Monte Carlo Inference for State–Space Models of Wild Animal Populations

Ken B. Newman,[1,*] **Carmen Fernández,**[2] **Len Thomas,**[1] **and Stephen T. Buckland**[1]

[1]School of Mathematics and Statistics, University of St. Andrews, The Observatory,
Buchanan Gardens, St. Andrews KY16 9LZ, Scotland
[2]Instituto Español de Oceanografía, Cabo Estai - Canido, Apdo. 1552, 36200 Vigo, Spain
[*]*Current address:* US Fish and Wildlife Service, Stockton FWO, 4001 North Wilson Way,
Stockton, California 95205, U.S.A.
[*]*email:* ken_newman@fws.gov

Summary. We compare two Monte Carlo (MC) procedures, sequential importance sampling (SIS) and Markov chain Monte Carlo (MCMC), for making Bayesian inferences about the unknown states and parameters of state–space models for animal populations. The procedures were applied to both simulated and real pup count data for the British grey seal metapopulation, as well as to simulated data for a Chinook salmon population. The MCMC implementation was based on tailor-made proposal distributions combined with analytical integration of some of the states and parameters. SIS was implemented in a more generic fashion. For the same computing time MCMC tended to yield posterior distributions with less MC variation across different runs of the algorithm than the SIS implementation with the exception in the seal model of some states and one of the parameters that mixed quite slowly. The efficiency of the SIS sampler greatly increased by analytically integrating out unknown parameters in the observation model. We consider that a careful implementation of MCMC for cases where data are informative relative to the priors sets the gold standard, but that SIS samplers are a viable alternative that can be programmed more quickly. Our SIS implementation is particularly competitive in situations where the data are relatively uninformative; in other cases, SIS may require substantially more computer power than an efficient implementation of MCMC to achieve the same level of MC error.

KEY WORDS: Auxiliary particle filter; British grey seals; Chinook salmon; Markov chain Monte Carlo; Parameter kernel smoothing; Rejection control; Sequential importance sampling.

1. Introduction

State–space models (SSMs) have been proposed as a unifying framework for modeling animal population dynamics and corresponding observations made on the population (Schnute, 1994; Buckland et al., 2004; Thomas et al., 2005). However, realistic SSMs for animal populations are often complex, nonlinear, and non-Gaussian and can be difficult to fit to real data. Fitting procedures are simulation based and often require writing software specific to the problem at hand (e.g., Trenkel, Elston, and Buckland, 2000; Schnute and Kronlund, 2002; Wikle, 2003; Clark and Bjørnstad, 2004; Ionides, Breto, and King, 2006). For SSMs for animal populations to gain wider usage, general purpose software for fitting a variety of SSMs is needed. Our primary aim was to determine whether Markov chain Monte Carlo (MCMC; Gilks, Richardson, and Spiegelhalter, 1996) or sequential importance sampling (SIS; Liu, 2001) would be a more promising inference engine for such software. In this article, we compare in a pragmatic way, and in the context of real animal populations, the use

of MCMC and SIS to make Bayesian inferences for SSMs of animal population dynamics.

MCMC has been the more commonly used procedure for fitting such SSMs. Meyer and Millar (1999) have used `WinBUGS` (Lunn et al., 2000), which is based on MCMC, to fit a relatively simple SSM for albacore biomass, while others (Rivot et al., 2004; Michielsens et al., 2006) have used it for more complex population SSMs. Due to the often high degree of dependence between state variables and parameters in an SSM, however, we have sometimes found `WinBUGS` either very slow to converge or very sensitive to the choice of initial values with slight changes causing the program to fail. To speed or ensure convergence, MCMC software tailored to the SSM is often required, with careful choices needed for proposal distributions, parameterization, and updating schemes.

Conversely, SIS has been seldom used to fit SSMs for population dynamics (e.g., Trenkel et al., 2000; Thomas et al., 2005; Newman et al., 2006). It is an alternative that biometricians may want to consider because in its most basic form

SIS is a more automatic procedure in that the state equations can be written down with little concern about the difficulty of evaluating the probability functions, in contrast to most MCMC implementations, which require such evaluations. SIS proceeds through a times series forward in time and, hence, it is a natural tool to consider for inference in population dynamics models. It also handles very easily the incorporation of new time points at the end of the series. On the other hand, Monte Carlo variation of SIS algorithms can be quite high, even for quite large samples. The root of the problem is particle depletion: a single simulated realization of the SSM's states and parameters is called a particle, and particle depletion is the phenomenon whereby the number of different particles with nonnegligible weights gets successively smaller as the simulation of the time series proceeds forward in time. Even when the number of simulated particles is very large (in the millions), the effective number of different particles can be very small (in the tens). This is especially problematic for static (as opposed to time varying) parameters.

Comparisons of MCMC and (nonsequential) importance sampling have been made from a theoretical perspective; e.g., Bassetti and Diaconis (2006) used eigen analysis to show that the Metropolis algorithm is comparable to or better than the importance sampler, in terms of precision, sometimes by an exponential amount, for the examples they tried. While relative Monte Carlo (MC) variation (across different runs of the algorithm) of the two procedures is important, so are implementation ease and computational efficiency. In particular, we wish to compare the degree of difficulty in setting up reasonably efficient procedures for fitting relatively complex animal population SSMs.

Although the article focuses on comparison of computational methods, the problem chosen for application is a real-life animal population with complex dynamics and substantive management issues, the British grey seal (*Halichoerus grypus*) metapopulation. The remainder of the article is organized as follows. Section 2 briefly describes SSMs and the general use of SIS and MCMC to fit SSMs. Section 3 defines the SSM for seals, describes the particular implementations of SIS and MCMC, and presents the results. Section 4 compares and contrasts our experience using SIS and MCMC to fit animal population SSMs. Web Appendix A contains an additional example using simulated data for a Chinook salmon (*Oncorhynchus tshawytscha*) population, whereas Web Appendix B provides additional comparisons of SIS and MCMC for simulated seal data.

2. SSMs for Animal Populations and Monte Carlo Inference

SSMs are models for two time series running in parallel: one, labeled the state process with value $\mathbf{n}_t$ at time t, that is unobserved, and a second, called the measurement or observation process with value $\mathbf{y}_t$ at time t, that is observed and is a function of the state process (Harvey, 1989). A realization of the state process from time a to time b is denoted $\mathbf{n}_{a:b} = (\mathbf{n}_a, \ldots, \mathbf{n}_b)$, with similar notation for the observation process.

For animal population dynamics SSMs, the state process often describes the true, but unobservable, population demographics as they change over time. Excepting models for population biomass (e.g., Meyer and Millar, 1999), the components

of $\mathbf{n}_t$ are often counts of subpopulations, for example, age 1 females. The observation process is a model for samples from or estimates of the population, and $\mathbf{y}_t$ has components that correspond to state components, or aggregated sets of them, for example, an estimate of all females.

The probabilistic structure for SSMs can be written as a set of three probability density or mass functions (for conciseness we refer to both as pdfs), as follows:

$$g_0(\mathbf{n}_0 \,|\, \eta) \quad \text{Initial state pdf,} \tag{1}$$

$$g_t(\mathbf{n}_t \,|\, \mathbf{n}_{t-1}, \eta) \quad \text{State process pdf,} \tag{2}$$

$$f_t(\mathbf{y}_t \,|\, \mathbf{n}_t, \psi) \quad \text{Observation process pdf,} \tag{3}$$

where $t = 1, 2, \ldots, T$ and η and ψ are vectors of parameters. We will use θ to denote (η, ψ).

