Econometric Reviews

ISSN: 0747-4938 (Print) 1532-4168 (Online) Journal homepage: http://www.tandfonline.com/loi/lecr20

A Survey of Sequential Monte Carlo Methods for Economics and Finance

Drew Creal

To cite this article: Drew Creal (2012) A Survey of Sequential Monte Carlo Methods for Economics and Finance, Econometric Reviews, 31:3, 245-296, DOI: 10.1080/07474938.2011.607333

To link to this article: http://dx.doi.org/10.1080/07474938.2011.607333

Econometric Reviews, 31(3):245–296, 2012

ISSN: 0747-4938 print/1532-4168 online
DOI: 10.1080/07474938.2011.607333

Taylor & Francis
Taylor & Francis Group

A SURVEY OF SEQUENTIAL MONTE CARLO METHODS FOR ECONOMICS AND FINANCE

Drew Creal

Booth School of Business, University of Chicago, Chicago, Illinois, USA

□ *This article serves as an introduction and survey for economists to the field of sequential Monte Carlo methods which are also known as particle filters. Sequential Monte Carlo methods are simulation-based algorithms used to compute the high-dimensional and/or complex integrals that arise regularly in applied work. These methods are becoming increasingly popular in economics and finance; from dynamic stochastic general equilibrium models in macro-economics to option pricing. The objective of this article is to explain the basics of the methodology, provide references to the literature, and cover some of the theoretical results that justify the methods in practice.*

Keywords Kalman filter; Markov chain Monte Carlo; Particle filter; Sequential Monte Carlo; State space models.

JEL Classification C11; C15; C32.

1. INTRODUCTION

Economic theory often prescribes fundamental nonlinear relationships between variables of interest. Nonlinear models for learning and strategic interaction among agents provide the modern foundation for microeconomic models. Building on these microfoundations, macroeconomists formulate their structural models as dynamic stochastic general equilibrium (DSGE) models, which have nonlinear first order conditions. Many important economic time series also exhibit strong patterns of non-Gaussian or time-varying behavior. Regime switching, stochastic volatility, and time-varying parameter models have become increasingly popular over the last decade.

Complex models often lead to integrals that cannot be solved analytically. This has created an increase in the popularity of Bayesian

Address correspondence to Drew Creal, Booth School of Business, University of Chicago, 5807 South Woodlawn Ave., Chicago, IL 60637, USA; E-mail: dcreal@chicagobooth.edu

methods that utilize Markov chain Monte Carlo (MCMC) algorithms. Sequential Monte Carlo (SMC) methods are alternative simulation-based algorithms for solving analytically intractable integrals. In these methods, a (partially) continuous probability distribution is approximated by a discrete distribution made of weighted draws termed particles. From one iteration of the algorithm to the next, particles are updated to approximate one distribution after another by changing the particle's location on the support of the distribution and their weights. SMC methods include the particle filter, which generalizes the Kalman filter and hidden Markov model (HMM) filter to nonlinear, non-Gaussian state space models. Particle filters were introduced into the economics literature by Kim et al. (1998) to study the volatility of asset prices. Their popularity has grown in economics since the publication of Fernández-Villaverde and Rubio-Ramírez (2005, 2007), who used them to estimate DSGE models. Particle filters also share a common mathematical structure with genetic algorithms which are popular in economics.

The standard reference for SMC methods is Doucet et al. (2001). A considerable number of advances have taken place since its publication; advances ranging from stimulating new applications, improved algorithms, and new theoretical results. Most of the methodological results have occurred outside economics, where nice reviews for engineers and applied mathematicians are provided by Cappé et al. (2007) and Doucet and Johansen (2009). This article provides a guide to the growing literature intended for economists. The presentation given here extends previous reviews by including a discussion of SMC methods applied outside state space models. The methods are also applied to several economic applications. To reach as wide an audience as possible, the survey has been split into two parts. The first half focuses on practical applications of particle filters to general state space models. The second half covers recent developments in the field with more emphasis on Bayesian computation as well as an overview of the theoretical properties of SMC methods.

The theoretical properties of SMC algorithms have been intensely studied since Del Moral (1996), who provided the first consistency proof for the original particle filter of Gordon et al. (1993). In SMC algorithms, the draws or particles interact and are therefore dependent. Traditional limit theorems for Monte Carlo methods, e.g., Geweke (1989) and Tierney (1994), do not apply. The main theoretical properties that are relevant for applied researchers are reviewed in the article while additional references are provided for those interested in further study. To make the article shorter, readers are assumed to be modestly familiar with linear, Gaussian state space models, importance sampling, accept-reject algorithms (acceptance sampling), and MCMC. Harvey (1989) and Durbin and Koopman (2001) provide introductions to linear, Gaussian state space

models while Robert and Casella (2004) and Geweke (2005) are good references for the traditional Monte Carlo methods.

In Section 2, SMC methods are introduced starting with the particle filter and its application to nonlinear, non-Gaussian state space models. This section contains a minimum of technical details and concentrates on best practices that a researcher should consider when implementing them. Some of the theoretical properties of SMC algorithms are reviewed in Section 3. In Section 4, more advanced SMC algorithms are discussed which are applicable outside the context of state space models. Both Sections 2 and 4 include economic applications to illustrate the relevance of the methods. The final section concludes.

2. PARTICLE FILTERS FOR STATE SPACE AND HIDDEN MARKOV MODELS

2.1. Definition of the Models

State space or hidden Markov models are a convenient means for studying dynamic systems. A state space model consists of two equations: the observation or measurement equation and the transition equation which are respectively given by

$$\mathbf{y}_n = m_n(\mathbf{x}_n, \varepsilon_n), \tag{1}$$

$$\mathbf{x}_n = h_n(\mathbf{x}_{n-1}, \eta_n). \tag{2}$$

The state variables $\mathbf{x}_n$ and observations $\mathbf{y}_n$ may be continuous-valued, discrete-valued, or a combination of the two. The functions m_n and h_n are possibly nonlinear but of known form. Time is denoted by the subscripts n. It is assumed that the distributions of the observations and state variable admit density functions with respect to appropriate dominating measures $d\mathbf{y}_n$ and $d\mathbf{x}_n$, respectively. These densities $p(\mathbf{y}_n \mid \mathbf{x}_n; \theta)$ and $p(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta)$ corresponding to (1) and (2), respectively, are called the observation (or measurement) and transition densities. The latter terminilogy stems from the fact that $\mathbf{x}_n$ is a Markov process. The densities will typically depend upon a vector of unknown parameters θ that need to be estimated from the observed data $\mathbf{y}_{1:T} = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$.

The sequence of state variables $\mathbf{x}_{0:n} = \{\mathbf{x}_0, \ldots, \mathbf{x}_n\}$ are generally unobserved and it is the aim of the researcher to estimate them using the observed data. Uncertainty about the state variable is formulated as a joint conditional probability distribution $p(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta)$ known as the joint smoothing distribution. It is defined as

$$p(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta) = \frac{p(\mathbf{x}_{0:n}, \mathbf{y}_{1:n}; \theta)}{p(\mathbf{y}_{1:n}; \theta)}, \tag{3}$$

where the constant of integration $p(\mathbf{y}_{1:n};\theta)$ is the likelihood of the state space model. Three of its marginal distributions are of interest: the one-step ahead predictive distribution $p\left(\mathbf{x}_n \,|\, \mathbf{y}_{1:n-1};\theta\right)$, the filtering distribution $p\left(\mathbf{x}_n \,|\, \mathbf{y}_{1:n};\theta\right)$, and the smoothing distribution $p\left(\mathbf{x}_n \,|\, \mathbf{y}_{1:T};\theta\right)$. Each distribution conditions on a different set of observations. In addition, researchers are often interested in computing the likelihood of the model at a point θ. Although θ is unknown, it is traditional in the literature to run filtering and smoothing algorithms assuming a fixed value of θ. Therefore, in Sections 2.1.2–2.7, the value of θ is assumed to be known. The estimation of θ is considered in later sections.

2.1.1. *Joint Smoothing Recursion*

The joint smoothing distribution can be written recursively as

$$\begin{aligned} p\left(\mathbf{x}_{0:n} \,|\, \mathbf{y}_{1:n};\theta\right) &= \frac{p\left(\mathbf{y}_n \,|\, \mathbf{x}_{0:n}, \mathbf{y}_{1:n-1};\theta\right) p\left(\mathbf{x}_{0:n} \,|\, \mathbf{y}_{1:n-1};\theta\right)}{p\left(\mathbf{y}_n \,|\, \mathbf{y}_{1:n-1};\theta\right)} \\ &= \frac{p\left(\mathbf{y}_n \,|\, \mathbf{x}_{0:n}, \mathbf{y}_{1:n-1};\theta\right) p\left(\mathbf{x}_n \,|\, \mathbf{x}_{0:n-1}, \mathbf{y}_{1:n-1};\theta\right)}{p\left(\mathbf{y}_n \,|\, \mathbf{y}_{1:n-1};\theta\right)} p\left(\mathbf{x}_{0:n-1} \,|\, \mathbf{y}_{1:n-1};\theta\right) \\ &= \frac{p\left(\mathbf{y}_n \,|\, \mathbf{x}_n;\theta\right) p\left(\mathbf{x}_n \,|\, \mathbf{x}_{n-1};\theta\right)}{p\left(\mathbf{y}_n \,|\, \mathbf{y}_{1:n-1};\theta\right)} p\left(\mathbf{x}_{0:n-1} \,|\, \mathbf{y}_{1:n-1};\theta\right). \end{aligned} \tag{4}$$

This decomposition will be an important component for a particle filter which will recursively approximate it through time. The marginal distributions of interest are then obtained as a by-product.

2.1.2. *Marginal Prediction and Filtering Recursions*

An alternative to working with the joint distribution is to calculate the one-step ahead predictive and filtering distributions recursively. These recursions are the traditional approach taken in the state space modeling literature. They begin under the assumption that the initial distribution of the state variable $p\left(\mathbf{x}_0;\theta\right)$ is known. At a future iteration n, the prediction step projects last period's filtering distribution $p\left(\mathbf{x}_{n-1} \,|\, \mathbf{y}_{1:n-1};\theta\right)$ forward using the dynamics of the model (2) and its transition density

$$p\left(\mathbf{x}_n \,|\, \mathbf{y}_{1:n-1};\theta\right) = \int p\left(\mathbf{x}_n \,|\, \mathbf{x}_{n-1};\theta\right) p\left(\mathbf{x}_{n-1} \,|\, \mathbf{y}_{1:n-1};\theta\right) d\mathbf{x}_{n-1}. \tag{5}$$

This distribution is a one-step ahead forecast of the state variable. With the addition of another observation $\mathbf{y}_n$, the update step computes the filtering

distribution by applying Bayes' rule

$$
\begin{aligned}
p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n}; \theta\right) &= \frac{p\left(\mathbf{y}_n, \mathbf{x}_n \mid \mathbf{y}_{1:n-1}; \theta\right)}{p\left(\mathbf{y}_n \mid \mathbf{y}_{1:n-1}; \theta\right)} \\
&= \frac{p\left(\mathbf{y}_n \mid \mathbf{x}_n, \mathbf{y}_{1:n-1}; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n-1}; \theta\right)}{\int p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n-1}; \theta\right) d\mathbf{x}_n} \\
&= \frac{p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n-1}; \theta\right)}{\int p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n-1}; \theta\right) d\mathbf{x}_n}. \qquad (6)
\end{aligned}
$$

This completes one iteration of the recursion which continues until the end of the dataset.

Difficulty arises in this approach because for most state space models the integrals in (5) and in the denominator of (6) cannot be calculated analytically. The latter integral $p\left(\mathbf{y}_n \mid \mathbf{y}_{1:n-1}; \theta\right)$ is the contribution to the likelihood since $p\left(\mathbf{y}_{1:n}; \theta\right) = p\left(\mathbf{y}_n \mid \mathbf{y}_{1:n-1}; \theta\right) p\left(\mathbf{y}_{1:n-1}; \theta\right)$. There are several known cases in which it is possible to solve the recursions analytically. The first case is when the functions in (1) and (2) are linear and the densities are Gaussian. The recursions can then be solved by the Kalman filter, see Kalman (1960) and Kalman and Bucy (1961). The other case is when the state variable $\mathbf{x}_n$ takes on a discrete number of values and the recursions can be solved by the HMM filter, see Baum and Petrie (1966) and Baum et al. (1970). The latter algorithm being rediscovered and extended to autoregressions by Hamilton (1989) in his influential model for the business cycle. For textbook treatments of these methods see Harvey (1989) and Durbin and Koopman (2001) for the linear, Gaussian state space model while Frühwirth-Schnatter (2006) covers models with discrete state variables.

Outside of these cases, computing the distributions of interest requires approximating the integrals in (5) and the denominator of (6). Deterministic and functional approximations to the integrals have been proposed in the literature, including Gaussian sum filters (Alspach and Sorenson, 1972), numerical integration (Kitagawa, 1987), extended Kalman filters (Anderson and Moore, 1979), and unscented Kalman filters (Julier and Uhlmann, 1997; Julier et al., 2000). These methods may work well on some problems however they all have the same limitation. The approximation of the integrals at time n is a function of the approximation of the integrals at the previous iteration. As noted by Künsch (2001), if the approximation at iteration $n-1$ is poor, the approximation error can effect the current period's estimate. Errors can accumulate over iterations and as the number of observations increases the algorithms might diverge from the true value. In the following sections, we discuss approximating the integrals in these recursions by Monte Carlo methods.

2.2. Importance Sampling

Consider approximating the entire joint distribution $p\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta\right)$. Given a function f of the state variable, a standard Monte Carlo estimator of the integral

$$\mathbb{E}\left[f(\mathbf{x}_{0:n})\right] = \int f(\mathbf{x}_{0:n}) p\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta\right) d\mathbf{x}_{0:n}$$

consists of drawing sequences $\mathbf{x}_{0:n}$ directly from the target distribution $p\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta\right)$. By the law of large numbers, sample averages will converge to population moments as the number of draws increases. This strategy is generally impossible for complex models because the target distribution is nonstandard and it is unknown how to draw directly from it.

Instead, a researcher could use importance sampling (IS) where random draws are taken from a proposal or importance distribution $g_{0:n}\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \psi\right)$ from which it is easy to sample. The importance distribution is chosen by the researcher with the restriction that its support covers the support of the target distribution. The variable ψ denotes a vector of tuning parameters which are used to make the proposal distribution approximate the target distribution as closely as possible. The integral of interest is now

$$\mathbb{E}\left[f(\mathbf{x}_{0:n})\right] = \int f(\mathbf{x}_{0:n}) \frac{p\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta\right)}{g_{0:n}\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \psi\right)} g_{0:n}\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \psi\right) d\mathbf{x}_{0:n}. \tag{7}$$

After taking N draws $\left\{\mathbf{x}_{0:n}^{(i)}\right\}_{i=1}^{N}$ from the proposal distribution, the draws are reweighted to correct for the fact that they were drawn from the wrong distribution. The importance weights $\left\{w_n^{(i)}\right\}_{i=1}^{N}$ are defined as the ratio of the target distribution to the proposal distribution

$$w_n^{(i)} \propto \frac{p\left(\mathbf{x}_{0:n}^{(i)} \mid \mathbf{y}_{1:n}; \theta\right)}{g_{0:n}\left(\mathbf{x}_{0:n}^{(i)} \mid \mathbf{y}_{1:n}; \psi\right)}. \tag{8}$$

Given the draws $\left\{\mathbf{x}_{0:n}^{(i)}, w_n^{(i)}\right\}_{i=1}^{N}$, the importance sampling estimator of (7) is given by

$$\mathbb{E}\left[f(\mathbf{x}_{0:n})\right] \approx \sum_{i=1}^{N} f\left(\mathbf{x}_{0:n}^{(i)}\right) \hat{w}_n^{(i)} \qquad \hat{w}_n^{(i)} = \frac{w_n^{(i)}}{\sum_{j=1}^{N} w_n^{(j)}}, \tag{9}$$

where the importance weights are self-normalized because the constant of integration in the target density may not be known.

IS was invented by Kahn and Marshall (1953) and Marshall (1956) and was first used in econometrics by Kloek and van Dijk (1978). To facilitate IS

for state space models, researchers need to find an importance distribution $g_{0:n}\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \psi\right)$ that closely approximates the target distribution. General procedures for finding proposal distributions that approximate the joint smoothing distribution implied by a state space model have been proposed by many authors. Shephard and Pitt (1997) and Durbin and Koopman (2001) use Laplace approximations of the integral as a proposal distribution. Alternatively, after selecting $g_{0:n}\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \psi\right)$ to be a parameteric class of distributions (e.g., multivariate normal), researchers have proposed using Newton–Raphson schemes to choose ψ to minimize a criterion function such as the Kullback–Leibler distance or the coefficient of variation that measures the separation between the two distributions; see, e.g., Oh and Berger (1993) and Liu (2001, p. 42). This strategy has been applied to state space models by Richard and Zhang (2007) who call their method efficient importance sampling (EIS). Other flexible methods for finding importance distributions that have recently been developed are Hoogerheide et al. (2007), Cappé et al. (2008), and Cornuet et al. (2009), although these have yet to be applied to state space models.

Geweke (1989) proved consistency and asymptotic normality of the IS estimator (9). In order for the central limit theorem governing asymptotic normality of the IS estimator to apply, the variance of the importance weights must be finite. IS for high dimensional integrals can be problematic because it can be shown that the variance of the importance weights grows with the time index n; see Kong et al. (1994) for a theorem. If the variance of the IS weights does not exist, estimates produced by IS can be highly unreliable. This does not mean IS should be avoided for all applications to state space models. However, researchers should always check the stability of the importance weights by graphical diagnostics or formal hypothesis tests; see, e.g., Koopman et al. (2009).

Recently, de Jong et al. (2009) proposed the EIS filter for estimating state space models. They successfully apply their method to several DSGE models. The EIS filter is an IS algorithm; however, it does not try to approximate the entire joint distribution $p\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta\right)$ at one time. Instead it approximates the marginal prediction and filtering distributions by applying the EIS algorithm of Richard and Zhang (2007) to the integrals in (5) and (6) at each time period. The method is similar in spirit to Kitagawa (1987) who uses numerical integration to approximate the integrals in each time period instead of IS; see also Tanizaki and Mariano (1994).

2.3. Sequential Importance Sampling

Another limitation of standard IS for state space models is that the entire expression for the importance weights (8) needs to be recomputed at each iteration. The computational demands of the algorithm grow over

time. To avoid evaluating the entire expression each period, a researcher can use sequential importance sampling (SIS). SIS is a special case of IS where instead of drawing $\mathbf{x}_{0:n}$ from a joint proposal distribution draws are made from a sequence of conditional distributions. By redefining the joint importance distribution, SIS iteratively approximates the joint smoothing recursion in (4).

