

## Appendix S1: In-depth description of the sequential importance sampling algorithm used

This appendix details the sequential importance sampling algorithm used to obtain the results shown in Figures 2-5. Source code to implement model S3, written using the statistical computing language R Version 2.0.1 (R Development Core Team, 2004), and accompanying data are given in Appendix S2.

### Initializing the model parameters

For each particle, each model parameter was initialized by drawing at random from its prior distribution. The prior distributions are shown in Table A1.

**Table A1:** Prior distributions for the parameters used in the state-space models

Parameter	Prior distribution	Prior expectation
$\phi_a$	$\text{logit}(\phi_a) \sim \text{Norm}(\text{mean}=3.178054, \text{sd}=0.5208333)$	0.96
$\phi_{pmax}$	$\text{Beta}(\text{shape1}=14.53, \text{shape2}=6.23)$	0.70
$\gamma$	$\text{Beta}(\text{shape1}=22.05, \text{shape2}=1.15)$	0.95
$\delta$	$\text{Gamma}(\text{shape}=4, \text{scale}=0.025)$	0.10
$\beta_1$	$\text{Gamma}(\text{shape}=4, \text{scale}=0.0004814821)$	0.00193
$\beta_2$	$\text{Gamma}(\text{shape}=4, \text{scale}=0.001444446)$	0.00578
$\beta_3$	$\text{Gamma}(\text{shape}=4, \text{scale}=0.0005777786)$	0.00231
$\beta_4$	$\text{Gamma}(\text{shape}=4, \text{scale}=0.0001444446)$	0.000578
$\alpha_0, \alpha_1$ and $\alpha_2$	Prior distributions and variance-covariance structure obtained from the capture-recapture analyses for the various movement model forms	-

### Initialising the states

The priors for the states were generated using the pup production estimate for 1984 together with the priors for the parameters (Thomas *et al.* 2005). The initial number of

pups for each particle,  $n_{0,i,0}^{[k]}$ , was generated by “reversing” the observation equation, i.e. by sampling from

$$n_{0,i,0}^{[k]} \sim \text{Normal}\left(y_{0,i,0}, \delta^{2[k]} y_{0,i,0}^2\right) \quad (\text{eqn A.1})$$

where  $y_{0,i,0}$  is the estimated pup production at colony  $i$  in 1984 and  $\delta^{[k]}$  is the value of the CV parameter  $\delta$  sampled from the prior for particle  $k$ . These values were further dispersed by resampling them from a uniform distribution with bounds given by the sampled value  $\div 1.3$  and the sampled value  $\times 1.3$ . This was done to make sure all likely values for the initial states were contained in the sample (the value of 1.3 was chosen by Thomas *et al.* (2005) by trial and error). Initial values for the age 1 females were generated by assuming that half of the pups were female and sampling from

$$n_{1,i,0}^{[k]} \sim \text{Binomial}\left(n_{0,i,0}^{[k]}, 0.5\phi_{p,i,0}^{[k]}\right) \quad (\text{eqn A.2})$$

and ages 2 to 5 females from

$$n_{a,i,0}^{[k]} \sim \text{Binomial}\left(n_{a-1,i,0}^{[k]}, \phi_a^{[k]}\right) \quad (\text{eqn A.3})$$

where  $a = 2, \dots, 5$ , for the subscript on  $n$ . Again these values were further dispersed using a uniform distribution. To generate the initial distribution of adult females the breeding sub-process was “reversed” by sampling from

$$n_{6+,i,0}^{[k]} \sim \text{Negative Binomial}\left(n_{0,i,0}^{[k]}, \gamma^{[k]}\right) + n_{0,i,0}^{[k]} \quad (\text{eqn A.4})$$

and once again these values were further dispersed using a uniform distribution.