In a Bayesian context, the inference objectives for SSMs include generating a sample from the posterior distribution for the states and unknown parameters conditional on the entire observation time series, $\mathbf{y}_{1:T}$. Denoting by $\pi(\theta)$ the prior pdf of the parameters, the joint posterior pdf can be written as

$$\pi(\mathbf{n}_{0:T}, \theta \,|\, \mathbf{y}_{1:T}) \propto \pi(\theta) g_0(\mathbf{n}_0 \,|\, \eta)$$
$$\times \left\{ \prod_{t=1}^{T} g_t(\mathbf{n}_t \,|\, \mathbf{n}_{t-1}, \eta) f_t(\mathbf{y}_t \,|\, \mathbf{n}_t, \psi) \right\}. \tag{4}$$

Direct evaluation of $g_t(\mathbf{n}_t \,|\, \mathbf{n}_{t-1}, \eta)$ can be difficult. The evolution from $\mathbf{n}_{t-1}$ to $\mathbf{n}_t$ is often the result of a sequence of stochastic and deterministic subprocesses. For example, suppose that three stochastic subprocesses take place in the following order: survival, movement between different areas and, finally, harvest. An intuitive way of modeling the subprocesses is to specify separate pdfs for each of them; e.g., $g_{1t}(\,)$ for survival, $g_{2t}(\,)$ for movement, and $g_{3t}(\,)$ for harvest. Consequently $g_t(\mathbf{n}_t \,|\, \mathbf{n}_{t-1}, \eta)$ is difficult to evaluate directly because the output from one process becomes input to the next process and $\mathbf{n}_t$ is the result of composite functions and convolutions. Computational procedures can be made easier by augmenting the state process to include intermediate vectors, say $\mathbf{u}_{1t}, \mathbf{u}_{2t}, \mathbf{u}_{3t}$, where $\mathbf{u}_{3t} = \mathbf{n}_t$, which describe the state vector immediately after each subprocess, and the corresponding transitional pdfs, g_{1t}, g_{2t}, g_{3t}, may be simple to evaluate.

2.1 *Sequential Importance Sampling*

Here we briefly describe SIS in general, beginning with a review of importance sampling. Let $\pi(x)$ denote the target pdf from which a sample is sought. Ordinary (nonsequential) importance sampling begins by generating a sample of size N from $h(x)$, the importance sampler, whose support includes the support of $\pi(x)$. Denoting each generated value by $x^{(i)}, i = 1, \ldots, N$, weights per value are calculated by $w^{(i)} = \frac{\pi(x^{(i)})}{h(x^{(i)})}$. Expected values of arbitrary functions, $m(X)$, with respect to the target pdf $\pi(x)$, are estimated by

$$\hat{E}\{m(X)\} = \frac{\sum_{i=1}^{N} m(x^{(i)}) w^{(i)}}{\sum_{i=1}^{N} w^{(i)}}. \tag{5}$$

The degree of MC variation between samples depends on how closely $h(x)$ matches $\pi(x)$. An unweighted sample from $\pi(x)$ is generated by resampling (with replacement) the $x^{(i)}$ values with weights proportional to $w^{(i)}$, where the $x^{(i)}$ with higher weights should appear more frequently.

For SSMs the target distribution is $\pi(\mathbf{n}_{0:T} \,|\, \mathbf{y}_{1:T})$ (ignoring θ here), which is typically of high dimension. Directly specifying an efficient high-dimensional importance sampler, $h(\mathbf{n}_{0:T})$, is difficult. The strategy behind SIS is one of divide and conquer: construct the importance sampler and weights incrementally, one time period at a time. Start with a sample of the initial state $\mathbf{n}_0$ with weights $w_0^{(i)}$, $i = 1, \ldots, N$. Then, sequentially for each time period $t = 1, \ldots, T$: (a) Generate a sample of size N for the state $\mathbf{n}_t$ using an importance sampler $h_t(\mathbf{n}_t)$, where h_t could depend on $\mathbf{n}_{0:t-1}$ and $\mathbf{y}_{1:t}$; (b) Calculate new weights $w_t^{(i)} = w_{t-1}^{(i)} \frac{g_t(\mathbf{n}_t^{(i)} \,|\, \mathbf{n}_{t-1}^{(i)}) f_t(\mathbf{y}_t \,|\, \mathbf{n}_t^{(i)})}{h_t(\mathbf{n}_t^{(i)})}$. Resampling at some intermediate time points is generally done, thereby "pruning" away values with relatively low weights and increasing the number of values with higher weights. The "resampling schedule" can be critical to the quality of the SIS performance. If the state pdf $g_t(\mathbf{n}_t \,|\, \mathbf{n}_{t-1})$ is used for $h_t(\mathbf{n}_t)$, the weights are updated simply by the likelihood, $f_t(\mathbf{y}_t \,|\, \mathbf{n}_t^{(i)})$, thereby avoiding the need to evaluate the state pdf. This will prove useful in certain cases with multiple subprocesses where the state pdf cannot be evaluated but it is possible to sample from it (as illustrated later in the seal application). Each generated sequence, $\mathbf{n}_{0:T}^{(i)}$, is sometimes referred to as a particle. SIS is an example of a particle filter in that it provides for each time point, $t = 1, 2, \ldots, T$, a sample from $\pi(\mathbf{n}_{0:T} \,|\, \mathbf{y}_{1:T})$ (Harvey, 1989).

2.2 *MCMC*

MCMC is also a divide and conquer strategy to generating a sample from $\pi(\mathbf{n}_{0:T} \,|\, \mathbf{y}_{1:T})$ (ignoring θ momentarily). In contrast to SIS, which generates N independent samples of state $\mathbf{n}_t$ in one iteration, MCMC generates one sample of $\mathbf{n}_{0:T}$ in each of $B + N$ sequential iterations. The first B iterations, the burn-in, are discarded, and the last N iterations are a dependent sample from $\pi(\mathbf{n}_{0:T} \,|\, \mathbf{y}_{1:T})$. Additional general background on MCMC methods and applications can be found in Gilks et al. (1996).

Here we briefly mention aspects of the most widely used MCMC algorithm, Metropolis–Hastings, in the context of an SSM. Let the superscript $i - 1$ denote values at the end of iteration $i - 1$. At iteration i, the state vectors $\mathbf{n}_0^{(i-1)}$, $\mathbf{n}_1^{(i-1)}, \ldots, \mathbf{n}_T^{(i-1)}$ are updated sequentially (divide and conquer strategy), updating $\mathbf{n}_t$ conditional on the other time periods, as follows. A candidate value $\mathbf{n}_t^{(c)}$ is generated from a proposal distribution $h_t(\mathbf{n}_t \,|\, \mathbf{n}_t^{(i-1)})$, which can depend on $\mathbf{n}_t^{(i-1)}$, the current values of the other states, and $\mathbf{y}_{1:T}$. The new value is $\mathbf{n}_t^{(i)} = \mathbf{n}_t^{(c)}$ with probability

$$MH\left(\mathbf{n}_t^{(c)}, \mathbf{n}_t^{(i-1)}\right)$$

$$= \min\left\{1, \frac{g_t\left(\mathbf{n}_t^{(c)} \,\big|\, \mathbf{n}_{t-1}^{(i)}\right) f_t\left(\mathbf{y}_t \,\big|\, \mathbf{n}_t^{(c)}\right) g_{t+1}\left(\mathbf{n}_{t+1}^{(i-1)} \,\big|\, \mathbf{n}_t^{(c)}\right)}{g_t\left(\mathbf{n}_t^{(i-1)} \,\big|\, \mathbf{n}_{t-1}^{(i)}\right) f_t\left(\mathbf{y}_t \,\big|\, \mathbf{n}_t^{(i-1)}\right) g_{t+1}\left(\mathbf{n}_{t+1}^{(i-1)} \,\big|\, \mathbf{n}_t^{(i-1)}\right)} \right.$$

$$\left. \times \frac{h_t\left(\mathbf{n}_t^{(i-1)} \,\big|\, \mathbf{n}_t^{(c)}\right)}{h_t\left(\mathbf{n}_t^{(c)} \,\big|\, \mathbf{n}_t^{(i-1)}\right)} \right\}, \tag{6}$$

whereas with the remaining probability, $\mathbf{n}_t^{(i)} = \mathbf{n}_t^{(i-1)}$. Note that evaluation of the acceptance probability for states requires evaluation of state pdfs corresponding to two consecutive time periods, t and $t + 1$. Evaluation of the state pdf can be impossible in certain cases with multiple subprocesses, in which case a common solution is the inclusion of latent states corresponding to intermediate subprocesses whose pdfs can be evaluated (as will be shown in the seal application).

Figure 1 compares SIS and MCMC graphically. Whereas SIS proceeds from left to right, on a column basis, MCMC moves from top to bottom, filling in one row at each iteration.