To see how this works, the importance distribution in a SIS algorithm is factored into two parts

$$g_{0:n}\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \psi\right) \equiv g_n\left(\mathbf{x}_n \mid \mathbf{x}_{0:n-1}, \mathbf{y}_{1:n}; \psi\right) g_{0:n-1}\left(\mathbf{x}_{0:n-1} \mid \mathbf{y}_{1:n-1}; \psi\right). \tag{10}$$

The second distribution $g_{0:n-1}\left(\mathbf{x}_{0:n-1} \mid \mathbf{y}_{1:n-1}; \psi\right)$ is per particle a Dirac measure $\delta_{\mathbf{x}_{0:n-1}}$ placing a unit probability mass on each path that has already been simulated in the previous iterations up to time $n-1$. A new set of values $\left\{\mathbf{x}_n^{(i)}\right\}_{i=1}^N$ are drawn at time n from the first part of the importance distribution $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{0:n-1}, \mathbf{y}_{1:n}; \psi\right)$. Consequently, a new sequence of paths is obtained by keeping the trajectories of the old draws up to time $n-1$ fixed and appending the newly simulated values to the end of the old trajectories, $\left\{\mathbf{x}_{0:n}^{(i)}\right\}_{i=1}^N = \left\{\mathbf{x}_{0:n-1}^{(i)}, \mathbf{x}_n^{(i)}\right\}_{i=1}^N$. The time subscript on the importance distribution g_n indicates that either it or its parameters ψ can potentially be chosen at time n and can change over time.

Substituting (4) and (10) into (8), one obtains

$$\begin{aligned} w_n &= \frac{p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta\right) p\left(\mathbf{x}_{0:n-1} \mid \mathbf{y}_{1:n-1}; \theta\right)}{p\left(\mathbf{y}_n \mid \mathbf{y}_{1:n-1}; \theta\right) g_n\left(\mathbf{x}_n \mid \mathbf{x}_{0:n-1}, \mathbf{y}_{1:n}; \psi\right) g_{0:n-1}\left(\mathbf{x}_{0:n-1} \mid \mathbf{y}_{1:n-1}; \psi\right)} \\ &\propto w_{n-1} \frac{p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta\right)}{g_n\left(\mathbf{x}_n \mid \mathbf{x}_{0:n-1}, \mathbf{y}_{1:n}; \psi\right)} \\ &\propto w_{n-1} \tilde{w}_n \end{aligned} \tag{11}$$

where

$$\tilde{w}_n = \frac{p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta\right)}{g_n\left(\mathbf{x}_n \mid \mathbf{x}_{0:n-1}, \mathbf{y}_{1:n}; \psi\right)}. \tag{12}$$

The densities $p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right)$ and $p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta\right)$ are determined by the state space model (1)-(2). The ratio of densities $\tilde{w}_n$ defined in (12) is known as the *incremental importance weight*. It is the only part of the importance weight that needs computed at each iteration. The conditioning information in the importance distribution in the denominator of (11) will typically be reduced to $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \psi\right)$ for computational convenience. Calculating the incremental weights then does not require the past observations or the entire past trajectories $\left\{\mathbf{x}_{0:n-2}^{(i)}\right\}_{i=1}^N$.

At the end of each iteration, the SIS algorithm produces N simulated paths and importance weights $\left\{\mathbf{x}_{0:n}^{(i)}, w_n^{(i)}\right\}_{i=1}^{N}$. In the literature on SMC methods which includes the particle filter, these draws are known as "particles." They provide a discrete distribution that approximates the (partially) continuous distribution. Estimates of the moments are the same as in standard IS and are

$$\mathbb{E}\left[f(\mathbf{x}_{0:n})\right] \approx \sum_{i=1}^{N} f\left(\mathbf{x}_{0:n}^{(i)}\right) \hat{w}_n^{(i)}, \qquad \hat{w}_n^{(i)} = \frac{w_n^{(i)}}{\sum_{j=1}^{N} w_n^{(j)}}. \tag{13}$$

An estimate of the target distribution is given by

$$p(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta) \approx \sum_{i=1}^{N} \hat{w}_n^{(i)} \delta_{\mathbf{x}_{0:n}^{(i)}}(\mathbf{x}_{0:n}), \tag{14}$$

where $\delta_{\mathbf{x}_{0:n}^{(i)}}$ denotes a Dirac measure located at $\mathbf{x}_{0:n}^{(i)}$. An estimate of the contribution to the likelihood at time n is

$$p\left(\mathbf{y}_n \mid \mathbf{y}_{1:n-1}; \theta\right) \approx \sum_{i=1}^{N} \hat{w}_{n-1}^{(i)} \tilde{w}_n^{(i)},$$

which is a function of the normalized weights from last period.

SIS was invented by Hammersley and Morton (1954) and first applied to state space models by Handschin and Mayne (1969) and Handschin (1970). It was first used in econometrics by Hendry and Richard (1991). While the SIS algorithm avoids evaluating the entire expression for the importance weights each period by computing them recursively, the variance of the importance weights will grow over iterations. In fact, it can be shown that the variance of the SIS weights grows exponentially in time, see Chopin (2004). This is because as the number of iterations increases all the probability mass will eventually be allocated to one particle; one particle's normalized importance weight converges to one while the normalized weights of the other particles are converging to zero. The SIS estimator will ultimately be a function of a single draw. This is known as *weight degeneracy* in the literature.

2.4. Particle Filters

In their seminal article introducing the particle filter, Gordon et al. (1993) added a resampling step within the SIS algorithm that mitigates the weight degeneracy problem. Resampling means that a new population of particles are replicated from the existing population in proportion to their normalized importance weights. In its simplest form, we draw

N random variables with replacement from a multinomial distribution with probabilities $\left\{\hat{w}^{(i)}\right\}_{i=1}^{N}$. Particles with large importance weights are randomly duplicated while particles with small probability are eliminated. Once resampled the particles' weights are set equal to any constant, e.g., $w_n^{(i)} = \frac{1}{N}$ for $i = 1, \ldots, N$. This forces the weights not to permanently degenerate as in the SIS algorithm.

This new algorithm called sequential importance sampling with resampling (SISR) combined the sampling importance resampling (SIR) method of Rubin (1987, 1988) with the SIS algorithm and applied it to filtering in state space models. The basic SISR particle filter with resampling applied in each time period is given as Algorithm 1 (see Table 1). Different particle filtering algorithms are obtained by different choices of the incremental importance distribution $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \psi\right)$ and different types of resampling algorithms, which are both chosen by the user. Additional early contributions to the particle filtering literature include Isard and Blake (1996), Liu and Chen (1995), Kitagawa (1996), and Berzuini et al. (1997).

Table 1 Algorithm 1: Sequential Importance Sampling with Resampling (SISR)

At $n = 0$, for $i = 1, \ldots, N$

Draw $\mathbf{x}_0^{(i)} \sim g_0(\mathbf{x}_0)$ and set $w_0^{(i)} = \frac{p(\mathbf{x}_0^{(i)})}{g_0(\mathbf{x}_0^{(i)})}$.

For $n = 1, \ldots, T$:

(i) For $i = 1, \ldots, N$ draw $\mathbf{x}_n^{(i)} \sim g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}^{(i)}, \mathbf{y}_n; \psi\right)$ and compute the importance weights $w_n^{(i)} \propto w_{n-1}^{(i)} \frac{p\left(\mathbf{y}_n \mid \mathbf{x}_n^{(i)}; \theta\right) p\left(\mathbf{x}_n^{(i)} \mid \mathbf{x}_{n-1}^{(i)}; \theta\right)}{g_n\left(\mathbf{x}_n^{(i)} \mid \mathbf{x}_{n-1}^{(i)}, \mathbf{y}_n; \psi\right)}$.

(ii) For $i = 1, \ldots, N$ normalize the importance weights: $\hat{w}_n^{(i)} = \frac{w_n^{(i)}}{\sum_{j=1}^{N} w_n^{(j)}}$.

(iii) Resample N particles with probabilities $\left\{\hat{w}_n^{(i)}\right\}_{i=1}^{N}$ and for $i = 1, \ldots, N$ set $w_n^{(i)} = \frac{1}{N}$.

Given the particles, quantities of interest (e.g., moments, quantiles, etc.) can be estimated during the algorithm. Estimates of the moments are

$$\mathbb{E}\left[f(\mathbf{x}_{0:n})\right] \approx \sum_{i=1}^{N} f\left(\mathbf{x}_{0:n}^{(i)}\right) \hat{w}_n^{(i)}, \qquad \hat{w}_n^{(i)} = \frac{w_n^{(i)}}{\sum_{j=1}^{N} w_n^{(j)}}, \tag{15}$$

and an estimate of the joint target distribution (3) is given by

$$p(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta) \approx \sum_{i=1}^{N} \hat{w}_n^{(i)} \delta_{\mathbf{x}_{0:n}^{(i)}}(\mathbf{x}_{0:n}). \tag{16}$$

Alternative estimators of these quantities can be obtained after the particles have been resampled when their importance weights are equal.

These are given by

$$\mathbb{E}\left[f(\mathbf{x}_{0:n})\right] \approx \frac{1}{N}\sum_{i=1}^{N} f\left(\mathbf{x}_{0:n}^{(i)}\right). \tag{17}$$

$$p(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta) \approx \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{x}_{0:n}^{(i)}}(\mathbf{x}_{0:n}). \tag{18}$$

Estimates of the state variable in (15) and the distribution in (16) should always be preferred to estimators calculated after resampling such as (17) and (18). Resampling introduces additional Monte Carlo variation into the algorithm. In the discussion that follows in Section 2.6, we will see that it is better not to resample at every iteration.

For an algorithm that resamples in every time period, an estimate of the contribution to the likelihood at time n is

$$\begin{aligned} p\left(\mathbf{y}_n \mid \mathbf{y}_{1:n-1}; \theta\right) &= \int p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n-1}; \theta\right) d\mathbf{x}_n \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \tilde{w}_n^{(i)}. \end{aligned} \tag{19}$$

If resampling is performed in random time periods, an estimator of contributions to the likelihood is given by

$$p\left(\mathbf{y}_n \mid \mathbf{y}_{1:n-1}; \theta\right) \approx \sum_{i=1}^{N} \hat{w}_{n-1}^{(i)} \tilde{w}_n^{(i)}, \tag{20}$$

which includes (19) as a special case when $\hat{w}_{n-1}^{(i)} = \frac{1}{N}$ for $i = 1, \ldots, N$.

While resampling is a crucial feature to the success of the particle filter, it is important to understand why particles are resampled and what the side-effects of resampling are. Resampling does not cure the degeneracy problem when it comes to the particle filter's estimate of the entire joint distribution $p\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta\right)$. Repeatedly resampling particles copied from previous generations reduces the number of distinct particles representing the early parts of the joint distribution. The past paths will eventually coalesce into a single particle. The particle filter can produce a good approximation of the marginal distribution $p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n}; \theta\right)$, the likelihood contribution $p\left(\mathbf{y}_n \mid \mathbf{y}_{1:n-1}; \theta\right)$, and for small k the joint distribution $p\left(\mathbf{x}_{n-k:n} \mid \mathbf{y}_{1:n}; \theta\right)$. However, its approximation at time n of the entire joint distribution $p\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta\right)$ and the earlier marginal distributions $p\left(\mathbf{x}_{n-l} \mid \mathbf{y}_{1:n}; \theta\right)$ will be poor as n and l increase. Because of this effect, in practice only the most recent generation of particles $\left\{\mathbf{x}_{n-k:n}^{(i)}\right\}_{i=1}^{N}$

are resampled and stored in memory. The purpose of resampling is to prevent *future* degeneracy by replicating those particles that appear relevant for estimating next period's marginal distribution.

The number of particles also does not need to remain constant during the algorithm. Particle size may vary either deterministically or at random over time. For example, the number of particles can change within each iteration. One can draw R particles (where $R = \alpha N$ for a positive integer α) from the importance distribution, compute the importance weights and any quantities of interest using the R particles, and then resample only N out of the R particles. The advantage is that a larger number of particles are used when the estimator is computed.

The popularity of particle filters has increased since the original article of Gordon et al. (1993). A simple explanation being that basic particle filters are extremely easy to implement. A second reason is that as long as the dimension of $\mathbf{x}_n$ is moderate the particle filters' estimator of the marginal filtering distribution $p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n}; \theta\right)$ has good properties. In Section 3, we will see that many particle filters are consistent and asymptotically normal and that they will "forget their past errors" sufficiently fast to ensure that past errors do not accumulate. In practice, this means that a particle filter can be applied to a long stretch of time series (e.g., a financial time series) and the precision of the estimator of the marginal $p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n}; \theta\right)$ will not deteriorate as one obtains more observations.

Example #1 (Stochastic Volatility Model). To provide some intuition about the SISR algorithm and the estimator that it produces, data were simulated from a standard stochastic volatility model

$$\begin{aligned} \mathbf{y}_n &= \exp(\mathbf{x}_n/2)\varepsilon_n, \qquad \varepsilon_n \sim \mathcal{N}(0,1), \\ \mathbf{x}_n &= \mu + \phi(\mathbf{x}_{n-1} - \mu) + \sigma_\eta \eta_n, \qquad \eta_n \sim \mathcal{N}(0,1), \end{aligned} \tag{21}$$

with parameter values chosen as $\mu = 0.5, \phi = 0.985$, and $\sigma_\eta^2 = 0.04$. Stochastic volatility models are popular in finance where they are used to model heteroskedasticity in financial asset returns. Shephard (2005) provides a recent review of the literature and discusses properties of the model. To implement the SISR algorithm, we selected the conditional proposal distribution at each iteration to be the transition density $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{0:n-1}, \mathbf{y}_{1:n}; \psi\right) = p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta\right)$ implied by the dynamics of the model (21). This means that the incremental weight function is equal to the measurement density, $\tilde{w}_n = p(\mathbf{y}_n \mid \mathbf{x}_n; \theta)$. The algorithm uses multinomial resampling at each iteration and is equivalent to the original particle filter of Gordon et al. (1993) called the *bootstrap filter*.

Figure 1 plots the true value of the state variable $\mathbf{x}_n$ from this model over the first 50 time periods together with the particles. For

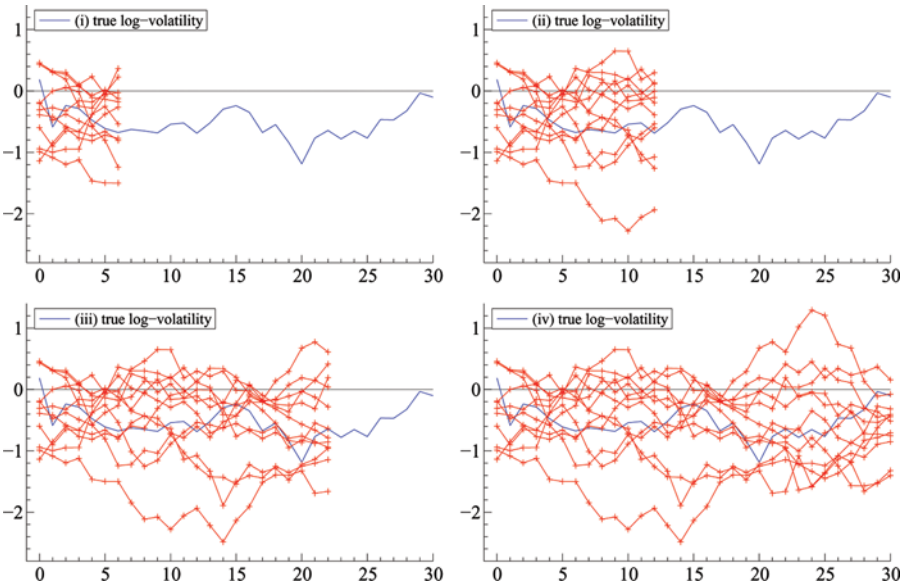

FIGURE 1 Fifty iterations of a discrete-time log-normal stochastic volatility model. Pictured is the true log-volatility (solid line) and a particle system with $N = 15$ particles after: (i) 5 time-steps; (ii) $n = 10$ time-periods; (iii) $n = 30$ time-periods; and (iv) $n = 50$ time-periods. (Figure available in color online.)

illustration purposes, the number of particles is $N = 15$. The panels show the evolution of a particle system $\left\{\mathbf{x}_{0:n}^{(i)}, \hat{w}_n^{(i)}\right\}_{i=1}^{N=15}$ through time. The graphs indicate how the particle filter approximates a continuous distribution $p\left(\mathbf{x}_{0:n} \,|\, \mathbf{y}_{1:n}; \theta\right)$ with a discrete distribution. The affects of repeated resampling on the early parts of the joint distribution can be seen by comparing panel (ii) and panel (iv). After $n = 10$ time periods in panel (ii), there are many distinct particles covering the support of the distribution. By the time $n = 50$ in panel (iv), particles estimating the first 10 time periods overlap one another as some particles have been duplicated and others eliminated. The surviving particles do not cover the support of early parts of the distribution resulting in a poor estimate of the joint distribution. This will happen over time for any fixed value of the number of particles.

The marginal filtering distribution $p\left(\mathbf{x}_n \,|\, \mathbf{y}_{1:n}; \theta\right)$ at the last iteration however can still be estimated well. To see this, Fig. 2 provides perhaps a more intuitive perspective of the estimator by plotting the empirical distribution function implied by (16) for different particle sizes. Panel (i) depicts the particles' approximation of the marginal filtering distribution at time $n = 50$ using $N = 15$ particles. The remaining panels in Fig. 2 demonstrate what happens as the number of particles increases. In this

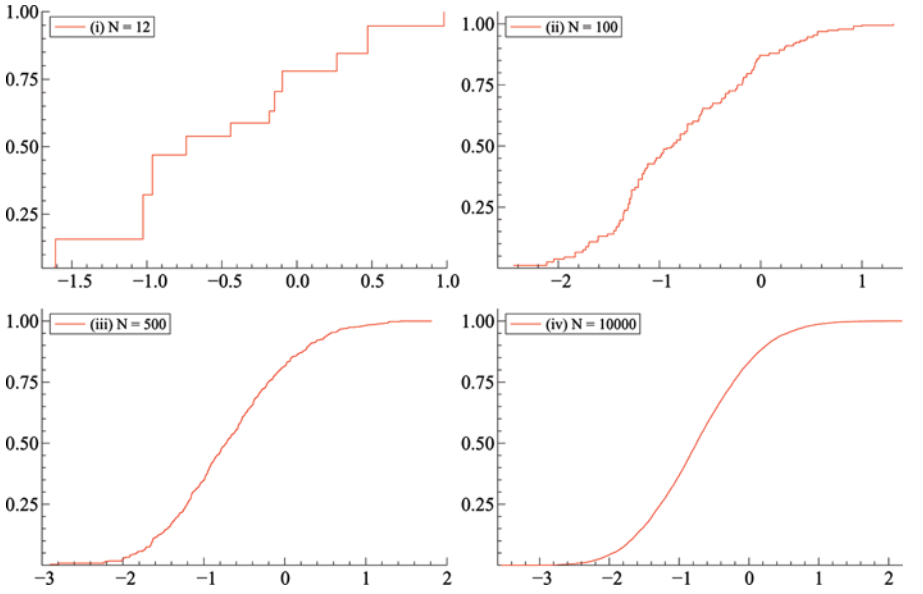

FIGURE 2 Empirical distribution functions created using the particles to approximate the marginal filtering distribution $p(\mathbf{x}_{50} \mid \mathbf{y}_{1:50}; \theta)$ for the stochastic volatility model. (i) $N = 15$ particles; (ii) $N = 250$ particles; (iii) $N = 1000$ particles; and (iv) $N = 10000$ particles. (Figure available in color online.)

example where the state variable $\mathbf{x}_n$ is a continuous random variable, the particles form a probability mass function that is converging towards a continuous distribution function.

