# Efficient methods for fitting nonlinear non-Gaussian state space models of wildlife population dynamics

Fanny Empacher

A thesis submitted for the degree of PhD
at the
University of St Andrews

2024

## Abstract

State-space models (SSMs) are a popular and flexible framework for modelling time series due to their ability to separate changes in the underlying state of a system from the noisy observations made on these states. This thesis explores methods for estimating states and model parameters in non-linear and non-Gaussian Bayesian SSMs. We focus on models of wildlife population dynamics, in particular a case study of the UK grey seal population.

Calculation of the likelihood is fundamental to Bayesian analysis, but direct calculation is typically intractable for non-linear non-Gaussian SSMs. We use a class of simulation-based methods, Sequential Monte Carlo (SMC), which build on repeated importance sampling of simulated states to deliver an unbiased estimate of the likelihood. We find that variance of the estimated likelihood can be high and explore techniques for variance reduction.

For parameter inference, we use particle marginal Metropolis-Hastings (PMMH), which embeds the SMC likelihood within a Markov chain Monte Carlo (MCMC) algorithm. Careful balance is needed between computational effort expended on the SMC step and the number of MCMC samples.

A much faster alternative is the Kalman filter, designed for linear and Gaussian SSMs. We applied the Kalman filter to an approximation of the seal model. The posterior distribution obtained was often close to the true posterior, while reducing computation time by a factor of 1790.

We show the seal model suffers from identifiability issues which cannot be resolved by increasing the accuracy of the observations or allowing more flexibility in the underlying biological process with random effects. However, estimation of underlying states (i.e., population sizes) is unaffected by these issues.

A reduction in PMMH computation time can be achieved by exploiting the structure of the state model: separately estimating likelihood components in each of the 4 seal regions led to a 5-fold increase in speed.

## Candidate's declaration

I, Fanny Empacher, do hereby certify that this thesis, submitted for the degree of PhD, which is approximately 61,000 words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for any degree. I confirm that any appendices included in my thesis contain only material permitted by the 'Assessment of Postgraduate Research Students' policy.

I was admitted as a research student at the University of St Andrews in September 2017.

I received funding from an organisation or institution and have acknowledged the funder(s) in the full text of my thesis.

Date 28.03.2024 Signature of candidate

## Supervisor's declaration

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of PhD in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree. I confirm that any appendices included in the thesis contain only material permitted by the 'Assessment of Postgraduate Research Students' policy.

Date 28/3/2024 Signature of supervisor

## Permission for publication

In submitting this thesis to the University of St Andrews we understand that we are giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. We also understand, unless exempt by an award of an embargo as requested below, that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that this thesis will be electronically accessible for personal or research use and that the library has the right to migrate this thesis into new electronic forms as required to ensure continued access to the thesis.

I, Fanny Empacher, confirm that my thesis does not contain any third-party material that requires copyright clearance.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis:

**Printed copy**

No embargo on print copy.

**Electronic copy**

No embargo on electronic copy.

| | | |
|---|---|---|
| Date | 28.03.2024 | Signature of candidate |
| Date | 28/3/2024 | Signature of supervisor |

**Underpinning Research Data or Digital Outputs**

**Candidate's declaration**

I, Fanny Empacher, hereby certify that no requirements to deposit original research data or digital outputs apply to this thesis and that, where appropriate, secondary data used have been referenced in the full text of my thesis.

Date 28.03.2024 Signature of candidate

## Acknowledgements

This PhD has only been possible because of the support I received from numerous people.

First and foremost, thank you to my supervisors, Len Thomas and Ken Newman. They have been the best supervisor team I could have hoped for. Len supported me unreservedly through an eventful PhD, interrupted by two parental leaves, a move abroad and a pandemic. Even before my official starting date, he made a research trip to the Banff International Research Station in Oaxaca possible, where I attended a workshop on state space models, only weeks after I had first learned about them. Our discussions were a highlight of my week and, put simply, made research fun. And thank you to Ken, who provided perspective on my various ideas, always knew where to find the information I was missing, and who thoroughly read and commented on my thesis drafts.

Thank you to Chris Morris and Callan Duck for providing the pup production data and the photos of grey seals in Chapter 1, and to Debbie Russell for providing the independent estimate data and for patiently answering my questions about grey seals.

CREEM has been a fantastic research environment. From the start, everyone was friendly and supportive. The daily CREEM cake and coffee session often turned into a discussion about some statistical method, an animal I had never heard of or a heated debate about the differences between Marmite and Vegemite. There are a number of people I want to mention: Valentin Popov, who initially suggested the idea of a PhD at St Andrews to me; my cohort Andy Seaton, Filippo Franchini, Nseobong Uto, Rick Camp and Wei Jing, who made the APTS training sessions enjoyable; my office mates Richard Glennie and Felix Petersma.

A special thank you to Calliste Fagard-Jenkin, for wading through the jungle of SMC methods with me, for providing me with the output from his GPU code when code on a CPU was too slow, and for always sharing the results of his baking endeavours.

For making life in St Andrews so much richer: Gerard O'Reilly, Ashley Clayton, Finn Smith, who shared the ups and downs of doing a PhD with me; everyone at the Music Centre, the Opera Society, and the Music Society; the Postgraduate Society and Mizuki Morisake and Jen Bré in particular who shared their year in office with me.

Before starting my PhD studies, I was lucky enough to have the support from mentors at every stage of my academic journey. Among these are Friedrich Platz, who let me get away with writing a thesis in statistics for my music degree, and who first planted the idea of an academic career in my head. And Uta Freiberg, who gave me the advice that the best remedy for being slightly unhappy with a degree outcome is to simply go for the next and better one.

Thank you to my whole family and especially my parents, for encouraging me in my decision to pursue a PhD and for supporting me throughout.

Lastly, to my daughters Thea and Elinor, who have dance parties with me after I've sat in front of a screen for too long, who think it's pretty cool that I've written a book, and who make my life infinitely better every day; and to Julian, who has supported me with kindness, patience and practical help every step of the way, who believed more in me than I did, who was excited about every fuzzy caterpillar plot, and without whom by my side I would not have wanted to do this PhD journey.

## Funding Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council, via a Doctoral Training Partnership grant to the University of St Andrews; and by the University of St Andrews (School of Mathematics and Statistics).

# Abbreviations and Notation

| **Abbreviation** | **Description** |
|---|---|
| APF | auxiliary particle filter |
| BF | bootstrap filter |
| cdf | cumulative distribution function |
| CV | coefficient of variation |
| ESS | effective sample size |
| GPU | Graphical Processing Unit |
| IH | Inner Hebrides |
| i.i.d. | independent and identically distributed |
| IS | importance sampling |
| KF | Kalman filter |
| KFMH | Kalman filter within Metropolis-Hastings |
| log-L | log-likelihood |
| MCMC | Markov chain Monte Carlo |
| MCMC-DA | MCMC algorithm with data augmentation |
| MH | Metropolis-Hastings |
| MLE | maximum likelihood estimate |
| NDLM | normal dynamic linear model |
| NS | North Sea |
| OH | Outer Hebrides |
| Ork | Orkney |
| pdf | probability density function |
| PF | particle filter |
| PMCMC | particle MCMC |
| PMMH | particle marginal Metropolis-Hastings |
| RMSE | root mean square error |
| SMC | sequential Monte Carlo |
| SRMSE | scale root mean square error |
| SSM | state space model |

| **Notation** | **Description** |
|---|---|
| Cov | covariance |
| $\xrightarrow{d}$ | convergence in distribution |
| E | expected value |
| $f$ | transition pdf |
| $g$ | observation pdf |
| $i:j$ | range of integer values from $i$ to $j$ |
| $L$ | likelihood |
| $M$ | number of iterations of an MCMC algorithm |
| $N$ | number of particles in a bootstrap filter |
| $p$ | model pdf |
| $q$ | proposal pdf |
| $r$ | region |
| $\hat{R}$ | potential scale reduction |
| $x$ | state |
| $y$ | observation |
| $t$ | time |
| Var | variance |
| $\alpha$ | parameter for fecundity |
| $\beta$ | alternative carrying capacity parameter, inversely related to $\chi$ |
| $\chi$ | parameter for carrying capacity |
| $\omega$ | parameter for sex ratio |
| $\phi_a$ | parameter for adult survival probability |
| $\phi_{p,\max}$ | parameter for maximum pup survival probability |
| $\phi_{p,r,t}$ | pup survival probability in region $r$ and year $t$ |
| $\rho$ | parameter for density dependence shape |
| $\tau$ | parameter for observation precision |
| $\theta$ | parameter or parameter vector |

