Supplementary Information One: Sequential Monte Carlo methods for data assimilation problems in ecology.

Kwaku Peprah Adjei[1,2], Rob Cooke[3], Nick Isaac[3], Robert B. O'Hara[1,2]

[1]Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway

[2]Centre of Biodiversity Dynamics, Norwegian University of Science and Technology, Trondheim, Norway

[3]Center of Ecology and Hydrology, Wallingford, UK

1 Introduction

This document provides some background information needed to understand the Algorithms implemented in the main paper. We first provide information about state space models (SSM), sequential Monte Carlo (SMC) methods, and then proceed to describe the bootstrap and auxiliary particle filters as well as the particle Markov Chain Monte Carlo (pMCMC) with their algorithms. It must be stated that some of the equations and terminologies are repeated to enhance the smooth understanding of this document, with fewer referrals to the main paper.

2 State space models

We assume we can obtain a (multivariate) observed time series data denoted by $y_{1:T} = \{y_1, y_2, \ldots, y_T\}$, where y_t is a $(k \times 1)$ observed vector and k number of observed points at each time step. These observations depend on latent (unobserved but in interest) states $x_{1:T} = \{x_1, x_2, \ldots, x_T\}$, which are assumed to have a first order Markov structure (the latent state at time t depends on latent state at time $t-1$ only) and the observations at each time t, y_t, given the latent state at that time x_t are independent of previous observations and states.

In summary, we have the following information for the SSM framework:

$$
\begin{aligned}
\text{Intial state distribution}: &\quad p(x_0|\theta); \quad t = 0 \\
\text{State model}: &\quad p(x_t|x_{t-1}, \theta); \quad t = 1, 2, \dots, T \\
\text{Observation model}: &\quad p(y_t|x_t, \theta); \quad t = 1, 2, \dots, T
\end{aligned}
\tag{1}
$$

where θ are model parameters (assumed to be constant in the SSM defined in equation 1).

Furthermore, we assume that we have already fitted a SSM with MCMC approach, to the observed data from time 1 to t. The posterior samples from the fitted SSMs are organised in a number of interations $\times$ t matrix of latent state posterior samples and number of interations $\times$ number of model parameters matrix of model parameter posterior samples as shown in equation (2).

$$
\begin{array}{ccc}
1 \quad \text{Old observation: } y_1, \dots, y_t & t & \text{New observation: } y_{t+1}, \dots, y_T \quad T \\
\underbrace{\begin{bmatrix} \hat{x}_1^{(1)} & \hat{x}_2^{(1)} & \dots & \hat{x}_t^{(2)} \\ \hat{x}_1^{(2)} & \hat{x}_2^{(2)} & \dots & \hat{x}_t^{(2)} \\ \vdots & \vdots & \dots & \vdots \\ \hat{x}_1^{(n.iter)} & \hat{x}_2^{(n.iter)} & \dots & \hat{x}_t^{(n.iter)} \end{bmatrix} \begin{bmatrix} \hat{\theta}_r^{(1)} \\ \hat{\theta}_r^{(2)} \\ \vdots \\ \hat{\theta}_r^{(n.iter)} \end{bmatrix}}_{\text{MCMC used for State Space models}} & & \underbrace{\begin{bmatrix} x_{t+1}^{(1)} & \dots & x_T^{(2)} \\ x_{t+1}^{(2)} & \dots & x_T^{(2)} \\ \vdots & \dots & \vdots \\ x_{t+1}^{(n.iter)} & \dots & x_T^{(n.iter)} \end{bmatrix} \begin{bmatrix} \theta_{upd}^{(1)} \\ \theta_{upd}^{(2)} \\ \vdots \\ \theta_{upd}^{(n.iter)} \end{bmatrix}}_{\text{SMC used to update latent states and parameters}}
\end{array}
\tag{2}
$$

where $\hat{\theta}_r$ and $\hat{\mathbf{x}}_{1:t} = \{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_t\}$ are posterior samples of model parameters and latent states from the already-fitted SSM to the observations $y_{1:t} = y_1, y_2, \dots, y_t$; θ_{upd} and $\mathbf{x}_{t+1:T} = \{\mathbf{x}_{t+1}, \dots, \mathbf{x}_T\}$ are posterior samples of parameters and latent states we are interested in obtaining after the new stream of observations $y_{t+1:T} = \{y_{t+1}, y_{t+2}, \dots, y_T\}$ are obtained.