2.5. Choosing an Importance Distribution

This section covers the major classes of importance distributions. To shorten the survey, detailed derivations of the algorithms are left to the references.

2.5.1. *Preliminary Comments*

Before describing the literature on importance distributions for particle filters, it is helpful to consider when a particle filter might run into problems. The resampling step in a particle filter ensures that the particles do not permanently degenerate as in SIS. However, the variance of the incremental importance weights $\tilde{w}_n$ may still be large. If the variance of the incremental importance weights is high, the marginal filtering distribution may be poorly estimated in some time periods. This can happen when the incremental importance distribution $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \psi\right)$ is a poor approximation of the incremental target $p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta\right)$.

When designing a particle filter, a researcher should try to understand the structure and properties of their model in order to understand how different proposal distributions might cause the variance of the incremental importance weights to be large.

Secondly, an important feature of the incremental importance distribution in basic particle filters is that it is a function of the current set of particles $\left\{\mathbf{x}_{n-1}^{(i)}\right\}_{i=1}^{N}$ whose locations represent the support of the marginal filtering distribution $p(\mathbf{x}_{n-1} \mid \mathbf{y}_{1:n-1}; \theta)$. Most of these particles will be located in the high probability mass regions of $p(\mathbf{x}_{n-1} \mid \mathbf{y}_{1:n-1}; \theta)$ with few particles in the tails. When two neighboring marginals are extremely different the majority of old particle locations will not form a good proposal distribution. This is because they are not located near and may not be informative about the high probability mass regions of the next marginal $p(\mathbf{x}_n \mid \mathbf{y}_{1:n}; \theta)$. Consequently, the new set of particles will need to be simulated far from their current locations $\left\{\mathbf{x}_{n-1}^{(i)}\right\}_{i=1}^{N}$.

For example, the performance of a particle filter may deteriorate when there exist observations that might be inconsistent with the proposed model (e.g., outliers), which is demonstrated clearly by Pitt and Shephard (1999). In this situation, it is important to incorporate the current observation $\mathbf{y}_n$ in the proposal distribution because it will carry more information about $\mathbf{x}_n$ than the past particles $\left\{\mathbf{x}_{n-1}^{(i)}\right\}_{i=1}^{N}$. The presence of outliers suggests a form of model miss-specification. The literature on particle filters has generally not focused on how substantial miss-specification of a model impacts the performance of particle filters. However, there is a literature on the effects of misspecifying the initial distribution, see Douc et al. (2009).

2.5.2. *The Bootstrap Filter*

The original particle filter of Gordon et al. (1993) called the *bootstrap filter* is also the simplest to implement. It uses the transition density as the proposal, $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \psi\right) = p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta\right)$. Many authors call this importance distribution the prior kernel or prior distribution given the Bayesian interpretation of a state space model. The bootstrap filter resamples in each time period making the incremental importance weights equal to $\tilde{w}_n = p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right)$. This particle filter can perform well for some models but notice that it does not use the information in the current observation $\mathbf{y}_n$ to propose new particles. Proposal distributions that do not use the current observation $\mathbf{y}_n$ are often called *blind* proposals. When the variance of the measurement density (as a function of $\mathbf{x}_n$) is large, we should expect the incremental weights to be unbalanced and for proposal distributions that incorporate $\mathbf{y}_n$ to outperform this choice.

Example #2 (Nonlinear Measurement Equation with Additive Noise). This example illustrates a potential limitation of this importance distribution and what it means to have incremental importance weights with a large variance. Consider the following state space model

$$\mathbf{y}_n = h(\mathbf{x}_n) + \varepsilon_n, \qquad \varepsilon_n \sim \mathcal{N}(0, \Sigma_\varepsilon),$$
$$\mathbf{x}_n = \mu + \phi(\mathbf{x}_{n-1} - \mu) + \eta_n, \qquad \eta_n \sim \mathcal{N}(0, \Sigma_\eta),$$

where the function $h(\mathbf{x}_n)$ in the measurement equation is potentially nonlinear and the noise ε_n is additively Gaussian. In economics, models that often fit within this framework are nonlinear DSGE models. Different second-order solution methods for approximating the DSGE model will result in different functions $h(\mathbf{x}_n)$. Many researchers add measurement noise ε_n and set the diagonal elements of the covariance matrix Σ_ε to small values such as 10^{-3} or 10^{-6}; see, e.g., An and Schorfheide (2007) and Amisano and Tristani (2010).

Using the bootstrap filter, the incremental weight function is the normal density $\tilde{w}_n = \mathcal{N}(\mathbf{y}_n \mid h(\mathbf{x}_n), \Sigma_\varepsilon)$ with mean $h(\mathbf{x}_n)$ and diagonal covariance matrix Σ_ε. When viewed as a function of $\mathbf{x}_n$ for a fixed value of $\mathbf{y}_n$, the properties of this function will depend on the covariance matrix Σ_ε. As $\Sigma_\varepsilon \to 0$, the weight function will become increasingly peaked as the observation $\mathbf{y}_n$ becomes increasingly informative. Relative to the bootstrap filter, we should expect the numerical efficiency gains of an alternative particle filter that intelligently incorporates $\mathbf{y}_n$ in the proposal to increase as $\Sigma_\varepsilon \to 0$. This illustrates that the efficiency of one proposal distribution relative to another will generally depend on the parameter values of the model.

To see why the variance of the incremental weights increases as $\Sigma_\varepsilon \to 0$, consider the incremental weight function for a univariate model. Let $\Sigma_\varepsilon = \sigma_\varepsilon^2$. Figure 3 plots $\tilde{w}_n$ for four different functional forms for $h(\mathbf{x}_n)$ and different degrees of measurement error. In each panel, $\mathbf{x}_n$ ranges from $[-2.5, 2]$ and we plot the function for different values $\sigma_\varepsilon = 1, \sigma_\varepsilon = 0.5$, and $\sigma_\varepsilon = 0.05$. Assume we observe $\mathbf{y}_n = 1$. The functional form $h(\mathbf{x}_n)$ and the observed value of $\mathbf{y}_n$ will determine the location and width of the peak. The height of the peak is determined by the amount of measurement noise. Consider panel (ii) where $h(\mathbf{x}_n) = \mathbf{x}_n^2$, we can see that when $\mathbf{y}_n = 1$ this functional form causes there to be two peaks at $\mathbf{x}_n = -1, 1$. If the proposal density blindly simulates most of the particles between the peaks and only a few particles around -1 and 1, the normalized importance weights will be highly unstable for small values of σ_ε. In this setting, the bootstrap filter will tend to degenerate more as $\sigma_\varepsilon \to 0$.

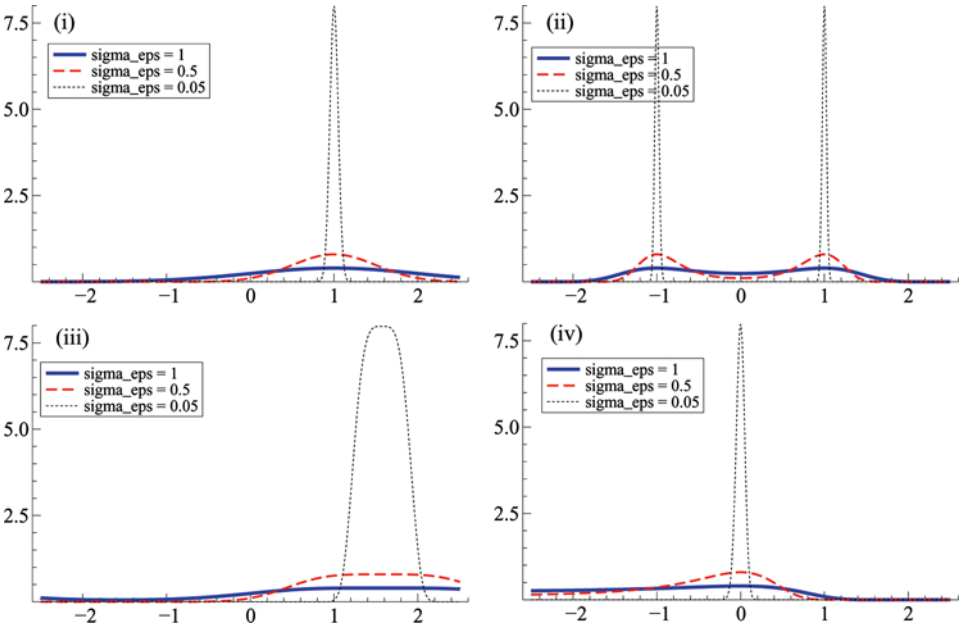

FIGURE 3 Incremental weight functions $\tilde{w}_n$ for three different levels of measurement error. $\sigma_\varepsilon = 1, \sigma_\varepsilon = 0.5, \sigma_\varepsilon = 0.05$. We observe $\mathbf{y}_n = 1$. Each panel plots a different functional form: (i) $h(\mathbf{x}_n) = \mathbf{x}_n$; (ii) $h(\mathbf{x}_n) = \mathbf{x}_n^2$; (iii) $h(\mathbf{x}_n) = \sin(\mathbf{x}_n)$; and (iv) $h(\mathbf{x}_n) = \exp(\mathbf{x}_n)$. (Figure available in color online.)

2.5.3. *Conditionally Optimal Importance Distribution*

The particle filtering literature includes the notion of a *conditionally* optimal importance distribution for any model. The conditionally optimal distribution is defined as the distribution that minimizes the Monte Carlo variation of the importance weights. The "conditional" portion of this statement emphasizes that the importance distribution is optimal if one only conditions on the current observation $\mathbf{y}_n$ and last period's particles $\left\{\mathbf{x}_{n-1}^{(i)}\right\}_{i=1}^{N}$. This idea was introduced by Liu and Chen (1995), although it exists in an earlier literature on SIS algorithms from Zaritskii et al. (1975) and Akashi and Kumamoto (1977). The conditionally optimal importance distribution is given by

$$g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \psi\right) = p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \theta\right),$$

$$= \frac{p\left(\mathbf{y}_n \mid \mathbf{x}_n, \mathbf{x}_{n-1}; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta\right)}{p\left(\mathbf{y}_n \mid \mathbf{x}_{n-1}; \theta\right)}, \tag{22}$$

$$= \frac{p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta\right)}{p\left(\mathbf{y}_n \mid \mathbf{x}_{n-1}; \theta\right)}. \tag{23}$$

A nice feature of $p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \theta\right)$ is that it uses the information in $\mathbf{y}_n$ and $\mathbf{x}_{n-1}$ simultaneously. The incremental weight function $\tilde{w}_n = p\left(\mathbf{y}_n \mid \mathbf{x}_{n-1}; \theta\right)$ is interestingly a function of the previous state $\mathbf{x}_{n-1}$ and not the current state $\mathbf{x}_n$. This importance distribution unfortunately requires drawing from $p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \theta\right)$ and evaluating $p\left(\mathbf{y}_n \mid \mathbf{x}_{n-1}; \theta\right)$. For most models, these are rarely known in closed-form except in special circumstances, e.g., when the measurement equation (1) is linear and its density is Gaussian see Doucet et al. (2000). However, researchers use this distribution as a benchmark and try to approximate it with sub-optimal choices. For an application of this idea to DSGE models, see Amisano and Tristani (2010).

2.5.4. *Auxiliary Particle Filters*

The auxiliary particle filter (APF) of Pitt and Shephard (1999, 2001) is a popular algorithm that is simple to implement and works well in many cases. The presentation given here is different than the original article, as this discussion does not explicitly include auxiliary variables. When proposing new particles at the beginning of each iteration, we would like to use the information available in the current observation $\mathbf{y}_n$. These authors call particle filters that incorporate $\mathbf{y}_n$ into their proposal *adapted* particle filters. In addition, since particles carried over from last period form part of this period's proposal distribution, some of the old particles provide more information about $\mathbf{x}_n$ than others.

Pitt and Shephard (1999, 2001) approximate the incremental target distribution in (4) with the importance distribution

$$\begin{aligned} p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta\right) &\approx g_{1,n}\left(\mathbf{y}_n \mid \mathbf{x}_n; \psi\right) g_{2,n}\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \psi\right), \\ &= g_{1,n}\left(\mathbf{y}_n \mid \mathbf{x}_{n-1}; \psi\right) g_{2,n}\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \psi, \mathbf{y}_n\right). \quad (24) \end{aligned}$$

The proposal distribution in (24) is decomposed into two parts implying that the sampling of new values $\left\{\mathbf{x}_n^{(i)}\right\}_{i=1}^N$ from this distribution can potentially be performed in two steps. The APF is given as Algorithm 2 (see Table 2).

Table 2 Algorithm 2: Auxiliary particle filter (APF)

At $n = 0$, for $i = 1, \ldots, N$

Draw $\mathbf{x}_0^{(i)} \sim g_0(\mathbf{x}_0)$ and set $w_0^{(i)} = \frac{p(\mathbf{x}_0^{(i)})}{g_0(\mathbf{x}_0^{(i)})}$.

For $n = 1, \ldots, T$:

(i) For $i = 1, \ldots, N$ compute $\tau^{(i)} = g_n(\mathbf{y}_n \mid \mathbf{x}_{n-1}^{(i)}; \psi) w_{n-1}^{(i)}$ and normalize $\widehat{\tau}^{(i)} = \frac{\tau^{(i)}}{\sum_{j=1}^N \tau^{(j)}}$.

(ii) Resample N particles $\left\{\mathbf{x}_{n-1}^{(i)}\right\}_{i=1}^N$ with probabilities $\left\{\widehat{\tau}^{(i)}\right\}_{i=1}^N$.

(iii) For $i = 1, \ldots, N$ draw $\mathbf{x}_n^{(i)} \sim g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}^{(i)}; \mathbf{y}_n, \psi\right)$.

(iv) For $i = 1, \ldots, N$ compute importance weights: $w_n^{(i)} = \frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)};\theta) p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)};\theta)}{g_n(\mathbf{y}_n|\mathbf{x}_{n-1}^{(i)};\psi) g_n\left(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)};\mathbf{y}_n,\psi\right)}$.

The APF nests other algorithms in the literature as special cases. If we select the proposal as $g_{1,n}(\mathbf{y}_n \mid \mathbf{x}_{n-1}; \theta) = 1$ and $g_{2,n}(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta, \mathbf{y}_n) = p(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta)$, the APF reduces to the bootstrap filter of Gordon et al. (1993). Many economic models have a special structure with non-Gaussian measurement densities and linear, Gaussian transition densities. In this case if the measurement density is log-concave, Pitt and Shephard (1999, 2001) suggest taking $g_{1,n}(\mathbf{y}_n \mid \mathbf{x}_{n-1}; \theta)$ to be the Taylor series expansion of $\log p(\mathbf{y}_n \mid \mathbf{x}_n; \theta)$ around a point μ_n and combining it with the transition density $g_{2,n}(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta, \mathbf{y}_n) = p(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta, \mathbf{y}_n)$. The resulting proposal distributions then locally approximate the conditionally optimal distribution. Smith and Santos (2006) apply this APF to several models demonstrating the improvement of a second-order expansions over first-order expansions when there are outliers in the data.

In settings where one can evaluate $p(\mathbf{y}_n \mid \mathbf{x}_{n-1}; \theta)$, one can select $g_{1,n}(\mathbf{y}_n \mid \mathbf{x}_{n-1}; \psi) = p(\mathbf{y}_n \mid \mathbf{x}_{n-1}; \theta)$ and $g_{2,n}(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta, \mathbf{y}_n) = p(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta, \mathbf{y}_n)$. The incremental importance weights are then equal to one. Pitt and Shephard (1999, 2001) call this situation *full adaption* or *perfect adaption*. As it incorporates $p(\mathbf{y}_n \mid \mathbf{x}_{n-1}; \theta)$, it is similar to the conditionally optimal distribution of Liu and Chen (1995). However, the APF is different than the conditionally optimal distribution of Liu and Chen (1995) because it uses $p(\mathbf{y}_n \mid \mathbf{x}_{n-1}; \theta)$ to resample first and "pre-select" from the existing set of particles. Pre-selecting from the previous period's particles using $\mathbf{y}_n$ can potentially improve the current period's proposal distribution because the resampled particles after step (ii) form a different importance distribution than the original particle filter that resamples at the end of each iteration. The APF can encompass more general algorithms than discussed here. Pitt and Shephard (1999, 2001) show that it is possible to use the accept-reject algorithm or alternatively MCMC moves within an APF as well.

In the original article, the APF contained a second resampling stage at the end of each iteration. In independent work by Johansen and Doucet (2008) and Douc et al. (2009c), the authors prove that if one keeps the particle size constant at each iteration then the second resampling stage in the original algorithm is unnecessary. Including it increases the asymptotic variance of the corresponding estimator. Johansen and Doucet (2008) also demonstrate that the APF can actually degrade the performance of a particle filter even in the case of perfect adaption. These authors show that the performance of the APF will depend upon the signal to noise ratio in the state space model.

2.5.5. *Particle Filters Built from Accept-Reject Algorithms (Acceptance Sampling)*

Accept-reject algorithms (see Robert and Casella, 2004, p. 47) can also be incorporated within a particle filter. This idea originates with

Hürzeler and Künsch (1998) and also Tanizaki and Mariano (1998). For example, when it is impossible to draw directly from the conditionally optimal importance distribution (22), one can draw N particles from this distribution using an accept–reject algorithm. The algorithms have been studied theoretically by Künsch (2005), who also proposes some extensions to the algorithms in the original articles. As an accept–reject algorithm will be run for each particle in each time period, the user needs to find a good proposal distribution within the accept–reject algorithm. If this is not chosen well, a large number of trial simulations may be needed for each particle to be accepted.

2.5.6. *Other Approximations to the Conditionally Optimal Distribution*

Other proposal distributions exist when either or both of the functions $m_n(\cdot)$ and $h_n(\cdot)$ in (1) and (2) are nonlinear but the disturbances η_n and ε_n are additively Gaussian. Proposal distributions can then be formulated by taking a Taylor series expansion of the non-linear function. These importance distributions, given in Doucet et al. (2000), van der Merwe et al. (2000), and Guo et al. (2005), make proposals using one-step of the extended or unscented Kalman filter applied to each particle. The extended and unscented Kalman filters are nonlinear filters that use analytical approximations; see, e.g., Anderson and Moore (1979) and Julier et al. (2000).

When both the observation and transition densities are log-concave, the conditionally optimal proposal distribution will also be log-concave. As suggested by Pitt and Shephard (1999) and Doucet et al. (2000), another option is to choose the parameters ψ of the importance distribution $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \psi\right)$ in each time period and for each particle so that its mode and curvature match the mode and curvature of the incremental target density $p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta\right)$. This will provide an approximation to the conditionally optimal distribution. Finding the mode of the target can be accomplished using Newton–Raphson methods. One can then choose the importance distribution $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \psi\right)$ to be a normal or Student's t distribution setting the parameters ψ to have this mode and an inflated variance to ensure its support includes the support of the incremental target distribution.