For model S4 in which no movement occurred the number of female seals at Fast Castle was initialised in 1997 (the first year of data for this colony) using the same methods as above. For the other three models (S1 to S3) this was not necessary as seals were able to move to this colony from the established colonies from 1990 onwards according to the various movement process models used.

### Sequential Importance Sampling algorithm

In the basic Sequential Importance Sampling (SIS) algorithm detailed in the Methods the likelihood,  $L_t^{[k]}$ , for particle  $k$  in step 3 is calculated as:

$$L_t^{[k]} = f_t(\mathbf{y}_t | \mathbf{n}_t^{[k]}, \Theta^{[k]}) = \prod_i \left( \frac{1}{\sqrt{2\pi} \delta^{[k]} n_{0,i,t}^{[k]}} \exp\left( \frac{-(y_{0,i,t} - n_{0,i,t}^{[k]})^2}{2\delta^{2[k]} n_{0,i,t}^{2[k]}} \right) \right) \quad (\text{eqn A.5})$$

where  $i = 1,3,4$ , for  $t = 1, \dots, 12$  and  $i = 1, \dots, 4$  for  $t = 13, \dots, 19$  and the superscript  $k$  represents the particle number,  $k = 1, \dots, 350,000$ .

The weight calculated in step 3 of the SIS algorithm for particle  $k$ ,  $w_t^{[k]}$ , and used in step 4 for resampling the particles is

$$w_t^{[k]} = \frac{L_t^{[k]}}{\sum_k L_t^{[k]}} \quad (\text{eqn A.6})$$

where  $k = 1, \dots, 350,000$ .

Four techniques were used to augment the basic SIS algorithm given in the Methods section with the aim of reducing the problem of particle depletion. These four techniques are detailed below and are followed by the augmented algorithm.

### *1. Rejection Control*

Rejection control is a technique to reduce the number of particles required to represent the distribution of the posterior at a given time point, without significantly increasing the Monte-Carlo error. It is described in detail by Liu (2001) and Liu, Chen & Wong (1998). We implemented rejection control in the second year of the SIS algorithm (the first year being used to initialize the states) because most of the particles generated from the priors are extremely unlikely given the data. In the second year particles with low likelihood were rejected probabilistically if their likelihood was less than a threshold of  $c$  (calculated as the 95<sup>th</sup> percentile of the likelihoods for the first run of the rejection controller). If the particle has a higher likelihood than the threshold it is accepted, if its likelihood is lower it is accepted with a probability equal to its likelihood divided by the threshold. Particles that survive, even though their likelihood  $< c$ , have their likelihood inflated to  $c$ . The discarded particles are replaced with new ones drawn from the original priors and states.

In this analysis rejection control was only implemented in the second year. It is, however, possible to incorporate other checkpoints during the time-series. With a lot of checkpoints, however, the algorithm can become very demanding in terms of computational time.

## 2. *Auxiliary Particle Filter*

Auxiliary particle filtering works by looking ahead in the data and concentrating in parts of the time-series where the model fit is expected to be particularly “good” (i.e. have high likelihood given the observation equation). The technique is described in detail by Liu & West (2001) and Pitt & Shepherd (1999).

The particles at time  $t-1$  are projected forward to time  $t$  deterministically according to their parameter values. The particles are then resampled according to their weights at time  $t$ , (i.e. those that are expected to be in an area of high density are given a higher weight at this stage than the others, the idea being here to sample from particles with high predictive likelihoods). These are the first weights. The parameter values of this new set of “auxilliary” particles are kernel smoothed (see 3, below) and their states are taken back to time  $t-1$  and projected forward stochastically. These particles are then resampled according to a second weight, which is proportional to the stochastic likelihood divided by the predictive (deterministic) likelihood for the particles – dividing by the deterministic likelihoods here corrects for having “looked ahead”. This causes those particles that do better than expected from the first filtering to get a higher final weight.