# Contents

**List of Figures** **viii**

**List of Tables** **xviii**

**1 Introduction** **1**

1.1 Ecological Motivation: Wildlife Population Dynamics . . . 1
1.1.1 Introduction to the grey seal case study . . . 2
1.1.2 Data . . . 6
1.2 Models . . . 8
1.2.1 Notation . . . 8
1.2.2 State Space Models . . . 9
1.2.3 Complete Seal Model . . . 10
1.2.4 Complete Seal Model Without Independent Estimate . . . 16
1.2.5 7-state model . . . 17
1.2.6 2-state model . . . 17
1.3 Inference for State Space Models . . . 18
1.3.1 Inference Methods . . . 19
1.4 Software . . . 20
1.5 Thesis Outline . . . 20

**2 Likelihood Estimation With SMC Methods** **23**

2.1 Introduction . . . 23
2.2 Review of SMC Methods for State Inference and Likelihood Estimation . . . 24
2.2.1 Importance Sampling . . . 24
2.2.2 Bootstrap Filter . . . 28
2.2.3 Bootstrap Filter as Likelihood Estimator . . . 31
2.2.4 Number of Particles . . . 33
2.2.5 Resampling Method . . . 34
2.2.6 Resampling Schedule . . . 35
2.2.7 Auxiliary Particle Filter . . . 36
2.2.8 Smoothing . . . 39
2.3 Simulation Study . . . 40
2.3.1 Simulation Methods . . . 40
2.3.2 Simulation Results and Interpretation . . . 45
2.4 Application to the 7-State Model . . . 52
2.4.1 Methods . . . 52
2.4.2 Results and Interpretation . . . 53
2.5 Application to the Complete Seal Model . . . 55
2.5.1 Methods . . . 55

2.5.2 Results and Interpretation . . . 57
2.6 Discussion and Conclusion . . . 58

**3 Parameter Inference** **61**
3.1 Introduction . . . 61
3.2 Review of SMC Methods for Parameter Inference . . . 61
3.2.1 Augmenting the unknown states with the parameters . . . 61
3.2.2 Particle MCMC and Variants . . . 67
3.2.3 SMC$^2$ . . . 78
3.3 Empirical Investigations of SMC Methods for Parameter Inference . . . 79
3.3.1 PMMH: Tuning the Proposal Distribution . . . 80
3.3.2 PMMH: Adapting the Number of Particles For Each Model . . . 84
3.3.3 PMMH: With Re-estimation . . . 92
3.3.4 SMC$^2$ for the Seal Model . . . 97
3.4 Analysis of Posterior Distribution . . . 103
3.5 Conclusion . . . 108

**4 Using the Kalman Filter to Approximate the Likelihood in Complex State-Space Models** **110**
4.1 Introduction . . . 110
4.2 Inference Methods . . . 111
4.2.1 Normal Dynamic Linear Models . . . 111
4.2.2 Kalman Filter . . . 111
4.2.3 Linearisation and Normalisation of the Seal Model . . . 113
4.2.4 Quality of the NDLM Approximation . . . 119
4.2.5 Algorithms for Parameter Estimation . . . 121
4.2.6 Assessment of Performance . . . 123
4.3 Simulation Study . . . 124
4.3.1 Simulation Scenarios . . . 124
4.3.2 Set-Up . . . 126
4.3.3 Results . . . 126
4.4 Application to Real Data . . . 135
4.4.1 Methods . . . 135
4.4.2 Results . . . 135
4.5 Improving the Linear Approximation . . . 141
4.5.1 Improving the Approximated Likelihood Estimate: The Unscented Kalman Filter . . . 141
4.5.2 Improving the approximated posterior with an SMC sampler . . . 145
4.6 Discussion and Conclusion . . . 146

**5 Effect of process and observation errors on inference in a nonlinear non-normal SSM** **147**
5.1 Introduction . . . 147
5.2 Methods . . . 149
5.2.1 Previous Results . . . 149
5.2.2 Changes to the Model . . . 150
5.2.3 Diagnostic Tools . . . 155
5.3 Results . . . 157
5.3.1 Changing Observation Variance . . . 157
5.3.2 Changing Process Stochasticity . . . 163

5.4 Conclusion . . . 167

**6 Factorising the Likelihood** **170**
6.1 Introduction . . . 170
6.2 Methods . . . 170
6.2.1 Factorisation of the Independent Estimate . . . 170
6.2.2 Incorporating the Independent Estimate in a Factorised Formulation 171
6.2.3 Assessing Gain in Efficiency . . . 174
6.3 Results . . . 175
6.3.1 Likelihood Estimation . . . 175
6.3.2 Posterior Distribution . . . 178
6.4 Discussion and Conclusion . . . 179

**7 Concluding Remarks and Further Research** **180**
7.1 Summary . . . 180
7.2 Potential Future Research . . . 181
7.3 Concluding Remarks . . . 182

**Bibliography** **183**

**A Chapter 1** **192**
A.1 Data . . . 192
A.2 Proofs . . . 193
A.2.1 Carrying Capacity . . . 193
A.2.2 Overdispersion of $x_0$ . . . 193

**B Chapter 2** **196**
B.1 Results for 2-State Model—Second Dataset . . . 196
B.2 Results for 7-State Model—Second Dataset . . . 201

**C Chapter 3: Additional Results** **203**

**D Chapter 4** **211**
D.1 First and second moments of $X_0$ in the seal model . . . 211
D.2 First and second moments of $X_t|X_{t-1}$ in the seal model . . . 213
D.3 Distribution of $Y$ when $Y|X \sim N(X, X^2/\tau)$ and $X \sim N(\mu, \sigma^2)$ . . . 214

**E Chapter 5: Additional Results** **216**

**F Chapter 6: Additional Results** **217**

# List of Figures

1.1 Female grey seal with newborn pup. Photo: Chris Morris, SMRU. . . . . . 3
1.2 Grey seals breeding on the beach on Stroma, in the Pentland Firth. Photos: Chris Morris, SMRU. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
1.3 Map showing the location of grey seal colonies in the UK and Isle of Man. Regularly monitored colonies, which are the ones included in the analysis undertaken in this thesis, are colour coded by region: Inner Hebrides (cyan), Outer Hebrides (pink), Orkney (blue) and North Sea (central – orange, south – red; these were combined in the analysis reported here). Image reproduced from Thomas et al. (2019, Supplementary Materials). . . . . . . 5
1.4 Regional pup production data from 1984 to 2010 (see Table S1 in Thomas et al., 2019). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
1.5 Beverton-Holt density dependent pup survival for different values of $\beta$ and $\rho$. The maximum pup survival probability $\phi_{p,\max}$ here is 1. . . . . . . . . . 12
1.6 Prior distributions for $\phi_{p,\max}$, $\phi_a$ and $\alpha$ (red lines) as induced by the Beverton-Holt density dependence. The black lines indicate the original independent prior distributions. . . . . . . . . . . . . . . . . . . . . . . . . . . 17
1.7 Paired density plots of the priors for $\phi_{p,\max}$, $\phi_a$ and $\alpha$ as induced by the Beverton-Holt density dependence. The correlation coefficients between the induced prior distributions are $\rho_{\phi_a,\phi_{p,\max}} = -0.312$, $\rho_{\phi_{p,\max},\alpha} = -0.024$ and $\rho_{\alpha,\phi_a} = -0.107$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

2.1 Means of 100 estimated means of $\mathcal{U}(0,4)$ through importance sampling and self-normalised importance sampling with proposal distribution $\mathcal{N}(0,1)$ for a varying number of particles. The shaded area shows the region between the 10th and 90th percentile. . . . . . . . . . . . . . . . . . . . . . . . . . . 27
2.2 Histogram of weighted samples of sample size 10000 from $\mathcal{U}(0,4)$ generated from two different normal distributions as proposal distribution, with the density of their proposal distribution superimposed. . . . . . . . . . . . . . 28
2.3 Histogram of estimated means of $\mathcal{U}(0,4)$ through self-normalised importance sampling with 5000 particles for two different proposal distribution. Note the very different x-axis scales on the two plots. . . . . . . . . . . . . . . 29
2.4 Likelihood estimates $\hat{p}(y_{1:t}|\theta)$ for the toy example given in Equations 2.12 with true parameter $\theta = 0.9$ (indicated by the black vertical line). The likelihood was estimated with a bootstrap filter (BF, red lines) 5 times at each parameter value with 100 particles each. The true likelihood was calculated with a Kalman filter (KF, blue line). . . . . . . . . . . . . . . . . . . 32