2.5.7. *Rao–Blackwellization*

Some state space models have a special structure, where a subset of the state vector may be integrated out analytically. Analytical integration of part of the state vector reduces the Monte Carlo variation of the resulting estimator and will always improve its numerical efficiency. It is known as Rao–Blackwellization in the Monte Carlo literature because it is an implication of the Rao–Blackwell Theorem; e.g., see Robert

and Casella (2004, p. 130). When this is possible, the structure of the model implies that the state vector can be separated into two parts $\mathbf{x}_n = \left(\mathbf{x}'_{1,n}, \mathbf{x}'_{2,n}\right)'$. The marginal filtering distribution can then be decomposed as $p(\mathbf{x}_{1,n}, \mathbf{x}_{2,n} \mid \mathbf{y}_{1:n}; \theta) = p(\mathbf{x}_{1,n} \mid \mathbf{x}_{2,n}, \mathbf{y}_{1:n}; \theta) p(\mathbf{x}_{2,n} \mid \mathbf{y}_{1:n}; \theta)$. Particles are only simulated randomly from $p(\mathbf{x}_{2,n} \mid \mathbf{y}_{1:n}; \theta)$, while conditional on each individual draw $\mathbf{x}_{2,n}^{(i)}$ the distribution $p(\mathbf{x}_{1,n} \mid \mathbf{x}_{2,n}^{(i)}, \mathbf{y}_{1:n}; \theta)$ can be evaluated analytically.

A class of models amenable to Rao–Blackwellization that is popular in economics is

$$\mathbf{y}_n = Z_n\left(\mathbf{x}_{2,n}\right)\mathbf{x}_{1,n} + \varepsilon_n, \qquad \varepsilon_n \sim \mathcal{N}\left(0, H_n\left(\mathbf{x}_{2,n}\right)\right), \tag{25}$$

$$\mathbf{x}_{1,n} = T_n\left(\mathbf{x}_{2,n}\right)\mathbf{x}_{1,n-1} + \eta_n, \qquad \eta_n \sim \mathcal{N}\left(0, Q_n\left(\mathbf{x}_{2,n}\right)\right), \tag{26}$$

$$p_{ij} = p\left(\mathbf{x}_{2,n} = j \mid \mathbf{x}_{2,n-1} = i\right), \qquad \mathbf{x}_{2,n} \in \{1, 2, \ldots, k\}, \tag{27}$$

which is a linear, Gaussian state space model where the parameters in the state space matrices Z_n, T_n, Q_n, H_n depend upon the value of an additional discrete state variable $\mathbf{x}_{2,n}$. The discrete state variables follow a first-order Markov process as in (27). These models are covered in Kim and Nelson (1999) and Frühwirth-Schnatter (2006).

Conditional on the discrete state variables $\left\{\mathbf{x}_{2,n}^{(i)}\right\}_{i=1}^{N}$ the resulting system is a linear, Gaussian state space model and $p(\mathbf{x}_{1,n} \mid \mathbf{x}_{2,n}, \mathbf{y}_{1:n}; \theta)$ can be evaluated by the Kalman filter. These particle filters are due to Chen and Liu (2000) who named them *mixture Kalman filters*, see also Doucet et al. (2001). de Freitas et al. (2004), Schön et al. (2005), and Bos and Shephard (2006) are additional references which apply some form of this methodology.

Other models that can be Rao–Blackwellized are partially observed Gaussian state space models as in Andrieu and Doucet (2002), which include dynamic probit and Tobit models with unobserved states. State space models where the functions in (1) and (2) are nonlinear but depend on both discrete and continuous-valued states can also be Rao–Blackwellized; e.g., see Andrieu et al. (2003).

Example #3 (Applying a Rao–Blackwellized Particle Filter)**.** Forecasting inflation is an important part of monetary policy-making and has a long history in economics. We illustrate the use of Rao–Blackwellized particle filters on a model recently proposed by Stock and Watson (2007) to forecast inflation π_n. It is a time-varying random walk plus noise or local level model

$$\pi_n = \mathbf{x}_{1,n} + \varepsilon_n, \qquad \varepsilon_n \sim \mathcal{N}(0, \exp(\mathbf{x}_{2,n})), \tag{28}$$

$$\mathbf{x}_{1,n+1} = \mathbf{x}_{1,n} + \eta_n, \qquad \eta_n \sim \mathcal{N}(0, \exp(\mathbf{x}_{3,n})), \tag{29}$$

$$\mathbf{x}_{2,n+1} = \mathbf{x}_{2,n} + \omega_{1,n}, \qquad \omega_{1,n} \sim \mathcal{N}(0, 0.2), \tag{30}$$

$$\mathbf{x}_{3,n+1} = \mathbf{x}_{3,n} + \omega_{2,n}, \qquad \omega_{2,n} \sim \mathcal{N}(0, 0.2), \tag{31}$$

where $\mathbf{x}_{1,n}$ is the unobserved time-varying mean of inflation and $\mathbf{x}_{i,n}$ for $i = 2, 3$ are unobserved log-variances. Stock and Watson (2007) argued that this specification improves forecasting because the model accounts for the structural breaks present in inflation. It can be shown, see, e.g., Harvey (1989, p. 68), that the local level model with constant variances is equivalent to an ARIMA(0,1,1) model with additional restrictions on the parameter space. The stochastic variances for the level and irregular components in (28)–(31) imply a time-varying variance and MA parameter in this ARIMA representation. The time-varying MA parameter conveniently summarizes how the model's forecast function changes through time.

Stock and Watson (2007) estimate the state variables of the model by MCMC, while it is (arguably) easier to implement a particle filter when there are no static parameters that need to be estimated. The model has a special structure because conditional on the log-variances $\mathbf{x}_{2,n}$ and $\mathbf{x}_{3,n}$ the remaining model is a linear, Gaussian state space model. The conditional distribution can then be calculated exactly by the Kalman filter. A good particle filter for this application is an APF version of the mixture Kalman filter; see, Chen and Liu (2000). Following the discussion in Section 2.5.7, the proposal distribution can be broken into two parts, $g_n(\mathbf{x}_{1,n} \mid \mathbf{x}_{1,n-1}, \mathbf{x}_{2,n}, \mathbf{x}_{3,n}, \mathbf{y}_n; \theta) g_n(\mathbf{x}_{2,n}, \mathbf{x}_{3,n} \mid \mathbf{x}_{2,n-1}, \mathbf{x}_{3,n-1} \mathbf{y}_n; \theta)$. The log-variances $\mathbf{x}_{2,n}$ and $\mathbf{x}_{3,n}$ are first simulated from the latter part of the proposal distribution which was chosen to be the transition densities of the state equations (30) and (31). Conditional on these values, the Kalman filter can update the sufficient statistics of $p_n(\mathbf{x}_{1,n} \mid \mathbf{x}_{1,n-1}, \mathbf{x}_{2,n}, \mathbf{x}_{3,n}, \mathbf{y}_n; \theta) = g_n(\mathbf{x}_{1,n} \mid \mathbf{x}_{1,n-1}, \mathbf{x}_{2,n}, \mathbf{x}_{3,n}, \mathbf{y}_n; \theta)$ which are the mean and covariance matrix of the Gaussian distribution denoted by $\mathbf{x}_{1,n|n}$ and $P_{1,n|n}$. Since part of the state vector $\mathbf{x}_{1,n}$ has been computed analytically, this algorithm should need fewer particles to achieve the same level of statistical efficiency relative to other simple particle filtering algorithms such as the SISR filter. This particle filter is given in detail as Algorithm 3 (see Table 3). It can be used for any model that is a conditionally Gaussian state space model.

In this example, the data are quarterly U.S. CPI inflation from Q1:1959–Q7:2008 constructed from the "real-time" price indices available from the U.S. Federal Reserve Bank of Philadelphia. The filtering algorithm was implemented with $N = 10{,}000$ particles and systematic resampling. Smoothed estimates of each of the components were computed by taking 5,000 draws using the simulation smoothing algorithm of Godsill et al. (2004) discussed in Section 2.7 below.

Table 3 Algorithm 3: Rao-Blackwellized APF for the time-varying local level model

At $n = 0$, for $i = 1, \ldots, N$

Draw $\mathbf{x}_{1,0}^{(i)}, P_{1,0|0}^{(i)} \sim p(\mathbf{x}_{1,0})$, $\mathbf{x}_{2,0}^{(i)} \sim p(\mathbf{x}_{2,0})$, $\mathbf{x}_{3,0}^{(i)} \sim p(\mathbf{x}_{3,0})$ and set $w_0^{(i)} = \frac{1}{N}$.

For $n = 1, \ldots, T$:

(i) For $i = 1, \ldots, N$, run the prediction step of the Kalman filter to obtain the prediction errors and prediction error variances $\left\{v_n^{(i)}, F_n^{(i)}\right\}_{i=1}^N$.

(ii) For $i = 1, \ldots, N$, compute the importance weights $w_n^{(i)} \propto \mathcal{N}\left(v_n^{(i)}, F_n^{(i)}\right)$ and normalize them: $\widehat{w}_n^{(i)} = \frac{w_n^{(i)}}{\sum_{j=1}^N w_n^{(j)}}$.

(iii) Resample N particles $\left\{\mathbf{x}_{1,n-1|n-1}^{(i)}, P_{1,n-1|n-1}^{(i)}, \mathbf{x}_{2,n-1}^{(i)}, \mathbf{x}_{3,n-1}^{(i)}\right\}_{i=1}^N$ with probabilities $\left\{\widehat{w}_n^{(i)}\right\}_{i=1}^N$ and for $i = 1, \ldots, N$ set $w_n^{(i)} = \frac{1}{N}$.

(iv) For $i = 1, \ldots, N$, draw $\mathbf{x}_{2,n}^{(i)} \sim \mathcal{N}(\mathbf{x}_{2,n-1}^{(i)}, 0.2)$ and $\mathbf{x}_{3,n}^{(i)} \sim \mathcal{N}(\mathbf{x}_{3,n-1}^{(i)}, 0.2)$ and run the Kalman filter on each particle to obtain $\left\{\mathbf{x}_{1,n|n}^{(i)}, P_{1,n|n}^{(i)}\right\}_{i=1}^N$.

The one-step ahead forecast of inflation and the filtered and smoothed estimates of the volatilities are pictured in Fig. 4. These estimates largely confirm the results of Stock and Watson (2007). The volatility of the

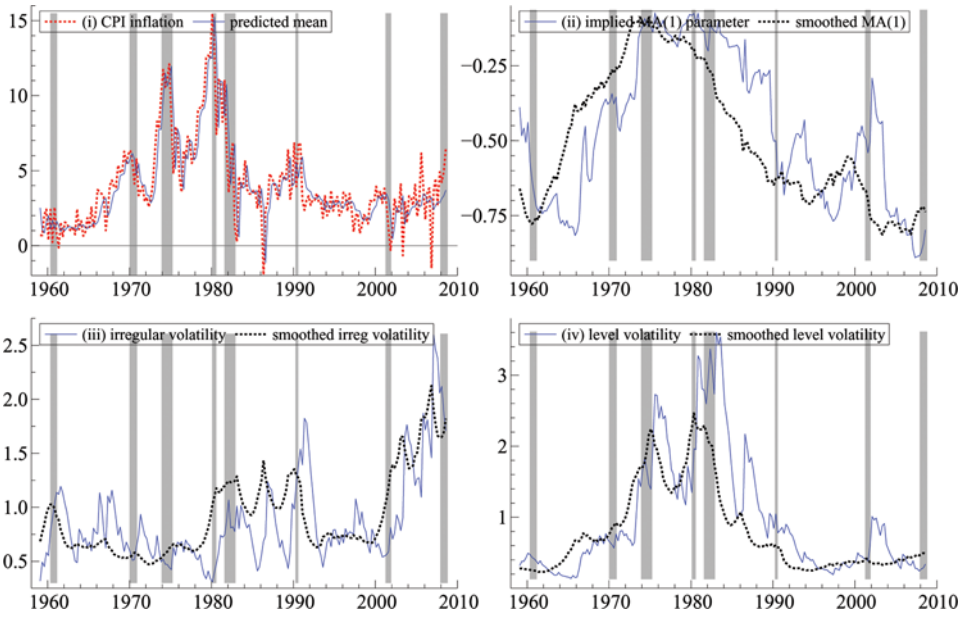

FIGURE 4 Estimates from the time-varying local level model applied to quarterly U.S. inflation Q1:1959-Q3:2008: (i) inflation and its one-step ahead forecast; (ii) filtered and smoothed estimates of the implied MA(1) parameter; (iii) filtered and smoothed estimates of the irregular volatility $\exp(\mathbf{x}_{2,n}/2)$; and (iv) filtered and smoothed estimates of the state volatility $\exp(\mathbf{x}_{3,n}/2)$. NBER recession dates are indicated by the vertical bars. (Figure available in color online.)

level or permanent component $\exp(\mathbf{x}_{3,n}/2)$ increased during the period of high-inflation in the 1970s, while the volatility of the irregular component $\exp(\mathbf{x}_{2,n}/2)$ was relatively more stable. Filtered and smoothed estimates of the implied MA(1) parameter are shown in panel (ii), and they indicate that it also increased during this period. The forecastability of inflation appears to have changed over time as argued by Stock and Watson (2007). This data set includes five additional years of inflation beyond that analyzed by these authors. The volatility of inflation has recently increased beginning in the middle of 2007. It appears to be concentrated in the irregular or transitory component.

2.5.8. *MCMC and Adaptive Proposals*

If the particles were resampled during an iteration of the particle filter, the resampling step causes some particles to be duplicated. The duplicated particles form part of the importance density at the next iteration. Resampling also causes the particles at the previous iterations to coalesce. Gilks and Berzuini (2001) proposed the *resample-move* algorithm which creates diversity among the particles by applying one iteration of a Metropolis–Hastings or Gibbs sampler Markov kernel to each particle conditional on the particles being resampled. At time n, the user can choose their Markov kernel such that the algorithm returns k periods into the past (for small k) and moves a block of variables $\left\{\mathbf{x}_{n-k:n}^{(i)}\right\}_{i=1}^{N}$ instead of only the last period's. The resulting resample-move algorithm will improve on a standard particle filter when it comes to estimating the joint distribution $p(\mathbf{x}_{n-k:n} \mid \mathbf{y}_{1:n}; \theta)$. This is because it introduces some diversity into the past paths whereas these paths are typically fixed in a standard algorithm.

Another recent line of research considers using the past particles $\left\{\mathbf{x}_{0:n-1}^{(i)}, \hat{w}_n^{(i)}\right\}_{i=1}^{N}$ to adapt the importance distribution over time. Cornebise et al. (2008) consider selecting the parameters ψ of $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \psi\right)$ each period to minimize an empirical estimate of the Shannon entropy or the coefficient of variation between the empirical distribution of the particles and the target distribution. Their article contains additional references to work on adaptive methods in particle filters. Using MCMC and adaptive proposals within particle filters are relatively more advanced methods. They are closely related to the algorithms discussed in Section 4.

2.5.9. *Block Sampling*

When using MCMC, it is well-known that better performing algorithms can be built if one can find a proposal distribution that enables joint sampling of blocks of variables from the target distribution. If the proposal is chosen well, sampling variables in blocks improves the speed by which

the Markov chain explores the support of the distribution. Doucet et al. (2006) propose a similar idea for particle filters. At the beginning of iteration n of a particle filter, the algorithm has already simulated and stored the paths $\left\{\mathbf{x}_{0:n-1}^{(i)}\right\}_{i=1}^{N}$. The goal is not only to extend each path at the endpoint but instead returning k time periods into the past (where k is say 5–10) and sample a block $\left\{\mathbf{x}_{n-k:n}^{(i)}\right\}_{i=1}^{N}$. Instead of using a proposal distribution for a single time period $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \psi\right)$, the proposal distribution is over the path $g_{n-k:n}\left(\mathbf{x}_{n-k:n} \mid \mathbf{x}_{n-k-1:n-1}, \mathbf{y}_{n-k:n}; \psi\right)$. Particle filters based on block-sampling proposal distributions do not directly approximate the joint smoothing recursion (4) like a standard particle filter. Instead, they rely on defining a sequence of articial target distributions and are related to the simulation algorithms discussed in Section 4.

In order to implement block sampling, the importance weight recursions (11) need to be rewritten to account for the alternative sequence of target distributions. This changes the definition of the incremental weights, see Doucet et al. (2006) for details. If the proposals $g_{n-k:n}\left(\mathbf{x}_{n-k:n} \mid \mathbf{x}_{n-k-1:n-1}, \mathbf{y}_{n-k:n}; \psi\right)$ are chosen well, this algorithm will perform better at estimating the joint distribution $p\left(\mathbf{x}_{n-k:n} \mid \mathbf{y}_{1:n}; \theta\right)$ than a standard particle filter for the same number of particles. However, selecting good joint proposals may be challenging in practice and the algorithms are more computationally expensive.

2.6. Resampling and Branching Algorithms

There are four resampling algorithms that dominate most of the literature: multinomial resampling of Gordon et al. (1993), stratified resampling of Kitagawa (1996), residual resampling of Liu and Chen (1998), and systematic resampling of Carpenter et al. (1999).[1] All of these algorithms can be performed in $O(N)$ operations.[2] The main point for applied researchers to note is that some resampling algorithms are preferable because they introduce less Monte Carlo variation into the particle filter's estimator. Douc et al. (2005) compare their efficiency in terms of Monte Carlo variation. They prove that the stratified resampling algorithm and the residual resampling scheme should be preferred to the original multinomial resampling. The Monte Carlo variation introduced by these algorithms is strictly smaller. The systematic resampling algorithm is the easiest to implement. It can also perform well in Monte Carlo studies

[1]The residual and systematic resampling schemes are also known in the genetic algorithm literature under alternative names, see Whitley (1994).

[2]Matlab code for each of the resampling algorithms can be found at Nando de Freitas' webpage at http://www.cs.ubc.ca/~nando/software.html.

but does not always dominate multinomial resampling in terms of variance, see Douc et al. (2005).

The residual and stratified resampling algorithms are also unbiased in the sense that the expected number of times a particle $\mathbf{x}_n^{(i)}$ will be resampled is equal to its importance weight. Thus, the algorithms satisfy the condition

$$\mathbb{E}\left[N_n^{\#(i)} \mid \left\{\hat{w}_n^{(i)}\right\}_{i=1}^N\right] = N\hat{w}_n^{(i)},$$

where $N_n^{\#(i)}$ is the number of times the ith particle is replicated during resampling. This condition is a maintained assumption in the consistency and asymptotic normality proofs behind most particle filters.

Other notable resampling algorithms include the optimal resampling algorithm of Fearnhead and Clifford (2003), which should be used for any model whose state variable has a discrete component, e.g., the Markov-switching state space model (25)–(27). The stopping-time resampling algorithm of Chen et al. (2005) is another recent alternative. The articles by Fearnhead and Clifford (2003) and Chen et al. (2005) illustrate the point that a resampling algorithm can be tailored for specific classes of models or even a specific application.

