov chain improves the samples in this sense (see [15]). The resulting algorithm is,

Algorithm 2 (*Particle filter with MCMC*). One iteration of the particle filter algorithm with MCMC correction step is carried out as follows.

1. Begin with N unweighted samples $\mathbf{x}^i(j)$ from $\mathbf{p}(\mathbf{x}(j)|\{\mathbf{h}(l)\}_1^j)$.
2. Generate N samples$\mathbf{x}'^i(j+1)$ from $p^j(\mathbf{x}(j+1)|\{\mathbf{x}(j)=\mathbf{x}^i(j)\})$.
3. Evaluate the weights,

$$W^i(j+1) = \frac{g(\mathbf{h}(j+1),\mathbf{x}'^i(j+1))}{\sum_1^N g(\mathbf{h}(j+1),\mathbf{x}'^k(j+1))}.$$

4. Generate N independent uniform random variables, $\{\Theta^i(j)\}_{i=1}^N$, in $(0,1)$. For $i = 1,\dots,N$ let $(\mathbf{x}^{*i}(j), \mathbf{x}^{i,0}(j+1)) = (\mathbf{x}^k(j), \mathbf{x}^k(j+1))$ where

$$\sum_{l=1}^{k-1} W^k(j+1) \leqslant \Theta^i(j) < \sum_{l=1}^{k} W^k(j+1).$$

5. For each i, construct a Markov chain $\{\mathbf{Y}^{i,n}\}$ with initial values

$$\mathbf{Y}^{i,0} = \mathbf{x}^{i,0}(j+1)$$

and stationary distribution

$$\mathbf{p}(\mathbf{x}(j+1)|\{\mathbf{x}(j) = \mathbf{x}^{*i}(j)\}, \mathbf{h}(j+1)) \propto p^j(\mathbf{x}(j+1)|\{\mathbf{x}(j) = \mathbf{x}^{*i}(j)\})g(\mathbf{h}(j+1), \mathbf{x}(j+1)).$$

6. Let $\mathbf{x}^i(j+1) = \mathbf{Y}^{i,K}$ for each i.
7. Return to Step 1 with $j+1$ in place of j.

In general Step 5 of Algorithm 2 will require that one can evaluate the density $p^j(\mathbf{x}(j+1)|\mathbf{x}(j))$ at least up to a normalization constant. In many cases this is not possible. The next section addresses this issue.

4. Continuous time problems

In the discussion so far it has been assumed that the hidden signal $\mathbf{x}$ is a discrete time process which is observed at each time step. In the problem studied in the next section it is assumed that the underlying Markov process $\mathbf{x}(j)$ is a discrete sampling at observation times $s_0 < s_1 < s_2 < \cdots$ of an approximation to a stochastic differential equation

$$d\mathbf{x}(t) = F(\mathbf{x}(t))\,dt + \sigma dB(t), \tag{9}$$

where $B(t)$ is a $d_{\mathbf{x}}$-dimensional Brownian motion. This introduces several complications.

In this context the transition density $p^j(\mathbf{x}(j+1)|\mathbf{x}(j))$ is the density of a transition from position $\mathbf{x}(j)$ at time s_j to position $\mathbf{x}(j+1)$ at time s_{j+1}. In general it is not possible to sample directly from $p^j(\mathbf{x}(j+1)|\mathbf{x}(j))$ and one must use some numerical approximation scheme. This is accomplished by approximating the probability density $p^j(\mathbf{x}(j+1)|\mathbf{x}(j))$ by a product of transition densities over shorter time intervals (say of length Δ). These new transition densities, p^j_Δ, describe the probability of a transition from some position $\mathbf{x}(j,n)$ at time $s_j + n\Delta$ to a new position $\mathbf{x}(j,n+1)$ at time $s_j + (n+1)\Delta$ where $0 \leqslant n \leqslant N_j$ and $s_j + N_j\Delta = s_{j+1}$.

In fact it is useful to simply assume that the hidden signal evolves according to these approximate transition densities p^j_Δ, and replace the true transition density by

$$p^j(\mathbf{x}(j+1)|\mathbf{x}(j)) = \int \left(\prod_{n=0}^{N_j-1} p^j_\Delta(\mathbf{x}(j,n+1)|\mathbf{x}(j,n))\right) \prod_{n=1}^{N_j-1} d\mathbf{x}(j,n). \tag{10}$$

For example, one might choose to approximate (9) using the familiar Euler discretization,

$$\mathbf{x}(j,n+1) = \mathbf{x}(j,n) + F(\mathbf{x}(j,n))\Delta + \sigma\sqrt{\Delta}\xi(j,n), \quad 0 \leqslant n < N_j,\ \mathbf{x}(j,0) = \mathbf{x}(j), \tag{11}$$

where the $\xi(j,n)$ are independent Gaussian random variables with mean 0 and identity covariance. In this case,

$$p^j_\Delta(\mathbf{x}(j,n+1)|\mathbf{x}(j,n)) \propto \exp\left(-\frac{(\mathbf{x}(j,n+1) - \mathbf{x}(j,n) - F(\mathbf{x}(j,n))\Delta)^2}{2}\sigma^2\Delta\right). \tag{12}$$

While formula (10) defines a transition density for the Markov process $\mathbf{x}$ which can be easily sampled (via (11)), one cannot easily evaluate $p^j(\mathbf{x}(j+1)|\mathbf{x}(j))$. This is an important requirement of the MCMC step in Algorithm 2. To see this observe that the Markov chain constructed in Step 5 of Algorithm 2 must preserve the distribution

$$\mathbf{p}(\mathbf{x}(j+1)|\mathbf{x}(j), \mathbf{h}(j+1)) \propto p^j(\mathbf{x}(j+1)|\mathbf{x}(j))g(\mathbf{h}(j+1), \mathbf{x}(j+1)). \tag{13}$$

In general constructing such a chain will require that (13) can be evaluated at least up to a normalization constant. Of course this requires that $p^j(\mathbf{x}(j+1)|\mathbf{x}(j))$ can be evaluated at least up to a normalization constant.

Two difficulties arise when one attempts to evaluate $p^j(\mathbf{x}(j+1)|\mathbf{x}(j))$ using formula (10). The first and most obvious is that the integral in (10) cannot be easily evaluated. This problem can be avoided by constructing a larger Markov process in Step 5 of Algorithm 2 which preserves the joint conditional density of the entire path $\mathbf{x}(j,1), \mathbf{x}(j,2), \dots, \mathbf{x}(j+1)$

$$\mathbf{p}(\mathbf{x}(j,1), \mathbf{x}(j,2), \dots, \mathbf{x}(j+1)|\mathbf{x}(j), \mathbf{h}(j+1)) \propto \left(\prod_{n=0}^{N_j-1} p^j_\Delta(\mathbf{x}(j,n+1)|\mathbf{x}(j,n))\right) g(\mathbf{h}(j+1), \mathbf{x}(j+1)) \tag{14}$$

instead of the joint conditional density of $\mathbf{x}(j+1)$ alone (13). Suppose, for example, that

$$P(\mathbf{y}(j,1),\mathbf{y}(j,2),\ldots,\mathbf{y}(j+1)|\mathbf{x}(j,1),\mathbf{x}(j,2),\ldots,\mathbf{x}(j+1))$$

is any transition density which preserves (14), i.e.

$$\mathbf{p}(\mathbf{y}(j,1),\mathbf{y}(j,2),\ldots,\mathbf{y}(j+1)|\mathbf{x}(j),\mathbf{h}(j+1)) = \int P(\mathbf{y}(j,1),\mathbf{y}(j,2),\ldots,\mathbf{y}(j+1)|\mathbf{x}(j,1),\mathbf{x}(j,2),\ldots,\mathbf{x}(j+1)) \times \mathbf{p}(\mathbf{x}(j,1),\mathbf{x}(j,2),\ldots,\mathbf{x}(j+1)|\mathbf{x}(j),\mathbf{h}(j+1)) \times \mathbf{dx}(j,1)\mathbf{dx}(j,2)\cdots\mathbf{dx}(j+1).$$

This implies that if the initial path $\mathbf{x}(j,1),\mathbf{x}(j,2),\ldots,\mathbf{x}(j+1)$ is drawn from (14) then the sample $\mathbf{y}(j+1)$ generated by P will be distributed according to

$$\int \mathbf{p}(\mathbf{x}(j,1),\mathbf{x}(j,2),\ldots,\mathbf{x}(j+1)|\mathbf{x}(j),\mathbf{h}(j+1))\,\mathbf{dx}(j,1)\mathbf{dx}(j,2)\cdots\mathbf{dx}(j,N_j-1) = \mathbf{p}(\mathbf{x}(j+1)|\mathbf{x}(j),\mathbf{h}(j+1)).$$

The second difficulty presented by formula (10) that must be overcome in order to construct the Markov chain in Steps 5 and 6 of Algorithm 2 is that for many discretizations of (9) the resulting densities $p^j_\Delta(\mathbf{x}(j,n+1)|\mathbf{x}(j,n))$ cannot be efficiently evaluated. However, in many cases, the approximation chosen will depend on a collection of simple random variables whose joint distribution is known. For example, consider the explicit trapezoidal discretization

$$\mathbf{x}(j,n+1) = \mathbf{x}(j,n) + \Big(F(\mathbf{x}(j,n)) + \Delta F(\mathbf{x}(j,n)) + \sigma\sqrt{\Delta}\xi(j,n)) + F(\mathbf{x}(j,n))\Big)\frac{\Delta}{2} + \sigma\sqrt{\Delta}\xi(j,n), \tag{15}$$

where as before, the $\xi(j,n)$ are independent Gaussian random variables with mean 0 and identity covariance. This discretization will be used in the next section to approximate the flow of the Kuroshio current and yields a representation of p^j of the form (10). However, in this case it is difficult to evaluate the resulting factors $p^j_\Delta(\mathbf{x}(j,n+1)|\mathbf{x}(j,n))$.