3. Modeling the British Grey Seal Metapopulation

British grey seals are an iconic conservation species in the United Kingdom. They were hunted nearly to extinction and became the first mammal to be protected by modern legislation in 1914. Since then their numbers have increased substantially and there are now conflicts between fishermen and conservationists over seal management. Scientific advice, based in part on the models and methods in this article, is provided to the Scottish and U.K. governments via a committee of scientists that meets annually.

Grey seals breed colonially, mostly on the offshore islands around Scotland. Because grey seals spend most of their time under water, they are very difficult to survey, with only data from aerial surveys of pups available. Pup numbers overall are estimated to have grown at about 6% per year since surveys began in the early 1960s, although the trajectory varies regionally. Management concerns include the effect of predation on fish populations (Buckland et al., 2007) and scientific questions include how local densities may affect pup survival and between-colony migration (Thomas et al., 2005).

The data available consisted of estimated numbers of pups born, based on 19 years of aerial surveys of seal breeding colonies (1984–2002). The numbers were aggregated into four geographically distinct regions: 1 (North Sea, 4 colonies), 2 (Inner Hebrides, 19 colonies), 3 (Outer Hebrides, 11 colonies), and 4 (Orkneys, 22 colonies). Observations for the first year (1984), denoted $t = 0$, were used as a basis for specifying the initial state pdf and are no longer considered as part of the observation vector. Relative to the available data, estimated numbers of pups born on each colony, the underlying population dynamics model (the state process defined below) is admittedly overparameterized, a reflection of the complexity of the dynamics as well as the questions being asked. Consequently, the influence of prior distributions for some parameters was expected to be sizeable and expert knowledge was used to specify several priors.

A simulation study was also conducted to evaluate bias and MC variation in posterior samples generated by MCMC and SIS. Mimicking the real data, we simulated five datasets, each with 19 years of data. The parameters chosen were equal to the prior means and the state and observation vectors were simulated according to the processes described below.

3.1 *SSM Formulation*

State process. Per region, there are seven components in the state vector: the number of pups (age 0 males and females), the number of age a females, $a = 1, \ldots, 5, 6+$. We denote by
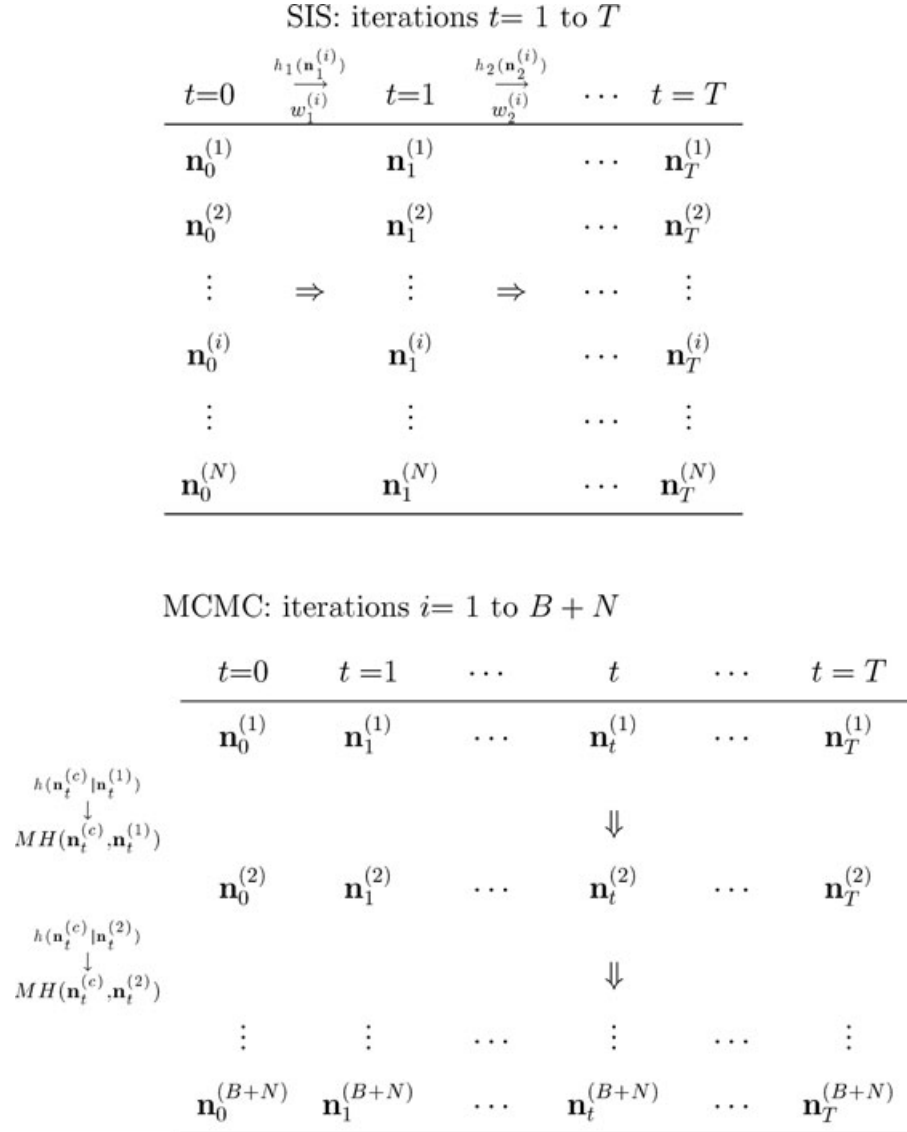
SIS: iterations $t=1$ to T

MCMC: iterations $i=1$ to $B+N$

Figure 1. Graphical comparison of SIS and MCMC algorithms for simulating a sample of size N from $\pi(\mathbf{n}_{0:T} \mid \mathbf{y}_{1:T})$. The superscripts indicate sample number and the subscripts denote the point in time; e.g., $\mathbf{n}_t^{(i)}$ is the ith sample of the state at time t. Whereas SIS proceeds from left to right, MCMC goes from top to bottom.

$n_{0,r,t}$ the male and female pups and by $n_{a,r,t}$ the number of female seals aged $a \in \{1, \ldots, 5, 6+\}$, in geographical region $r \in \{1, \ldots, 4\}$, in year $t \in \{0, 1, \ldots, 18\}$. The yearly state process consists of four subprocesses in the following sequence: survival and age incrementation, gender assignment, movement between regions, and birth. Following birth, a new time period starts.

Survival, birth, and gender assignment are Bernoulli processes. The survival probability for pups in region r at time $t, \phi_{r,t}$, is density dependent and varies between regions according to the Beverton–Holt model (Quinn and Deriso, 1999)

$$\phi_{r,t} = \frac{\phi_j}{1 + \beta_r \, n_{0,r,t-1}}, \tag{7}$$

where ϕ_j is an upper bound on juvenile (pup) survival. Age 1+ female seals have a constant annual survival probability, ϕ_{ad}. Age 6+ females give birth to a single pup with probability α, and pups are female with probability 0.5.

Regarding movement, we assume that only age-5 (prebreeding) females can move from their natal regions, and once they move, they remain faithful to their new region. Let $n_{5,r,t}^*$ and $n_{5,r,t}$ denote numbers in region r before and after movement, respectively, and $p_{r,t}$ the probability that an age-5 female is in region r after movement. We assume that

$$(n_{5,1,t}, n_{5,2,t}, n_{5,3,t}, n_{5,4,t})$$

$$\sim \text{Multinomial}\left(\sum_r n_{5,r,t}^*; p_{1,t}, p_{2,t}, p_{3,t}, p_{4,t}\right), \tag{8}$$

with probabilities

$$p_{r,t} = \frac{\sum_{i=1}^{4} n_{5,i,t}^{*} \rho_{i \to r,t}}{\sum_{i=1}^{4} n_{5,i,t}^{*}},$$

where $\rho_{i \to r,t}$ is proportional to 1 for $r = i$ and, for $r \neq i$, to $\gamma_{dd} \max(\phi_{r,t} - \phi_{i,t}, 0) \exp\{-\gamma_{dist} d(i, r)\}$, where the proportionality factor ensures that $\sum_{r=1}^{4} \rho_{i \to r,t} = 1$. The parameters γ_{dd} (where dd stands for density dependence) and γ_{dist} are both positive and $d(i, r)$ denotes sea distance between regions i and r. Hence, the probability odds of migrating to a different region are zero for regions whose expected pup survival probability is not larger than in the original region. The odds of migrating to a better region are proportional to the increment in pup survival probability multiplied by a decreasing function of the distance between the regions. In other words, better pup survival probability and distance between regions stimulate and deter migration, respectively.