The original particle filter of Gordon et al. (1993) carries out resampling every time period. To lower the degree of Monte Carlo variation introduced into the estimator, Liu and Chen (1995) suggested resampling only after time periods where the importance weights are unstable. They suggested using the effective sample size (ESS) as a measure of weight instability. The ESS is given by

$$\text{ESS} = \frac{1}{\sum_{i=1}^N \left(\hat{w}_n^{(i)}\right)^2}, \tag{32}$$

and is a number between 1 and N. If the ESS $= N$, the interpretation is that the weights are equally balanced and that all N particles are contributing to the estimator. At each iteration of the algorithm, the user calculates the ESS and, if it drops below a user chosen threshold, then resampling is performed. Resampling is therefore performed at random times, see Algorithm 4 (see Table 4). The threshold for the ESS is typically chosen to be a percentage of the number of particles, say 0.5 to 0.75.

Two other commonly used measures of weight instability are the coefficient of variation (CV) of Kong et al. (1994) and the Shannon entropy of the weights. These can be substituted into Algorithm 4 instead of the ESS. The CV is defined as

$$\text{CV} = \left[\frac{1}{N}\sum_{i=1}^N \left(N\hat{w}_n^{(i)} - 1\right)^2\right]^{0.5}, \tag{33}$$

Table 4 Algorithm 4: SISR filter with resampling at random times

At $n = 0$, for $i = 1, \ldots, N$

Draw $\mathbf{x}_0^{(i)} \sim g_0(\mathbf{x}_0)$ and set $w_0^{(i)} = \frac{p(\mathbf{x}_0^{(i)})}{g_0(\mathbf{x}_0^{(i)})}$.

For $n = 1, \ldots, T$:

(i) For $i = 1, \ldots, N$ draw $\mathbf{x}_n^{(i)} \sim g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}^{(i)}, \mathbf{y}_n; \psi\right)$ and compute the importance weights $w_n^{(i)} \propto w_{n-1}^{(i)} \frac{p\left(\mathbf{y}_n \mid \mathbf{x}_n^{(i)}; \theta\right) p\left(\mathbf{x}_n^{(i)} \mid \mathbf{x}_{n-1}^{(i)}; \theta\right)}{g_n\left(\mathbf{x}_n^{(i)} \mid \mathbf{x}_{n-1}^{(i)}, \mathbf{y}_n; \psi\right)}$.

(ii) For $i = 1, \ldots, N$ normalize the importance weights: $\widehat{w}_n^{(i)} = \frac{w_n^{(i)}}{\sum_{j=1}^N w_n^{(j)}}$.

(iii) Compute the ESS.

(iv) If ESS $\leq$ threshold, resample N particles with probabilities $\left\{\widehat{w}_n^{(i)}\right\}_{i=1}^N$ and for $i = 1, \ldots, N$ set $w_n^{(i)} = \frac{1}{N}$ else, if ESS $>$ threshold, set $w_n^{(i)} = \widehat{w}_n^{(i)}$ for $i = 1, \ldots, N$.

and is a number between zero and $\sqrt{N-1}$.[3] If all the weights are equal, then CV = 0, and if one particle has all the probability mass, then CV = $\sqrt{N-1}$. Note that ESS = $N/(1 + \text{CV}^2)$. The Shannon entropy (SE) is

$$\text{SE} = -\sum_{i=1}^{N} \hat{w}_n^{(i)} \log_2 \hat{w}_n^{(i)}, \tag{34}$$

which is minimal at zero when one particle has all the mass. Its largest value is $\log_2 N$ when all the weights are equal. When using the CV and SE criteria to determine when to resample, the threshold will depend upon the model and on the particle size N.

An alternative to resampling algorithms for rejuvenating the particles are "branching" algorithms, which are popular in the theoretical probability literature and are reasonably simple to implement. In most implementations, the number of particles will be random over time N_n and therefore these methods are not as common in applications. For more details on branching algorithms; see, e.g., Crisan et al. (1999) and Del Moral and Miclo (2000).

2.7. Particle Smoothing and Maximum a Posteriori Estimation

The marginal smoothing distribution $p\left(\mathbf{x}_n \mid \mathbf{y}_{1:T}; \theta\right)$ characterizes the state variable given all T observations in the dataset. Computing the distribution $p\left(\mathbf{x}_n \mid \mathbf{y}_{1:T}; \theta\right)$ for all possible n while T is held fixed is the most common form of smoothing in economics. This is known as *fixed-interval smoothing* in the engineering literature; see, e.g., Anderson and Moore

[3]The squared coefficient of variance CV^2 is equal to the estimator of the asymptotic variance for the self-normalized IS estimator (see Geweke, 1989, 2005), where the function being integrated is equal to one, i.e., $f(\mathbf{x}_{0:n}) = 1$.

(1979). Fixed-interval smoothing algorithms for state space models are historically based upon one of two frameworks known as *forward-filtering backward-smoothing* or *two-filter formula smoothing*. Both types of algorithms compute the same sequence of marginal distributions $\left\{p\left(\mathbf{x}_n \mid \mathbf{y}_{1:T}; \theta\right)\right\}_{n=1}^{T}$. Particle smoothing algorithms have been created using both approaches. A good reference for this material is Briers et al. (2004) on which my discussion is based while Chapter 3 of Cappé et al. (2005) contains a more general, measure-theoretic treatment.

After running a filtering algorithm forward and computing each of the predictive and filtering distributions $\left\{p\left(\mathbf{x}_{n+1} \mid \mathbf{y}_{1:n}; \theta\right), p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n}; \theta\right)\right\}_{n=1}^{T}$, the sequence of marginal smoothing distributions can be computed from $n = T-1, \ldots, 1$ using the following backward recursion

$$\begin{aligned} p\left(\mathbf{x}_n \mid \mathbf{y}_{1:T}; \theta\right) &= \int p\left(\mathbf{x}_n, \mathbf{x}_{n+1} \mid \mathbf{y}_{1:T}; \theta\right) d\mathbf{x}_{n+1}, \\ &= \int p\left(\mathbf{x}_{n+1} \mid \mathbf{y}_{1:T}; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{x}_{n+1}, \mathbf{y}_{1:T}; \theta\right) d\mathbf{x}_{n+1}, \\ &= \int p\left(\mathbf{x}_{n+1} \mid \mathbf{y}_{1:T}; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{x}_{n+1}, \mathbf{y}_{1:t}; \theta\right) d\mathbf{x}_{n+1}, \\ &= p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n}; \theta\right) \int \frac{p\left(\mathbf{x}_{n+1} \mid \mathbf{y}_{1:T}; \theta\right) p\left(\mathbf{x}_{n+1} \mid \mathbf{x}_n; \theta\right)}{p\left(\mathbf{x}_{n+1} \mid \mathbf{y}_{1:n}; \theta\right)} d\mathbf{x}_{n+1}. \end{aligned} \tag{35}$$

The backward recursion is initialized using the last filtering distribution $p\left(\mathbf{x}_T \mid \mathbf{y}_{1:T}; \theta\right)$ and the predictive distribution $p\left(\mathbf{x}_{T+1} \mid \mathbf{y}_{1:T}; \theta\right)$ from the forward filtering recursions. The smoothing algorithms for the linear, Gaussian state space model that are popular in economics, e.g., Harvey (1989), Kim and Nelson (1999), and Durbin and Koopman (2001), are versions of this approach based upon original work by Rauch et al. (1965). Doucet et al. (2000) invented a particle smoother using this framework, but it is an $O(N^2T)$ operation making it uncompetitive with MCMC. A second shortcoming is the particles' locations on the support of the distributions are fixed on the forward filtering pass. These particles are then simply reweighted by changing their importance weights on a backwards pass using the information in the future data. Although future observations are available, new particle locations are not simulated on the backwards pass. These one-sided particle locations may not be representative of the marginal smoothing distributions given more data.

Two-filter formula smoothing consists of running two filters that are independent of one another and using their output to construct the marginal smoothing distributions. This method was proposed by Fraser and Potter (1969) for linear, Gaussian models. The first filter

calculates the one-step ahead predictive and filtering distributions $\left\{p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n-1};\theta\right), p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n};\theta\right)\right\}_{n=1}^{T}$ running forward in time and the second filter calculates a series of functions $\left\{p\left(\mathbf{y}_{n:T} \mid \mathbf{x}_n;\theta\right)\right\}_{n=1}^{T}$ running backward in time. Together these can compute the marginal smoothing distributions using the forward recursion

$$\begin{aligned} p\left(\mathbf{x}_n \mid \mathbf{y}_{1:T};\theta\right) &= p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n-1}, \mathbf{y}_{n:T};\theta\right), \\ &= \frac{p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n-1};\theta\right) p\left(\mathbf{y}_{n:T} \mid \mathbf{y}_{1:n-1}, \mathbf{x}_n;\theta\right)}{p\left(\mathbf{y}_{n:T} \mid \mathbf{y}_{1:n-1};\theta\right)}, \\ &\propto p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n-1};\theta\right) p\left(\mathbf{y}_{n:T} \mid \mathbf{x}_n;\theta\right), \\ &\propto p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n};\theta\right) p\left(\mathbf{y}_{n+1:T} \mid \mathbf{x}_n;\theta\right). \end{aligned}$$

The set of backward functions $p\left(\mathbf{y}_{n:T} \mid \mathbf{x}_n;\theta\right)$ can be computed recursively via

$$p\left(\mathbf{y}_{n:T} \mid \mathbf{x}_n;\theta\right) = \int p\left(\mathbf{y}_{n+1:T} \mid \mathbf{x}_{n+1};\theta\right) p\left(\mathbf{x}_{n+1} \mid \mathbf{x}_n;\theta\right) p\left(\mathbf{y}_n \mid \mathbf{x}_n;\theta\right) d\mathbf{x}_{n+1},$$

which is known as the *backward information filter* and was first proposed by Mayne (1966). Difficulty may sometimes arise with this approach because $p\left(\mathbf{y}_{n:T} \mid \mathbf{x}_n;\theta\right)$ is not a probability density. The integral of this function can grow without bound (the integral is infinite). Practical implementations of two-filter formula smoothing are therefore based on normalization of $p\left(\mathbf{y}_{n:T} \mid \mathbf{x}_n;\theta\right)$ to ensure that it is a density.

Kitagawa (1996) proposed the first particle smoother based upon two-filter formula smoothing. However, this algorithm implicitly assumes that $p\left(\mathbf{y}_{n:T} \mid \mathbf{x}_n;\theta\right)$ is integrable. Briers et al. (2004) develop a two-filter formula particle smoother that solves the integrability problem. Their method also simulates fresh particle locations on the backward pass, but it remains an $O(N^2T)$ operation. Building on this work, Fearnhead et al. (2008) have recently shown how to apply a two-filter formula particle smoother which is only an $O(NT)$ operation making it competitive with MCMC. This smoother does not solve the problem for all general state space models (1) and (2) but applies to only those models whose state equation is linear and Gaussian. This is typically the case in economics. Details of the implementation of the algorithm are relatively lengthy, and therefore, we refer to their article for further discussion.

Godsill et al. (2004) developed a simulation smoothing algorithm for a general nonlinear, non-Gaussian state space model using particle filters that is an $O(NT)$ operation. A simulation smoother is an algorithm that takes random draws of a sequence of state variables $\mathbf{x}_{0:T}$ from the joint smoothing distribution $p\left(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T};\theta\right)$. Their method can be viewed as an extension of the simulation smoothing algorithms for linear, Gaussian

models of Carter and Kohn (1994), Frühwirth-Schnatter (1994), de Jong and Shephard (1995), and Durbin and Koopman (2002). By repeatedly drawing samples from this distribution, smoothed estimates of the state variable can be computed by averaging across the simulations as in standard i.i.d. Monte Carlo methods. The algorithm is particularly simple, and therefore, we refer to Godsill et al. (2004) for its implementation. Recently, Douc et al. (2009) have provided a theoretical analysis of this algorithm and have suggested ways of improving its implementation.

Particle smoothing can be used to approximate the mean of the marginal or joint smoothing distributions. This is the optimal estimator if the user has a quadratic loss function. Viewing the joint smoothing distribution as a posterior distribution, it is also possible for particle filters to approximate the *maximum a posteriori* (MAP) estimator. This is the sequence $\mathbf{x}_{0:T}$ that maximizes the posterior distribution $p\left(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T}; \theta\right)$ and is the optimal estimator under a zero-one loss function. Godsill et al. (2002) solve this problem by extending the well-known Viterbi (1967) algorithm for discrete-state HMM models to the context of particle filters. The algorithm is a simple dynamic programming problem and is an $O(N^2 T)$ operation.

2.8. Parameter Estimation and Hypothesis Testing Using Particle Methods

2.8.1. *Computing the Likelihood for a General State Space Model*

The log-likelihood of a time series model is given by the prediction error decomposition

$$\log L\left(\theta \,|\, \mathbf{y}_{1:T}\right) = \log p\left(\mathbf{y}_1, \ldots, \mathbf{y}_T; \theta\right) = \sum_{n=1}^{T} \log p\left(\mathbf{y}_n \,|\, \mathbf{y}_{1:n-1}; \theta\right).$$

The particle filter's approximation of the likelihood function for a single observation was given in (20). It is an unbiased estimator. Taking the log of this approximation and summing over all the observations gives

$$\log L\left(\theta \,|\, \mathbf{y}_{1:T}\right) \approx \sum_{n=1}^{T} \log \left[\sum_{i=1}^{N} w_{n-1}^{(i)} \tilde{w}_n^{(i)} \right].$$

The particle filters estimator of the likelihood can be used in hypothesis testing as well as in parameter estimation.

2.8.2. *Hypothesis Testing and Model Evaluation*

In a frequentist setting, estimates of the likelihood can be used for model diagnostics such as likelihood ratio and Ljung–Box statistics; see

Andrieu et al. (2004, p. 429) for details. When θ is treated as a random variable with prior $p(\theta)$, the likelihood can be combined with the prior to obtain the posterior $p(\theta \mid \mathbf{y}_{1:T})$. Given a set of models $m = 1, \ldots, M$, the particle filter can be used to help compute the marginal likelihood $p(\mathbf{y}_{1:T} \mid m) = \frac{p(\mathbf{y}_{1:T} \mid \theta, m) p(\theta \mid m)}{p(\theta \mid \mathbf{y}_{1:T}, m)}$. The marginal likelihood is needed for Bayesian hypothesis testing; see, e.g., Geweke (2005) and Chapter 5 of Frühwirth-Schnatter (2006) for a general discussion of different ways to compute marginal likelihoods. Kim et al. (1998) and Chib et al. (2002) among others use the particle filter's approximation of the likelihood to compute the marginal likelihood and compare different stochastic volatility models.

2.8.3. *Frequentist Likelihood-Based Parameter Estimation*

Frequentist parameter estimation of nonlinear, non-Gaussian state space models by particle filters remains a current research topic. The two major issues to consider are computing the maximum likelihood (ML) estimator in a computationally efficient way and its statistical properties (i.e., consistency and asymptotic normality) once it is computed. Although the particle filter's approximation of the likelihood function at a point θ is consistent asymptotically in the number of particles, the log-likelihood function is not a continuous function of the parameters. This is true even if one tries to use common random numbers. The discontinuity is created from the resampling stage within a particle filter and can cause problems for gradient-based optimizers; e.g., see Hürzeler and Künsch (2001) for a detailed example of the problem.

Pitt (2002) developed a new algorithm called the smooth particle filter to overcome the problem of a non-smooth log-likelihood function. This algorithm replaces the standard resampling algorithm with a new resampling method. It builds a continuous c.d.f. using piecewise linear approximations between particles instead of the discrete c.d.f. used in the standard resampling algorithms. Pitt's algorithm is only viable when the state dimension is equal to one or perhaps two because smoothing the c.d.f. requires the ordering of the state variables during each iteration of the filter. The method becomes an $O(N^2 T)$ operation beyond a one-dimensional state vector.

Olsson and Rydén (2008) consider maximization of the log-likelihood and also address the resulting estimator's theoretical properties. They approximate the parameter space using a discrete grid of points and evaluate the log-likelihood function by particle filter at each point. They then prove what conditions are needed on the grid size, the number of particles, and the state space model in order to guarantee consistency and asymptotic normality of the resulting ML estimator. This appears to be the first result of this kind.

Otherwise, most work on ML estimation using particle filters has focused on using approaches other than gradient-based optimizers that avoid the discontinuity problem. These methods include stochastic gradient-based methods, recursive maximum likelihood methods (Doucet and Tadić, 2003; Poyiadjis et al., 2005a,b) and Monte Carlo expectation maximization (MCEM) methods (Cappé et al., 2005; Olsson et al., 2008). The last article also analyzes the statistical properties of the estimator. To my knowledge, none of these methods have been applied in the economics literature.

2.8.4. *Alternative Methods and Online Estimation*

A number of other proposals have been made for estimating the parameters of general state space models using particle methods. In particular, researchers are interested in accounting for parameter uncertainty by approximating the distributions $p\left(\mathbf{x}_{0:n}, \theta \,|\, \mathbf{y}_{1:n}\right)$ and $p\left(\mathbf{x}_n, \theta \,|\, \mathbf{y}_{1:n}\right)$ online as data arrives. These provide estimates of the marginal distributions $p\left(\theta \,|\, \mathbf{y}_{1:n}\right)$ and $p\left(\mathbf{x}_n \,|\, \mathbf{y}_{1:n}\right)$ instead of the traditional marginal filtering distribution $p\left(\mathbf{x}_n \,|\, \mathbf{y}_{1:n}; \theta\right)$ which does not account for parameter uncertainty.

Research in this area is still on-going. Some of the earlier methods are reviewed in Doucet et al. (2001). Assume throughout this section that we are in a Bayesian setting with an initial prior for the parameters $p(\theta)$ at time $n = 0$. Kitagawa (1998) proposed simulating from $p(\theta)$ and then placing the parameters in the state vector with the variance set to zero, i.e., $\theta_n = \theta \;\forall\; n$. Unfortunately, the parameter space is then only explored at initialization of the algorithm and after several stages of resampling the particles will consist of only one value of θ making for a poor estimator. Kitagawa (1998) and Liu and West (2001) proposed making the parameters dynamic, $\theta_n = \theta_{n-1} + \omega_n$, where ω_n is artificial noise whose covariance is converging to zero asymptotically as $n \to \infty$. The parameters are then added to the state vector. Liu and West (2001) considered an extension of this where they approximate the distribution $p(\theta \,|\, \mathbf{y}_{1:n})$ using a nonparametric kernel, e.g., Epanechnikov or normal. This has been a popular approach taken in the literature. However, some researchers argue that this changes the original problem of interest as the true parameters in the model of interest are fixed and not dynamic. Selecting the tuning parameters within the nonparametric kernel may also be difficult in practice.