2.5 Histogram of 10,000 estimates $\hat{p}(y_{1:t}|\theta)/p(y_{1:t}|\theta)$ for two values of $\theta$. The estimates $\hat{p}$ were produced by a bootstrap filter with 100 particles. The mean of the estimates is indicated with a blue vertical line (■). . . . . . . . . 33
2.6 Histograms of 10,000 likelihood estimates $\hat{p}(y_{1:t}|\theta)/p(y_{1:t}|\theta)$ for $\theta = 1.5$ for numbers of particles $N$. The mean of the estimates is indicated with a blue vertical line (■). Note the difference in x-axis scale. . . . . . . . . . . . . . . 34
2.7 Two sets of pup and adult numbers used in the simulation study, generated using the 2-state model with the true parameter in Table 2.1. Observations of pup numbers are displayed as blue circles and the pup carrying capacity is indicated by a black horizontal line. . . . . . . . . . . . . . . . . . . . . . . . 42
2.8 Measures of variation of 100 likelihood estimates of three different parameter values in the 2-state model computed by a standard bootstrap filter with varying numbers of particles $N$. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates. . . . . . . . . . . . . . . . . . . . . . . . . . . 46
2.9 Boxplots of 100 likelihood estimates of three different parameter values in the 2-state model for varying numbers of particles $N$. . . . . . . . . . . . . 47
2.10 Bootstrap filter runtime per particle for varying number of particles $N$. . . 47
2.11 Measures of variation of 1000 likelihood estimates of two parameters in the 2-state model computed by a bootstrap filter with varying ESS thresholds $\text{ESS}_{\text{min}}/N$ for resampling. False parameter 2 is not included in the plot because the values are too large and show no discernible trends. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates. . . . . . . . . . . . 48
2.12 Boxplots of 1000 log-likelihood estimates of three different parameter values in the 2-state model comparing a bootstrap filter (BF) and an auxiliary particle filter (APF) using the expected value in the adjustment multiplier. 49
2.13 Measures of variation of 1000 likelihood estimates of three different parameter values in the 2-state model computed by an auxiliary particle filter using tempered weights with varying exponents in the resampling step. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates. . . . . . . . . . . . 51
2.14 Boxplots of 1000 log-likelihood estimates computed by an auxiliary particle filter using tempered weights with varying exponents in the resampling step, for three different parameter values in the 2-state model. . . . . . . . . . . . 51
2.15 Two sets of simulated counts from the 7-state model. Only numbers of pups and adult females aged at least 6 years are shown. Observations of pup numbers are shown as blue circles and the pup carrying capacity is indicated with a black horizontal line. . . . . . . . . . . . . . . . . . . . . . . . 53
2.16 Measures of variation of 100 likelihood estimates of the 7-state model computed by a standard bootstrap filter with varying numbers of particles $N$. The first figure gives the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates. . . . . . . . 54
2.17 Boxplots of 100 likelihood estimates of the 7-state model for varying numbers of particles $N$ at three different parameter values. . . . . . . . . . . . . 54

2.18 Measures of variation of 100 likelihood estimates computed by a standard bootstrap filter with varying numbers of particles $N$. The likelihood was estimated at the true parameter value for the joint formulation and the factorised formulation of the simplified full model, using simulated data, and at the posterior mean for the real full model using real data. The first figure gives the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates. . . . . . . . . 57
2.19 Effective sample sizes of the particles at each iteration from 3 runs of a bootstrap filter with $N = 1000$ particles, run with the complete seal model using real data with the independent estimate, with the parameter set to the posterior mean. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58

3.1 Decrease of unique parameter values in the bootstrap filter with augmented state algorithm. Observations were simulated from the model in Equation 3.1 with $\theta = 0.9$ (black horizontal line) and $T = 30$. The top plot shows the number of unique parameter values over time. The bottom plot shows the values of the remaining parameter values, with the size and color indicating how many copies of this values are present in the sample. . . . . . . . . . . 63
3.2 The acceptance rate and effective sample size per second when running the PMMH for the 2-state model with $N = 3$ particles and $M = 10,000$ iterations while varying the scale $h$ of the covariance of the proposal distribution. 74
3.3 Trace plots for estimating $p(\theta|y = 1)$ with three different MCMC methods. The chains have length $M = 1000$. The probability of sampling $z = 1$ is set to $c = 0.1$. The correct posterior probability for $\mathbb{P}(\theta = 0.2|y = 1)$ is 0.2. . . 77
3.4 The effective sample size per second when running the PMMH for the 2-state model while varying the number of particles $N$ the scale $h$ of the covariance of the proposal distribution. . . . . . . . . . . . . . . . . . . . . . . . 81
3.5 ESS/sec for the 2-state model using the PMMH algorithm with two proposal distributions. The number of particles $N$ and the covariance scale $h$ of the proposal distribution were varied. Blue lines represent results with a multivariate normal distribution on the original parameter scale, while red lines show results after transforming parameters to the real line. Missing values indicate non-convergence. . . . . . . . . . . . . . . . . . . . . . . . . . . 82
3.6 Trace plots of the last 25% of iterations from running 2 chains with the PMMH for the parameter $\phi_a$, with $h = 1$ and $N = 30$. . . . . . . . . . . . . 83
3.7 Results generated by running the PMMH for the 2-state model with 2 chains with $M = 100,000$ iterations each. The scaling factor $h$ was set to 0.8 and the number of particles was $N = 30$ for estimating the likelihood. The runtime was 945.6s$\approx$ 15.76min, resulting in an effective sample size of 3932.0. 85
3.8 The effective sample size per second when running the PMMH for the 7-state model while varying the number of particles $N$ the scale $h$ of the covariance of the proposal distribution. . . . . . . . . . . . . . . . . . . . . . . . 86
3.9 Results generated by running the PMMH for the 7-state model with 2 chains with $M = 300,000$ iterations each. The scaling factor was set to $h = 0.8$ and the number of particles was $N = 10$. The runtime was 1982.53s$\approx$ 33.0min, resulting in an effective sample size of 13073.27. . . . . . . . . . . . . . . . . . 87

3.10 Relationship between the number of particles $N$ and ESS per runtime when sampling from the posterior distribution of the complete seal model without independent estimate. The proposal distribution is a multivariate normal distribution on the transformed parameter space with a covariance $h = 0.8$ times the posterior covariance. . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
3.11 Results generated by running the PMMH for the complete seal model without independent estimate with 2 chains with $M = 300,000$ iterations each. The scaling factor was set to $h = 0.8$ and the number of particles was $N = 100$. The runtime was 13920.59s≈ 3.87h, resulting in an effective sample size of 1389.41. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
3.12 Relationship between the number of particles $N$ and ESS per runtime when sampling from the posterior distribution of the complete seal model with independent estimate using the real data. The proposal distribution is a multivariate normal distribution on the transformed parameter space with a covariance $h = 0.8$ times the posterior covariance. . . . . . . . . . . . . . . . 90
3.13 Trace plots and density plots from the sample generated from the PMMH with two different values of $N$, with two chains each. The first plot shows the trace plot of the parameter for adult survival, $\phi_a$, and the second shows the log-likelihood estimate at each iteration. The third plot shows a plot of the estimated posterior density. . . . . . . . . . . . . . . . . . . . . . . . . . . 91
3.14 Plot of the four marginal posterior densities for all 6 parameters, as estimated by running a PMMH four times. The two red lines show the estimated posterior density when the likelihood is not re-estimated, the two blue lines show the same when the likelihood is re-estimated at every iteration. The black vertical line indicates the true parameter value with which the data were simulated. The black density line indicates the prior distribution. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
3.15 Plot of the two marginal posterior densities for all 10 parameters, as estimated by running a PMMH. The red lines shows the estimated posterior densities when the likelihood is not re-estimated, the blue lines show the same when the likelihood is re-estimated at every iteration. The black density line indicates the prior distribution. . . . . . . . . . . . . . . . . . . . . . . . 96
3.16 ESS of the parameter particles through the iterations $t$ of the SMC$^2$. The total number of parameter particles was $N_\theta = 30,000$ and a resampling step was triggered when the ESS fell below $3,000$. The red lines indicate the iterations where resampling ocurred. . . . . . . . . . . . . . . . . . . . . . . 97
3.17 Inter-quartile range and median of the distributions $p(\theta|y_{1:t})$ through time $t$ for all six parameters of the 2-state model, as estimated by the SMC$^2$ algorithm. The black dots represent the inter-quartile range and median of the prior distribution. The vertical red lines indicate the iterations where a resampling step occurred. The horizontal line indicates the true parameter value that was used to simulate the data. . . . . . . . . . . . . . . . . . . . . . 98
3.18 Estimated densities of all six parameters in the 2-state model from 12 runs of the SMC$^2$ algorithm in blue. The black vertical line shows the parameter values used to simulate the data. The black density line indicates the prior distribution. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 100