Thus, for $t = 1, \ldots, 18$, the state equations are as follows: For each of the four regions (r), we have the following independent survival and birth processes:

$$n_{1,r,t} \sim \text{Binomial}(n_{0,r,t-1}, 0.5\phi_{r,t}), \tag{9}$$

$$n_{a,r,t} \sim \text{Binomial}(n_{a-1,r,t-1}, \phi_{ad}), \text{ for } a = 2, 3, 4, \tag{10}$$

$$n_{5,r,t}^{*} \sim \text{Binomial}(n_{4,r,t-1}, \phi_{ad}), \tag{11}$$

$$n_{6+,r,t} \sim \text{Binomial}(n_{5,r,t-1} + n_{6+,r,t-1}, \phi_{ad}), \tag{12}$$

$$n_{0,r,t} \sim \text{Binomial}(n_{6+,r,t}, \alpha), \tag{13}$$

whereas the movement process (8) links all four regions.

Observation process. For each year $t = 1, \ldots, 18$, the observations $y_{r,t}$ consist of an estimate of pup numbers for each region r. We assume that they are independent and normally distributed with mean $n_{0,r,t}$ and coefficient of variation $\psi^{-1/2}$:

$$y_{r,t} \sim \text{N}\left(n_{0,r,t}, n_{0,r,t}^{2}/\psi\right), \quad \psi > 0. \tag{14}$$

Prior distributions and initial state pdf. Independent prior distributions for the model parameters were developed from consultation with grey seal biologists (below). The distribution for initial pup numbers, $n_{0,r,0}$, was the product of a Gaussian variable centered on the pup estimates for 1984 (taken as $t = 0$) and an independent Uniform(0.7,1.3) variable, to incorporate additional variation. The distributions for initial age-1 to age-5 seal numbers used $n_{0,r,0}$, equations (9) and (10) repeatedly, with $t - 1$ (on the right-hand sides) replaced by 0. Lastly, $n_{6+,r,0}$ was taken to be equal to $n_{0,r,0}$ plus a negative binomial distribution with parameters $n_{0,r,0}$ and α. As was done for age 0, extra variation was incorporated in all ages via multiplication by independent Uniform distributions. This was followed by rounding to the nearest integer. More details can be found in Thomas et al. (2005).

Many parameters and states will be poorly identified from the data, in the sense that there can be alternative fits that are consistent with the observations; this will lead to severe correlations in the posterior distribution. The latter coupled with the high dimensionality of the posterior distribution, makes obtaining a sampling from it very challenging. Next, we indicate some main features of the computational algorithms we implemented.

3.2 *SIS Implementation*

The SIS algorithm used was essentially that of Liu and West (LW; 2001). Their method combines inference for unknown states and parameters simultaneously, using the auxiliary particle filter (Pitt and Shephard, 1999) in combination with kernel smoothing of parameters. For additional details of the LW algorithm we recommend Liu and West (2001) and the tutorial-style explanation given by Newman et al. (2006). Here we note the modifications we made to increase algorithm efficiency.

To begin, the dimension of the parameter space to sample via SIS was reduced by analytically integrating out the unknown precision parameter of the observation pdf, ψ, from the posterior distribution $\pi(\mathbf{n}_{0:T}, \eta, \psi \mid \mathbf{y}_{1:T})$. Thus SIS was used to generate a sample from $\pi(\mathbf{n}_{0:T}, \eta \mid \mathbf{y}_{1:T})$, where η denotes the parameters in the state process. The conditional distribution of $\psi, \pi(\psi \mid \mathbf{n}_{0:T}, \eta, \mathbf{y}_{1:T})$ was Gamma and a sample from it was directly generated using the posterior sample for $\mathbf{n}_{0:T}, \eta$.

To initialize the SIS routine, samples of the parameters (excluding ψ) and states in the initial year were generated directly from the prior and initial state pdfs. For the remaining years, the state process pdfs in Section 3.1 were used as the importance sampler, generating the entire state vector with 28 components (ages 0 to 6+ for each of the four regions). Age-5 numbers before movement are considered as a result of an intermediate subprocess (survival from age 4) and not part of the state vector. Kernel smoothing of the parameters in the LW algorithm was done using a multivariate normal distribution. Parameters restricted to a portion of the real number line were transformed to $\mathcal{R}^1$ before smoothing; i.e., the survival and birth probabilities and the density-dependent parameters, β_r, (which were a posteriori always smaller than 1) were logit transformed, and the movement parameters were log transformed. The degree of kernel smoothing was quite small with the weight of over 98% given to the original value and less than 2% to the total sample average.

To attempt to reduce particle depletion, and consequently MC variation, we modified the LW algorithm to include rejection control (Liu, 2001). The general idea is to oversample, generate $M > N$ particles, fix a cutoff value for weights, say the 95th percentile, keep all particles in the top 5% of weights, randomly subsample particles in the lower 95% of weights, and then adjust the weights of the selected particles appropriately. For the analysis of the simulated and the real data, multiple parallel runs were made with each run using the rejection control procedure twice: it was used after the first year, $t = 1$, then after the LW algorithm was carried out for years $t = 2, \ldots, 18$, it was used for the final year, with the entire process repeated until a sample of size N was attained.

3.3 *MCMC Implementation*

The MCMC implementation was tailored for the seal SSM to speed convergence and increase precision in the posterior samples.

Analytical integration of parameters and states was performed whenever possible, substantially reducing the dimension of the posterior distribution being sampled by MCMC. In addition to the parameter ψ which was also integrated out analytically for SIS, the birth probability parameter, α, and the states $n_{1,r,t}$, $n_{2,r,t}$, $n_{3,r,t}$, and $n_{4,r,t}$ were integrated out. These parameters and states were then directly sampled from the MCMC generated posterior samples for the other parameters and states. However, unlike for SIS, the state vector must be augmented with the number of age-5 seals before movement, $n_{5,r,t}^{*}$, to be able to evaluate the state pdfs intervening in the Metropolis–Hastings acceptance probabilities. This is because it is not possible to derive analytically an expression for the state process pdf to go from numbers age 4 in one year to numbers age 5 (after movement) in the following year.

Reparameterization was used to facilitate sampling. Instead of sampling γ_{dd} we used a new parameter $\gamma_m = \gamma_{dd}\phi_j$ which effectively removed ϕ_j from the movement distribution. To summarize, the parameters and states sampled via MCMC were $\phi_{ad}, \phi_j, \gamma_m, \gamma_{dist}, \beta_1, \ldots, \beta_4, n_{0,r,t}, n_{5,r,t}, n_{5,r,t}^{*}$, and $n_{6+,r,t}$ for each region r and time period t.

The conditional posterior densities of ϕ_{ad} and ϕ_j are log concave and these parameters were sampled directly using Gibbs sampling via adaptive rejection sampling (Gilks and Wild, 1992). For the remaining parameters and states, the Metropolis–Hastings algorithm was used, using t distributions as proposals (truncated to the relevant domain and discretized, as appropriate), matching the mode and curvature at the mode of the corresponding conditional posterior pdf.

The previous paragraphs describe very succinctly our MCMC sampling strategy; however, the details were extremely intricate. Much of the added complication arose from the movement model (from the fact that for $i \neq r, \rho_{i \to r,t}$ is strictly positive only if $\phi_{r,t} > \phi_{i,t}$) and from the distribution of states in the initial year.

Figure 2 provides a schematic side-by-side comparison of our SIS and MCMC implementations for the seal application.

3.4 *Results*

The SIS code was written in `ANSI C` and MCMC code in `FORTRAN 77`; both are available on request from the authors. Computation times reported here are from program runs on the same computer (a dual AMD Operon 248 processor with 2GB RAM).