Storvik (2002) proposed learning the parameters sequentially in time by simulating new parameter values $\theta = (\theta_1, \ldots, \theta_k)$ each period along with the state variables. New parameter values are drawn from their posterior conditional distribution which he summarizes by a set of low-dimensional sufficient statistics. Fearnhead (2002), Johannes et al. (2006), Carvalho

et al. (2008), and Johannes et al. (2008) apply more advanced versions of these methods to several applications. The basic idea behind these methods is that in many models the posterior conditional distribution of each individual parameter $p(\theta_i \mid \mathbf{y}_{1:n}, \mathbf{x}_{0:n})$ can often be summarized by a low dimensional set of sufficient statistics denoted by $\mathbf{s}_n = s(\mathbf{y}_{1:n}, \mathbf{x}_{0:n})$. The sufficient statistics can often be written as a recursive function of the past period's sufficient statistics, the new data, and the state variable from the current time period such that $\mathbf{s}_n = s(\mathbf{y}_{1:n}, \mathbf{x}_{0:n}) = \tilde{s}(\mathbf{y}_n, \mathbf{x}_n, \mathbf{s}_{n-1})$. Instead of storing the entire history of the particles $\left\{\mathbf{x}_{0:n-1}^{(i)}\right\}_{i=1}^{N}$, the algorithm only needs to store the sufficient statistics $\left\{\mathbf{s}_n^{(i)}\right\}_{i=1}^{N}$ which provides computational savings. Redefine each particle as $\tilde{\mathbf{x}}_n = (\mathbf{s}_n', \mathbf{x}_n', \theta')'$. At each iteration, new state variables are simulated from their proposal distribution conditional on the existing parameters and then the sufficient statistics are updated using the recursion $\mathbf{s}_n = \tilde{s}(\mathbf{y}_n, \mathbf{x}_n, \mathbf{s}_{n-1})$. Conditional on the sufficient statistics, new parameter values are simulated from their posterior conditional distribution. The particles $\tilde{\mathbf{x}}_n$ are reweighted given the particle filter's importance weights and then resampled (potentially conditional on the ESS as noted above). The methodology outlined in these articles has the advantage of being extremely simple to implement.

Andrieu et al. (1999), Andrieu et al. (2005), and Künsch (2006) note, however, that the success of these methods will depend upon the mixing properties of the Markov kernels within the algorithm. Past errors produced by the particle filter's approximations need to be forgotten and not accumulated over time. Keep in mind that the particle filter does not generally give a good approximation of the joint distribution $p\left(\mathbf{x}_{0:n} \mid \mathbf{y}_{1:n}; \theta\right)$ for large n because there will be few particles representing early parts of the distribution (see Example #1 and Fig. 1). As n increases, the particles representing $p\left(\mathbf{x}_{0:n-k} \mid \mathbf{y}_{1:n}; \theta\right)$ for large k should contribute information toward estimating the parameters θ. However, information about θ may not always accumulate if there exist few particle paths representing this part of the joint distribution, see Andrieu et al. (2005) for further discussion.

3. THEORETICAL PROPERTIES

Early reviews of the theoretical properties of particle filters can be found in Chapters 2 and 3 of Doucet et al. (2001) and Crisan and Doucet (2002), while a full-length treatment is given by Del Moral (2004). Recent papers on consistency and asymptotic normality for more general classes of algorithms are Chopin (2004), Künsch (2005), Douc and Moulines (2008), and Del Moral et al. (2008). The goal of this section is to discuss some of the main results at an intuitive level and describe what their implications are for applied researchers.

3.1. Consistency and Asymptotic Normality

At each iteration, a particle filter produces samples $\left\{\mathbf{x}_{0:n}^{(i)}, w_n^{(i)}\right\}_{i=1}^{N}$ that can be used to approximate the expectation of a function f with respect to the joint smoothing distribution. The exact conditions for consistency and asymptotic normality of the estimator depend upon the particle filter one implements. Proofs in the literature vary accordingly with different types of regularity conditions favored by different authors. It is not possible to cover all the results in the literature and the different types of regularity conditions. Instead, the discussion here is limited to Theorem 1 from Chopin (2004), whose regularity conditions are relatively simple. This theorem covers the SISR and APF algorithms under multinomial sampling at each iteration (i.e., Algorithms 1, 2, and 3).

Standard IS algorithms require some regularity conditions on the importance weights and the set of functions f within the integrals for which the estimator will be well defined; see, e.g., Geweke (2005, p. 114). These are that the importance weights remain bounded so that the estimator remains well-behaved. In addition, the function f within the integral must have finite variance when evaluated under the target distribution. There are similar conditions for particle filters limiting both the set of functions f that are valid and conditions to ensure that the variance of the importance weights is finite at each iteration. In a particle filter, the importance weights are determined recursively through the weight recursion (11). The variability of the importance weights w_n depends on the Monte Carlo variation introduced at the current iteration as well as any variability that is carried over from previous periods. This is due to the fact that particles simulated at previous iterations form part of the future joint importance distribution through the Dirac measure on past paths, see (10).

A particle filter produces several estimators of interest. The results of Chopin (2004) cover three cases: the estimator of the moments prior to resampling given by (15), the estimator of the moments after resampling given by (17), and the estimator of the marginal distribution of the unweighted particles. We consider here only the case of the marginal filtering distribution $p(\mathbf{x}_n \mid \mathbf{y}_{1:n}; \theta)$, and we denote by $\mathbb{E}_{p_n}$ the expectation with respect to this distribution. The estimator of the marginal distribution of the unweighted particles $\mathbf{x}_n$ is generally not of interest, but it is used as an intermediate quantity in the theorem. This marginal can be defined recursively as

$$g_n(\mathbf{x}_n) = \int g_n(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \theta) g_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1},$$

where we omit the fact that this distribution is implicitly a function of the observations and θ. Let $\mathbb{E}_{g_n}$ denote the expectation of a function f with

respect to this distribution, and let $\mathbb{E}_{g_{0:n}}$ be the expectation with respect to the joint distribution (10).

The initial iteration of a particle filter is a standard importance sampling iteration. Therefore, the standard importance sampling assumptions apply to the first iteration $n = 0$. Given the results in Geweke (1989), these are given by:

(i) $\mathbb{E}_{p_0}[f(\mathbf{x}_0)]$ exists;
(ii) $\mathbb{V}_{p_0}[f(\mathbf{x}_0)]$ exists;
(iii) The support of the initial importance distribution $g_0(\mathbf{x}_0)$ includes the target $p(\mathbf{x}_0)$;
(iv) The initial importance weights w_0 are bounded.

Proofs of consistency and asymptotic normality for later iterations of a particle filter follow by induction. If one starts with a sample that produces a consistent and asymptotically normal estimator for a function f, then one iteration of the sampling and resampling operations produce a new sample whose estimator is also consistent and asymptotically normal for any function f within a class of functions. The basic idea behind the results of Chopin (2004) (and many other results in the literature) follows by noting that, conditional on the past draws, each new iteration of the algorithm produces a set of draws that are independent but not identically distributed. It is then possible to apply a conditional law of large numbers (LLN) and central limit theorem (CLT) to each iteration.

Let $\|.\|$ denote the Euclidean norm in $\mathbf{R}^m$. Denote by $\Phi_n^{(m)}$ the set of measurable functions for which the estimator will be consistent and asymptotically normal at iteration n. This set of functions will be determined recursively. The initial set of functions $\Psi_0^{(m)}$ contains those measurable functions whose second moments are finite with respect to the initial proposal distribution (implied by conditions (ii) and (iv) above). At later iterations, the set $\Phi_n^{(m)}$ is constrained by the following two conditions:

(a) For some $\delta > 0$, $\mathbb{E}_{g_n}\|w_n f(\mathbf{x}_n)\|^{2+\delta} < \infty$;
(b) The function $\mathbb{E}_{g_{0:n}(\mathbf{x}_{n-1},.)}[w_n(.)f(.)] \in \Phi_{n-1}$;

Condition (a) implies that the importance weights (for a function f) must have finite moments of order $2 + \delta$ with respect to g_n. Additional assumptions necessary for future iterations are:

(v) For all $n \geq 0$, the constant function belongs to $\Phi_n^{(1)}$;
(vi) For all $n \geq 0$, the support of the incremental importance distribution $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \psi\right)$ includes the support of the incremental target $p\left(\mathbf{y}_n \mid \mathbf{x}_n; \theta\right) p\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta\right)$;

The last assumption is a necessary condition for all IS algorithms.

Given the discussion above, Theorem 1 of Chopin (2004) states that the three estimators produced by the particle filter are consistent at iteration n for any measurable function $f \in \Phi_n^{(m)}$ meaning that as $N \to \infty$

$$\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_n^{(i)}) \xrightarrow{a.s.} \mathbb{E}_{g_n}\left[f(\mathbf{x}_n)\right], \tag{36}$$

$$\sum_{i=1}^{N} \hat{w}_n^{(i)} f(\mathbf{x}_n^{(i)}) \xrightarrow{a.s.} \mathbb{E}_{p_n}\left[f(\mathbf{x}_n)\right], \tag{37}$$

$$\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_n^{(i)}) \xrightarrow{a.s.} \mathbb{E}_{p_n}\left[f(\mathbf{x}_n)\right]. \tag{38}$$

The estimators (37) and (38) differ by the fact that the former is computed before the particles are resampled while the latter is computed using the particles after the resampling step. The two estimators (37) and (38) are asymptotically normal for any measurable function $f \in \Phi_n^{(m)}$ meaning that as $N \to \infty$

$$\sqrt{N}\left[\sum_{i=1}^{N} \hat{w}_n^{(i)} f(\mathbf{x}_n) - \mathbb{E}_{p_n}\left[f(\mathbf{x}_n)\right]\right] \xrightarrow{d} \mathcal{N}\left(0, V_n(f)\right), \tag{39}$$

$$\sqrt{N}\left[\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_n) - \mathbb{E}_{p_n}\left[f(\mathbf{x}_n)\right]\right] \xrightarrow{d} \mathcal{N}\left(0, \widehat{V}_n(f)\right), \tag{40}$$

where $V_n(f)$ and $\widehat{V}_n(f)$ are the respective asymptotic variances. The notation makes explicit that the asymptotic variances are a function of the function f being integrated.

The asymptotic variances within the CLTs are written recursively as

$$\widetilde{V}_n(f) = \widehat{V}_{n-1}\left\{\mathbb{E}_{g_{0:n}}[f(\mathbf{x}_n)]\right\} + \mathbb{E}_{p_{n-1}}\left\{\mathrm{Var}_{g_{0:n}}[f(\mathbf{x}_n)]\right\}, \qquad n > 0, \tag{41}$$

$$V_n(f) = \widetilde{V}_n\left\{w_n(f(\mathbf{x}_n) - \mathbb{E}_{p_n}[f(\mathbf{x}_n)])\right\}, \qquad n \geq 0, \tag{42}$$

$$\widehat{V}_n(f) = V_n(f) + \mathrm{Var}_{p_n}[f(\mathbf{x}_n)], \qquad n \geq 0. \tag{43}$$

where the recursions are initialized with $\widetilde{V}_0(f) = \mathrm{Var}_{g_0}[f(\mathbf{x}_0)]$. A particle filter consists of three basic steps: (i) drawing new particles at each iteration; (ii) weighting the draws with their importance weights; and (iii) resampling the particles. The asymptotic variance is broken into three parts and written recursively to demonstrate the impact each of these three steps has on the variability of the estimator in each period. For example, the difference between $V_n(f)$ and $\widehat{V}_n(f)$ in (42) and (43) is the additional

variance created by the resampling step. As noted by Chopin (2004), the resampling step has an additive effect and will always increase the variance of the current period's estimator (for any nonconstant function f) by the factor $\mathrm{Var}_{p_n}[f(\mathbf{x}_n)]$. This is why the estimator computed from the particles before resampling is preferred to the estimator available after resampling.

The expressions for the asymptotic variance will depend on the particle filter one implements, including what type of resampling. Several authors have evaluated the relative numerical efficiency of different algorithms by comparing these theoretical expressions; see, e.g., Chopin (2004) and Künsch (2005). For example, both these authors prove that the asymptotic variance of the SISR algorithm with residual resampling will be strictly smaller than the SISR filter with multinomial resampling. Chopin (2004) also verifies the discussion from Section 2.5.7 that Rao–Blackwellization of a particle filter will always increase its numerical efficiency. In applications of standard importance sampling, an estimator of the asymptotic variance within the CLT is often used to measure the ex-post numerical efficiency of the simulation algorithm; these are the numerical standard errors, see, e.g., Geweke (2005). The asymptotic variance expressions for particle filters are complicated enough that estimators of the asymptotic variances have not been seriously evaluated in the particle filtering literature, although one was proposed by Gilks and Berzuini (2001). Most practioners simply use the ESS to measure instability. Under some simplifying assumptions, Johansen and Doucet (2008) show that it is possible to write the asymptotic variance expressions explicitly for the SISR and APF algorithms.

For standard importance sampling, the variance of the importance weights grows to infinity as the time dimension increases. A theoretical result that has important implications for applied researchers using particle filters is that the asymptotic variance in the CLT can be proven to be finite and bounded by a constant that is not a function of time. In practice, this means that a particle filter can be applied to a long stretch of time series (e.g., financial time series) and the precision of the estimator will not systematically deteriorate as one obtains more observations.[4] In this sense, the particle filter forgets its past errors. We note that these results have been established for the marginal filtering distribution but generally will not hold for the joint smoothing distribution (due to the resampling step causing the past particles to coalesce). In addition, they do not apply for *all* particle filters. These results generally require additional regularity conditions such that the transition density $p(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta)$ of the state space model as well as the Markovian transition

[4]Keep in mind that the estimator may be poor in some time periods when $g_n(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \theta)$ is a poor approximation of $p(\mathbf{y}_n \mid \mathbf{x}_n; \theta)\, p(\mathbf{x}_n \mid \mathbf{x}_{n-1}; \theta)$.

kernel $g_n(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \theta)$ have good mixing properties.[5] Bounds on the asymptotic variance have been obtained by many authors for different types of particle filtering algorithms and under different metrics, see Del Moral and Guionnet (2001), Le Gland and Oudjane (2004), Del Moral (2004), Künsch (2005), Chapter 9 of Cappé et al. (2005), and Douc et al. (2009c).

Although the asymptotic variance within the CLT can be proven to be bounded by a finite constant, this constant is (generally) a function of the dimension of the state vector and will get larger as the state vector gets larger. In practice, this means that as the state vector's dimension grows the only way to keep the same level of efficiency is to increase the number of particles. Note that this is a problem shared by all Monte Carlo estimators including MCMC, IS, etc., and is not necessarily a criticism of only particle filters.

3.2. Additional References on Consistency and Asymptotic Normality

The first consistency proof of the bootstrap filter was given by Del Moral (1996) with asymptotic normality established in Del Moral and Guionnet (1999) and Del Moral and Miclo (2000). The CLT was later extended to include more advanced algorithms in Chopin (2004) and Künsch (2005). These include the SISR, APF, and resample move algorithms under multinomial and residual resampling. Künsch (2005) provides a CLT for particle filters based on the accept–reject algorithm. Del Moral (2004) includes consistency and asymptotic normality results for the particle filters' estimator of the likelihood $p(\mathbf{y}_n; \theta)$ of the state space model. Douc and Moulines (2008) and Del Moral et al. (2008) both prove consistency and asymptotic normality for algorithms that resample at random times (see Algorithm 4). Douc and Moulines (2008) consider algorithms that resample at random times via the coefficient of variation (CV) while Del Moral et al. (2008) covers algorithms that resample according to the effective sample size (ESS). Douc et al. (2009c) is an in-depth analysis of the auxiliary particle filter using the limit theorems from Douc and Moulines (2008). In the discussion above, we focused

[5]For example, Theorem 5 of Chopin (2004) states that the asymptotic variance (42) will remain bounded if there exist constants C, $\underline{p}$, $\overline{p}$ such that for any $n \geq 0$:

(a) For any $x, x', x'' \in \mathrm{X}$, the transition density satisfies $\frac{p(x' \mid x;\theta)}{p_n(x'' \mid x;\theta)} \leq \mathrm{C}$;
(b) For any $x, x', x'' \in \mathrm{X}$, the incremental importance density satisfies $\frac{g_n(x' \mid x;\psi)}{g_n(x'' \mid x;\psi)} \leq \mathrm{C}$;
(c) For any $x \in \mathrm{X}$, $y \in \mathrm{Y}$, the observation density satisfies $0 < \underline{p} \leq p(y \mid x; \theta) \leq \overline{p}$;

where X and Y are the state spaces of the Markov chain.

on consistency and asymptotic normality of estimators of the moments. Some of the authors, particularly Del Moral (2004) and Künsch (2005), investigate empirical process results and the convergence of the empirical distribution function to the true distribution.

3.3. Properties Other Than Consistency and Asymptotic Normality

Del Moral (2004) includes more advanced coverage of particle systems including properties other than consistency and asymptotic normality. These include results on Berry–Esseen theorems for the CLT, empirical process theory, large deviations, and propogation of chaos properties. Del Moral and Doucet (2009) provides a recent review of theoretical properties using Feynman-Kac path integral theory as a unifying framework. There also exists a literature on forgetting of the initial condition and initial distribution, see Douc et al. (2009) with additional references therein. This is relevant when the initial condition is miss-specified, and we would like to know how this miss-specification affects the performance of the algorithm. Finally, this survey does not cover contributions in the applied probability theory literature that analyze optimal filtering algorithms for continuous-time models; see, e.g., Bain and Crisan (2008) and Xiong (2008).

4. RECENT DEVELOPMENTS IN SEQUENTIAL MONTE CARLO

This section covers two more recent developments that extend SMC outside the context of traditional particle filtering. In the first extension, researchers working in Monte Carlo methods recognized that SMC could be used to simulate from sequences of distributions other than the filtering distributions defined by a state space model. These methods are particularly applicable to Bayesian inference problems because they provide an alternative to MCMC for simulating from complex distributions. They can also be applied to models for cross-sectional data. Sections 4.1–4.2 review this research. The second extension of standard particle filtering uses the particle filter to provide a proposal distribution for draws within MCMC algorithms. This type of algorithm is currently being used in the macroeconomics literature on Bayesian estimation of DSGE models. Section 4.3 covers this material.

4.1. SMC Samplers

Recognizing that the particles form a collection of interacting Markov chains on a sequence of general state spaces is the key to building other types of SMC algorithms. Leading references in this field include Gilks and

Berzuini (2001), Chopin (2002), Liang (2002), and Cappé et al. (2004). Del Moral et al. (2006a,b) built a framework titled *SMC samplers* that encompasses a number of the algorithms in the literature.

Research in this area of Monte Carlo methods is on-going. There are several key themes in this research: (i) an emphasis on building adaptive Monte Carlo algorithms that learn from their previous draws; (ii) understanding the practical circumstances where allowing the Markov chains to interact is beneficial relative to MCMC; and (iii) developing the necessary limit theory to justify the methods in practice.