3 Sequential Monte Carlo (SMC) methods

The SMC methods use sequential importance sampling (SIS) technique to estimate the filtering distributions (Doucet, De Freitas, and Gordon 2001; Michaud et al. 2021). At each time step t, the latent state x_t is proposed from the previous state x_{t-1} from a proposal distribution or importance function, $\pi(x_t|x_{t-1}, y_{1:t}, \theta)$, and posterior samples of x_t are drawn from the proposed samples using importance weights w_t:

$$w_t^{(i)} \propto \frac{p(x_t^{(i)}|x_{t-1}^{(i)}, y_{1:t}, \theta)}{\pi(x_t^{(i)}|x_{t-1}^{(i)}, y_{1:t}, \theta)} \tag{3}$$

and iteratively as:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(x_t^{(i)}|x_{t-1}^{(i)}, \theta) p(y_{1:t}|x_t^{(i)})}{\pi(x_t^{(i)}|x_{t-1}^{(i)}, y_{1:t}, \theta)} \tag{4}$$

for $i = 1, 2, \dots, M$ particles.

Since the importance function chosen for SIS is critical to the performance of the SMC method (Arulampalam et al. 2002; Doucet, De Freitas, and Gordon 2001; Michaud et al. 2021), the prior distribution of the latent states are chosen as the importance function. In this case, the importance weights in equation (4) simplifies to:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \times p(y_t|x_t^{(i)}, \theta); \tag{5}$$

for $i = 1, 2, \dots, M$ particles. See Doucet, De Freitas, and Gordon (2001) for the details of the simplification.

This suggests that from time steps $t + 1$ to T (where we are have new data), we only need information about the weights at t and the SSM distributions in equation (1) (which are assumed to be known as shown in equation 2), we can use any of the SMC algorithms, such as bootstrap or auxiliary particle filter to be briefly discussed in sections 3.1 and 3.2 respectively, to estimate posterior distribution of latent states and update the model parameters. The saved posterior samples of latent state for time steps $1, 2, \dots, t$ are assumed to be equally weighted samples (i.e. they have equal weights and we choose $w_t^{(i)} = w_{t-1}^{(i)} \times p(y_t|x_t, \theta), \forall i$). In summary, our proposed framework uses the following information for generating the posterior samples of the latent states: for the j^{th} ($j = 1, 2, \dots, N$) draw of $x_{1:t}$ from the posterior distribution of the reduced model, the following information is used for the particle algorithms in the updating process:

$$\begin{cases} w_s^{(j)} = w_{s-1} p(y_s | x_s, \theta_r) & x_s^{(i)} = x_s^{(j)} & \text{for } s = 1, 2, \dots, t \text{ and } 1 \leq i \leq M \\ w_s^{(i)} \text{ defined by any SMC algorithm} & x_s^{(i)} \text{ from SMC} & \text{for } s = t+1, t+2, \dots, T \text{ and } 1 \leq i \leq M \end{cases}. \tag{6}$$

3.1 Bootstrap particle filter

The bootstrap particle filter (hereafter, BPF) re-samples with replacement the M particles $(x_{1:T}^{(i)}; i = 1, 2, \dots, M)$ from the set of proposed samples $(\tilde{x}_{1:T}^{(i)}; i = 1, 2, \dots, M)$ according the importance weights $(w_t^{(i)}; i = 1, 2, \dots, M)$ defined in equations 3 to 5. This re-sampling approach mitigates the particle degeneracy problem, where unimportant particles are propagated through time (Arulampalam et al. 2002; Doucet, De Freitas, and Gordon 2001).

Algorithm 1 shows the changes made to the bootstrap particle filter algorithm implemented in `nimbleSMC` (Michaud et al. 2021).

3.2 Auxiliary particle filter

The auxiliary particle filter (APF hereafter; Pitt and Shephard (1999)) uses the new observation to generate more likely states by including an additional "look-ahead step" (Michaud et al. 2021). At each time step t, this is done by first sampling M particles from weights from time $t-1$ which are calculated using a rough estimate of the likelihood of the current data given the particles from the previous time point (Michaud et al. 2021). The sampled particles are then propagated in time by the proposal distribution $\pi(x_t^{(i)} | \tilde{x}_{t-1}^{(i)}, y_t, \theta)$ and re-weighted again using another weights $w_t^{(i)}$, for $i = 1, 2, \dots, M$ particles. See Pitt and Shephard (1999) and Doucet, De Freitas, and Gordon (2001) for detailed explanation to the APF algorithm.