Of course for each value of $\mathbf{x}(j)$ and each sequence of noise variables

$$\xi(j) = \xi(j,0),\ldots,\xi(j,N_j-1)$$

there is a unique value of $\mathbf{x}(j+1) = \mathbf{x}(j,N_j)$ determined by (15). This value will be denoted by $\mathbf{x}_\Delta(j+1,\xi(j))$. One could just as easily express $p^j(\mathbf{x}(j+1)|\mathbf{x}(j))$ in terms of the random variables $\xi(j,n)$ in (15) instead of the $\mathbf{x}(j,n)$, so that expression (10) becomes

$$p^j(\mathbf{x}(j+1)|\mathbf{x}(j)) \propto \int \exp\left(-\sum_{n=0}^{N_j-1}\frac{\xi(j,n)^T\xi(j,n)}{2}\right)\delta(\mathbf{x}_\Delta(j+1,\xi)-\mathbf{x}(j+1)) \times d\xi(j,0)\ldots,d\xi(j,N_j-1), \tag{16}$$

where the symbol δ in this expression represents the Dirac delta function.

Exactly as before, to avoid computing the integral in (16) one can construct a Markov chain in Steps 5 and 6 of Algorithm 2 which preserves the joint conditional density

$$\mathbf{p}(\xi(j,0),\ldots,\xi(j,N_j-1)|\mathbf{x}(j),\mathbf{h}(j+1)) \propto \exp\left(-\sum_{n=0}^{N_j-1}\frac{\xi(j,n)^T\xi(j,n)}{2}\right)g(\mathbf{h}(j+1),\mathbf{x}_\Delta(j+1,\xi))$$

instead of $\mathbf{p}(\mathbf{x}(j+1)|\mathbf{x}(j),\mathbf{h}(j+1))$. If one begins with a sample of the path $\xi(j,0),\ldots,\xi(j,N_j-1)$ which is drawn from $\mathbf{p}(\xi(j,0),\ldots,\xi(j,N_j-1)|\mathbf{x}(j),\mathbf{h}(j+1))$ then each sample generated by such a Markov chain would correspond to a value $\mathbf{x}_\Delta(j+1,\xi)$ which is distributed according to $\mathbf{p}(\mathbf{x}(j+1)|\mathbf{x}(j),\mathbf{h}(j+1))$.

These modifications of Algorithm 2 are summarized in the following algorithm.

Algorithm 3 (*Particle filter with MCMC for SDE with discrete observations*). One iteration of the particle filter with an MCMC correction for a continuous time Markov process is carried out as follows.

1. `Begin with` N `unweighted samples` $\mathbf{x}^i(j)$ `from` $\mathbf{p}(\mathbf{x}(j)|\{\mathbf{h}(l)\}_1^j)$.
2. `For each` i `generate a sample` $(\xi^i(j,0),\ldots,\xi^i(j,N_j-1))$ from the joint density proportional to

$$\exp\left(-\sum_{n=0}^{N_j-1}\frac{\xi(j,n)^T\xi(j,n)}{2}\right).$$

3. `Evaluate the weights`,

$$W^i(j+1) = \frac{g(\mathbf{h}(j+1),\mathbf{x}_\Delta(j+1,\xi^i))}{\sum_1^N g(\mathbf{h}(j+1),\mathbf{x}_\Delta(j+1,\xi^k))}.$$

4. `Generate` N `independent uniform random variables`, $\{\Theta(j)\}_{i=1}^N$, in $(0,1)$. For $i=1,\ldots,N$ let

$$(\mathbf{x}^{*i}(j),\xi^{i,0}(j,0)\ldots,\xi^{i,0}(j,N_j-1)) = (\mathbf{x}^k(j),\xi^k(j,0),\ldots,\xi^k(j,N_j)),$$

where

$$\sum_{l=1}^{k-1} W^k(j+1) \leqslant \Theta(j) < \sum_{l=1}^{k} W^k(j+1).$$

5. For each i, construct a Markov chain $\{\mathbf{Y}^{i,n}\}$ with initial value

$$\mathbf{Y}^{i,0} = (\xi^{i,0}(j,0), \ldots, \xi^{i,0}(j,N_j-1))$$

and stationary distribution

$$\mathbf{p}(\xi(j,0), \ldots, \xi(j,N_j-1) | \{\mathbf{x}(j) = \mathbf{x}^{*i}(j)\}, \mathbf{h}(j+1)) \propto \exp\left(-\sum_{n=0}^{N_j-1} \frac{\xi(j,n)^T \xi(j,n)}{2}\right) g(\mathbf{h}(j+1), \mathbf{x}_{\Delta}(j+1,\xi))).$$

6. Let $\mathbf{x}^i(j+1) = \mathbf{x}_{\Delta}(j+1, \mathbf{Y}^{i,K})$.
7. Return to Step 1 with $j+1$ in place of j.

The implementation and choice of MCMC method in Steps 5 and 6 of Algorithms 2 and 3 are, of course, key to the success or failure of this filtering strategy. This point cannot be overemphasized. It is certain that any naive choice of an MCMC method will produce a filter that is extremely expensive and/or ineffective. The MCMC method chosen for the computations reported on in Section 6 is a combination of the hybrid Monte Carlo method (HMC) and parallel marginalization (PMMC). In order to move on to a description of the model problem and the results of the numerical tests, a discussion of these techniques is postponed until Appendix A.

5. A bimodal ocean current model

The filtering approach outlined above will be tested on a discrete stochastic system obtained informally from the stochastic partial differential equation,

$$\frac{\partial}{\partial t}\mathbf{x} = -\nabla \cdot (u\mathbf{x}, v\mathbf{x}) - f\left(\frac{f_x}{f} - \frac{h_x}{h}\right)u - f\left(\frac{f_y}{f} - \frac{h_y}{h}\right)v + \nu\Delta\mathbf{x} + \sigma\omega \tag{17}$$

in the domain D of Fig. 3, where $\omega(t,x,y)$ is a white noise in space and time with covariance $\mathbf{E}[\sigma w(t-t',x-x',y-y')\sigma w(t-t',x-x',y-y')] = 6\times 10^{-13}\delta(t-t')\delta(x-x')\delta(y-y')s^{-4}$ and u and v are velocities found from $\mathbf{x}(t,x,y)$. This equation should not be strictly interpreted. Indeed the behavior of solutions, or even the sense in which they might exist, is not the purpose of the current study. I am more interested in the discrete system obtained informally from this system. I do not assume that my discretization converges in any meaningful way to an exact trajectory of (17). However, as I show later, the discrete system that emerges demonstrates a bimodal behavior that is qualitatively similar to that of the Kuroshio current. This characteristic, along with its size and complexity, make the resulting discrete system an interesting validation for the filtering technique advocated in this paper.

The system (17) is a stochastic perturbation of the barotropic vorticity equation shown by Chao in [13] to model the large and small meander states exhibited by the Kuroshio current (see Fig. 4). It The coordinates x, y are rotated 20° counter-clockwise from North–South. The viscosity is set to $\nu = 0.8\times 10^7$ cm^2 s^{-1} and the noise parameter σ is. The Coriolis parameter is given by

$$f = f_0 + f_x x + f_y y,$$

where

$$f_x = \beta\sin(20\text{deg}) \quad \text{and} \quad f_y = \beta\cos(20\text{deg})$$

and β and f_0 are given by $\beta = 2\times 10^{-13}$ cm^{-1} s^{-1} and $f_0 = 7\times 10^{-5}$ s^{-1}. The function $h(x,y)$ is the water depth and is 1000 m away from the two bumps that model the Izu Ridge. The northern bump is defined by

$$h_N(x,y) = 500\text{ m}\cos\left(\frac{\pi}{2}\frac{\sqrt{(x-1410\text{ km})^2 + (y-1020\text{ km})^2}}{90\text{ km}}\right)$$

for

$$\sqrt{(x-1410\text{ km})^2 + (y-1020\text{ km})^2} \leqslant 90\text{ km}$$

and the southern bump is defined by

$$h_S(x,y) = 500\text{ m}\cos\left(\frac{\pi}{4}\sqrt{\left(\frac{x'}{120\text{ km}}\right)^2 + \left(\frac{y'}{90\text{ km}}\right)^2}\right)$$

for

$$\sqrt{\left(\frac{x'}{120\text{ km}}\right)^2+\left(\frac{y'}{90\text{ km}}\right)^2}\leqslant 1,$$

where

$$x'=\frac{(x-1410\text{ km})+(y-780\text{ km})}{\sqrt{2}}$$

and

$$y'=\frac{(x-1410\text{ km})-(y-780\text{ km})}{\sqrt{2}}.$$

The horizontal velocities u and v satisfy

$$(hu,hv)=(-\psi_y,\psi_x),$$

where the volume transport streamfunction ψ solves

$$\boldsymbol{x}=\frac{\partial}{\partial x}\left(\frac{1}{h}\psi_x\right)+\frac{\partial}{\partial y}\left(\frac{1}{h}\psi_y\right).$$