3.19 Inter-quartile range and median of the distributions $p(\theta|y_{1:t})$ through time $t$ from one of the 20 runs for all ten parameters of the 28-state model with independent estimate, using the real data, as estimated by the SMC$^2$ algorithm. The black dots represent the inter-quartile range and median of the prior distribution. The vertical red lines indicate the iterations where a resampling step occurred. The horizontal line indicates the posterior mean estimated in Thomas et al. (2019). . . . . . . . . . . . . . . . . . . . . . . . . 101
3.20 Results from running the SMC$^2$ 20 times for the complete seal model. For each of the runs the number of parameter particles was $N_\theta = 1,000$ and the number of state particles for each parameter was $N_x = 3,000$. The resampling step for the parameter particles was triggered when the ESS fell below 100. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 102
3.21 Density plots of the marginal posterior densities for all 10 parameters, generated by running the PMMH for the complete seal model with independent estimate with $N = 3,000$ particles and 20 chains with $M = 1,000,000$ iterations each. The scaling factor for the covariance of the proposal distribution was set to $h = 0.8$. The black density line indicates the prior distribution. For $\phi_{p,\max}, \phi_a$ and $\alpha$, the dashed line indicates the theoretical marginal prior and the solid line indicates the prior induced by the Beverton-Holt density dependence. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 104
3.22 Posterior correlation coefficients of the 10 parameters of the complete seal model, generated from running the PMMH with $N = 3,000$ particles and 20 chains with $M = 1,000,000$ iterations each. . . . . . . . . . . . . . . . . . 106
3.23 Smoothed posterior state estimates of pup counts in all 4 regions and of total adult population size, produced by the PMMH. The red dashed lines show the 95% credible interval and the red solid line shows the mean. The black dashed and solid lines show the same with the estimates from Thomas et al. (2019). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107

4.1 Histogram of 10,000 samples of $Y$ where $Y|X \sim \mathrm{N}(X, X^2)$ and $X \sim \mathrm{N}(0,1)$ 117
4.2 Density plots of $y_{30}|y_{1:29}$ (generated from 100,000 samples) for fixed parameter values ($\phi_{p,\max} = 0.7, \phi_a = 0.9, \alpha = 0.9, \chi = 2500, \rho = 6, \tau = 100$) and 12 different simulations of $y_{1:29}$ in the 7-state model (🟥) and its NDLM approximation (🟦). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119
4.3 Filtered estimates of number of pups $x_t|y_{1:t}$ for $t = 0, \ldots, 30$ with fixed parameter values ($\phi_{p,\max} = 0.7, \phi_a = 0.9, \alpha = 0.9, \chi = 2500, \rho = 6, \tau = 100$) and 12 different simulations of $y_{1:30}$ (●) with a bootstrap filter (🟥) and with a Kalman filter using the NDLM approximation (🟦). . . . . . . . . . . . . . . 120
4.4 Boxplot of the differences between the filtered estimates of number of pups $x_t|y_{1:t}$ for $t = 0, \ldots, 30$, produced by the Kalman filter (KF) applied to the NDLM approximation and a particle filter (BF), with fixed parameter values ($\phi_{p,\max} = 0.7, \phi_a = 0.9, \alpha = 0.9, \chi = 2500, \rho = 6, \tau = 100$) for 1000 different simulations of $y_{1:30}$. . . . . . . . . . . . . . . . . . . . . . . . . . . 121
4.5 Observation density of the independent estimate for fixed $x_{t=24}$ in the 7-state model (🟥) and its NDLM approximation (🟦). . . . . . . . . . . . . . . . . 122
4.6 Simulated trajectories of pup numbers (🟦) and annual observations (🟦) in Scenarios 0 through 3. The values in Scenario 4 (estimating only one parameter) were simulated in the same way as in Scenario 0 (baseline) and are not shown here. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 125

4.7 Scenario 0: posterior distributions of all six parameters for simulated observations in Scenario 0 (baseline) with the 2-state model, estimated by the Particle Marginal Metropolis-Hastings algorithm (PMMH), a MCMC with data augmentation (MCMC) algorithm and the KFMH algorithm. The parameter used for the simulation is indicated by the black vertical line, and the prior distribution is drawn as a black curve. . . . . . . . . . . . . . . . . . 127
4.8 Scenario 0: correlations between the parameters in the posterior distributions in Scenario 0 (baseline) with the 2-state model, as estimated by (from left to right) the MCMC-DA algorithm, the PMMH algorithm and the KFMH algorithm. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 127
4.9 Scenario 1:Posterior distributions of all six parameters for simulated observations in Scenario 1 (uninformative data) with the 2-state model, estimated by the Particle Marginal Metropolis-Hastings algorithm (PMMH), an MCMC with data augmentation algorithm and the KFMH algorithm. The parameter used for the simulation is indicated by the black vertical line, and the prior distribution is drawn as a black curve. . . . . . . . . . . 129
4.10 Scenario 1: correlations between the parameters in the posterior distributions in Scenario 1 (uninformative data) with the 2-state model, as estimated by (from left to right) an MCMC with data augmentation algorithm, the PMMH algorithm and the KFMH algorithm. . . . . . . . . . . . . . . . . 129
4.11 Scenario 2: posterior distributions of all six parameters for simulated observations in Scenario 2 (edge of parameter support) with the 2-state model, estimated by the Particle Marginal Metropolis-Hastings algorithm (PMMH), an MCMC with data augmentation algorithm and the KFMH algorithm. The parameter used for the simulation is indicated by the black vertical line, and the prior distribution is drawn as a black curve. . . . . . . . . . . 130
4.12 Scenario 2: correlations between the parameters in the posterior distributions in Scenario 2 (edge of parameter support) with the 2-state model, as estimated by (from left to right) an MCMC with data augmentation algorithm, the PMMH algorithm and the KFMH algorithm. . . . . . . . . 131
4.13 Scenario 3: posterior distributions of all five parameters for simulated observations in Scenario 3 (mis-specified model) with the 2-state model, estimated by an MCMC with data augmentation algorithm and the KFMH algorithm. The density plot from the PMMH algorithm is not included, because this algorithm never converged. The prior distribution is drawn as a black curve. There is no vertical line to indicate the true parameter value, as the data were simulated from a different model. . . . . . . . . . . . . . . 132
4.14 Scenario 3: correlations between the parameters in the posterior distributions in Scenario 3 (mis-specified model) with the 2-state model, as estimated by (from left to right) an MCMC with data augmentation and the KFMH algorithm. The areas of the circles are proportional to the absolute value of corresponding correlation coefficients. . . . . . . . . . . . . . . . . . . 133
4.15 Scenario 4: posterior distributions of the parameter $\phi_a$ for simulated observations in Scenario 4 (estimating only one parameter) with the 2-state model, estimated by an MCMC with data augmentation algorithm and the KFMH algorithm. The prior distribution is drawn as a black curve and the true value used to simulate the observation is indicated with a black vertical line. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 134

4.16 Without $y_{IE}$: posterior distributions of all nine parameters for simulated observations in the complete seal model without the independent estimate, estimated by the KFMH algorithm and the PMMH algorithm. The prior distribution is drawn as a black curve. The black vertical lines indicate the posterior mean that was obtained by Thomas et al. (2019) with the Liu-West algorithm. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 136
4.17 Without $y_{IE}$: correlations between the parameters in the posterior distributions in the complete seal model without the independent estimate, as estimated by the KFMH algorithm and the PMMH algorithm. . . . . . . . 138
4.18 With $y_{IE}$: posterior distributions of all ten parameters in the complete seal model using the real data with the independent estimate, estimated by the KFMH algorithm and the PMMH algorithm. The prior distribution is drawn as a black curve. The black vertical lines indicate the posterior mean that was obtained by Thomas et al. (2019) with the Liu-West algorithm. . . 139
4.19 With $y_{IE}$: correlations between the parameters in the posterior distributions in the complete seal model with the independent estimate, as estimated by the KFMH algorithm and the PMMH algorithm. . . . . . . . . . 141
4.20 Marginal posterior distributions of all six parameters for simulated observations in Scenario 0 (baseline), estimated by the unscented Kalman filter (UKF) and compared with the estimated posterior distributions by the PMMH, the MCMC-DA and the KFMH algorithm. The parameter used for the simulation is indicated by the black vertical line, and the prior distribution is drawn as a black curve. . . . . . . . . . . . . . . . . . . . . . . . . 144