Algorithm 1 Bootstrap filter for the updating process θ

for s in $1:t$ **do**
  Calculate $W_s := w_{s-1} \times p(y_s|\hat{x}_s, \theta)$ (where $\hat{x}_s$ is a posterior sample of latent state in equation 2)
  for m in $1:M$ **do**
    Set $w_s^{(m,i)} := W_s^{(i)}$; $x_s^{(m,i)} := \hat{x}_s^{(i)}$ (the i^{th} posterior sample of latent state in equation (2))
  end for
end for
for s in $t+1:T$ **do**
  for m in $1:M$ **do**
    Generate $\tilde{x}_s^{(m)} \sim q(x_s|x_{t-1}^{(m)}, y_s)$

    Calculate unnormalised weight $w_s^{(m)} = \frac{f(\tilde{x}_s^{(m)}|x_{s-1}^{(m)})g(y_s|\tilde{x}_s^{(m)})}{q(x_s|x_{s-1}^{(m),y_s})}\pi_{s-1}^{(m)}$
  end for
  for m in $1:M$ **do**
    Normalize $w_s^{(m)}$ as $\pi_s^{(m)} := \frac{w_s^{(m)}}{\sum_{i=1}^{M} w_s^{(m)}}$
  end for
  for m in $1:M$ **do**
    Sample an index j from the set of $i, \ldots, M$ with probabilities $\{\pi_s^{(m)}\}_{m=1}^{M}$
    Set $x_s^{(m)} = \tilde{x}_s^{(m)}$
    $\pi_s^{(m)} = \frac{1}{M}$
  end for
  Calculate $\tilde{p}(y_{s|1:s-1}) = \frac{1}{M}\sum_{m=1}^{M} w_s^{(m)}$
end for

Algorithm (2) shows the changes made to the auxiliary particle filter algorithm implemented in `nimbleSMC` (Michaud et al. 2021).

Algorithm 2 Auxiliary filter for updating process

```
for s in 1 : t do
    Set θ = θ̂_r (where θ̂_r is a posterior sample of model parameters in equation 2)
    Calculate W_s := w_{s−1}^{(1)} × p(y_s | x̂_s, θ) ( where x̂_s the posterior sample of latent state in equation (2))
    for m in 1 : M do
        Set w_s^{(m)} := W_s; x_s^{(m)} := x̂_s (the posterior sample of latent state in equation (2))
    end for
end for
for s in t + 1 : T do
    Set θ = θ̂_upd (where θ̂_upd is a sample of model parameters for the updated process)
    for m in 1 : M do
        Generate x̃_{s|s−1}^{(m)} from either x̃_{s|s−1}^{(m)} ∼ f(x_s | x_{s−1}^{(m)}) or x̃_{s|s−1}^{(m)} = E(x_s | x_{s−1}^{(m)})

        Calculate p̂(y_s | x_{s−1}^{(m)}) = p(y_s | x̃_{s−1}^{(m)})

        Calculate unnormalised weight w_{s|s−1}^{(m)} = π_{s−1}^{(m)} p̂(y_s | x_{s−1}^{(m)})

    end for
    for m in 1 : M do
        Normalize w_{s|s−1}^{(m)} as π_{s|s−1}^{(m)} := w_{s|s−1}^{(m)} / Σ_{i=1}^{M} w_{s|s−1}^{(m)}
    end for
    for m in 1 : M do
        Sample an index j_m from the set of 1, ..., M with probabilities {π_{s|s−1}^{(m)}}_{m=1}^{M}
        Set x̃_{s−1}^{(m)} = x_s^{(j_k)}
        Generate x_s^{(m)} ∼ q(x_s | x̃_{s−1}^{(m)}, y_s)
        Calculate unnormalised weight w_s^{(m)} = f(x̃_s^{(m)} | x_{s−1}^{(m)}) g(y_s | x̃_s^{(m)}) / ( q(x_s | x_{s−1}^{(m)}, y_s) p̂(y_s | x_{s−1}^{(m)}) )
    end for
    for m in 1 : M do
        Normalise w_s^{(m)} as π_s^{(m)} := w_s^{(m)} / Σ_{i=1}^{M} w_s^{(m)}
    end for
    Calculate p̃(y_{s|1:s−1}) = ( (1/M) Σ_{m=1}^{M} w_s^{(m)} ) ( Σ_{m=1}^{M} w_{s|s−1}^{(m)} )
end for
```