3.4.1 *Simulated data.* For each of the five simulated datasets we did five MCMC and SIS runs. After initial experimentation with various lengths, we found that MCMC runs of $B = 1.5$ million and $N = 1.5$ million iterations (taking about 12 hours) appeared acceptable in terms of getting consistent results from widely dispersed starting values. For comparability with the MCMC results, we used the same total run time (12 hours) for SIS, which meant generating 20 million particles. These were generated in 20 batches of 1 million particles, to avoid computer memory issues. As a final check, we performed a single long run for each dataset of MCMC ($B = 5$ million and $N = 15$ million, about 80 hours computation time per run) and SIS (300 million particles, about 180 hours computation time per run), which led to results that were consistent within datasets to within two significant figures for

posterior means and standard deviations of parameter and states.

For the MCMC implementation, the proposal distributions for Metropolis–Hastings led to very high acceptance probabilities, always above 70% and in the vast majority of cases around 95%. Most of the time was spent numerically computing modes of target distributions.

Table 1 is a summary of the results for inference (based on the five shorter runs for each of the five datasets) about the parameters and some of the states. When averaged across the 25 runs, the differences between SIS and MCMC in posterior means and standard deviations for parameters were typically less than 5% (Table 1), with the exception of the mean for the movement density dependence parameter, γ_{dd}, and the standard deviation for the birth probability parameter α. There were, however, small but consistent differences in posterior means for parameters between MCMC and SIS within datasets (see Web Appendix B), which we suspect are due to bias in SIS caused by the kernel smoothing. For the states shown (pups and age 6+), the posterior means were quite similar (within 1% of each other); however, the posterior standard deviations tended to be larger for SIS. MC variation (meaning the variation obtained from different runs of an algorithm given a dataset) for parameters was generally greater for SIS than for MCMC, as can be seen from the average coefficients of variation of the posterior means and standard deviations displayed in Table 1, with the exception of the average coefficient of variation of standard deviations for α. The pattern of MC variation in the states, however, was less consistent, with SIS having greater variation for pups and MCMC having greater variation for age 6+ females. The larger MC variation for α and age 6+ abundances can be largely attributed to the severe autocorrelation in the MCMC draws (e.g., for α the value was above 0.9 at lag 50) and a negative correlation between α and age 6+ abundances, resulting in slow mixing and a need for very long runs. The correlation between α and age 6+ abundances is due to only having pup estimates and no estimates of adult female seals: the number of pups observed can be explained by many combinations of birth probabilities and number of mature females.

3.4.2 *Real data.* As with the longer MCMC and SIS runs for the simulated datasets, MCMC was carried out with $B = 5$ million and $N = 15$ million and SIS results were based on 300 million particles. Marginal prior and posterior distributions for selected parameters are shown in Figure 3. Posterior means for the parameters from the SIS and MCMC runs were quite close in value (Table 2) with the exception of the density-dependent movement parameter, γ_{dd}, and the observation error parameter, ψ. The MCMC results indicated stronger site fidelity (i.e., smaller γ_{dd} value) than SIS; the reason for the difference is not understood, especially in light of the opposite ordering between MCMC and SIS in estimates of γ_{dd} based on the simulated data. The mean for ψ was 8% lower for SIS than for MCMC and is possibly due to bias in kernel smoothing.

Results for the pup numbers are shown in Figure 4. Posterior 95% central credible intervals are much narrower than a priori. A similar reduction in uncertainty was attained for other unobserved age categories (not shown here). In other words, just tying down the value of one of the states (numbers

SIS

Initialization   Generate $\mathbf{n}_0^{(i)}, \eta^{(i)}$ and set

$$w_0^{(i)} = 1, i = 1, \ldots, N.$$

Iteration   For $t=1$ to T {

    For $i=1$ to N {

$$w_{1,t}^{(i)} = w_{t-1}^{(i)} F_1(\mathbf{y}_{1:4,1:t}|\mathbf{n}_{0,1:4,1:t}^{(i)})$$

    }

    Resample $\mathbf{n}_{0:t}$ and η using $w_{1,t}$

    Kernel smooth η

    For $i=1$ to N {

      Generate $\mathbf{n}_t^{(i)}$ from $g_t(\mathbf{n}_t|\mathbf{n}_{t-1}^{(i)})$

$$w_t^{(i)} = \frac{F_2(\mathbf{y}_{1:4,1:t}|\mathbf{n}_{0,1:4,1:t}^{(i)})}{F_1(\mathbf{y}_{1:4,1:t}|\mathbf{n}_{0,1:4,1:t}^{(i)})}$$

    }

    If resample (optional):

      Set $w_t^{(i)} = 1, i = 1, \ldots, N$

}

MCMC

Initialization   Generate $\mathbf{n}_{0,1:4,0:T}^{(0)}, \mathbf{n}_{5,1:4,0:T}^{*(0)}, \mathbf{n}_{5,1:4,0:T}^{(0)}, \mathbf{n}_{6+,1:4,0:T}^{(0)},$

$$\phi_{ad}^{(0)}, \phi_j^{(0)}, \gamma_m^{(0)}, \gamma_{dist}^{(0)}, \beta_1^{(0)}, \beta_2^{(0)}, \beta_3^{(0)}, \beta_4^{(0)}$$

Iteration   For $i=1$ to $(B+N)$ {

    For $t=0$ to T, $r=1$ to 4 {

      For $a=0, 5$, and $6+$ {

        Generate $n_{a,r,t}^{(c)}$ from $h_t(n_{a,r,t}|n_{a,r,t}^{(i-1)})$

        Accept candidate wp $MH\left(n_{a,r,t}^{(c)}, n_{a,r,t}^{(i-1)}\right)$

        If accept, set $n_{a,r,t}^{(i)} = n_{a,r,t}^{(c)}$

        If reject, set $n_{a,r,t}^{(i)} = n_{a,r,t}^{(i-1)}$

      }

      Perform similar step for $n_{5,r,t}^*$

    }

    Sample $\phi_{ad}^{(i)}$ and $\phi_j^{(i)}$ via ARS

    Generate $\gamma_m^{(c)}$ from $h(\gamma_m|\gamma_m^{(i-1)})$

    Accept candidate wp $MH\left(\gamma_m^{(c)}, \gamma_m^{(i-1)}\right)$

    If accept, set $\gamma_m^{(i)} = \gamma_m^{(c)}$

    If reject, set $\gamma_m^{(i)} = \gamma_m^{(i-1)}$

    Perform similar step for each of $\gamma_{dist}, \beta_1, \beta_2, \beta_3, \beta_4$

}

For each SIS particle kept, sample ψ from the corresponding conditional posterior distribution

For each MCMC sample kept, sample ages 1 to 4, α and ψ from the corresponding conditional posterior distribution

Inference $\hat{E}(\mathbf{n}_t) = \dfrac{\sum_{i=1}^N w_T^{(i)} \mathbf{n}_t^{(i)}}{\sum_{i=1}^N w_T^{(i)}}, \ \hat{E}(\eta) = \dfrac{\sum_{i=1}^N w_T^{(i)} \eta^{(i)}}{\sum_{i=1}^N w_T^{(i)}}$

$\hat{E}(\mathbf{n}_t) = \frac{1}{N}\sum_{i=B+1}^{B+N} \mathbf{n}_t^{(i)}, \ \hat{E}(\eta) = \frac{1}{N}\sum_{i=B+1}^{B+N} \eta^{(i)}$

Figure 2. Comparison of SIS and MCMC algorithms used to simulate a sample of size N from the posterior distribution from $\pi(\mathbf{n}_{0:T}, \eta \,|\, \mathbf{y}_{1:T})$. Notation corresponds to Sections 2 and 3; "wp" means with probability, and ARS is adaptive rejection sampling. With ψ integrated out of the SIS algorithm, the weights are updated using complicated functions of $\mathbf{y}_{1:4,1:t}$, $E[\mathbf{n}_{0,1:4,t} \,|\, \mathbf{n}_{5:6,1:4,t-1}]$, and $\mathbf{n}_{0,1:4,1:t}$, denoted $F_1(\mathbf{y}_{1:4,1:t} \,|\, \mathbf{n}_{0,1:4,1:t}^{(i)})$ and $F_2(\mathbf{y}_{1:4,1:t} \,|\, \mathbf{n}_{0,1:4,1:t}^{(i)})$; rejection control is not shown.

age 0, which are the only ones observed) at each time point has helped substantially to reduce uncertainty about the values of the other states. Note that the recorded numbers of pups are subject to observation errors, whereas the credible intervals reflect uncertainty in the true numbers of pups. Hence more than 5% of the observations fall outside the bands, as would be expected.