5.1 Posterior (blue) and induced prior (black) densities for the 2-state model. The true parameter values used to simulate the data are indicated by the vertical red line (🟥), the prior means by a vertical line in black (⬛), and the posterior means by a vertical line in blue (🟦). For $\phi_a$, the red line indicating the true parameter value is hidden behind the black vertical line. 148
5.2 Simulated pup and adult numbers (🟥 line) with circles for the observations with a CV of 10% (🟦), 1% (🟦) and 0.5% (🟦) . . . . . . . . . . . . . . . . . . . 152
5.3 Distributions of $\alpha$ in different model formulations. 🟪 shows the prior density in the seal model $\alpha \sim 0.6 + 0.4\text{Beta}(2, 1.5)$, 🟩 shows the prior induced by the Beverton-Holt density dependence. 🟥 shows the distribution of $\alpha$ as a random effect, with $\alpha \sim \text{Beta}(a, b)$ with priors $a \sim \text{Ga}(40, 0.1)$ and $b \sim \text{Ga}(40, 0.4)$. 🟦 shows the distribution of $\alpha$ as a random effect $\alpha \sim \text{Beta}(a, b)$ when the shape parameters are fixed to $a = 400$ and $b = 100$. 154
5.4 Density of the distribution of the random effect on $\alpha$ when $\alpha \sim \text{Beta}(a, b)$ with $a$ and $b$ as in Table 5.7. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 154
5.5 Simulated trajectories of pup and adult numbers with and without (🟥) a random effect on the fecundity $\alpha$. For the two simulations with a random effect, the parameters of the Beta distribution for $\alpha$ were chosen such that the mean of $\alpha$ was 0.8, and the standard deviation was 0.03 (🟦) or 0.2 (🟩). 155
5.6 Marginal posterior distributions and means (vertical lines) for the 2-state model, estimated using observations with three different CVs. The marginal prior distributions and means are indicated in black. The true value used to simulate the data is shown as the vertical black dashed line (hidden behind the red vertical line for $\chi$). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 158

5.7 Samples from the prior and posterior distributions for the 2-state model, estimated using observations with three different CVs. The mean of each joint distribution is indicated with a black circle, other than for the prior of $\chi$, which is outside the plot limits. The true value used to simulate the data is shown as a black cross. For the parameter $\chi$, the axis limits were restricted to focus on the joint posterior distributions, resulting in an omission of some of the sampled values from the prior distribution. . . . . . 160
5.8 Pearson correlation coefficient between the four parameters for the prior distribution and the posterior distributions estimated using observations with three different CVs. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 161
5.9 Smoothed posterior state estimates of both pups and adults for the three different observation variances. The white circles show the observed pup and adult counts. In the second row, both axis limits are reduced to better illustrate the reduction in variance of the pup state estimates as the CV decreases. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
5.10 Standard deviations of the smoothed state estimates across time. The standard deviations are shown for both the adult and pup estimates and for all three observation CVs. The solid line shows the standard deviations when the parameters are sampled from the posterior distributions, and the dashed line shows it when the parameter is fixed to the true value. Note the change in y-axis scale between the three plots. . . . . . . . . . . . . . . . . . . . . . . . 162
5.11 Marginal posterior distributions for the 2-state model, using simulated state values with 5 different standard deviations for the random effect on $\alpha$. The marginal prior distribution is indicated in black. The induced marginal prior distributions for $\phi_{p,\max}$ and $\phi_a$ are indicated in yellow. The true value used to simulate the data is shown as the vertical solid black line. . . . . . . 164
5.12 Boxplots of posterior means from 50 simulations for each of the 5 values for the random effect standard deviation. The black dashed line indicates the true parameter value used to simulate the data, the black solid line indicates the mean of the prior distribution. . . . . . . . . . . . . . . . . . . . . 164
5.13 Samples from the prior and posterior distributions for the 2-state model, using simulated state values with 5 different standard deviations for the random effect on $\alpha$. The mean of each joint distribution is indicated with a black circle. The true value used to simulate the data is shown as a black cross. For the parameter $\chi$, the axis limits were restricted to focus on the joint posterior distributions, resulting in an omission of some of the sampled values from the prior distribution. . . . . . . . . . . . . . . . . . . . . . . . . . 165
5.14 Boxplots of the posterior correlation coefficients $\rho$ using 50 simulations each, for each of the 5 different standard deviations for the random effect on $\alpha$ and for each of the three parameter pairs. . . . . . . . . . . . . . . . . . . . . . 166
5.15 Smoothed posterior state estimates of both pups and adults, using simulated state values with 5 different standard deviations for the random effect on $\alpha$. The white circles show the observed pup and adult counts and the black lines indicate the true counts. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 167
5.16 Boxplots (🟥) of the scaled RMSEs of the smoothed posterior state estimates from 50 simulations with 5 different random effect standard deviations. The boxplots in blue (🟦 show the scaled RMSEs when the same observations are used but parameters are fixed to the true values.) . . . . . . . . . . . . . . 168

6.1 Boxplots of 100 likelihood estimates of four different parameter values (see Table 6.1) in the complete seal model with independent estimate for varying numbers of particles $N$ and for the two different model formulations. . . . . 176
6.2 Measures of variation of 100 likelihood estimates of four different parameter values (see Table 6.1, Parameters 1 to 4 correspond to $\theta_1$ to $\theta_4$) in the complete seal model with independent estimate for varying numbers of particles $N$ and for the two different model formulations. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates. . . . . . . . . . . . . . . . . . . 177
6.3 Relationship between the number of particles $N$ and ESS per runtime when using the PMMH to generate samples from the posterior distribution of the complete seal model with independent estimate, comparing the joint formulation (blue line) with the factorised formulation (red line). The proposal distribution is a multivariate normal distribution on the transformed parameter space with a covariance $h = 0.8$ times the posterior covariance. . . 178

B.1 Measures of variation of 100 likelihood estimates computed by a standard bootstrap filter with varying numbers of particles $N$ at three different parameter values. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 197
B.2 Boxplots of 100 likelihood estimates for varying numbers of particles $N$ at three different parameter values. . . . . . . . . . . . . . . . . . . . . . . . . . 197
B.3 Measures of variation of 1000 likelihood estimates computed by a bootstrap filter with varying ESS threshold for resampling. False parameter 2 is not included in the plot because the values are too large and show no discernible trends. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates. . . . . . 198
B.4 Boxplots of 1000 log-likelihood estimates comparing a bootstrap filter (BF) and an auxiliary particle filter (APF) using the expected value in the adjustment multiplier, for three different parameter values. . . . . . . . . . . . 198
B.5 Measures of variation of 1000 likelihood estimates computed by an auxiliary particle filter using tempered weights with varying exponents in the resampling step. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 199
B.6 Boxplots of 1000 log-likelihood estimates computed by an auxiliary particle filter using tempered weights with varying exponents in the resampling step, for three different parameter values. . . . . . . . . . . . . . . . . . . . . . . . 199
B.7 Measures of variation of 100 likelihood estimates of the 7-state model computed by a standard bootstrap filter with varying numbers of particles $N$. The first figure gives the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates. . . . . . . . . 202
B.8 Boxplots of 100 likelihood estimates of the 7-state model for varying numbers of particles $N$ at three different parameter values. . . . . . . . . . . . . 202

C.1 Density plots of the marginal posterior densities for all 10 parameters, generated by running the PMMH for the complete seal model with independent estimate with $N = 3$ particles and 2 chains with $M = 10,000,000$ iterations each. The scaling factor for the covariance of the proposal distribution swas set to $h = 0.8$. The black density line indicates the prior distribution. . . . 205
C.2 Density plots of the marginal posterior densities for all 10 parameters, generated by running the PMMH for the complete seal model with independent estimate with $N = 3,000$ particles and 2 chains with $M = 100,000$ iterations each. The scaling factor for the covariance of the proposal distribution was set to $h = 0.8$. The black density line indicates the prior distribution. . 206
C.3 Posterior correlation coefficients of the 10 parameters of the complete seal model, as generated by running the PMMH with two different number of particles $N$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 207
C.4 Scatter plots and joint densities of the posterior distribution for the complete seal model using the real data, generated by running the PMMH for the complete seal model with independent estimate with $N = 3,000$ particles and 20 chains with $M = 1,000,000$ iterations each. . . . . . . . . . . . 210

E.1 Smoothed posterior state estimates of both pups and adults when the true parameter value is used in combination with adult and pup observations with a varying CV. The white circles show the observed pup and adult counts. In the second row, both axis limits are reduced to better illustrate the reduction in variance of the pup state estimates as the CV decreases. . 216

# List of Tables

1.1 Prior distributions and summary statistics for the parameters of the complete seal model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