The boundary conditions are

$$\begin{aligned}
&I \quad \psi=0, \quad \boldsymbol{x}=0, \quad \text{at } y=0,\\
&II \quad \psi=-33\text{ Sv}, \quad \psi_n=0, \quad \text{along the northern boundary},\\
&III \quad \psi_x=0, \quad \psi_{xx}=0, \quad \text{at } x=0,\\
&IX \quad \psi=K(y), \quad \psi_{xx}=0, \quad \text{at } x=2220\text{ km},
\end{aligned}$$

where

$$K(y)=0 \quad \text{for } y\leqslant 870\text{ km},$$

and

$$K(y)=-33\,Sv\frac{y-870\text{ km}}{150\text{ km}} \quad \text{for } y>870\text{ km}.$$

In the above formulas an Sv is a Sverdrup and represents a volume transport of $10^6\text{ m}^3\text{ s}^{-1}$.

Now let $\Delta_x=30$ km denote the spatial mesh size which is the same in both the x and y directions. For any function g on D define

$$g_{k+\alpha_x,l+\alpha_y}=g((k+\alpha_x)\Delta_x,(l+\alpha_y)\Delta_x)$$

for $\alpha_x,\alpha_y\in[-1,1]$. Define the operators

$$\begin{aligned}
&\delta_x g=\frac{g_{k+1/2,l}-g_{k-1/2,l}}{\Delta_x} \quad \delta_y g=\frac{g_{k,l+1/2}-g_{k,l-1/2}}{\Delta_x},\\
&\mu_x g=\frac{g_{k+1/2,l}+g_{k-1/2,l}}{2}, \quad \mu_y g=\frac{g_{k,l+1/2}+g_{k,l-1/2}}{2},\\
&D_x^0=\mu_x\delta_x, \quad D_y^0=\mu_y\delta_y,
\end{aligned}$$

and

$$L^0=\delta_x\left(\frac{\delta_x}{h}\right)+\delta_y\left(\frac{\delta_y}{h}\right). \tag{18}$$

First, (17) is discretized in space using a simple centered difference scheme which involves only values of $\boldsymbol{x}$ at points $(k\Delta_x,j\Delta_x)$. After replacing $\boldsymbol{x}$ by its restriction to these points the system becomes a set of ordinary stochastic differential equations,

$$d\mathbf{x}_{k,m}(t)=F_{k,m}(\mathbf{x}(t))dt+\frac{1}{\Delta_x}\sigma dB_{k,m}(t), \tag{19}$$

where

$$F_{k,m}(\mathbf{x}(t))=-D_x^0(u_{k,j}\mathbf{x}_{k,j})-D_y^0(v_{k,m}\mathbf{x}_{k,m})-f_{k,m}\left(\frac{f_x}{f_{k,m}}+D_x^0\left(\frac{1}{h}\right)\right)u_{k,m}-f_{k,m}\left(\frac{f_y}{f_{k,m}}+D_y^0\left(\frac{1}{h}\right)\right)v_{k,m}+\nu(\delta_x\delta_x+\delta_y\delta_y)\mathbf{x}_{k,m}, \tag{20}$$

where ψ solves

$$L^0\psi = \mathbf{x}, \tag{21}$$

and

$$u_{k,m} = -\frac{D_y^0\psi_{k,m}}{h}, \quad v_{k,m} = \frac{D_x^0\psi_{k,m}}{h}.$$

The $B_{k,m}$ are independent Brownian motions. Consistent with the previous sections, $\mathbf{x}(j)$ denotes the solution of (22) at the time of jth observation s_j, and $\mathbf{x}(j,n)$ denotes the solution at the nth time step after the jth observation, $s_j + n\Delta$.

These stochastic ordinary differential equations cannot be solved explicitly and therefore require numerical solution. Here the spatially discretized system (19) is discretized in time as

$$\mathbf{x}_{k,m}(j,n+1) = \mathbf{x}_{k,m}(j,n) + (F_{k,m}(\check{\mathbf{x}}_{k,m}(j,n+1)) + F_{k,m}(\mathbf{x}_{k,m}(j,n)))\frac{\Delta}{2} + \frac{\sqrt{\Delta}}{\Delta_x}\sigma\xi_{k,m}(j,n), \tag{22}$$

where

$$\check{\mathbf{x}}_{k,m}(j,n+1) = \mathbf{x}_{k,m}(j,n) + F_{k,m}(\mathbf{x}_{k,m}(j,n))\Delta + \frac{\sqrt{\Delta}}{\Delta_x}\sigma\xi_{k,m}(j,n),$$

and for each k, m, j, and n, $\xi_{k,m}(j,n)$ is an independent Gaussian random variable with mean 0 and variance 1. The resulting method is adequate for the relatively low Reynolds number flow considered here. A Crank Nicholson type scheme was not applied to the linear part of the equation because the stiffness of the system is not dominated by the diffusion term at the level of discretization used here.

The discrete system above exhibits two metastable states that are qualitatively similar to the small and large meanders of the actual Kuroshio current. Fig. 4 shows typical states in both of these meanders. They were found by varying the northern boundary condition as in [13]. The noise parameter in (22) (σ) was set to 0 for the purposes of generating Fig. 4.

Let $(\mathbf{x}^*, \mathbf{y}^*)$ denote the point in D that is 990 km from the western boundary and 860 km from the southern boundary. This point is pictured in Fig. 3. Let (k^*, m^*) denote the location of this point on the discrete grid. The bimodality of the system is evident in Fig. 5 which shows a long trajectory of the system projected onto the variable ψ_{k^*,m^*}. The state for which $\psi_{k^*,m^*} \approx 15\,Sv$ roughly corresponds to the small meander and the state for which $\psi_{k^*,m^*} \approx -20\,Sv$ roughly corresponds to the large meander. As can also be seen in Fig. 5 the discrete stochastic system tends to remain in each of its meanders for roughly 10 years. Transitions between the two meanders usually occur in a time span of a few months.

The observation process is given by

$$\mathbf{h}(j) = \psi(j)_{k^*,m^*} + \chi(j) \quad \chi(j) \sim \mu, \tag{23}$$

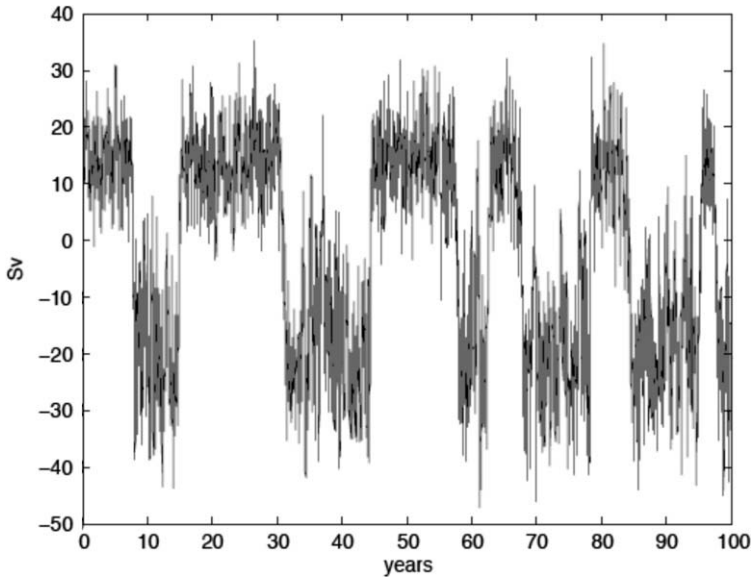

Fig. 5. Time series of approximation to $\psi(\mathbf{x}^*, \mathbf{y}^*)$ where $(\mathbf{x}^*, \mathbf{y}^*)$ denotes the point in D that is 990 km from the western boundary and 860 km from the southern boundary. Notice the transitions between a metastable state near $15\,Sv$ and one near $-20\,Sv$.

where

$$\mu(\mathbf{x}) = \exp\left(-\frac{\mathbf{x}^2}{200}\right).$$

Thus the discrete vorticity process $\mathbf{x}(j,n)$ is observed through the value of the discrete volume transport process $\psi(j)$ at the single point (k^*, m^*).