## 3. *Kernel Smoothing*

Kernel smoothing involves perturbing the parameter values of the particles, to generate new parameter values in the vicinity of parameter space supported by the data (this is akin to “mutation” in a Darwinian analogy). In this analysis a multivariate normal kernel, which smoothes out the parameters with respect to their variance-covariance plane, was used. This form of kernel smoothing preserves the first and second order moments of the parameters (i.e. their means and variances), and the correlations between them. Kernel smoothing of the states is not necessary, as these change stochastically as the particles are sent through the state process for each time step (Trenkel *et al.* 2000; Liu & West 2001; Thomas *et al.* 2005; Newman *et al.* in press).

Perturbing the parameters increases overall variation. One symptom is that those close to the edges of the variance-covariance plane will be pushed beyond its boundaries. To overcome this problem, the parameter values are shrunk back in towards their overall mean. Liu & West (2001) suggest that auxiliary particle filters should be used in combination with kernel smoothing.

Many parameters are bounded; in order to deal with this, parameters are transformed before kernel smoothing (probabilities getting a logit transform and parameters that can only be positive getting a log transform). After smoothing the parameters are then back-transformed.

The amount of kernel smoothing done is governed by a smoothing parameter  $a$ . In the work described here  $a$  was set to 0.7 (a relatively moderate amount of smoothing), the same value used by Thomas *et al.* (2005) in their SSM for the British grey seal

population and very close to the value used by Trenkel *et al.* (2000) in their SSM for red deer in Scotland.

Kernel smoothing makes the fitting algorithm more efficient and can compensate for poor choices of prior distributions and outliers in the data. However, as a result of the perturbations, some bias is introduced into the posterior estimates (Trenkel *et al.* 2000).

#### *4. Residual Resampling*

Residual resampling (Liu 2001) reduces the Monte-Carlo variance introduced during the resampling stage relative to simple random sampling with replacement. Samples are first drawn deterministically from the particles in the set and then the leftovers (the residuals) are drawn probabilistically. For example, in a set of 100 particles, a particle with a weight of 0.501 will be sampled 50 times deterministically. At the end of the deterministic stage there may be 1 particle left to make up the new set of 100 particles. The residual for the aforementioned particle is 0.001. The residuals for all the particles are then normalised to sum to 1, and the particle set is made up by sampling stochastically from these residuals. Residual resampling gives the same expected distribution of particles as standard stochastic resampling (Thomas *et al.* 2005) but gives smaller Monte Carlo variance and uses less computational time (Liu 2001). It was therefore used when resampling particles as part of the auxiliary particle filter.

#### *The augmented SIS algorithm*

The time points in the model are  $t = 0, \dots, T$ , where  $T = 19$ ,  $t = 0$  corresponds to 1984 and  $t = T$  corresponds to 2003. Step 1 of the augmented algorithm builds up a swarm of relatively probably particles by implementing rejection control at time point  $t = 1$ . For all subsequent time points we use an auxiliary particle filter with kernel smoothing of parameter values as detailed in steps 2-11.

**Step 1a Initialising:**  $K$  particles  $\{n_0^{[k]}, \Theta^{[k]}\}$  are drawn from  $g_0(n_0 | \Theta)$  and  $g_0(\Theta)$ , where  $k = 1, \dots, K$ .  $K = 350,000$ . Set  $t = 1$ .

**Step 1b Stochastic projection:** The particle swarm is projected forward stochastically through the state process model from time  $t-1$  to time  $t$  to give  $\{n_t^{[k]}, \Theta^{[k]}\}$ . A likelihood is calculated for each of the particles,  $L_t^{[k]}$ , given the data at time  $t$  and  $\{n_t^{[k]}, \Theta^{[k]}\}$ .