1.2 Definition of names for relevant distributions for inference in state space models. Note that the likelihood is not technically a distribution when considered a function of $\theta$ but is featured in this table for convenience. This table is modified from a table in Schön and Lindsten (2017). . . . . . . 18

2.1 Parameter vectors used in the simulation study . . . . . . . . . . . . . . . . . 41

2.2 Choices for each of the four simulations . . . . . . . . . . . . . . . . . . . . 43

2.3 Measures of variation of 100 likelihood estimates computed by a standard bootstrap filter with varying numbers of particles $N$. The likelihood was estimated at the true parameter value and at two false parameter values (defined in Table 2.1). The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 30,000$ particles. . . . . . . . . . . . . . . . . . . . . . 45

2.4 Measures of variation of 1000 likelihood estimates computed by a bootstrap filter with varying ESS thresholds $\text{ESS}_{\text{min}}/N$ for resampling. The likelihood was estimated at the true parameter value and at two false parameter values for the 2-state model. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. . 48

2.5 Measures of variation of 1000 likelihood estimates of three different parameter values in the 2-state model computed by a bootstrap filter (BF) and by an auxiliary particle filter (APF) using the expected value in the adjustment multiplier. The likelihood was estimated at the true parameter value and at two false parameter values. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49

2.6 Measures of variation of 1000 likelihood estimates computed by an auxiliary particle filter using tempered weights with varying exponents in the resampling step. The likelihood was estimated for the 2-state model at the true parameter value and at two false parameter values. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. . . . . . . . . . . . . . . . . . . . . . . . . . . 50

2.7 Parameter vectors used in the simulation study for the 7-state model . . . . 52

2.8 Measures of variation of 100 likelihood estimates of the 7-state model computed by a standard bootstrap filter with varying numbers of particles $N$. The likelihood was estimated at the true parameter value and at two false parameter values. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 30,000$ particles. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
2.9 Measures of variation of 100 likelihood estimates computed by a particle filter with varying numbers of particles $N$. The likelihood was estimated at the true parameter value for two different formulations of the simplified full model, using simulated data, and at the posterior mean for the real full model using real data. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 30,000$ particles. . . . . . . . . . . . . . . . . . . . . . . 57

3.1 Transformations from different domains to $\mathbb{R}$ and back . . . . . . . . . . . . 64
3.2 Results from running the PMMH for the 2-state model with $N = 3$ particles and $M = 10,000$ iterations while varying the scale $h$ of the proposal distribution. For $h = 0.01$, the chains did not converge. . . . . . . . . . . . 74
3.3 Results from running the PMMH for the complete seal model without independent estimate with relatively constant runtime, while varying the number of particles $N$. The proposal distribution is a multivariate normal distribution on the transformed parameter space with a covariance $h = 0.8$ times the posterior covariance. The ESS for $N = 3$ was not calculated due to its high $\hat{R}$ value. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
3.4 Results from running the PMMH for the complete seal model with independent estimate and with the real data with relatively constant runtime, while varying the number of particles $N$. The proposal distribution is a multivariate normal distribution on the transformed parameter space with a covariance $h = 0.8$ times the posterior covariance. The ESS for $N = 30000$ was not calculated due to its high $\hat{R}$ value. . . . . . . . . . . . . . . . . . . . . 88
3.5 Settings that led to the highest observed ESS per runtime for each of the four models when running a PMMH with a multivariate normal proposal distribution with covariance $h\Sigma_{\theta|y_{1:T}}$. $N$ stands for the number of particles to estimate the likelihood, $h$ is the scaling factor of the proposal covariance matrix and $M$ stands for the number of iterations of the MCMC. . . . . . . 92
3.6 Results from running the PMMH for the 2-state model, comparing the effect of re-estimating the likelihood term in the MH acceptance rate. The algorithms settings were the ones that proved the best choices in Section 3.3.1, so $h = 0.8, N = 30, M = 1,000,000$. Each of the two choices was repeated to determine whether any differences were only due to Monte Carlo error. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
3.7 Percentages of overlapped area of the marginal densities for each parameter, measuring the shared area for both the two runs with re-estimation, the two runs without re-estimation, and finally the shared area when these two versions are compared. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94

3.8 Results from running the PMMH for the complete seal model with independent estimate, comparing the effect of re-estimating the likelihood term in the MH acceptance rate. The algorithms settings were the ones that led to the two best results in Section 3.3.2.3, so $h = 0.8$, $N = 3$ and $N = 3,000$ with $M$ adapted to ensure comparable computation times. . . . . . . . . . . 95
3.9 Percentages of overlapped area of the marginal densities for each parameter, measuring the shared area between the run when the likelihood was re-estimated at every iteration and the run when this was not done. The number of particles was $N = 3,000$. . . . . . . . . . . . . . . . . . . . . . . . . . 95
3.10 Range and mean of percentages of overlapped area of the marginal densities for each parameter from 12 different runs each for the SMC$^2$ and the PMMH algorithm. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 99
3.11 Results from running the PMMH for the complete seal model with independent estimate and with the real data, using $N = 3,000$ particles and 20 chains with $M = 1,000,000$ iterations each. The algorithm settings are as determined in Section 3.3.2.3. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103
3.12 Mean (with standard deviation in parentheses) for the marginal prior and posterior distributions of the 10 parameters in the complete seal model. For the parameters $\phi_{p,\max}, \phi_a$ and $\alpha$, the induced prior was used to calculate the prior summary statistics and overlap percentages. The previous posterior statistics are the ones given in Thomas et al. (2019) using Algorithm 4. . . 105

4.1 Scenario 0: measures to assess the convergence speed of the three algorithms in Scenario 0 (baseline) with the 2-state model. For each entry, first the value in replicate 1 is given and then the range of values across all 5 replicates. 128
4.2 Scenario 1: measures to assess the convergence speed of the three algorithms in Scenario 1 (uninformative data) with the 2-state model. For each entry, first the value in replicate 1 is given and then the range of values across all 5 replicates. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the effective sample size by ESS. . . . . . . . . . . . . . . . 130
4.3 Scenario 2: measures to assess the convergence speed of the three algorithms in Scenario 2 (edge of parameter support) with the 2-state model. For each entry, first the value in replicate 1 is given and then the range of values across all 5 replicates. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the effective sample size by ESS. . . . . . . . . . 131
4.4 Scenario 3: measures to assess the convergence speed of the three algorithms in Scenario 3 (mis-specified model) with the 2-state model. For each entry, first the value in replicate 1 is given and then the range of values across all 5 replicates. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the effective sample size by ESS. . . . . . . . . . . . . . . . 133
4.5 Scenario 4: measures to assess the convergence speed of the three algorithms in Scenario 4 (estimating only one parameter) with the 2-state model. For each entry, first the value in replicate 1 is given and then the range of values across all 5 replicates if there was some variation. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the effective sample size by ESS. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 134
4.6 Without $y_{IE}$: measures to assess the convergence speed of the KFMH algorithm and the PMMH algorithm applied to the complete seal model without independent estimate. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the multivariate effective sample size by ESS. . . 135

4.7 Without $y_{IE}$: mean and standard deviation of the posterior distributions for the complete seal model without the independent estimate, as estimated by the PMMH algorithm and the KFMH algorithm. . . . . . . . . . . . . . 137
4.8 With $y_{IE}$: measures to assess the convergence speed of the KFMH algorithm and the PMMH algorithm applied to the complete seal model with independent estimate. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the mutlivariate effective sample size by ESS. . . 138
4.9 With $y_{IE}$: mean and standard deviation of the posterior distributions for the complete seal model with the independent estimate, as estimated by the PMMH algorithm, the KFMH algorithm, and the Liu-West algorithm (Thomas et al., 2019). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 140
4.10 Measures to assess the convergence speed of the UKF in Scenario 0 (baseline), compared to the MCMC-DA, PMMH and Kalman filter. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the effective sample size by ESS. . . . . . . . . . . . . . . . . . . . . . . . . . 144

5.1 Means (with standard deviations in parentheses) of the posterior and induced prior distributions of the 2-state model, as well as the true parameter values used to simulate the data. . . . . . . . . . . . . . . . . . . . . . . . . 147
5.2 Parameter values of $a$ and $b$ for a Gamma distribution for $\alpha$, with mean 0.8 and varying standard deviation. . . . . . . . . . . . . . . . . . . . . . . . . 154
5.3 Summaries from running the PMMH for the 2-state model with four unknown parameters with three different observation CVs. . . . . . . . . . . . 157
5.4 Means (with standard deviations in parentheses) of the prior distribution of the 2-state model, as well as the the posterior distributions estimated using observations with three different CVs and the true parameter values used to simulate the data. . . . . . . . . . . . . . . . . . . . . . . . . . . . 158
5.5 Empirical cumulative distribution function of the marginal prior and three posterior distributions, evaluated at the true parameter value. . . . . . . . . 159
5.6 Scaled root mean squared error (SRMSE) of the state estimates compared to the true state values. The SRMSE was calculated for all three posterior distributions, as well as when the true parameter value is assumed to be known and the three different sets of observations are used. . . . . . . . . . 161
5.7 Details from fitting the 2-state model with a random effect on $\alpha$ with varying variance. ESS, $\hat{R}$, and runtime are given for one randomly selected run each.163