6. Numerical results and discussion

In this section, I present the results of an application of Algorithm 3 to a filtering problem for the discrete system (22) given above with observation model (23). The MCMC method chosen for Step 5 of Algorithm 3 is a combination of hybrid Monte Carlo and parallel marginalization as in Algorithm 6 in Appendix A.3. The choice of hybrid Monte Carlo is motivated in part by the presence in (20) of the velocities u and v which depend on the volume transport ψ. In light of Eq. (21), if one makes even a $1-d$ perturbation, say $\mathbf{x}_{k,m}(j,n)+\epsilon$, then in order to calculate π at the new state the drift term, (20), must be recomputed for all components. This fact virtually rules out any method that cannot maintain reasonable acceptance rates while making global proposals (perturbations of $\mathbf{x}_{k,m}(j,n)$ for all (k,m) at once). Hybrid Monte Carlo is capable of making such global proposals with high acceptance rates.

The other motivating factor in the choice of hybrid Monte Carlo is the use of the change of variables from the position variables $\mathbf{x}(j,\cdot)$ to the noise variables $\xi(j,\cdot)$ discussed in Section 4. Suppose a Metropolis-Hastings MCMC proposal density suggests perturbations of only one of the noise terms $\xi(j,n)$ at a time, i.e. perturbations of the form

$$\xi(j,n) \to \xi(j,n)+\epsilon \quad \text{and} \quad \xi(j,m) \to \xi(j,m) \quad \text{for } m \neq n.$$

Then, in order to evaluate the acceptance probability for this proposal, the discrete system (22) must be evolved from time step (j,n) to time step (j,N_j), making evaluation of the acceptance prohibitively expensive. The hybrid Monte Carlo method avoids this difficulty by perturbing all increments in one step. The HMC proposals also take into account the effect of the observation on each increment. The use of hybrid Monte Carlo for the path smoothing problem has been suggested in [7,8]. The change of variables to the noise variables can be thought of as a very simple preconditioning of the HMC sampling (see [16]). The system is easier to sample in the new variables because the correlations between the noise variables given the observations are much weaker than the correlations between neighboring time steps of $\mathbf{x}$. In fact, due to this preconditioning, the effect of incorporating parallel marginalization is not nearly as pronounced as observed in [9]. Nevertheless, in repeated runs on this test problem, the addition of the PMMC step seems to reduce the equilibration time (measured in CPU time) of the Markov chain in Steps 5 and 6 of Algorithm 3 by a factor between 2 and 3.

In this test, the observations are fixed at $\mathbf{h}(j) = 19.2918\ Sv$ for all j which is the value of $\psi(\mathbf{x}^*, \mathbf{y}^*)$ for the small meander state shown in Fig. 4. The system is started in the large meander state shown in Fig. 4. These choices test the ability of the filter to adjust to a sudden change of the system from one metastable state to another. The time step Δ_j is chosen to be 0.00526 days and the observation times are $s_1 = 2.63$ days, $s_2 = 2(2.63) = 5.26$ days, $\ldots$, $s_{10} = 26.3$ days.

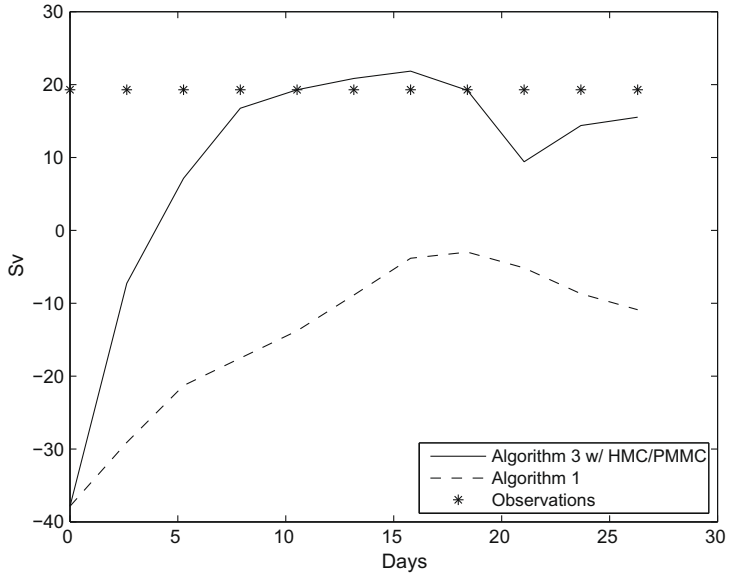

Fig. 6. Trajectory of estimate of $\psi(\mathbf{x}^*, \mathbf{y}^*)$ given observation for Algorithm 3 with HMC/PMMC and a particle filter. $(\mathbf{x}^*, \mathbf{y}^*)$ is the point in D that is 990 m from the western boundary and 860 m from the southern boundary.

The performance of Algorithm 3 with HMC/PMMC on this test problem is compared to the performance of a standard particle filter (see Algorithm 1). For other applications of particle filters to models related to geophysical problems see, for example, [17,18]. Fig. 6 shows the paths of the estimators generated by Algorithm 3 with $N = 10$ particles and by a standard particle filter with $N = 1000$ particles. The number of parallel marginalization levels ($L + 1$ in Algorithms refpm1 and 6) is chosen to be 4 and the number of MCMC iterations, K, is 3. The acceptance rates for HMC were higher at the higher levels (l in Algorithms 5 and 6) for a given step size δ and number of steps in the proposal mapping (φ_δ) described in Section A.1. However, it proved much more efficient to use more iterations of (φ_δ) at the higher levels. For this reason the number or iterations of the proposal mapping φ_δ (M) is set to 1, 2, 4, and 8 on levels 0,1,2, and 3 respectively. This results in similar HMC acceptance rates at all levels as well as similar cost per HMC iteration.

As programed, Algorithm 3 requires about 100 times more work per particle than the standard particle filter (when the state of all samples at each time between the current and future observation are stored). However, as is evident in Fig. 6, Algorithm 3 requires many fewer observations to adjust to the new state of the system. This is reinforced by the results shown in Fig. 7. This figure shows the weighted empirical densities of $\psi(\mathbf{x}^*, \mathbf{y}^*)$ at each observation time generated by the two algorithms. Clearly the weighted empirical density generated by the standard particle filter rarely has more than one statistically significant sample. Therefore, despite the relatively low number of particles used in Algorithm 3, it offers a higher resolution. These results indicate that the particle filter requires many more particles (than the 1000 used in this run) to

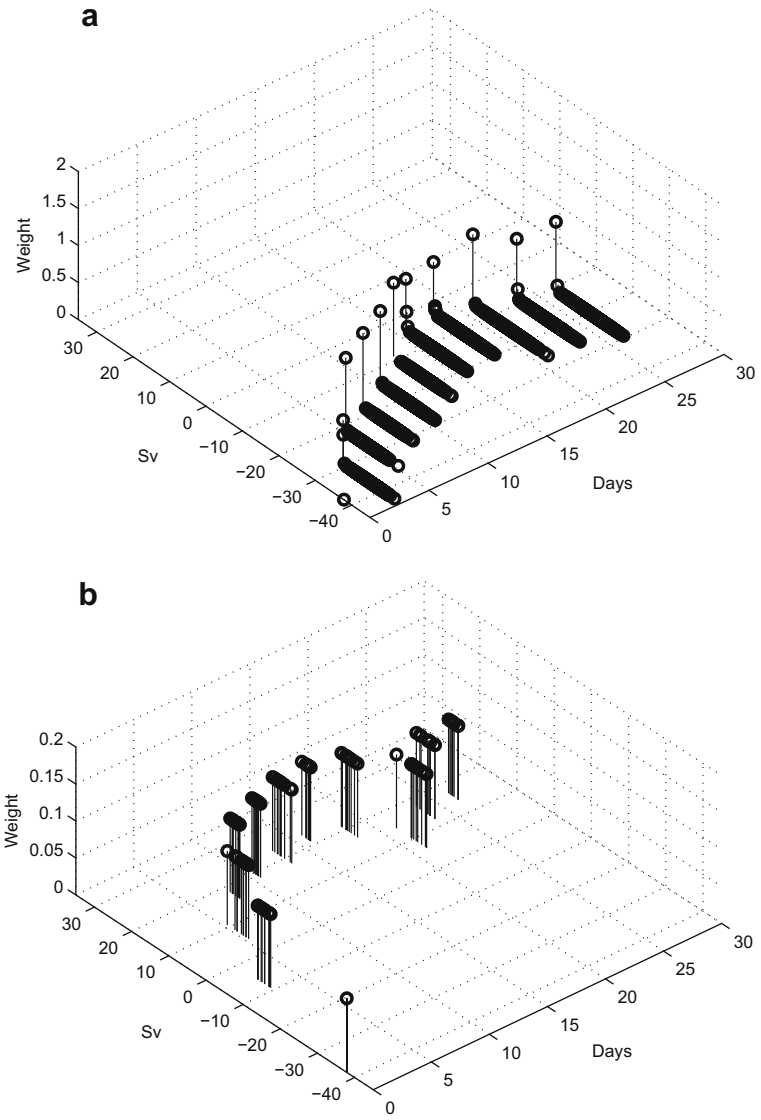

Fig. 7. (a) Weighted empirical distribution of $\psi(\mathbf{x}^*, \mathbf{y}^*)$ at each observation time generated by the standard particle filter with 1000 particles. (b) Weighted empirical distribution of $\psi(\mathbf{x}^*, \mathbf{y}^*)$ at each observation time generated by Algorithm 3 with 10 particles.

give an accurate estimate of the state of the system and would therefore be much more expensive than Algorithm 3. In fact in tests with as many as 4000 particles the performance of the standard particle filter did not improve appreciably. This is promising since like a particle filter, Algorithm 3 is very general and can be applied to problems with significant non-linear and non-Gaussian effects. It is interesting to note that while Algorithm 3 seems to have converged, neither method produces an estimate of the state of the full system which is consistent with the small meander state. This indicates that the value of the single variable $\psi(\mathbf{x}^*, \mathbf{y}^*)$ is not enough to fully specify the mode of the full system.