3.3 Particle Markov Chain Monte Carlo

We now discuss how posterior the posterior distribution of model parameters are obtained. Using the Bayesian framework, the joint likelihood of the latent states and these parameters is $p(\theta, x_{1:t}|y_{1:t}) =$

$p_\theta(x_{1:t}|y_{1:t})p(\theta)$; where $p(\theta)$ is the prior distribution for the model parameters .

pMCMC (Andrieu, Doucet, and Holenstein 2010) makes it possible to jointly sample from the posterior distribution of the latent states and the model parameters θ. This algorithm first proposes a value for the model parameter ($\theta^\star$) from a proposal distribution ($\pi(\theta^\star|\theta)$) and fits SMC method described above with the proposed value $\theta^\star$. This proposed value is accepted or rejected based on a Metropolis Hasting acceptance ratio (MHAR) defined as:

$$\begin{aligned} MHAR &= \frac{p_\theta(y_{1:t})p(\theta^\star)\pi(\theta|\theta^\star)}{p_{\theta^\star}(y_{1:t})p(\theta)\pi(\theta^\star|\theta)}; \\ \text{where} \quad p_\theta(y_{1:t}) &= p_\theta(y_1)\prod_{t=2}^{T} p_\theta(y_n|y_{n-1}) \\ &= p_\theta(y_1)\prod_{t=2}^{T}\sum_{i=1}^{M} w^i_{t|\theta} \end{aligned} \tag{7}$$

where $p_\theta(y_{1:t})$ is the marginal distribution of the observed data given the parameter θ, $w^i_{t|\theta}$ is the i^{th} importance weight at time t given the value of θ, $p(\theta^\star)$ and $p(\theta)$ is the prior distribution of $\theta^\star$ and θ respectively and $\pi(.|.)$ is the proposal distribution for the parameters θ.

Adapting the MHAR defined in equation (7) and weights in equation (6) to our proposed framework, we propose $\theta^\star$ from $\pi(\theta^\star|\hat{\theta}_r)$ and the proposed parameter value ($\theta^\star$) is now accepted or rejected with an updated MHAR:

$$MHAR_{upd} = \frac{p(\theta^\star)\pi(\hat{\theta}_r|\theta^\star)\prod_{s=t+1}^{T}\sum_{i=1}^{M} w^i_{t|\theta^\star}}{p(\hat{\theta}_r)\pi(\theta^\star|\hat{\theta}_r)\prod_{s=t+1}^{T}\sum_{i=1}^{M} w^i_{t|\hat{\theta}_r}}; \tag{8}$$

where $w^i_{s|\theta}$ is the i^{th} importance weight at time t given the value of θ, $p(\theta^\star)$ and $p(\hat{\theta}_r)$ is the prior distribution of $\theta^\star$ and $\hat{\theta}_r$ respectively, $\pi(.|.)$ is the proposal distribution for the parameters θ and M is the number of parameters.

We implement this by making changes to the *random walk block sampler* in the `nimbleSMC` package (Michaud et al. 2021) as shown in Algorithm (3).

As discussed in the main paper, we proposed a Gibbs - Metropolis Hasting sampler for the models with more than five model parameters. To do so, we split the set of model parameters θ into two

Algorithm 3 Particle MCMC to sample model paramters

for i in $1 : n.iterations$ **do**

Run a particle filter using either Algoirithm (1) or (2) to estimate the marginal likelihood $\tilde{p}(y_{1:T}|\theta_r^{(i)})$

At the last time step T, draw $\hat{x}_{1:T} \sim p(x_{1:T}|y_{1:T}, \hat{\theta}_r^{(i)})$ from the full particle filter history.