The paucity of data relative to model complexity was manifested in two ways. Firstly, the strong similarity of the prior and posterior distributions for some parameters (e.g., ϕ_j, in Figure 3) suggests strong prior influence. Secondly, strong correlations exist between some of the parameters and states, reflecting identifiability issues. For example, as for the simulated data results, α and $n_{6+,r,t}$ are negatively correlated (the

correlation coefficient ranged from -0.75 to -0.32), which is due to the lack of observations on adult female seals. The adult survival parameter ϕ_{ad} displays a correlation of about -0.9 with juvenile survival probabilities $\phi_{r,t}$ for each region and time period, while the latter are positively correlated among themselves. There is also extreme positive correlation (around 0.99) between numbers of adult female seals of different ages in the same cohort, but not with the number of pups, a consequence of having an observation on the latter quantity.

A major goal of the modeling has been to estimate the total population size using pup counts alone. Such estimates are used as inputs to models that then estimate fish consumption by seals. Even with such relatively sparse data and

Table 1

Simulation study results for inference about parameters and some of the states. All results are averaged over the five simulated datasets. Per dataset, five runs were carried out of MCMC and SIS. Averages of means and SDs are the average values of the posterior means and standard deviations for the 25 estimates (five datasets × five estimation runs). Average of CVs is the average (across datasets) of the coefficient of variation (%) corresponding to five runs of each given dataset. For example, for each dataset SIS was carried out five times and the CV of posterior means was calculated. Subsequently, the CVs from the five datasets are averaged resulting in the values shown in the table. The true β values, averages, and standard deviations have been multiplied by 10^4. The true states are averages over the five datasets.

Parameter	True	Average of Means		Avg. of CVs for Means		Average of SDs		Avg. of CVs for SDs	
		SIS	MCMC	SIS	MCMC	SIS	MCMC	SIS	MCMC
α	0.95	0.95	0.96	0.63	0.67	0.047	0.032	8.91	19.40
ϕ_{ad}	0.95	0.95	0.95	0.11	0.12	0.012	0.011	7.28	5.97
ϕ_j	0.70	0.68	0.67	1.75	1.24	0.099	0.099	6.14	4.00
γ_{dd}	3.89	3.57	3.88	6.19	5.17	2.72	2.71	12.29	7.70
γ_{dist}	0.50	0.49	0.49	6.73	0.68	0.33	0.32	11.27	0.75
β_1	8.30	8.42	8.88	2.51	1.22	2.00	2.05	10.02	2.31
β_2	11.80	13.08	13.73	2.78	1.40	3.14	3.27	10.89	2.11
β_3	3.00	3.52	3.69	2.72	1.78	0.95	0.94	12.35	3.72
β_4	2.30	2.20	2.29	2.80	1.10	0.57	0.57	10.85	2.42
ψ	140	133.2	139.0	0.92	0.45	24.19	24.85	1.44	0.31
States	True	SIS	MCMC	SIS	MCMC	SIS	MCMC	SIS	MCMC
$n_{0,1,9}$	2518	2493	2489	0.22	0.07	71	61	2.24	1.90
$n_{0,2,9}$	1937	1900	1896	0.18	0.11	56	48	4.27	1.93
$n_{0,3,9}$	7957	7903	7872	0.25	0.16	216	183	2.73	3.63
$n_{0,4,9}$	7932	7946	7906	0.25	0.16	211	184	4.86	3.93
$n_{0,1,18}$	3173	3189	3190	0.24	0.20	132	117	3.13	2.86
$n_{0,2,18}$	2265	2227	2229	0.23	0.20	89	80	3.04	2.31
$n_{0,3,18}$	9045	8663	8650	0.28	0.23	353	313	3.12	3.92
$n_{0,4,18}$	10,730	10,976	11,006	0.32	0.20	456	406	3.32	4.90
$n_{6+,1,9}$	2644	2627	2586	0.69	0.74	154	112	8.45	15.45
$n_{6+,2,9}$	2038	2002	1970	0.72	0.71	117	86	8.47	16.09
$n_{6+,3,9}$	8380	8328	8182	0.80	0.83	493	351	9.56	15.04
$n_{6+,4,9}$	8349	8373	8215	0.66	0.78	478	346	8.40	17.29
$n_{6+,1,18}$	3324	3367	3315	0.73	0.76	238	172	7.46	12.24
$n_{6+,2,18}$	2389	2351	2317	0.60	0.76	161	117	5.94	12.38
$n_{6+,3,18}$	9510	9145	8990	0.78	0.82	626	462	7.25	11.99
$n_{6+,4,18}$	11,275	11,587	11,438	0.72	0.77	821	584	9.82	13.56

a relatively complex model, we have still gained useful knowledge. Our simulations show that adult female numbers are estimated quite accurately when the correct model is used, and the results on real data show considerable reduction in uncertainty on total population size relative to priors. However, additional work (not reported here) has shown extreme sensitivity of estimates of adult population size to the biological process model used. For example, an equally plausible model includes density dependence in fecundity, rather than in juvenile survival. This model produces an almost indistinguishable fit to the pup count data, but posterior estimates of adult population size are around four times larger. A major outcome of this work has been to focus discussion on alternative methods of obtaining a single, independent estimate of total population size to calibrate the biological models.

Of additional biological interest, we note the shift in the posterior for γ_{dd} toward smaller values than the prior indicate stronger site fidelity than thought a priori. Allowing for the influence of priors, we also note that the degree of density dependence in juvenile survival is largest in the Inner Hebrides colonies (region 2), as the posterior distribution is centered on larger β_2 values.

4. Discussion and Conclusions

Our primary motivation was to answer the question of whether MCMC or SIS is a better underlying engine for software for ecologists to fit SSMs to animal population data. To answer this, we applied both methods to two problems, one for seals and one for salmon (described in Web Appendix A). We were successful with both approaches in the sense that we produced samples from the posterior distribution for parameters and states with acceptable levels of MC variation, and based on simulated data, with acceptable accuracy.