6.1 Parameter vectors used for the likelihood estimation . . . . . . . . . . . . . . 175
6.2 Measures of variation of 100 likelihood estimates computed by a particle filter with varying numbers of particles $N$. The likelihood was estimated with a particle filter at the posterior mean ($\theta_4$ in Table 6.2), using the factorised and the joint formulation. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 30,000$ particles. . . . . . . . . . . . . . . 175
6.3 Results from running the PMMH for the complete seal model with independent estimate, using the factorised formulation for the likelihood estimation. The algorithms settings were the ones that led to the two best results in Section 3.3.2.3. The number of particles $N$ for the likelihood estimation was varied between $N = 3$ and $N = 3,000$ with the number of iterations $M$ of the Markov chain adapted to ensure comparable computation times. . 178

A.1 Regional pup production estimates from 1984 to 2010. In 2009, there is a missing value for the Inner Hebrides due to too few aerial surveys in that region. The data was provided by the Sea Mammal Research Unit, University of St Andrews, and is published in Table S1 in Thomas et al. (2019). . . . 192
A.2 Parameters of the shifted Gamma distribution used in Thomas et al. (2019) to model the uncertainty in the independent estimate derived by Russell et al. (2016). . . . 193

B.1 Measures of variation of 100 likelihood estimates computed by a standard bootstrap filter with varying numbers of particles $N$ for the 2-state model. The likelihood was estimated at the true parameter value and at two false parameter values (defined in Table 2.1). The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 30,000$ particles. . . . 196
B.2 Measures of variation of 1000 likelihood estimates computed by a bootstrap filter with varying ESS threshold for resampling. The likelihood was estimated at the true parameter value and at two false parameter values for the 2-state model. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. . . . 200
B.3 Measures of variation of 1000 likelihood estimates of three different parameter values in the 2-state model computed by a bootstrap filter (BF) and by an auxiliary particle filter (APF) using the expected value in the adjustment multiplier. The likelihood was estimated at the true parameter value and at two false parameter values. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. . . . 200
B.4 Measures of variation of 1000 likelihood estimates computed by an auxiliary particle filter using tempered weights with varying exponents in the resampling step. The likelihood was estimated for the 2-state model at the true parameter value and at two false parameter values. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. . . . 200
B.5 Measures of variation of 100 likelihood estimates computed by a standard bootstrap filter with varying numbers of particles $N$for the 7-state model. The likelihood was estimated at the true parameter value and at two false parameter values. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 30,000$ particles. . . . 201

C.1 Results from running the PMMH for the 2-state model with constant $NM$, while varying the number of particles $N$ and the scaling factor $h$. The proposal distribution is a multivariate normal distribution with a covariance proportional to the posterior covariance. . . . 204

C.2 Results from running the PMMH for the 2-state model with constant $NM$, while varying the number of particles $N$ and the scaling factor $h$. The proposal distribution is a multivariate normal distribution on the transformed parameter space with a covariance proportional to the (transformed) posterior covariance. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 208
C.3 Results from running the PMMH for the 7-state model with constant $NM$, while varying the number of particles $N$ and the scaling factor $h$. The proposal distribution is a multivariate normal distribution with a covariance proportional to the posterior covariance. . . . . . . . . . . . . . . . . . . . . . 209

F.1 Measures of variation of 100 likelihood estimates computed by a particle filter with varying numbers of particles $N$. The likelihood was estimated with a particle filter at the prior mean (Parameter 1 in Table 6.1), using the factorised and the joint formulation. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 100,000$ particles. . . . . . . . . . . . . . . . 217
F.2 Measures of variation of 100 likelihood estimates computed by a particle filter with varying numbers of particles $N$. The likelihood was estimated with a particle filter at Parameter 2 in Table 6.1, using the factorised and the joint formulation. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 100,000$ particles. . . . . . . . . . . . . . . . . . . . . . . 218
F.3 Measures of variation of 100 likelihood estimates computed by a particle filter with varying numbers of particles $N$. The likelihood was estimated with a particle filter at Parameter 3 in Table 6.1, using the factorised and the joint formulation. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 100,000$ particles. . . . . . . . . . . . . . . . . . . . . . . 218

# Chapter 1

# Introduction

This thesis is about methods for fitting wildlife population dynamics models to data, in particular non-linear non-Gaussian state space models in a Bayesian framework. We do this through the case study of a model for the UK grey seal (*Halichoerus grypus*) population. The fitting method that is currently used for this model is computationally intensive and produces biased inference. We explore different alternatives to this fitting method by using sequential Monte Carlo (SMC) methods (Chapters 2, 3 and 6) and by employing an approximation to the model for fast inference (Chapter 4). In addition, using one of these fitting methods, we explore how different sources of uncertainty in the model affect posterior distributions (Chapter 5).

In this chapter, we first give an overview of the purposes of wildlife population dynamics modelling and introduce the case study of the UK grey seal population that is central to this thesis (Section 1.1). We then describe state space models, which form the framework for all models in this thesis; we also define the particular state space model used for grey seal population dynamics and introduce three simplifications to this model that will be used throughout this thesis (Section 1.2). The next section (Section 1.3) is devoted to inference for state-space models. We discuss some of the distributions that are commonly of interest for these models and summarise the available methods to carry out inference. Section 1.4 gives an overview of the software that can be used for inference and describes what was used in this thesis. Finally, we present an outline of the remainder of the thesis (Section 1.5).

## 1.1 Ecological Motivation: Wildlife Population Dynamics

Population dynamics models are tools to study the development of wildlife populations as dynamical systems. The work by Malthus (1798) is often regarded as the beginning of the field. Malthus stated that a population in an unchanging environment will grow exponentially, and the simple exponential growth model with $x_t = x_0 e^{rt}$, where $x_t$ is the population size at time $t$ and $r$ is the population growth rate, is often called the Malthusian growth model (Turchin, 2001). The modelling tools in this thesis are typically used for larger vertebrate species, whose population growth can often be modelled as an annual cycle and exhibits modest process stochasticity, i.e., modest year-to-year variation in changes of population size. That distinguishes these populations from other animals, e.g., insects, where other tools such as stochastic differential equations are commonly used (Soulsby and Thomas, 2012). The types of models considered here can model size and age composition of a population by incorporating information about the animals such as survival, ageing and growth, birth, and sex assignment (Newman et al., 2014d). It is possible to allow annual variation in the processes, e.g., by including external environmental factors in the model and treating them as covariates that the model parameters depend upon, or by placing random effects on parameters. These models can also describe populations at different sites and their interaction, and even track multiple species, to model, e.g., the competition between two species or their prey-predator relationship.

Modelling population dynamics can have many purposes. Newman et al. (2014c) mention some examples, such as managing fish stocks and setting levels of take (anthropogenic mortality) for a sustainable population, reintroducing a locally extinct animal such that the new population is viable, managing populations that have grown beyond the environment's capacity and cause a deterioration of their habitat, and exploring scenarios for reversing a population decline by testing the effect of various interventions on the abundance predicted by a model.

These different purposes require similar information about the population in question. Often, abundance is the primary emphasis and previous and current population sizes need to be estimated from noisy observations. It might also be of interest to predict the future development of the population, and to obtain estimates not just for the total population size but also for a structured population, namely subsets grouped by age, size, sex or location (Newman et al., 2014b). Next, it is often necessary to understand the processes that drive these dynamics, both as a means to an end when abundance is estimated, or as a question of interest in its own right, for example to know which parameters to target when seeking to make a conservation intervention. This can mean determining likely ranges of parameters, such as survival probabilities, for a fixed model or comparing several models with each other to assess which of them best explains the observed population numbers.