**Step 1c Rejection Control:** Accept particle  $k$  with probability

$$r^{[k]} = \min\left(1, \frac{L_t^{[k]}}{c}\right)$$

where  $c$  is the 95<sup>th</sup> percentile of the likelihoods obtained on the first iteration of the rejection controller. Repeat Steps 1a to 1c until  $K$  particles have been accepted. Denote this new set of particles  $\{n_t^{[k]*}, \Theta^{[k]*}\}$  and set their likelihoods to

$$L_t^{[k]*} = \max(L_t^{[k]}, c)$$

**Step 1d Correction:** Calculate a weight for each particle

$$w_t^{[k]*} = \frac{L_t^{[k]*}}{\sum_k L_t^{[k]*}}$$

**Step 1e** *Resampling:* Use these weights in residual resampling of the particles to obtain a new set of particles denoted  $\{n_t^{[k]**}, \Theta^{[k]**}\}$ .

**Step 1f** For  $k = 1, \dots, K$  redefine  $\{n_t^{[k]**}, \Theta^{[k]**}\} = \{n_t^{[k]}, \Theta^{[k]}\}$ .

**Step 1g** Increment  $t$  to  $t + 1$ .

**Step 2** *Deterministic projection:* The particle swarm is projected forward deterministically through the state process model from time  $t-1$  to time  $t$  to give  $\{n_t^{[k]}, \Theta^{[k]}\}$ .

**Step 3** Calculate a kernel location,  $m^{[k]}$ , for the parameter vector, a weighted combination of the particle and the mean vector for the parameter values across all particles,  $\bar{\Theta}$ :

$$m^{[k]} = a\Theta^{[k]} + (1-a)\bar{\Theta}$$

herein  $a = 0.7$ . Then calculate a likelihood for each of the particles,  $L_{\text{determin},t}^{[k]}$ , given the data at time  $t$  and  $\{n_t^{[k]}, m^{[k]}\}$ .

**Step 4** *Correction:* Calculate a first weight for each particle

$$w_{1,t}^{[k]} = \frac{L_{\text{determin},t}^{[k]}}{\sum_k L_{\text{determin},t}^{[k]}}$$

**Step 5** *Resampling:* Use the first weights in residual resampling of the particles to obtain a new set of “auxiliary” particles denoted  $\{n_t^{[k]*}, m^{[k]*}\}$ .

**Step 6** *Parameter mutation:* Kernel smooth the parameter values and denote the mutated parameters  $\Theta^{[k]*}$ :

$$\Theta^{[k]*} \sim \text{Multivariate Normal}(m^{[k]*}, h^2\mathbf{V})$$

where  $h^2 = 1 - a^2$  and  $\mathbf{V}$  is the variance-covariance matrix of the parameter vector  $\Theta^{[k]}$  prior to resampling.

**Step 7** *Stochastic projection:* Trace the particle swarm back to time  $t-1$  to give  $\{n_{t-1}^{[k]*}, \Theta^{[k]*}\}$ . Project these particles forward stochastically through the state process model to time  $t$ . A likelihood is then calculated for each of the particles,  $L_{\text{stoch},t}^{[k]*}$ , given the data at time  $t$  and  $\{n_t^{[k]*}, \Theta^{[k]*}\}$

**Step 8** *Correction:* Calculate a second weight for each particle

$$w_{2,t}^{[k]*} = \frac{\left( L_{\text{stoch},t}^{[k]*} / L_{\text{determ},t}^{[k]*} \right)}{\sum_k \left( L_{\text{stoch},t}^{[k]*} / L_{\text{determ},t}^{[k]*} \right)}$$

**Step 9** *Resampling:* Use residual resampling on the particles according to the second weights to get the final set of  $K$  particles for time  $t$  denoted  $\{n_t^{[k]**}, \Theta^{[k]**}\}$ .

**Step 10** For  $k = 1, \dots, K$  redefine  $\{n_t^{[k]**}, \Theta^{[k]**}\} = \{n_t^{[k]}, \Theta^{[k]}\}$ .

**Step 11** *Iteration:* If  $t < T$ , increment to  $t+1$ , go to step 2, and repeat.