Consider a setting where a researcher would like to sample from a sequence of probability distributions, $\{p_n(\mathbf{x}_n)\}_{n=1}^{J}$. The iteration number n in the sequence is a counter that may or may not represent calendar time. The number of observations in the researcher's sample (not necessarily a time series) is denoted by T while J is the number of distributions in the sequence. The random variable or particle $\mathbf{x}_n$ is no longer restricted to denote a state variable in a state space model as in Section 2. It is simply a quantity of interest with its interpretation depending upon the application. Some examples are:

(i) Filtering in state space models: the sequence of target distributions are equal to the marginal filtering distributions with $p_n(\mathbf{x}_n) = p\left(\mathbf{x}_n \,|\, \mathbf{y}_{1:n}\theta\right)$. A particle is equal to the state variable at time n;
(ii) Sequential Bayesian estimation: Consider a Bayesian model resulting in a posterior distribution $p(\theta \,|\, \mathbf{y}_{1:n})$ where θ denotes a $(k \times 1)$ vector of unknown parameters. The sequence of target distributions are equal to the posterior distribution given the number of observations $p_n(\mathbf{x}_n) = p(\theta \,|\, \mathbf{y}_{1:n})$. A particle is equal to the vector of parameters $\mathbf{x}_n = \theta$ for all $n = 1, \ldots, J$.

In the former problem, the sequence of distributions is naturally defined by the problem. The second example demonstrates that the researcher can artificially define the sequence of distributions.

Each density in the sequence is defined as

$$p_n(\mathbf{x}_n) = \frac{\gamma_n(\mathbf{x}_n)}{Z_n}, \tag{44}$$

where $\gamma_n(\mathbf{x}_n)$ is the unnormalized density which can be calculated for any realization of $\mathbf{x}_n$. The normalizing constant Z_n in the denominator of (44) typically includes integrals that cannot be solved analytically.

An SMC sampler begins by drawing N particles $\left\{\mathbf{x}_1^{(i)}\right\}_{i=1}^{N}$ from an initial importance density $g_1(\mathbf{x}_1)$ and reweighting the particles using standard

importance weights. Importance weights at the first iteration are

$$w_1 = \frac{\gamma_1(\mathbf{x}_1)}{g_1(\mathbf{x}_1)}, \tag{45}$$

which can be computed explicitly because the user knows the initial importance density $g_1(\mathbf{x}_1)$. Beginning at the second iteration and continuing forward, each particle is sampled from a forward nonhomogenous Markov transition kernel $\mathbf{x}_n^{(i)} \sim K_n\left(\mathbf{x}_{n-1}^{(i)}, .\right)$. This Markov kernel is simply a generalization of the Markovian importance distribution $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \psi\right)$ within a standard particle filter from Section 2. The marginal distribution of the unweighted particles after drawing from the transition kernel K_n is

$$g_n(\mathbf{x}_n) = \int g_{n-1}(d\mathbf{x}_{n-1}) K_n(\mathbf{x}_{n-1}, \mathbf{x}_n). \tag{46}$$

The importance weights at the nth iteration are the ratio of the target density to the importance density and are given by

$$w_n = \frac{\gamma_n(\mathbf{x}_n)}{g_n(\mathbf{x}_n)}. \tag{47}$$

Unfortunately, the integral in (46) cannot usually be solved analytically for an arbitrary choice of the transition kernel K_n. This makes it impossible to directly calculate the importance weights.

Del Moral et al. (2006a) solve the problem of having to evaluate the unknown importance density $g_n(\mathbf{x}_n)$ to compute importance weights beyond the first iteration by introducing new artificial target densities $p_{1:n}(\mathbf{x}_{1:n})$. The sequence of artificial targets $\{p_{1:n}(\mathbf{x}_{1:n})\}_{n=1}^{J}$ are the joint distributions associated with the sequence of random variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$.[6] The artificial joint densities in an SMC sampler are not of interest in themselves but their introduction allows the importance weights to be computed. An artificial target must be defined up to a normalizing constant

$$p_{1:n}(\mathbf{x}_{1:n}) = \frac{\gamma_{1:n}(\mathbf{x}_{1:n})}{Z_n}, \tag{48}$$

where the new target is intentionally designed to admit $p_n(\mathbf{x}_n)$ as a marginal density. The expanded target is similar to the earlier presentation of the particle filter which operated on the joint smoothing distributions to approximate the marginal filtering distributions. By sampling in a

[6]In a standard particle filter from Section 2, the joint smoothing densities are analagous to the artificial joint densities described here.

larger space, estimates of the marginal using the particles' locations and importance weights can be computed as a by-product.

Del Moral et al. (2006a) provide a framework for choosing both the artificial target densities $p_{1:n}(\mathbf{x}_{1:n})$ as well as the forward Markov kernels. As in Jarzynski (1997) and Neal (2001), they suggest defining the artificial targets as a sequence of artificial backward Markov kernels $L_n(\mathbf{x}_{n+1}, \mathbf{x}_n)$ which can be written as

$$\gamma_{1:n}(\mathbf{x}_{1:n}) = \gamma_n(\mathbf{x}_n) \prod_{k=1}^{n-1} L_k(\mathbf{x}_{k+1}, \mathbf{x}_k). \tag{49}$$

Given particles $\left\{w_{n-1}^{(i)}, \mathbf{x}_{1:n-1}^{(i)}\right\}_{i=1}^{N}$ that approximate the artificial target $\gamma_{1:n-1}(\mathbf{x}_{1:n-1})$, the next artificial target $\gamma_{1:n}(\mathbf{x}_{1:n})$ can be approximated by sampling from the forward Markov kernel. The (unweighted) particles' joint distribution after n transitions is

$$g_{1:n}(\mathbf{x}_{1:n}) = g_1(\mathbf{x}_1) \prod_{j=2}^{n} K_j\left(\mathbf{x}_{j-1}, \mathbf{x}_j\right). \tag{50}$$

Reweighting the particles using the importance weights changes their distribution from $g_{1:n}(\mathbf{x}_{1:n})$ to $p_{1:n}(\mathbf{x}_{1:n})$.

The unnormalized importance weights w_n for the joint distribution are defined as the ratio of the (unnormalized) joint target density (49) to the joint importance density (50) and are given by

$$w_n = \frac{\gamma_{1:n}(\mathbf{x}_{1:n})}{g_{1:n}(\mathbf{x}_{1:n})}. \tag{51}$$

These can be written recursively such that at each iteration one only calculates the incremental importance weights $\tilde{w}_n$ given by

$$w_n = w_{n-1}\tilde{w}_n \qquad \tilde{w}_n = \frac{\gamma_n(\mathbf{x}_n) L_{n-1}(\mathbf{x}_n, \mathbf{x}_{n-1})}{\gamma_{n-1}(\mathbf{x}_{n-1}) K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)}. \tag{52}$$

Notice the similarities between this recursion and (11). After normalizing the importance weights, an estimator of the moments is

$$\mathbb{E}[f(\mathbf{x}_n)] \approx \sum_{i=1}^{N} f\left(\mathbf{x}_n^{(i)}\right) \hat{w}_n^{(i)}, \qquad \hat{w}_n^{(i)} = \frac{w_n^{(i)}}{\sum_{j=1}^{N} w_n^{(j)}}. \tag{53}$$

Estimates of a marginal target distribution can be calculated as

$$p_n(\mathbf{x}_n) \approx \sum_{i=1}^{N} \hat{w}_n^{(i)} \delta_{\mathbf{x}_n^{(i)}}(\mathbf{x}_n).$$

and estimates of the ratio of normalizing constants can be computed as

$$\frac{\widehat{Z}_n}{Z_{n-1}} = \sum_{i=1}^{N} w_{n-1}^{(i)} \tilde{w}_n^{(i)}. \tag{54}$$

If the user chooses an initial distribution where the normalizing constant Z_1 can be calculated, then they obtain an estimate of the normalizing constant for any distribution in the sequence including the final iteration $\widehat{Z}_J$. For example, this could be the marginal likelihood in a Bayesian context or the likelihood of a general state space model.

Like the standard particle filter described previously, it is usually not optimal to resample the particles at each iteration of an SMC sampler. Particles should only be resampled when the variance of the importance weights grows and becomes unstable. This can be measured by any of the criterion described in Section 2.6. A standard SMC sampler is given by Algorithm 5 (see Table 5).

Table 5 Algorithm 5: Sequential Monte Carlo sampler

At $n = 1$, for $i = 1, \ldots, N$

Draw $\mathbf{x}_1^{(i)} \sim g_1(\mathbf{x}_1)$ and set $w_1^{(i)} = \frac{p(\mathbf{x}_1^{(i)})}{g_1(\mathbf{x}_1^{(i)})}$.

For $n = 2, \ldots, J$:

(i) For $i = 1, \ldots, N$ draw $\mathbf{x}_n^{(i)} \sim K_n\left(\mathbf{x}_{n-1}^{(i)}, .\right)$.
and compute the importance weights: $w_n^{(i)} \propto w_{n-1}^{(i)} \tilde{w}_n^{(i)}$.

(ii) For $i = 1, \ldots, N$ normalize the importance weights: $\widehat{w}_n^{(i)} = \frac{w_n^{(i)}}{\sum_{j=1}^{N} w_n^{(j)}}$.

(iii) Calculate the effective sample size (ESS).

(iv) If ESS $\leq$ threshold, resample N particles with probabilities $\left\{\widehat{w}_n^{(i)}\right\}_{i=1}^{N}$ and for $i = 1, \ldots, N$ set $w_n^{(i)} = \frac{1}{N}$ else, if ESS $>$ threshold, set $w_n^{(i)} = \widehat{w}_n^{(i)}$.

Although an SMC sampler is simply a particle filter in a more general context, it requires more input and experience from the user. In a standard particle filter, the sequence of target densities (and implicitly the backwards kernels) are already defined for the user by their state space model. This leaves only the choice of the forward Markov kernel (i.e., the importance distribution $g_n\left(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{y}_n; \psi\right)$) which is relatively easy to select. Conversely in an SMC sampler, the user will have to define the sequence of target densities and choose the forward and backward Markov kernels. Different choices for the forward and backward Markov kernels also determine how challenging it is to compute the incremental weight (52). Del Moral et al. (2006a) provide suggestions to users for choosing each of these quantities in practice. Although these authors consider many options, the easiest algorithms to implement for practitioners with experience using MCMC will be to choose $K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)$ to be a Gibbs or

Metropolis–Hastings kernel. Del Moral et al. (2006a,b) give the equations to calculate the incremental weights (52) when using these kernels, see their article for details. Many of these only involve evaluating the unnormalized target density (44) as in a standard MCMC algorithm.

4.2. Additional References and Comments

A number of algorithms in the Monte Carlo methods literature are special cases of an SMC sampler for specific choices of the sequence of target distributions and forward and backward Markov transition kernels. These include the resample move particle filtering algorithm of Gilks and Berzuini (2001), which has been applied either explicitly or implicitly by a number of authors including Chopin (2002), Chopin and Pelgrin (2004), and Carvalho et al. (2008). Chopin (2002) introduced the concept of applying SMC for static parameter estimation in models without latent state variables; his applications also included cross-sectional data. Chopin and Pelgrin (2004) and Chopin (2007) estimate discrete-state HMM models with the unique ability to estimate the number of states in the HMM as the data-set gets processed. Carvalho et al. (2008) focus on learning the parameters sequentially in time using the sufficient statistics structure proposed by Storvik (2002) as described in Section 2.8.4.

Another special case of an SMC sampler are the Population Monte Carlo (PMC) algorithms developed in a series of papers by Cappé et al. (2004), Celeux et al. (2006), Douc et al. (2007a), Douc et al. (2007b). In most of these papers, the sequence of target distributions are equal at each iteration. The purpose of introducing iterations into the Monte Carlo algorithm is to try and adapt the proposal distribution (i.e., the forward Markov kernel $K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)$) over iterations by using the information in the previously simulated draws.

Additional applications of SMC samplers include Johansen et al. (2006), who consider applications to rare event simulation. Jasra et al. (2008b) use adaptive SMC samplers to estimate Lévy-driven SV models. Jasra et al. (2008a) extended SMC samplers one step further by allowing the particles to be defined on different state spaces at different iterations of the algorithm.

The theoretical analysis of adaptive SMC algorithms is a current area of research. Del Moral et al. (2006a) provide a LLN and a CLT for their SMC sampler under some simplying assumptions. For the PMC algorithm, Douc et al. (2007a) consider the conditions under which adapting the conditional importance distribution using the past particles will and will not improve the algorithm.

4.3. Using SMC Algorithms as Proposal Distributions Within MCMC Algorithms

The discussion in Sections 4.1–4.2 assumed that SMC algorithms are used as an alternative to MCMC. Another possibility is to use an SMC algorithm as a proposal distribution within a MCMC algorithm. This computational method has been used within the macroeconomics literature for the Bayesian analysis of second-order approximations to DSGE models; see, e.g., Fernández-Villaverde and Rubio-Ramírez (2007) and An and Schorfheide (2007). In these papers, a particle filter is used to approximate the likelihood function of the nonlinear DSGE model. The log-likelihood approximation is then used within a standard random-walk Metropolis algorithm.

Recently, Andrieu et al. (2010) have given a formal proof for the convergence of the algorithm. These authors prove that as long as the estimate of the likelihood function is unbiased then the estimation error produced by the approximation does not change the equilibrium distribution of the Markov chain being simulated. These authors label their algorithms Particle Markov chain Monte Carlo (PMCMC). Andrieu et al. (2010) call the PMCMC algorithm currently being used in the macroeconomics literature a *particle marginal Metropolis–Hastings sampler*. In addition to providing convergence results for this algorithm, they also establish the results for a particle Gibbs sampler and a particle independent Metropolis–Hastings algorithm. They note that the particle Gibbs sampler should not be treated like a standard Gibbs sampler. Additional care needs to be used when implementing an MCMC algorithm that uses a particle filter within it and has steps other than random-walk Metropolis. Flury and Shephard (2008) apply the methodology to several simple economic models to demonstrate its applicability.

For practictioners, it is important to use a resampling algorithm that is known to be unbiased (e.g., multinomial, residual, or stratified resampling). This is one of the assumptions needed to apply the convergence results established by Andrieu et al. (2010). In addition, the performance of the PMCMC algorithm will depend on the quality of the approximation of the normalizing constant (i.e., the log-likelihood). If the variance of the approximation is high, the performance may deteriorate. Finally, it is possible to use the more advanced SMC algorithms such as an SMC sampler from Section 4.1 within the PMCMC algorithm.

5. SUMMARY

This article surveyed SMC methods that are applicable for economics and finance. From either a frequentist or Bayesian perspective, particle filters enable researchers to perform prediction and filtering in nonlinear,

non-Gaussian state space models. Particle filters and other SMC methods may play a larger role in risk management, option pricing, and high-frequency financial econometrics. Following recent trends in macroeconomics, particle filters are appearing more frequently to estimate structural models. Applied econometricians can use the particle filter in testing situations; i.e., to compute marginal likelihoods, likelihood-ratio statistics, or Ljung-Box statistics. Maximum likelihood estimation of nonlinear, non-Gaussian state space models using particle filters still remains an open research area. No single method has demonstrated an overwhelming computational or theoretical advantage for a reasonably large class of models. Work remains to be done on the statistical properties of the estimators as well.

SMC methods are likely to have a continued impact on Bayesian inference. SMC opens many new research avenues for estimating challenging models. These include trans-dimensional models, models that result in multimodal posteriors, and models with potentially a large number of parameters. The emphasis in this literature is currently on developing adaptive Monte Carlo algorithms that learn from previous draws. Understanding how the algorithms should be built in practice to make adaption work and its comparison with MCMC is part of this research. The limit theorems needed to justify their use is another. A second theme is the introduction of particle filters as proposal distributions within MCMC algorithms.

ACKNOWLEDGMENTS

I would like to thank Charles Bos, Siem Jan Koopman, Michael Massmann, Herman van Dijk, Eric Zivot, participants at the Emerging Methods in Bayesian Econometrics Workshop at Erasmus Universiteit Rotterdam, and two anonymous referees for constructive comments. I would also like to acknowledge financial support from the Grover and Creta Ensley Fellowship, which funded part of this research while I was a graduate student at the University of Washington. All the computations reported in this article were carried out using the OxMetrix 6.0 programming environment of Doornik (2009). Ox and some Matlab code are available upon request from the author.

REFERENCES

Akashi, H., Kumamoto, H. (1977). Random sampling approach to state estimation in switching environment. *Automatica* 13:429–434.

Alspach, D. L., Sorenson, H. W. (1972). Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control* 17:439–448.

Amisano, G., Tristani, O. (2010). Euro area inflation persistence in an estimated nonlinear DSGE model. *Journal of Economic Dynamics and Control* 34:1837–1858.

An, S., Schorfheide, F. (2007). Bayesian analysis of DSGE models. *Econometric Reviews* 26:113–172.
Anderson, B. D. O., Moore, J. B. (1979). *Optimal Filtering*. New Jersey: Prentice-Hall.
Andrieu, C., Davy, M., Doucet, A. (2003). Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions. *IEEE Transactions on Signal Processing* 51:1762–1770.
Andrieu, C., De Freitas, J. F. G., Doucet, A. (1999). Sequential Markov chain Monte Carlo for Bayesian model selection. *Proc. Workshop Higher Order Statistics* 130–134.
Andrieu, C., Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society, Series B* 64:827–836.
Andrieu, C., Doucet, A., Holenstein, R. (2010). Particle Markov chain Monte Carlo methods (with discussion). *Journal of the Royal Statistical Society, Series B* 72:1–33.
Andrieu, C., Doucet, A., Singh, S. S., Tadić, V. B. (2004). Particle methods for change detection, system identification, and control. *Proceedings of the IEEE* 92:423–438.
Andrieu, C., Doucet, A., Tadić, V. B. (2005). On-line simulation-based algorithms for parameter estimation in general state-space models. *Proc. Conf. Dec. Control*. New York: Institute of Electrical and Electronics Engineers.
Bain, A., Crisan, D. (2008). *Fundamentals of Stochastic Filtering*. New York, NY: Springer.
Baum, L. E., Petrie, T. P. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics* 37:1554–1563.
Baum, L. E., Petrie, T. P., Soules, G., Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics* 41:164–171.
Berzuini, C., Best, N. G., Gilks, W. R., Larizza, C. (1997). Dynamic conditional independence models and Markov chain Monte Carlo methods. *Journal of the American Statistical Association* 92:1403–1412.
Bos, C. S., Shephard, N. (2006). Inference for adaptive time series models: stochastic volatility and conditionally Gaussian state space form. *Econometric Reviews* 25:219–244.
Briers, M., Doucet, A., Maskell, S. (2004). Smoothing algorithms for state space models. Technical Report, University of Cambridge, Department of Engineering.
Cappé, O., Godsill, S., Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *IEEE Proceedings in Signal Processing* 95:899–924.
Cappé, O., Guillin, A., Marin, J.-M., Robert, C. P. (2004). Population Monte Carlo. *Journal of Computational and Graphical Statistics* 13:907–929.
Cappé, O., Guillin, A., Marin, J.-M., Robert, C. P. (2008). Adaptive importance sampling in general mixture classes. *Statistics and Computing* 18:447–459.
Cappé, O., Moulines, E., Rydén, T. (2005). *Inference in Hidden Markov Models*. New York: Springer Press.
Carpenter, J., Clifford, P., Fearnhead, P. (1999). An improved particle filter for non-linear problems. *IEE Proceedings. Part F: Radar and Sonar Navigation* 146:2–7.
Carter, C. K., Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika* 81:541–553.
Carvalho, C. M., Johannes, M., Lopes, H. F., Polson, N. G. (2008). Particle learning and smoothing. Working Paper, Booth School of Business, University of Chicago.
Celeux, G., Marin, J.-M., Robert, C. P. (2006). Iterated importance sampling in missing data problems. *Computational Statistics and Data Analysis* 50:3386–3404.
Chen, R., Liu, J. S. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society, Series B* 62:493–508.
Chen, Y., Xie, J., Liu, J. S. (2005). Stopping-time resampling for sequential Monte Carlo methods. *Journal of the Royal Statistical Society, Series B* 67:199–217.
Chib, S., Nardari, F., Shephard, N. (2002). Markov chain Monte Carlo methods for stochastic volatility models. *Journal of Econometrics* 108:281–316.
Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika* 89:539–551.
Chopin, N. (2004). Central limit theorem for sequential Monte Carlo and its application to Bayesian inference. *The Annals of Statistics* 32:2385–2411.
Chopin, N. (2007). Inference and model choice for sequentially ordered hidden Markov models. *Journal of the Royal Statistical Society, Series B* 69:269–284.