To determine which is better, we use the following criteria: ease of implementation (how difficult to translate SSM equations into computer code), computational efficiency (how much computing time and resources are needed to achieve an acceptable level of MC variation), and accuracy (how close
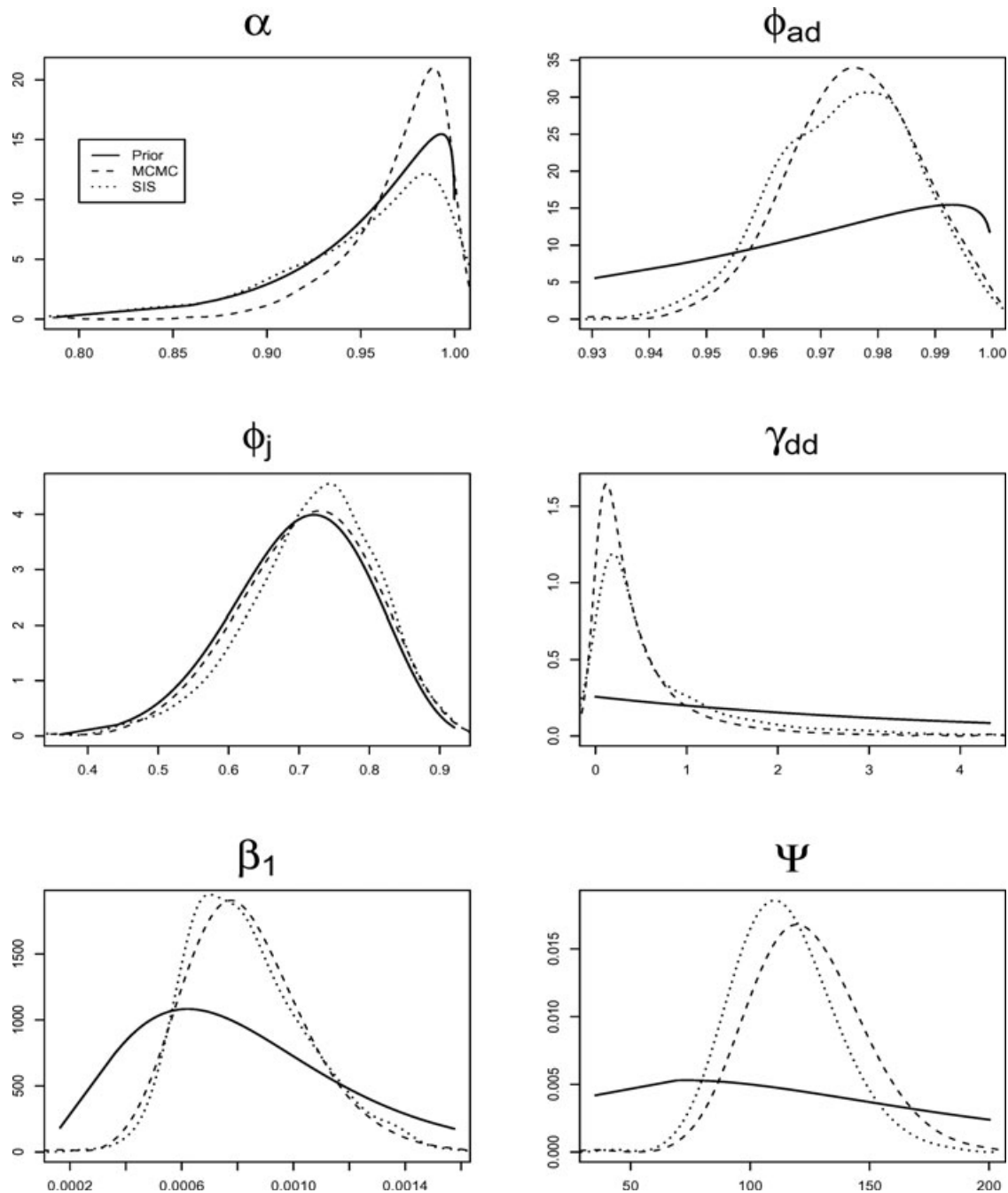
Figure 3. Posterior distributions for some parameters using MCMC (dashed line) and SIS (dotted lines). Solid lines are the prior densities.

the sample posterior distribution is to the true distribution). These criteria can be at odds with each other, e.g., an easily implemented algorithm can be relatively inefficient and inaccurate. Thus one needs to balance time spent selecting and coding a SIS or MCMC algorithm with computational time and resources.

Our conclusion is that there is no clearcut choice. For both SIS and MCMC, ease, efficiency, and accuracy are a function of both the SSM formulation and the available data. For the salmon, a relatively straightforward population dynamics model with ample data, design, and implementation of suitably efficient and accurate MCMC and SIS algorithms were relatively easy, although MCMC was less mechanical. For the seals, a complex metapopulation dynamics model with survival, birth and movement processes, and very limited data, considerable effort was needed to produce satisfactory MCMC and SIS implementations.

We close with some general observations and conclusions about using MCMC and SIS to fit SSMs, for animal populations in particular.

Table 2
Prior distributions and posterior means and standard deviations for seal model parameters based on SIS and MCMC. SIS results are based on a total of 300 million particles, while MCMC results are based on a burn-in of 5 million followed by a run of 15 million. The values for the β's have been multiplied by 10^4.

Param	Prior	E[Prior]	Mean		SD	
			SIS	MCMC	SIS	MCMC
α	Beta(22.05, 1.15)	0.950	0.948	0.970	0.052	0.027
ϕ_{ad}	Beta(22.05, 1.15)	0.950	0.974	0.976	0.012	0.011
ϕ_j	Beta(14.53, 6.23)	0.700	0.719	0.710	0.091	0.093
γ_{dd}	Gamma(1.00, 3.89)	3.890	0.711	0.459	0.942	0.572
γ_{dist}	Gamma(2.25, 0.22)	0.500	0.548	0.579	0.374	0.381
β_1	Gamma($4, 2.07 \times 10^{-4}$)	8.30	8.34	8.28	2.13	2.13
β_2	Gamma($4, 2.96 \times 10^{-4}$)	11.80	12.80	13.14	3.14	3.20
β_3	Gamma($4, 7.40 \times 10^{-5}$)	3.00	3.88	4.07	1.11	1.13
β_4	Gamma($4, 5.76 \times 10^{-5}$)	2.30	1.46	1.39	0.41	0.40
ψ	Gamma(2.1, 66.67)	140	114.0	123.7	21.2	23.4

1. *Similarities.* First, both are "divide and conquer" strategies for generating samples from high-dimensional posteriors: both work with just one time period at a time, although SIS generates a sample of the desired size for one time period before moving to the next whereas MCMC moves through all the time periods in each single sample (see Figure 1). Second, the choice of a proposal distribution in MCMC is analogous to the choice of an importance distribution in the SIS methods. In both cases poor choices can lead to high MC error: a poor proposal distribution in MCMC leads to slow or little movement through the parameter and state space ("slow mixing"), and a poor sampler in SIS will have high particle depletion yielding just a small number of different values.

2. *Ease of implementation and effect on model formulation.* For SSMs with a complex state process pdf, SIS samplers are relatively mechanical to implement, so long as the importance sampler simulates the individual subprocesses, e.g., survival, movement, harvest, birth, that characterize the state process.

   MCMC can be mechanical, too, so long as the state process pdf does not need to be evaluated; e.g., when Gibbs sampling is possible for all the states and all the parameters. However, this will typically not be the case for SSMs. Implementing a Metropolis–Hastings algorithm can also be mechanical at the proposal stage (e.g., if candidate values are generated from Normal distributions centered at current values), but the acceptance probability formula requires evaluation of the state pdf, which can be impossible in cases with intermediate subprocesses. A solution is to extend the state vector to include intermediate or latent components (as was done for the seal model, for example, by adding premovement age-5 females) and thus evaluate tractable subprocesses pdfs.

   With several subprocesses, this can become cumbersome and it may be more attractive to reformulate a more tractable model. This was what happened for the seal model, where an earlier formulation of the move-

ment subprocess involved four multinomial distributions, one for the age-5 females at each colony (Thomas et al., 2005); e.g., the numbers of age-5 females at colony 1 that stay at 1 or move to colonies 2, 3, or 4 were multinomially distributed. This formulation was subsequently modified to a single, somewhat complicated multinomial (equation (8)) to simplify the evaluation of the state process pdf, which otherwise would involve evaluating the pdf for a convolution of four multinomials. For SIS, however, it was equally easy to work with four multinomial distributions, and transparency between the original conceptual formulation and the implementation remained high.

   SIS also has the advantage of providing the posterior distribution after each intermediate time step $t < T$. This can be a useful diagnostic for possible model deficiencies, such as a parameter being treated as fixed that should be time varying. MCMC can also provide such information by fitting an SSM using just part of the sample, $\mathbf{y}_{1:t}$, but this requires making another run of the MCMC algorithm, which is time consuming.

3. *Computational and statistical efficiency.* The most easily implemented SIS algorithms, which simulate from the state process alone and resample periodically (e.g., the bootstrap filter; Liu, 2001), can be very inefficient, with sometimes impossibly large sample sizes required to achieve acceptable MC error. For example, a bootstrap filter applied to the seal example yielded an effective sample size of 4.9 unique samples after 600 million particles had been generated. The LW algorithm is more sophisticated and relatively easy to implement, but to get sufficient accuracy for the seal example, additional and complicated refinements were needed, such as analytically integrating parameters in the observation pdf and using rejection control, along with still large sample sizes.