It is likely that the method could be implemented much more efficiently. For example, the force appearing in the HMC step is very expensive to evaluate exactly. A possible solution to this problem is offered by the surrogate transition method described in [19]. However, the increased cost due to the MCMC step indicates that this algorithm is best suited for problems with posterior distributions which can be effectively represented by a small number of samples, but which nonetheless require a large number of particles using a standard approach.

Acknowledgments

I am grateful to Professor A. Chorin for his guidance during this research, which was carried out while I was a Ph.D. student at U. C. Berkeley. I would also like to thank Professor O. Hald, Professor P. Stinis, and Dr. Xuemin Tu, for their very helpful comments. This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098 and National Science Foundation grant DMS0410110 as well as by the Applied Mathematical Sciences Program of the U.S. Department of Energy under Contract DEFG0200ER25053.

Appendix A. Parallel marginalization and hybrid Monte Carlo

There are many choices for the Markov chain Monte Carlo method in Steps 5 and 6 of Algorithms 2 and 3. A particularly effective choice seems to be the combination of parallel marginalization (PMMC) and hybrid Monte Carlo (HMC) employed in the numerical study discussed in Section 6. The next two subsections contain a brief description of both PMMC and HMC in the general context of constructing a Markov chain to sample from some target density π_0. In the setting of Section 4 the target density π_0 is the conditional density,

$$\begin{aligned}\pi_0(\xi(j,0),\ldots,\xi(j,N_j-1)) &= \mathbf{p}(\xi(j,0),\ldots,\xi(j,N_j-1)|\mathbf{x}(j),\mathbf{h}(j+1))\\ &\propto \exp\left(-\sum_{n=0}^{N_j-1}\frac{\xi(j,n)^T\xi(j,n)}{2}\right)g(\mathbf{h}(j+1),\mathbf{x}_\Delta(j+1,\xi)))\end{aligned}$$

as required in Step 5 of Algorithm 3. Section A.3 below focuses on the implementation of HMC and PMMC in this setting.

A.1. Hybrid Monte Carlo

The hybrid Monte Carlo (HMC) scheme is a variation of the Metropolis Hastings scheme (see [20]) and was first introduced by [21] as a way to make large Metropolis-Hastings proposals without suffering low acceptance rates. Note that the target density can be written $\pi_0(\mathbf{x}) = \exp(-\mathscr{V}(\mathbf{x}))$ where $\mathscr{V}(\mathbf{x}) = -\log(\pi_0(\mathbf{x}))$. The first step is to augment the system by a vector of random variables, $\mathbf{r} \in \mathbb{R}^d$, such that the joint density of $\mathbf{x}$ and $\mathbf{r}$ is given by,

$$\frac{\exp\left(-\mathscr{H}(\mathbf{x},\mathbf{r})\right)}{\mathscr{Z}},$$

where

$$\mathscr{H}(\mathbf{x},\mathbf{r}) = \mathscr{V}(\mathbf{x}) + \mathscr{K}(\mathbf{r})$$

for some potential function $\mathscr{K}(\mathbf{r})$ and

$$\mathscr{Z} = \int e^{-\mathscr{K}(\mathbf{r})}\,\mathbf{dr}.$$

Notice that the marginal distribution of the $\mathbf{x}$ variables is $\pi_0(\mathbf{x})$, i.e.

$$\int \frac{\exp\left(-\mathscr{H}(\mathbf{x},\mathbf{r})\right)}{\mathscr{Z}}\,\mathbf{dr} = \pi_0(\mathbf{x}).$$

Recall that the solution to the Hamiltonian system,

$$\begin{aligned}\frac{d\mathbf{x}}{dt} &= \nabla_r\mathscr{H}(\mathbf{x},\mathbf{r}) = \nabla\mathscr{K}(\mathbf{r}),\\ \frac{d\mathbf{r}}{dt} &= -\nabla_x\mathscr{H}(\mathbf{x},\mathbf{r}) = -\nabla\mathscr{V}(\mathbf{x})\end{aligned} \tag{24}$$

preserves the value of $\mathscr{H}(\mathbf{x},\mathbf{r})$, and thus the density $\frac{\exp(-\mathscr{H}(\mathbf{x},\mathbf{r}))}{\mathscr{Z}}$ is stationary for this system of ordinary differential equations. Of course the fact that the value of $\mathscr{H}$ is conserved along trajectories of (24) also implies that these trajectories cannot visit all configurations. However, the properties of (24) can be used to define a viable Markov chain Monte Carlo method to sample $\pi_0(\mathbf{x})$.

For simplicity choose,

$$\mathscr{K}(\mathbf{r}) = \frac{\mathbf{r}^T\mathbf{r}}{2}.$$

Define the evolution map $\varphi_\delta : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^d$ by $\varphi_\delta = (\varphi^x_\delta(\mathbf{x},\mathbf{r}), \varphi^r_\delta(\mathbf{x},\mathbf{r}))$ where

$$\varphi^x_\delta(\mathbf{x},\mathbf{r}) = \mathbf{x} + \delta\mathbf{r} - \frac{\delta^2}{2}\nabla\mathscr{V}(\mathbf{x}),$$

$$\varphi^r_\delta(\mathbf{x},\mathbf{r}) = \mathbf{r} - \frac{\delta}{2}(\nabla\mathscr{V}(\mathbf{x}) + \nabla\mathscr{V}(\varphi^x_\delta(\mathbf{x},\mathbf{r}))).$$

This is the velocity Verlet discretization of the Hamiltonian system (see [22,23]). The important features of this discretization are that it is time reversible and area preserving. The hybrid Monte Carlo step from the point $\mathbf{Y}^n = \mathbf{x}$ consist of first generating an independent sample of $\mathbf{r}$ from the density proportional to $e^{-\mathscr{K}(\mathbf{r})}$ and then evolving the point $(\mathbf{x},\mathbf{r})$ under φ_δ for M steps to generate the point $(\mathbf{y},\mathbf{r}') = (\varphi_\delta)^M(\mathbf{x},\mathbf{r})$. We then set $\mathbf{Y}^{n+1} = \mathbf{y}$ with probability

$$A = \min\left\{1, \frac{\pi_0(\mathbf{y},\mathbf{r}')}{\pi_0(\mathbf{x},\mathbf{r})}\right\},$$

and $\mathbf{Y}^{n+1} = \mathbf{x}$ with probability $1 - A$. The properties of φ_δ imply that $\pi_0(\mathbf{x})$ is the stationary distribution for $\mathbf{Y}^n$ (see [20]). By decreasing δ and increasing M, the method can maintain large global proposals without suffering from low acceptance rates. The cost, of course, is the expense of increasing the iterations of φ_δ and therefore the increased time required to generate a proposal.

Suppose the current position of the Markov chain $\{\mathbf{Y}^n\}$ is $\mathbf{Y}^n = \mathbf{x}$.

Algorithm 4 (*HMC*). The chain moves from $\mathbf{Y}^n$ to $\mathbf{Y}^{n+1}$ as follows:

1. `Generate` d `independent Gaussian random variables with mean 0 and variance 1 (i.e., a sample of` $\mathbf{r}$`).`
2. `Evaluate` $(\mathbf{y},\mathbf{r}') = (\varphi_\delta)^M(\mathbf{x},\mathbf{r})$.
3. `Set` $\mathbf{Y}^{n+1} = \mathbf{y}$ `with probability`

$$A = \min\left\{1, \frac{\pi_0(\mathbf{y},\mathbf{r}')}{\pi_0(\mathbf{x},\mathbf{r})}\right\}$$

and $\mathbf{Y}^{n+1} = \mathbf{x}$ with probability $1 - A$.