Generate $\theta^\star \sim \pi(\theta|\hat{\theta}_r^{(i)})$

Run a particle filter with proposed SMC in Algoirithm (1) or (2) to estimate the marginal likelihood $\tilde{p}(y_{1:T}|\theta^\star)$

At the last time step T, draw $x^\star_{1:T} \sim p(x_{1:T}|y_{1:T}, \theta^\star)$ from the full particle filter history

Compute the Metropolis Hasting acceptance ratio in an equation (8) and choose $a^\star =$ min(1, MHARupd)

Generate $r \sim unif(0, 1)$

if $a^\star > r$ **then**

Set $\theta^{(i)} := \theta^\star$; $x^{(i)}_{1:T} := x^\star_{1:T}$

else

Set $\theta^{(i)} := \hat{\theta}_r^{(i)}$; $x^{(i)}_{1:T} := \hat{x}_{1:T}$

end if

end for

mutually exclusive sets θ_1 and θ_2. We assume that θ_2 depends on θ_1 and latent state distribution x. Therefore, employing the theory behind Gibbs sampling, we define a Metropolis Hastings block sampler to sample values of θ_1 and given the value(s) of θ_1, define a Metropolis Hastings block sampler to sample values of θ_2. We implement this by making changes to the *random walk block sampler* in the `nimbleSMC` package (Michaud et al. 2021) as shown in Algorithm (4):

References

Andrieu, Christophe, Arnaud Doucet, and Roman Holenstein. 2010. "Particle Markov Chain Monte Carlo Methods." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72 (3): 269–342.

Arulampalam, M Sanjeev, Simon Maskell, Neil Gordon, and Tim Clapp. 2002. "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking." *IEEE Transactions on Signal Processing* 50 (2): 174–88.

Doucet, Arnaud, Nando De Freitas, and Neil Gordon. 2001. "An Introduction to Sequential Monte Carlo Methods." *Sequential Monte Carlo Methods in Practice*, 3–14.

Michaud, Nicholas, Perry de Valpine, Daniel Turek, Christopher J Paciorek, and Dao Nguyen. 2021.

Algorithm 4 Particle MCMC using the proposed updated model via random walk block sampler

for i in $1 : n.iterations$ **do**

Run a particle filter using either Algoirithm (1) or (2) to estimate the marginal likelihood $\tilde{p}(y_{1:T}|\theta_r^{(i)})$

At the last time step T, draw $\hat{x}_{1:T} \sim p(x_{1:T}|y_{1:T}, \hat{\theta}_r^{(i)})$ from the full particle filter history.

Generate $\theta_1^\star \sim \pi(\theta_1|\hat{\theta}_{1|r}^{(i)})$

Compute the Metropolis Hastings acceptance ratio in equation (7) and choose $a^\star = \min(1, \text{MHAR})$

Generate $r \sim unif(0, 1)$

if $a^\star > r$ **then**

Set $\theta_1^{(i)} := \theta_1^\star$

else

Set $\theta_1^{(i)} := \hat{\theta}_{1|r}^{(i)}$

end if

Generate $\theta_2^\star \sim \pi(\theta_2|\hat{\theta}_{2|r}^{(i)})$

Set $\theta^\star := (\theta_1^{(i)}, \theta_2^\star)$

Run a particle filter using either Algoirithm (1) or or (2) to estimate the marginal likelihood $\tilde{p}(y_{1:T}|\theta^\star)$

At the last time step T, draw $x_{1:T}^\star \sim p(x_{1:T}|y_{1:T}, \theta^\star)$ from the full particle filter history

Compute the Metropolis Hasting acceptance ratio in equation (8) and choose $a^\star = \min(1, \text{MHARupd})$

Generate $r \sim unif(0, 1)$

if $a^\star > r$ **then**

Set $\theta^{(i)} := \theta^\star$; $x_{1:T}^{(i)} = x_{1:T}^\star$

else

Set $\theta^{(i)} := \hat{\theta}_r^{(i)}$; $x_{1:T}^{(i)} := \hat{x}_{1:T}$

end if

end for

"Sequential Monte Carlo Methods in the Nimble and nimbleSMC r Packages." *Journal of Statistical Software* 100: 1–39.

Pitt, Michael K, and Neil Shephard. 1999. "Filtering via Simulation: Auxiliary Particle Filters." *Journal of the American Statistical Association* 94 (446): 590–99.