   A well-designed, which may mean difficult to set up, MCMC sampler can be relatively efficient even for complex SSMs. For the salmon model and a relatively informative observation vector, given the same computing
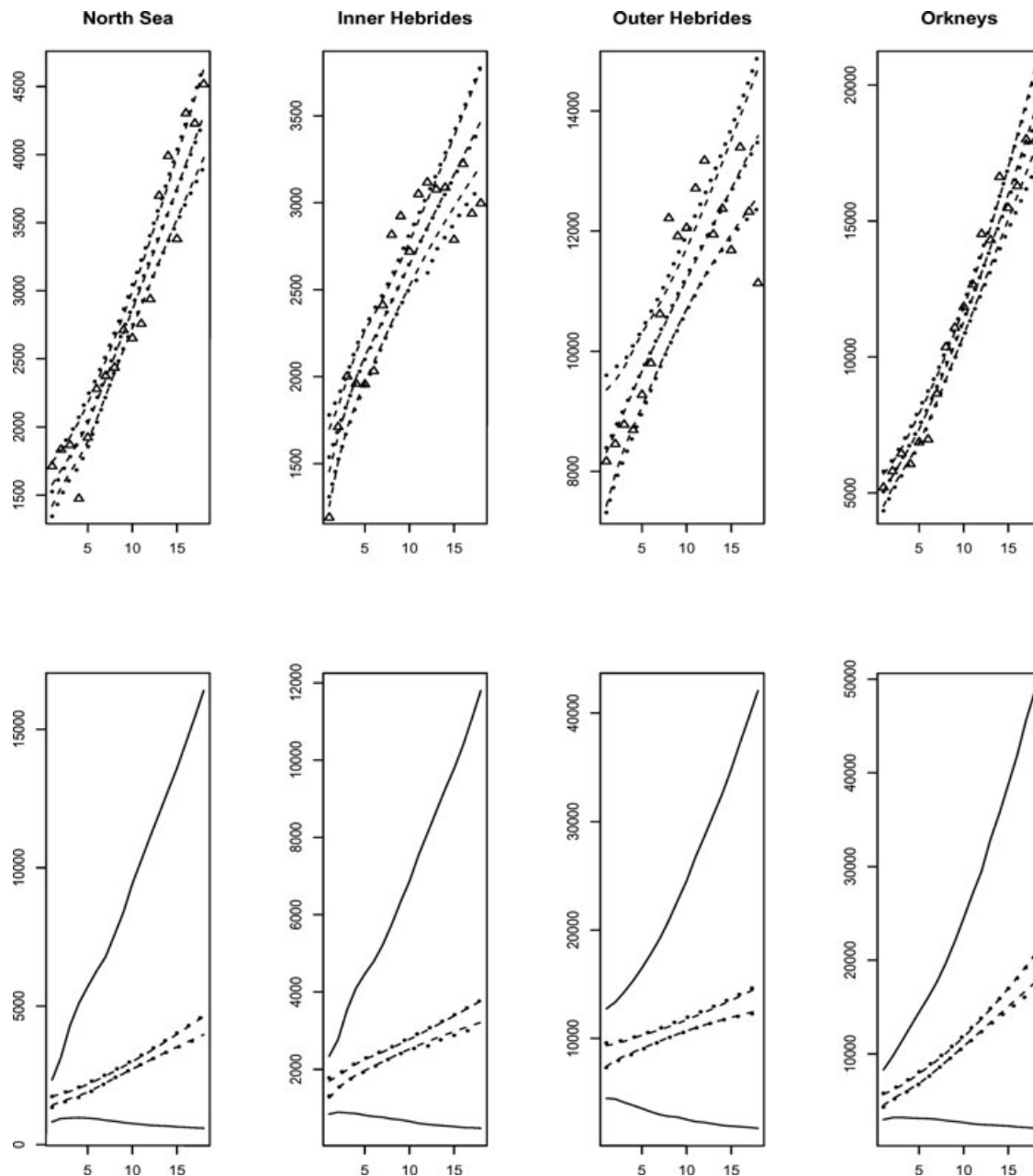
Figure 4. Top row: Observed data (triangles) and evolution over time of posterior 2.5, 50, and 97.5 percentiles of number of pups (dashed lines–MCMC, dotted lines–SIS). Bottom row: Prior (solid lines) and posterior (dashed lines–MCMC, dotted lines–SIS) 2.5 and 97.5 quantiles for number of pups over time.

time, the MCMC sampler had 5 to 10 times less MC variation for parameters and states than the SIS sampler (Web Appendix A). For the relatively complex seal model with seven age classes but only pup estimates, the MCMC sampler had less MC variation than SIS for all parameter estimates but probability of birth (α), but more MC variation for the age 6+ females (Web Appendix B), due presumably to the high correlation with α. Thus the SIS sampler was competitive in a situation where the data were relatively uninformative in relation to the priors. From our experiences here we conclude that problems of a level of difficulty (in terms of dimensionality and correlation) similar to that displayed by the grey seals problem are, at present, not amenable to more or less automatic computational implementations. In general, MCMC, tailored specifically for a particular SSM where data are informative relative to the priors, remains the gold standard, but SIS with repeated runs to assess MC variation could prove a useful alternative approach.

5. Supplementary Materials

The Web Appendices, referenced in Sections 1, 3, and 4 are available under the Paper Information link at the *Biometrics* website, `http://www.biometrics.tibs.org`.

References

Bassetti, F. and Diaconis, P. (2006). Examples comparing importance sampling and the Metropolis algorithm. *Illinois Journal of Mathematics* **50,** 67–91.

Buckland, S. T., Newman, K. B., Thomas, L., and Koesters, N. (2004). State-space models for the dynamics of wild animal populations. *Ecological Modelling* **171,** 157–175.

Buckland, S. T., Newman, K. B., Fernández, C., Thomas, L., and Harwood, J. (2007). Embedding population dynamics models in inference. *Statistical Science* **22,** 44–58.

Clark, J. S. and Bjørnstad, O. N. (2004). Population time series: Process variability, observation errors, missing values, lags, and hidden states. *Ecology* **85,** 3140–3150.

Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Journal of the Royal Statistical Society, Series C* **41,** 337–348.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. London: Chapman & Hall.

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

Ionides, E. L., Breto, C., and King, A. A. (2006). Inference for nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* **103,** 18438–18443.

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

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

Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). `WinBUGS`—a Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing* **10,** 325–337.

Meyer, R. and Millar, R. B. (1999). Bayesian stock assessment using a state-space implementation of the delay difference model. *Canadian Journal of Fisheries and Aquatic Sciences* **56,** 37–52.

Michielsens, C. G. J., McAllister, M. K., Kuikka, S., Pakarinen, T., Karlsson, L., Romakkaniemi, A., Perä, I., and Mäntyniemi, S. (2006). A Bayesian state-space mark-recapture model to estimate exploitation rates in mixed-stock fisheries. *Canadian Journal of Fisheries and Aquatic Sciences* **63,** 321–334.

Newman, K. B., Buckland, S. T., Lindley, S. T., Thomas, L., and Fernández, C. (2006). Hidden process models for animal population dynamics. *Ecological Applications* **16,** 74–86.

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

Quinn, T. J. and Deriso, R. B. (1999). *Quantitative Fish Dynamics*. New York: Oxford University Press.

Rivot, E., Prévost, E., Parent, E., and Baglinière, J. L. (2004). A Bayesian state-space modelling framework for fitting a salmon stage-structured population model to multiple time series of field data. *Ecological Modelling* **179,** 463–485.

Schnute, J. (1994). A general framework for developing sequential fisheries models. *Canadian Journal of Fisheries and Aquatic Science* **51,** 1676–1688.

Schnute, J. and Kronlund, A. R. (2002). Estimating salmon stock-recruitment relationships from catch and escapement data. *Canadian Journal of Fisheries and Aquatic Science* **59,** 433–449.

Thomas, L., Buckland, S. T., Newman, K. B., and Harwood, J. (2005). A unified framework for modelling wildlife population dynamics. *Australian and New Zealand Journal of Statistics* **47,** 19–34.

Trenkel, V. M., Elston, D. A., and Buckland, S. T. (2000). Fitting population dynamics models to count and cull data using sequential importance sampling. *Journal of the American Statistical Association* **95,** 363–374.

Wikle, C. K. (2003). Hierarchical Bayesian models for predicting the spread of ecological processes. *Ecology* **84,** 1382–1394.