A.2. Parallel marginalization

It is well known that for many distributions, appropriately chosen marginal distributions exhibit reduced spatial correlations. Spatial correlations often translate to long temporal correlations and slow convergence for MCMC methods. Recently a new Markov chain Monte Carlo method has been introduced (see [9]) which uses approximate marginal distributions of π_0 to accelerate MCMC sampling. Auxiliary Markov chains that sample approximate marginal distributions are evolved simultaneously with the Markov chain that samples the distribution of interest. By swapping their configurations, these auxiliary chains pass information between themselves and with the chain sampling the original distribution. For details of the construction the reader is directed to reference [9,24]. Parallel marginalization is closely related to work in [25,26]. It bears some resemblance to the multigrid Monte Carlo method suggested in [27].

Suppose that, by the Metropolis-Hastings or any other method (see [20]), one can construct a Markov chain, $\mathbf{Y}^n_0 \in \mathbb{R}^d$, which has π_0 as its stationary measure. That is, for two points $\mathbf{x}_0, \mathbf{y}_0 \in \mathbb{R}^d$

$$\int \tau_0(\mathbf{y}_0|\mathbf{x}_0)\pi_0(\mathbf{x}_0)\, d\mathbf{x}_0 = \pi_0(\mathbf{y}_0),$$

where $\tau_0(\mathbf{y}_0|\mathbf{x}_0)$ is the probability density of a move to $\{\mathbf{Y}^{n+1}_0 = \mathbf{y}_0\}$ given that $\{\mathbf{Y}^n_0 = \mathbf{x}_0\}$.

In order to take advantage of the shorter spatial correlations exhibited by marginal distributions of π_0, a collection of lower dimensional Markov chains which approximately sample marginal distributions of π_0 is considered. Let $\mathbf{x}_0$ be distributed according to π_0. In other words, $\mathbf{x}_0$ is the random variable to be simulated. Decompose the d components of $\mathbf{x}_0$ into two subsets,

$$\mathbf{x}_0 = (\bar{\mathbf{x}}_0, \tilde{\mathbf{x}}_0),$$

where $\bar{\mathbf{x}}_0$ has d_1 components and $\tilde{\mathbf{x}}_0$ has $d - d_1$ components. Recall that the $\bar{\mathbf{x}}_0$ variables are distributed according to the marginal density,

$$\bar{\pi}_0(\bar{\mathbf{x}}_0) = \int \pi_0(\bar{\mathbf{x}}_0, \tilde{\mathbf{x}}_0)\, d\tilde{\mathbf{x}}_0, \tag{25}$$

and that given the value of the $\bar{\mathbf{x}}_0$ variables, the $\tilde{\mathbf{x}}_0$ variables are distributed according to the conditional density,

$$\pi(\tilde{\mathbf{x}}_0|\bar{\mathbf{x}}_0) = \frac{\pi_0(\bar{\mathbf{x}}_0, \tilde{\mathbf{x}}_0)}{\bar{\pi}_0(\bar{\mathbf{x}}_0)}. \tag{26}$$

Now suppose that an approximation to the marginal distribution of the $\bar{\mathbf{x}}_0$ variables,

$$\pi_1(\bar{\mathbf{x}}_0) \approx \bar{\pi}_0(\bar{\mathbf{x}}_0)$$

is available. Let $\mathbf{x}_1 \in \mathbb{R}^{d_1}$ be independent of the $\mathbf{x}_0$ random variables and drawn from $\pi_1(\bar{\mathbf{x}}_0)$. Notice that $\mathbf{x}_1$ represents the same physical variables as $\bar{\mathbf{x}}_0$ though its probability density is not the exact marginal density. One can continue in this way to remove variables from the system by decomposing $\mathbf{x}_l \in \mathbb{R}^{d_l}$ into proper subsets as

$$\mathbf{x}_l = (\bar{\mathbf{x}}_l, \tilde{\mathbf{x}}_l),$$

and defining $\mathbf{x}_{l+1} \in \mathbb{R}^{d_{l+1}}$ to be independent of the $\{\mathbf{x}_0, \ldots, \mathbf{x}_l\}$ random variables and drawn from an approximation π_{l+1} to $\bar{\pi}_l(\bar{\mathbf{x}}_l)$. Clearly each $\mathbf{x}_{l+1}$ represents fewer physical variables than $\mathbf{x}_l$.

Just as one can construct a Markov chain $\mathbf{Y}_0^n \in \mathbb{R}^d$ to sample $\mathbf{x}_0$, one can also construct Markov chains $\mathbf{Y}_l^n \in \mathbb{R}^{d_l}$ to sample π_l. In other words, for each $\mathbf{Y}_l^n$ choose a transition probability density τ_l, such that

$$\int \tau_l(y_l|\mathbf{x}_l)\pi_l(\mathbf{x}_l)\, d\mathbf{x}_l = \pi_l(y_l)$$

for all i.

The chains $\mathbf{Y}_l^n$ can be arranged in parallel to yield a larger Markov chain,

$$\mathbf{Y}^n = (\mathbf{Y}_0^n, \ldots, \mathbf{Y}_L^n) \in \mathbb{R}^d \times \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_L}.$$

The probability density of a move to $\{\mathbf{Y}^{n+1} = \mathbf{y}\}$ given that $\{\mathbf{Y}^n = \mathbf{x}\}$ for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d \times \mathbb{R}^{d_1} \times \ldots \times \mathbb{R}^{d_L}$ is given by

$$\tau(\mathbf{y}|\mathbf{x}) = \prod_{l=0}^{L} \tau_l(\mathbf{y}_l|\mathbf{x}_l). \tag{27}$$

Since

$$\int \left(\tau(\mathbf{y}|\mathbf{x}) \prod_{l=0}^{L} \pi_l(\mathbf{x}_l) \right) d\mathbf{x}_0 \cdots d\mathbf{x}_L = \prod_{l=0}^{L} \pi_l(\mathbf{y}_l)$$

the stationary distribution of $\mathbf{Y}^n$ is

$$\Pi(\mathbf{x}_0, \ldots, \mathbf{x}_L) = \pi_0(\mathbf{x}_0) \cdots \pi_L(\mathbf{x}_L).$$

This chain would have no interaction between its various components and will therefore not equilibrate quickly. The next step in the construction is to design a new set of transition kernels ψ_l which allow for interactions between the chains sampling from the τ_l and to thereby pass information from the rapidly equilibrating chains on the lower dimensional spaces (large l) down to the chain on the original space ($l = 0$). This is accomplished by swap moves. In a swap move between levels l and $l + 1$, a subset, $\bar{\mathbf{x}}_l \in \mathbb{R}^{d_{l+1}}$, of the $\mathbf{x}_l$ variables is exchanged with the $\mathbf{x}_{l+1} \in \mathbb{R}^{d_{l+1}}$ variables. For the full chain, this swap takes the form of a move from $\{\mathbf{Y}^n = \mathbf{x}\}$ to $\{\mathbf{Y}^{n+1} = \mathbf{y}\}$ where

$$\mathbf{x} = (\ldots, (\bar{\mathbf{x}}_l, \tilde{\mathbf{x}}_l), \mathbf{x}_{l+1}, \ldots)$$

and

$$\mathbf{y} = (\ldots, (\mathbf{x}_{l+1}, \tilde{\mathbf{y}}_l), \bar{\mathbf{x}}_l, \ldots).$$

The $\tilde{\mathbf{y}}_l$ variables are drawn from some reference density $\gamma_l(\tilde{\mathbf{x}}_l|\mathbf{x}_{l+1})$ and the ellipses represent components of $\mathbf{Y}^n$ that remain unchanged in the transition. In order to ensure that these swaps are undertaken in a way that preserves the detailed balance condition for Π they are accepted with probability

$$A_l = \min\left\{1,\ \frac{\Pi(\mathbf{y})\gamma(\tilde{\mathbf{x}}_l|\bar{\mathbf{x}}_l)}{\Pi(\mathbf{x})\gamma(\tilde{\mathbf{y}}_l|\mathbf{x}_{l+1})}\right\} = \min\left\{1, \frac{\pi_l(\mathbf{x}_{l+1}, \tilde{\mathbf{y}}_l)\pi_{l+1}(\bar{\mathbf{x}}_l)\gamma(\tilde{\mathbf{x}}_l|\bar{\mathbf{x}}_l)}{\pi_l(\bar{\mathbf{x}}_l, \tilde{\mathbf{x}}_l)\pi_{l+1}(\mathbf{x}_{l+1})\gamma(\tilde{\mathbf{y}}_l|\mathbf{x}_{l+1})}\right\}, \tag{28}$$

and reject the swap with probability $1 - A_l$.