Chopin, N., Pelgrin, F. (2004). Bayesian inference and state number determination for hidden Markov models: an application to the information content of the yield curve about inflation. *Journal of Econometrics* 123:327–344.

Cornebise, J., Moulines, E., Olsson, J. (2008). Adaptive methods for sequential importance sampling with application to state space models. Unpublished manuscript, Ecole Nationale Supérieure des Télécommunications, Paris, France.

Cornuet, J.-M., Marin, J.-M., Mira, A., Robert, C. P. (2009). Adaptive multiple importance sampling. Working paper, Universite Montpellier 2 and CREST, INSEE.

Crisan, D., Del Moral, P., Lyons, T. (1999). Discrete filtering using branching and interacting particle systems. *Markov Processes and Related Fields* 5:293–318.

Crisan, D., Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing* 50:736–746.

de Freitas, N., Dearden, R., Hutter, F., Morales-Menendez, R., Mutch, J., Poole, D. (2004). Diagnosis by a waiter and a Mars explorer. *Proceedings of the IEEE* 92:1–11.

de Jong, D., Dharmarajan, H., Liesenfeld, R., Moura, G. V., Richard, J. F. (2009). Efficient likelihood evaluation of state space representations. Unpublished manuscript, Department of Economics, University of Pittsburgh.

de Jong, P., Shephard, N. (1995). The simulation smoother for time series models. *Biometrika* 82:339–350.

Del Moral, P. (1996). Nonlinear filtering: interacting particle solution. *Markov Processes and Related Fields* 2:555–579.

Del Moral, P. (2004). *Feyman–Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. New York, NY: Springer.

Del Moral, P., Doucet, A. (2009). Particle methods: an introduction with applications. Unpublished manuscript, Insitut National de Recherche en Informatique et en Automatique, Bordeaux, France.

Del Moral, P., Doucet, A., Jasra, A. (2006a). *Sequential Monte Carlo for Bayesian Computation*. Bayesian Statistics 8. Oxford, UK: Oxford University Press.

Del Moral, P., Doucet, A., Jasra, A. (2006b). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B* 68:1–26.

Del Moral, P., Doucet, A., Jasra, A. (2008). An adaptive sequential Monte Carlo method for approximate Bayesian computation. Working paper, Department of Statistics, University of British Columbia.

Del Moral, P., Guionnet, A. (1999). Central limit theorem for nonlinearing filtering and interacting particle systems. *The Annals of Applied Probability* 9:275–297.

Del Moral, P., Guionnet, A. (2001). On the stability of interacting processes with applications to filtering and genetic algorithms. *Ann. Inst. H. Poincaré Probab. Statist.* 37:155–194.

Del Moral, P., Miclo, L. (2000). Branching and interacting particle systems approximations of Feynman–Kac formulae with applications to non-linear filtering. *Séminaire de Probabilités XXXVII* Lecture Notes in Mathematics 1729. Berlin: Springer, pp. 1–145.

Doornik, J. A. (2009). *An Object-Oriented Matrix Language Ox 6*. London: Timberlake Consultants Press.

Douc, R., Cappé, O., Moulines, E. (2005). Comparison of resampling schemes for particle filtering. *4th International Symposium on Image and Signal Processing and Analysis (ISPA)*. Zagreb, Croatia.

Douc, R., Fort, G., Moulines, E., Priouret, P. (2009). Forgetting of the initial distribution in hidden markov models. *Stochastic Processes and Their Applications* 119:1235–1256.

Douc, R., Garivier, A., Moulines, E., Olsson, J. (2009). On the forward filtering backward smoothing particle approximations of the smoothing distribution in general state space models. Unpublished manuscript, Institut Télécom, Paris, France.

Douc, R., Guillin, A., Marin, J.-M., Robert, C. P. (2007a). Convergence of adaptive sampling schemes. *The Annals of Statistics* 35:1–35.

Douc, R., Guillin, A., Marin, J.-M., Robert, C. P. (2007b). Minimum variance importance sampling via population Monte Carlo. *ESAIM: Probability and Statistics* 11:420–448.

Douc, R., Moulines, E. (2008). Limit theorems for weighted samples with applications to sequential Monte Carlo methods. *The Annals of Statistics* 36:2344–2376.

Douc, R., Moulines, E., Olsson, J. (2009c). Optimality of the auxiliary particle filter. *Probability and Mathematical Statistics* 29:1–28.

Doucet, A., Briers, M., Sénécal, S. (2006). Efficient block sampling strategies for sequential Monte Carlo. *Journal of Computational and Graphical Statistics* 15:693–711.

Doucet, A., de Freitas, N., Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer-Verlag.

Doucet, A., Godsill, S. J., Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing* 10:197–208.

Doucet, A., Gordon, N. J., Krishnamurthy, V. (2001). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing* 49:613–624.

Doucet, A., Johansen, A. (2009). A tutorial on particle filtering and smoothing: fifteen years later. In: *The Oxford Handbook of Nonlinear Filtering*. Oxford, UK: Oxford University Press.

Doucet, A., Tadić, V. (2003). Parameter estimation in general state space models using particle methods. *Annals of the Institute of Statistical Mathematics* 55:409–422.

Durbin, J., Koopman, S. J. (2001). *Time Series Analysis by State Space Methods*. Oxford, UK: Oxford University Press.

Durbin, J., Koopman, S. J. (2002). A simple and efficient simulation smoother for state space time series analysis. *Biometrika* 89:603–616.

Fearnhead, P. (2002). MCMC, sufficient statistics, and particle filters. *Journal of Computational and Graphical Statistics* 11:848–862.

Fearnhead, P., Clifford, P. (2003). On-line inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society, Series B* 65:887–899.

Fearnhead, P., Wyncoll, D., Tawn, J. (2008). A sequential smoothing algorithm with linear computational cost. Working Paper. Department of Statistics, University of Lancaster.

Fernández-Villaverde, J., Rubio-Ramírez, J. F. (2005). Estimating dynamic equilibrium economies: linear versus nonlinear likelihood. *Journal of Applied Econometrics* 20:891–910.

Fernández-Villaverde, J., Rubio-Ramírez, J. F. (2007). Estimating macroeconomic models: a likelihood approach. *The Review of Economic Studies* 74:1059–1087.

Flury, T., Shephard, N. (2008). Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models. Unpublished Paper, Department of Economics, University of Oxford.

Fraser, D., Potter, J. (1969). The optimum linear smoother as a combination of two optimum linear filters. *IEEE Transactions on Automatic Control* 4:387–390.

Frühwirth-Schnatter, S. (1994). Data augmentation and dynamic linear models. *Journal of Time Series Analysis* 15:183–202.

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models*. New York, NY: Springer Press.

Geweke, J. F. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* 57:1317–1139.

Geweke, J. F. (2005). *Contemporary Bayesian Econometrics and Statistics*. New Jersey: Wiley Press.

Gilks, W., Berzuini, C. (2001). Following a moving target – Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society, Series B* 63:126–146.

Godsill, S. J., Doucet, A., West, M. (2002). Maximum a posteriori sequence estimation using Monte Carlo particle filters. *Annals of the Institute of Statistical Mathematics* 53:82–96.

Godsill, S. J., Doucet, A., West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association* 99:156–168.

Gordon, N. J., Salmond, D. J., Smith, A. F. M. (1993). A novel approach to nonlinear and non-Gaussian Bayesian state estimation. *IEE Proceedings. Part F: Radar and Sonar Navigation* 140:107–113.

Guo, D., Wang, X., Chen, R. (2005). New sequential Monte Carlo methods for nonlinear dynamic systems. *Statistics and Computing* 15:135–147.

Hamilton, J. D. (1989). A new approach to the econometric analysis of nonstationary time series and the business cycle. *Econometrica* 57:357–384.

Hammersley, J. M., Morton, K. W. (1954). Poor man's Monte Carlo. *Journal of the Royal Statistical Society, Series B* 16:23–38.

Handschin, J. (1970). Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica* 6:555–563.

Handschin, J., Mayne, D. Q. (1969). Monte Carlo techniques to estimate the conditional expectations in multi-stage non-linear filtering. *International Journal of Control* 9:547–559.

Harvey, A. C. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge, UK: Cambridge University Press.

Hendry, D. F., Richard, J. F. (1991). Likelihood evaluation for dynamic latent variable models. In: Amman, D. A. B., Pau, L. F., eds. *Computational Economics and Econometrics*. Amsterdam: Kluwer, pp. 3–17.

Hoogerheide, L. F., van Dijk, H. K., van Oest, R. D. (2007). On the shape of posterior distributions and credible sets in instrumental variables regression models with reduced rank: an application of flexible sampling methods using neural networks. *Journal of Econometrics* 139:154–180.

Hürzeler, M., Künsch, H. R. (1998). Monte Carlo approximations for general state space models. *Journal of Computational and Graphical Statistics* 7:175–193.

Hürzeler, M., Künsch, H. R. (2001). Approximating and maximising the likelihood for a general state space model. In: Doucet, A., De Freitas, N., Gordon, N., eds. *Sequential Monte Carlo Methods in Practice*. New York: Springer, pp. 159–175.

Isard, M., Blake, A. (1996). Contour tracking by stochastic propagation of conditional density. *Proceedings of the European Conference on Computer Vision* Cambridge, pp. 343–356.

Jarzynski, C. (1997). Nonequilibrium equality for free energy differences. *Physical Review Letters* 78:2690–2693.

Jasra, A., Doucet, A., Stephens, D., Holmes, C. (2008a). Interacting sequential Monte Carlo samplers for trans-dimensional simulation. *Computational Statistics and Data Analysis* 52:1765–1791.

Jasra, A., Stephens, D. A., Doucet, A., Tsagaris, T. (2008b). Inference for Lévy-driven stochastic volatility models via adaptive sequential Monte Carlo. Unpublished manuscript, Institute for Statistical Mathematics, Tokyo, Japan.

Johannes, M., Korteweg, A., Polson, N. G. (2008). Sequential learning, predictive regressions, and optimal portfolio returns. Working paper, Graduate School of Business, Stanford University.

Johannes, M., Polson, N. G., Stroud, J. R. (2006). Interacting particle systems for sequential parameter learning. Working paper, Graduate School of Business, Columbia University.

Johansen, A. M., Del Moral, P., Doucet, A. (2006). Sequential Monte Carlo samplers for rare events. *Proc. 6th International Workshop on Rare Event Simulation*.

Johansen, A. M., Doucet, A. (2008). A note on auxiliary particle filters. *Statistics and Probability Letters* (forthcoming).

Julier, S. J., Uhlmann, J. K. (1997). A new method for nonlinear transformation of means and covariances in filters and estimators. *The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II*. SPIE.

Julier, S. J., Uhlmann, J. K., Durrant-White, H. F. (2000). A new method for nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control* 45:477–482.

Kahn, H., Marshall, A. W. (1953). Methods of reducing sample size in Monte Carlo computations. *Journal of the Operational Research Society of America* 1:263–271.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering, Transactions of the ASME* 82:35–46.

Kalman, R., Bucy, R. (1961). New results in linear filtering and prediction theory. *Journal of Basic Engineering, Transactions of the ASME* 83:95–108.

Kim, C.-J., Nelson, C. R. (1999). *State Space Models with Regime Switching Classical and Gibbs Sampling Approaches with Applications*. Cambridge, MA: MIT Press.

Kim, S., Shephard, N., Chib, S. (1998). Stochastic volatility: likelihood inference and comparison with ARCH models. *The Review of Economic Studies* 65:361–393.

Kitagawa, G. (1987). Non-Gaussian state space modeling of nonstationary time series. *Journal of the American Statistical Association* 82:1023–1063.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics* 5:1–25.

Kitagawa, G. (1998). A self-organizing state-space model. *Journal of the American Statistical Association* 93:1203–1215.

Kloek, T., van Dijk, H. K. (1978). Bayesian estimations of equation system parameters: an application of Monte Carlo integration. *Econometrica* 46:1–20.

Kong, A., Liu, J. S., Wong, W. (1994). Sequential imputation and Bayesian missing data problems. *Journal of the American Statistical Association* 89:278–288.

Koopman, S. J., Shephard, N., Creal, D. (2009). Testing the assumptions behind importance sampling. *Journal of Econometrics* 149:2–11.

Künsch, H. R. (2001). State space and hidden Markov models. In: Barndorff-Nielson, O. E., Cox, D. R., Klüppelburg, C., eds. *Complex Stochastic Systems*. Chapman Hall CRC, London: pp. 109–173.

Künsch, H. R. (2005). Recursive Monte Carlo filters: algorithms and theoretical analysis. *The Annals of Statistics* 33:1983–2021.

Künsch, H. R. (2006). Filtering, smoothing and parameter estimation for general state space models. Lecture notes, Troisieme Cycle Romand, Course at EPFL, http://stat.ethz.ch/~kuensch.

Le Gland, F., Oudjane, N. (2004). Stability and uniform approximation of nonlinear filters using the Hilbert metric, and application to particle filters. *The Annals of Applied Probability* 14:144–187.

Liang, F. (2002). Dynamically weighted importance sampling in Monte Carlo computation. *Journal of the American Statistical Association* 93:807–821.

Liu, J., West, M. (2001). Combined parameter and state estimation in simulation-based filtering. In: Doucet, A., De Freitas, N., Gordon, N., eds. *Sequential Monte Carlo Methods in Practice*. New York: Springer, pp. 197–223.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. New York, NY: Springer Press.

Liu, J. S., Chen, R. (1995). Blind deconvolutions via sequential imputations. *Journal of the American Statistical Association* 90:567–576.

Liu, J. S., Chen, R. (1998). Sequential Monte Carlo computation for dynamic systems. *Journal of the American Statistical Association* 93:1032–1044.

Marshall, A. W. (1956). The use of multi-stage sampling schemes in Monte Carlo computations. In: Meyer, M. ed. *Symposium on Monte Carlo Methods*. New York: Wiley, pp. 123–140.

Mayne, D. Q. (1966). A solution of the smoothing problem for linear dynamic systems. *Automatica* 4:73–92.

Neal, R. (2001). Annealed importance sampling. *Statistics and Computing* 11:125–139.

Oh, M. S., Berger, J. O. (1993). Integration of multimodal functions by Monte Carlo importance sampling. *Journal of the American Statistical Association* 88:450–456.

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

Olsson, J., Rydén, T. (2008). Asymptotic properties of particle filter-based maximum likelihood estimators for state space models. *Stochastic Processes and their Applications* 118:649–680.

Pitt, M. K. (2002). Smooth particle filters for likelihood evaluation and maximisation. Unpublished manuscript, Department of Economics, Warwick University.

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

Pitt, M. K., Shephard, N. (2001). Auxiliary variable based particle filters. In: Doucet, A., De Freitas, N., Gordon, N., eds. *Sequential Monte Carlo Methods in Practice*. New York: Springer, pp. 273–292.

Poyiadjis, G., Doucet, A., Singh, S. S. (2005a). Maximum likelihood parameter estimation in general state space models using particle methods. Unpublished manuscript, Department of Engineering, Cambridge University.

Poyiadjis, G., Doucet, A., Singh, S. S. (2005b). Particle methods for optimal filter derivative: application to parameter estimation. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 925–928.

Rauch, H., Tung, F., Striebel, C. (1965). Maximum likelihood estimation of linear dynamic systems. *AIAA Journal* 3:1445–1450.

Richard, J. F., Zhang, W. (2007). Efficient high-dimensional importance sampling. *Journal of Econometrics* 141:1385–1411.

Robert, C. P., Casella, G. (2004). *Monte Carlo Statistical Methods*. 2nd ed. New York: Springer Press.

Rubin, D. B. (1987). A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when the fraction of missing information is modust: the SIR algorithm (discussion of Tanner and Wong). *Journal of the American Statistical Association* 82:543–546.

Rubin, D. B. (1988). Using the SIR algorithm to simulate posterior distributions. In: Bernardo, D. V. L. J. M., DeGroot, M. H., Smith, A. F. M., eds. *Bayesian Statistics 3*. New York: Oxford University Press, pp. 395–402.

Schön, T., Gustafsson, F., Nordlund, P.-J. (2005). Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing* 53:2279–2289.

Shephard, N. (2005). *Stochastic Volatility: Selected Readings*. Oxford, UK: Oxford University Press.

Shephard, N., Pitt, M. K. (1997). Likelihood analysis of non-Gaussian measurement time series. *Biometrika* 84:653–667.

Smith, J. Q., Santos, A. A. F. (2006). Second-order filter distribution approximations for financial time series with extreme outliers. *Journal of Business and Economic Statistics* 24:329–337.

Stock, J. H., Watson, M. W. (2007). Why has U.S. inflation become harder to forecast? *Journal of Money, Credit, and Banking* 39:3–34.

Storvik, G. (2002). Particle filters in state space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing* 50:281–289.

Tanizaki, H., Mariano, R. S. (1994). Prediction, filtering and smoothing in non-linear and non-normal cases using Monte Carlo integration. *Journal of Applied Econometrics* 9:163–179.

Tanizaki, H., Mariano, R. S. (1998). Nonlinear and non-Gaussian state-space modeling with Monte-Carlo simulations. *Journal of Econometrics* 83:263–290.

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *The Annals of Statistics* 22:1701–1762.

van der Merwe, R., Doucet, A., de Freitas, N. (2000). The unscented particle filter. In: Leen, T. G. D. T. K., Tresp, V., eds. *Advances in Neural Information Processing Systems*. Cambridge: MIT Press, p. 13.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions in Information Theory* 13:260–269.

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing* 4:65–85.

Xiong, J. (2008). *An Introduction to Stochastic Filtering Theory*. Oxford, UK: Oxford University Press.

Zaritskii, V., Svetnik, V., Shimelevich, L. (1975). Monte Carlo techniques in problems of optimal data processing. *Automation and Remote Control* 12:95–103.