Summarizing the discussion above, one swap step at level l of the PMMC algorithm proceeds as follows:

Algorithm 5 (*PMMC swap step at level l*). The chain moves from $\mathbf{Y}^n$ to $\mathbf{Y}^{n+1}$ as follows:

1. Let $\mathbf{U}$ be sampled from $\gamma_l(\tilde{\mathbf{x}}_l|\mathbf{x}_{l+1})$.
2. Set

$$\mathbf{Y}^{n+1} = (\ldots,(\mathbf{x}_{l+1},\mathbf{U}),\bar{\mathbf{x}}_l,\ldots)$$

with probability

$$A_l = \min\left\{1,\ \frac{\pi_l(\mathbf{x}_{l+1},\mathbf{U})\pi_{l+1}(\bar{\mathbf{x}}_l)\gamma(\tilde{\mathbf{x}}_l|\bar{\mathbf{x}}_l)}{\pi_l(\bar{\mathbf{x}}_l,\tilde{\mathbf{x}}_l)\pi_{l+1}(\mathbf{x}_{l+1})\gamma(\mathbf{U}|\mathbf{x}_{l+1})}\right\}, \tag{29}$$

and

$$\mathbf{Y}^{n+1} = \mathbf{Y}^n = (\ldots,(\bar{\mathbf{x}}_l,\tilde{\mathbf{x}}_l),\mathbf{x}_{l+1},\ldots)$$

with probability $1 - A_l$.

For many applications the swap steps described above will be rejected with overwhelming probability rendering the procedure pointless. In general finding a reference density $\gamma_l(\tilde{\mathbf{x}}_l|\bar{\mathbf{x}}_l)$ that yields reasonable swap acceptance rates is very difficult. This problem is addressed by several modifications of the PMMC algorithm (see [24]). For example, it is possible to replace the sampling from $\gamma_l(\tilde{\mathbf{x}}_l|\bar{\mathbf{x}}_l)$ by MCMC sampling. However, these generalizations were found to be unnecessary for the application in this paper and will not be pursued here.

A.3. PMMC and HMC in Algorithm 3

In this subsection, the discussion is specialized to the setting of Section 4. Steps 5 and 6 of Algorithm 3 require an MCMC scheme which preserves the distribution

$$\begin{aligned}\pi_0(\xi(j,0),\ldots,\xi(j,N_j-1)) &= \mathbf{p}(\xi(j,0),\ldots,\xi(j,N_j-1)|\mathbf{x}(j),\mathbf{h}(j+1))\\ &\propto \exp\left(-\sum_{n=0}^{N_j-1}\frac{\xi(j,n)^T\xi(j,n)}{2}\right)\times g(\mathbf{h}(j+1),\mathbf{x}_{\Delta}(j+1,\xi))),\end{aligned}$$

where, as in Section 4, $\mathbf{x}_{\Delta}(j+1,\xi))$ is the value of $\mathbf{x}(j,N_j)$ determined by the recursion

$$\begin{aligned}&\mathbf{x}(j,n+1) = \mathbf{x}(j,n) + (F(\mathbf{x}(j,n) + \Delta F(\mathbf{x}(j,n)) + \sigma\sqrt{\Delta}\xi(j,n)) + F(\mathbf{x}(j,n)))\frac{\Delta}{2} + \sigma\sqrt{\Delta}\xi(j,n), \quad 0 \leqslant n \leqslant N_j,\\ &\mathbf{x}(j,0) = \mathbf{x}(j)\end{aligned}$$

for a specific value of $\mathbf{x}(j)$ and sequence $\xi(j,0),\ldots,\xi(j,N_j-1)$. In order to use PMMC within such a scheme one must first define approximate marginal distributions π_l of π_0. To that end let $\xi(j,\cdot)$ denote the sequence $(\xi(j,0),\ldots,\xi(j,N_j-1))$ and define the transformation R_0 by the formula

$$R_0\xi(j,\cdot) = (\bar{\xi}(j,\cdot),\tilde{\xi}(j,\cdot)),$$

where, for $n = 0,\ldots,\frac{N_j}{2}-1$,

$$\bar{\xi}(j,n) = \frac{\xi(j,2n+1)+\xi(j,2n)}{\sqrt{2}}$$

and

$$\tilde{\xi}(j,n) = \frac{\xi(j,2n+1)-\xi(j,2n)}{\sqrt{2}}.$$

The choice of marginalization is motivated by the fact that each $\sqrt{\Delta_j}\xi(j,n)$ represents an increment of the Brownian motion B so that

$$\sqrt{2\Delta_j}\bar{\xi}(j,n) = \sqrt{2\Delta_j}\frac{\xi(j,n+1)+\xi(j,n)}{\sqrt{2}} = B(s_j+(n+2)\Delta_j) - B(s_j+n\Delta_j).$$

That is, the $\bar{\xi}$ variables represent increments of the same realization of the Brownian motion over longer time intervals.

Now let $\xi_1(j,\cdot)$ be independent of $\xi(j,\cdot)$ and distributed according to the density

$$\pi_1(\xi_1(j,0),\ldots,\xi_1(j,N_j/2-1)) \propto \exp\left(-\sum_{n=0}^{N_j/2-1}\frac{\xi_1(j,n)^T\xi_1(j,n)}{2}\right)g(\mathbf{h}(j+1),\mathbf{x}_{2\Delta}(j+1,\xi_1))),$$

where $\mathbf{x}_{2\Delta}(j+1,\xi_1))$ represents the value of $\mathbf{x}(j,N_j/2)$ determined by the recursion

$$\mathbf{x}(j,n+1) = \mathbf{x}(j,n) + (F(\mathbf{x}(j,n) + 2\Delta F(\mathbf{x}(j,n)) + \sigma\sqrt{2\Delta}\xi_1(j,n)) + F(\mathbf{x}(j,n)))\Delta + \sigma\sqrt{2\Delta}\xi_1(j,n), \quad 0 \leqslant n \leqslant N_j/2,$$
$$\mathbf{x}(j,0) = \mathbf{x}(j)$$

for a specific value of $\mathbf{x}(j)$ and the sequence $\xi_1(j,0),\ldots,\xi_1(j,N_j/2-1)$, i.e. determined by the same recursion as $\mathbf{x}_\Delta(j,\xi)$ but with twice the time step (2Δ instead of Δ.). Thus the approximate marginal density is the conditional density obtained by doubling the step size in the discretization of (19). The density π_1 is an approximation to the marginal density of the $\bar{\xi}(j,\cdot)$ variables.

These steps can be repeated for $l = 1,\ldots,L-1$ by letting $\xi_l(j,\cdot)$ be independent of $\{\xi(j,\cdot),\xi_1(j,\cdot),\ldots,\xi_{l-1}(j,\cdot)\}$ and distributed according to π_l and for $n = 0,\ldots,\frac{N_j}{2^{l+1}}-1$, defining the variables

$$\bar{\xi}_l(j,n) = \frac{\xi_l(j,2n+1)+\xi_l(j,2n)}{\sqrt{2}},$$

and

$$\tilde{\xi}_l(j,n) = \frac{\xi_l(j,2n+1)-\xi_l(j,2n)}{\sqrt{2}}.$$

One can then define an approximation π_{l+1} to the marginal density of the $\bar{\xi}_l(j,\cdot)$ variables by

$$\pi_{l+1}(\xi_{l+1}(j,0),\ldots,\xi_{l+1}(j,N_j/2^{l+1}-1)) \propto \exp\left(-\sum_{n=0}^{N_j/2^{l+1}-1}\frac{\xi_{l+1}(j,n)^T\xi_{l+1}(j,n)}{2}\right) g(\mathbf{h}(j+1),\mathbf{x}_{2^{l+1}\Delta}(j+1,\xi_{l+1}))), \tag{30}$$

where $\mathbf{x}_{2^{l+1}\Delta}(j+1,\xi_{l+1}))$ represents the value of $\mathbf{x}(j,N_j/2^{l+1})$ determined by the recursion

$$\mathbf{x}(j,n+1) = \mathbf{x}(j,n) + \left(F\left(\mathbf{x}(j,n) + 2^{l+1}\Delta F(\mathbf{x}(j,n)) + \sigma\sqrt{2^{l+1}\Delta}\xi_{l+1}(j,n)\right) + F(\mathbf{x}(j,n))\right)2^l\Delta + \sigma\sqrt{2^{l+1}\Delta}\xi_{l+1}(j,n),$$
$$0 \leqslant n \leqslant N_j/2^{l+1},$$
$$\mathbf{x}(j,0) = \mathbf{x}(j)$$

for a specific value of $\mathbf{x}(j)$ and the sequence $\xi_{l+1}(j,0),\ldots,\xi_{l+1}(j,N_j/2^{l+1}-1)$,

Recall that in the PMMC algorithm one must also choose a reference density $\gamma_l(\tilde{\xi}_l|\bar{\xi}_l)$, and evaluate the acceptance probability

$$A_l = \min\left\{1,\ \frac{\pi_l(\xi_{l+1},\mathbf{U})\pi_{l+1}(\bar{\xi}_l)\gamma(\tilde{\xi}_l|\bar{\xi}_l)}{\pi_l(\bar{\xi}_l,\tilde{\xi}_l)\pi_{l+1}(\xi_{l+1})\gamma(\mathbf{U}|\xi_{l+1})}\right\}, \tag{31}$$

where $\mathbf{U}$ is a sample from $\gamma_l(\tilde{\xi}_l|\bar{\xi}_l)$. The density of ξ_l can be factored as,

$$\pi_l(\xi_l) \propto \rho_l(\bar{\xi}_l)\rho_l(\tilde{\xi}_l)g(\mathbf{h}(j+1),\mathbf{x}_{2^l\Delta}(j+1,\xi_l)),$$

where the density ρ_l is defined by

$$\rho_l(\bar{\xi}_l) \propto \exp\left(-\sum_{n=0}^{N_j/2^l-1}\frac{\bar{\xi}_l(j,n)^T\bar{\xi}_l(j,n)}{2}\right).$$

The choice $\gamma_l(\tilde{\xi}_l|\bar{\xi}_l) = \rho_l(\tilde{\xi}_l)$ results in a particularly simple form of the PMMC acceptance probability. Indeed, notice that

$$\frac{\pi_l(\xi_{l+1},\tilde{\xi}_l)}{\gamma_l(\tilde{\xi}_l|\xi_{l+1})} = \rho_l(\xi_{l+1})g(\mathbf{h}(j+1),\mathbf{x}_{2^l\Delta}(j+1,R_l^{-1}(\xi_{l+1},\tilde{\xi}_l))),$$

and

$$\frac{\pi_l(\bar{\xi}_l,\tilde{\xi}_l)}{\gamma_l(\tilde{\xi}_l|\bar{\xi}_l)} = \rho_l(\bar{\xi}_l)g(\mathbf{h}(j+1),\mathbf{x}_{2^l\Delta}(j+1,R_l^{-1}(\bar{\xi}_l,\tilde{\xi}_l))),$$

where the maps R_l are defined by

$$R_l\ \xi_l(j,\cdot) = (\bar{\xi}_l(j,\cdot),\tilde{\xi}_l(j,\cdot)). \tag{32}$$

Therefore, the PMMC acceptance probability, (31), becomes

$$A_l = \min\left\{1,\ \frac{\pi_l(\xi_{l+1},\mathbf{U})\pi_{l+1}(\bar{\xi}_l)\gamma(\tilde{\xi}_l|\bar{\xi}_l)}{\pi_l(\bar{\xi}_l,\tilde{\xi}_l)\pi_{l+1}(\xi_{l+1})\gamma(\mathbf{U}|\xi_{l+1})}\right\}$$
$$= \min\left\{1,\ \frac{g(\mathbf{h}(j+1),\mathbf{x}_{2^{l+1}\Delta}(j+1,\bar{\xi}_l))}{g(\mathbf{h}(j+1),\mathbf{x}_{2^{l+1}\Delta}(j+1,\xi_{l+1}))} \times \frac{g(\mathbf{h}(j+1),\mathbf{x}_{2^l\Delta}(j+1,R_l^{-1}(\xi_{l+1},\mathbf{U})))}{g(\mathbf{h}(j+1),\mathbf{x}_{2^l\Delta}(j+1,R_l^{-1}(\bar{\xi}_l,\tilde{\xi}_l)))}\right\}.$$

In order to construct a Markov chain capable of exploring all configurations the PMMC swap steps must be combined with transition densities τ_l that preserve the densities π_l. The particular way in which the PMMC swap steps and τ_l are combined can effect the equilibration time of the resulting chain. In the computations discussed in Section 6 each τ_l is the transition density defined by the hybrid Monte Carlo algorithm described in the previous section (with the parameters announced in Section 6). They are combined with swap steps via a recursion similar to the one defining the familiar multigrid W-cycle. The state vector

$$\xi = (\xi(j,\cdot), \xi_1(j,\cdot), \ldots, \xi_L(j,\cdot))$$

is evolved one step by calling **pmmc**$(0,\xi)$ where the routine **pmmc** is defined as follows.

Algorithm 6. **pmmc**$(l,\boldsymbol{\xi})$

{
  if l < L
    for $i = 1,2$
      pmmc(l + 1,ξ);
  end if
  if l > 0
    attempt a swap between levels l and $l-1$ *as follows*:
    generate a sample $\mathbf{U}$ from ρ_{l-1}
    set $\boldsymbol{\xi} = (\boldsymbol{\xi}_0, \ldots, \xi_{l-2}, (\xi_l, \mathbf{U}), \bar{\boldsymbol{\xi}}_{l-1}, \xi_{l+1}, \ldots, \xi_L)$ with probability

$$A_l = \min\left\{1, \frac{g(\mathbf{h}(j+1), \mathbf{x}_{2^{l+1}\Delta}(j+1, \bar{\xi}_l))}{g(\mathbf{h}(j+1), \mathbf{x}_{2^{l+1}\Delta}(j+1, \xi_{l+1}))} \times \frac{g\left(\mathbf{h}(j+1), \mathbf{x}_{2^l\Delta}(j+1, R_l^{-1}(\xi_{l+1}, \mathbf{U}))\right)}{g}(\mathbf{h}(j+1), \mathbf{x}_{2^l\Delta}(j+1, R_l^{-1}(\bar{\xi}_l, \tilde{\xi}_l)))\right\}$$

  end if
  evolve ξ_l one step according to τ_l;
}.

References

[1] N.D. Freitas, A. Doucet, N. Gordon, Sequential Monte Carlo Methods in Practice, Springer, 2005.
[2] R. Kalman, A new approach to linear filtering and prediction problems, J. Basic Eng. 82 (1960) 35–45.
[3] G. Evensen, The ensemble Kalman filter: theoretical formulation and practical implementation, Ocean Dyn. 53 (2003) 343–367.
[4] A. Apte, C. Jones, A. Stuart, A bayesian approach to Lagrangian data assimilation, Tellus A 60 (2) (2008) 336–347.
[5] N. Gordon, D. Salmond, A. Smith, Novel approach to nonlinear non-Gaussian Bayesian state estimation radar and signal processing, IEE Proc. F 140 (1993) 107–113.
[6] G. Kitagawa, Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, J. Comput. Graph. Stat. 5 (1) (1996) 1–25.
[7] F. Alexander, G. Eyink, J. Restrepo, Accelerated Monte Carlo for optimal estimation of time series, J. Stat. Phys. 119 (2004) 1331–1345.
[8] A.M. Stuart, J. Voss, P. Wiberg, Fast communication conditional path sampling of SDEs and the Langevin MCMC method, Commun. Math. Sci. 2 (4) (2004) 685–697.
[9] J. Weare, Efficient conditional path sampling of stochastic differential equations by parallel marginalization, Proc. Nat. Acad. Sci. USA 104 (3) (2007) 12657–12662.
[10] B. Qiu, W. Miao, Kuroshio path variations south of Japan: bimodality as a self-sustained internal oscillation, J. Phys. Oceanogr. 30 (2000) 2124–2137.
[11] B.A. Taft, Characteristics of the flow of the Kuroshio south of Japan, in: H. Stommel, K. Yoshida (Eds.), Kuroshio Physical Aspects of the Japan Coast, University of Washington Press, New York, 1972, pp. 165–214.
[12] K. Yoshida, On the variations of Kuroshio and cold water mass of Enshu Nada, Hydrogr. Bull. 67 (1961) 54–57.
[13] S.-Y. Chao, Bimodality of the Kuroshio, J. Phys. Oceanogr. 14 (1) (1984) 92–103.
[14] W. Gilks, C. Berzuini, Following a moving target – Monte Carlo inference for dynamic Bayesian models, J. Royal Stat. Soc. B 63 (1) (1999) 127–146.
[15] S.P. Meyn, R.L. Tweedie, Markov Chains and Stochastic Stability, Communications and Control Engineering Series, Springer-Verlag London Ltd., London, 1993.
[16] R. Toral, A. Ferreira, A general class of hybrid Monte Carlo methods, Proc. Phys. Comput. 94 (1994) 265–268.
[17] R. Miller, E. Carter, S. Blue, Data assimilation into nonlinear stochastic models, Tellus A 51 (2) (1999) 167–194.
[18] A.J. Chorin, P. Krause, Dimensional reduction for a Bayesian filter, Proc. Natl. Acad. Sci. USA 101 (42) (2004) 15013–15017. electronic.
[19] J.S. Liu, R. Chen, Sequential Monte Carlo methods for dynamic systems, J. Am. Stat. Assoc. 93 (443) (1998) 1032–1044.
[20] J.S. Liu, Monte Carlo Strategies in Scientific Computing, Springer, 2002.
[21] S. Duane, A. Kennedy, B. Pendleton, D. Roweth, Hybrid Monte Carlo, Phys. Lett. B. 195 (1987) 216–222.
[22] M.P. Allen, D.J. Tildesley, Computer Simulation of Liquids, Clarendon Press, Oxford, 1987.
[23] D. Frenkel, B. Smit, Understanding Molecular Simulation, Academic Press, San Diego, 1996.
[24] J. Weare, Parallel marginalization Monte Carlo with applications to conditional path sampling, Preprint.
[25] A.J. Chorin, Conditional expectations and renormalization, Multiscale Model. Simulat. 1 (1) (2003) 105–118. electronic.
[26] P. Okunev, Renormalization Methods with Applications to Spin Physics, U. C. Berkeley Math. Dept., 2005.
[27] J. Goodman, A. Sokal, Multigrid Monte Carlo methods for lattice field theories, Phys. Rev. Lett. 56 (1986) 1015–1018.