### 1.1.1 Introduction to the grey seal case study

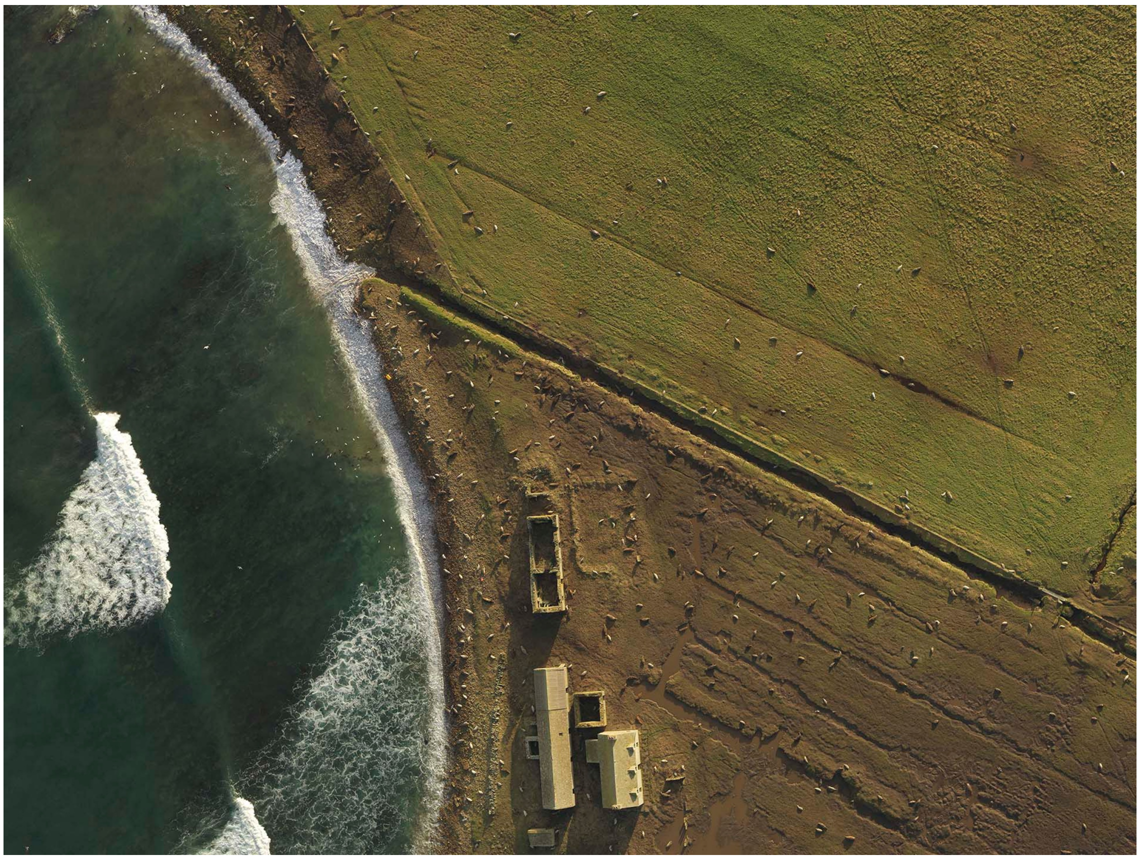
Grey seals (*Halichoerus grypus*) are a member of the family Phocidae or true seals. They inhabit parts of the North Atlantic, North Sea, Baltic Sea and Arctic Ocean and feed on various fish species, cephalopods and sand eels, and occasionally even on larger animals such as harbor seals (van Neer et al., 2015). Adult male seals weigh up to 300kg and adult females up to 180kg, with lengths of up to 200cm and 180cm respectively (Duck, 2007). Male seals can live for over 20 years and female seals for over 30 years, beginning to breed at around age 5 (Special Committee on Seals, 2021). The breeding cycle is 12 months long and consists of a 3-4-month period between conception and implantation of the fertilized egg, and an 8-month gestation period after which females give birth to a single pup. For breeding, females aggregate in large colonies, typically in the UK on remote beaches and uninhabited islands but increasingly also on beaches that are visited by the public (Hall and Russell, 2018). Figure 1.2 shows an aerial photo of a breeding colony and the map in Figure 1.3 shows the distribution of the breeding colonies in the UK. Newborn pups are fed by their mothers for approximately 2-3 weeks (the lactation period), after which they are abandoned. The breeding season occurs between September and December, starting in the south-west and moving clockwise around the UK. Mating takes place towards the end of the lactation period (Russell et al., 2019) with some males mating with up to 10 females while others are excluded. Figure 1.1 shows a female grey seal with a newborn pup. Grey seals are capital breeders: lactating mothers fast throughout the weaning period and rely on fat stores accumulated through the year to provision their offspring (Houston et al., 2006). During this time, their mass reduces significantly. Smout et al. (2020) estimated

**Figure 1.1:** Female grey seal with newborn pup. Photo: Chris Morris, SMRU.

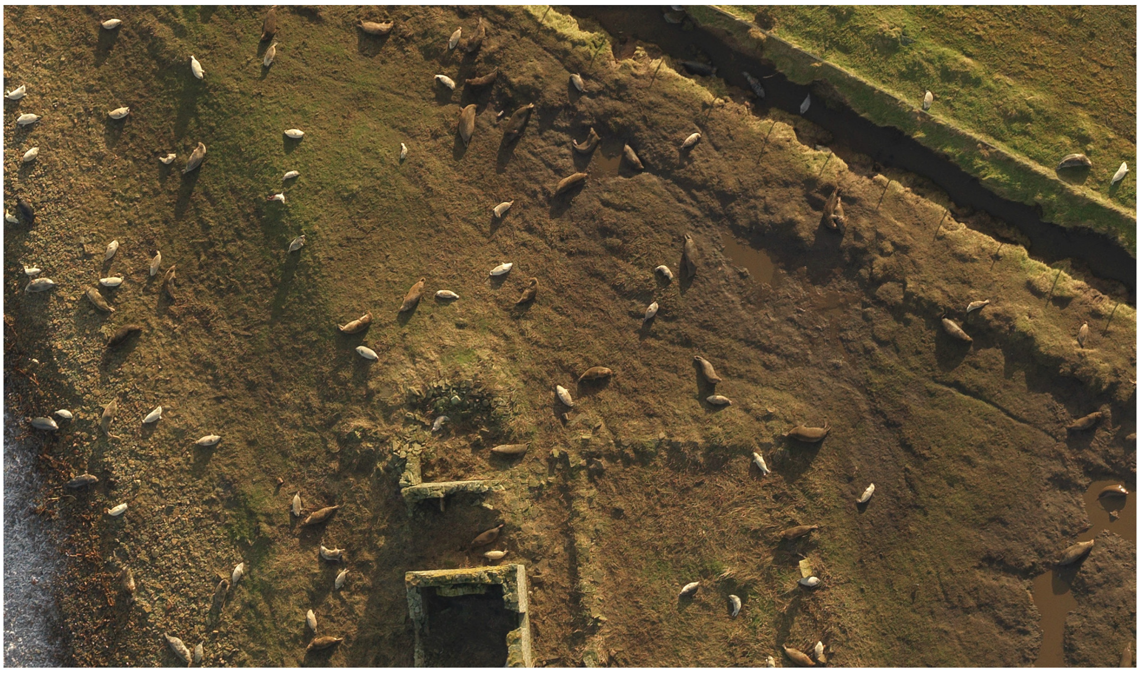
the mean loss of maternal mass to be 45%, using a sample of 584 adult females on North Rona, Outer Hebrides, and 273 adult females on the Isle of May, Firth of Forth. They also found that low body mass after weaning and poor environmental conditions negatively affected the probability of breeding in the following year, while skipping breeding one year increased the probability of subsequent breeding. Pups are born weighing between 11 and 20 kg and initially stay on land in breeding colonies for a 15-21 day-long lactation period where they increase their weight up to 40 kg. A fasting period lasting around 21 days follows which is suspected to be related to diving ability before leaving the colony (Hall and Russell, 2018). After the breeding season, seals disperse and spend most of the time at sea for foraging trips lasting several days, with shorter periods in-between spent hauled out ashore (Lonergan et al., 2011). These haul-out sites are much larger in number and more widely dispersed than the breeding sites (Russell et al., 2016).

In Special Committee on Seals (2021) an estimate of the total world-wide pup production is given as 191,270. The ratio of the number of adult seals to pups is not known exactly but estimated to be between 3 and 4.5. The UK grey seal population makes up around 35% of the global population abundance. The other major populations are in the western Atlantic (57%) and Europe excluding the UK (8%) (Special Committee on Seals, 2021).

The conservation management of grey seals in the UK has an eventful history. In 1914, after a decline of the population due to commercial and subsistence hunting, they became the first mammal protected by an act of Parliament (Grey Seals Protection Act 1914) due to concerns raised by sportsmen who suspected the total number of seals to be below 500



**(a)** Full image.



**(b)** Close up of grey seals, including pups.

**Figure 1.2:** Grey seals breeding on the beach on Stroma, in the Pentland Firth. Photos: Chris Morris, SMRU.

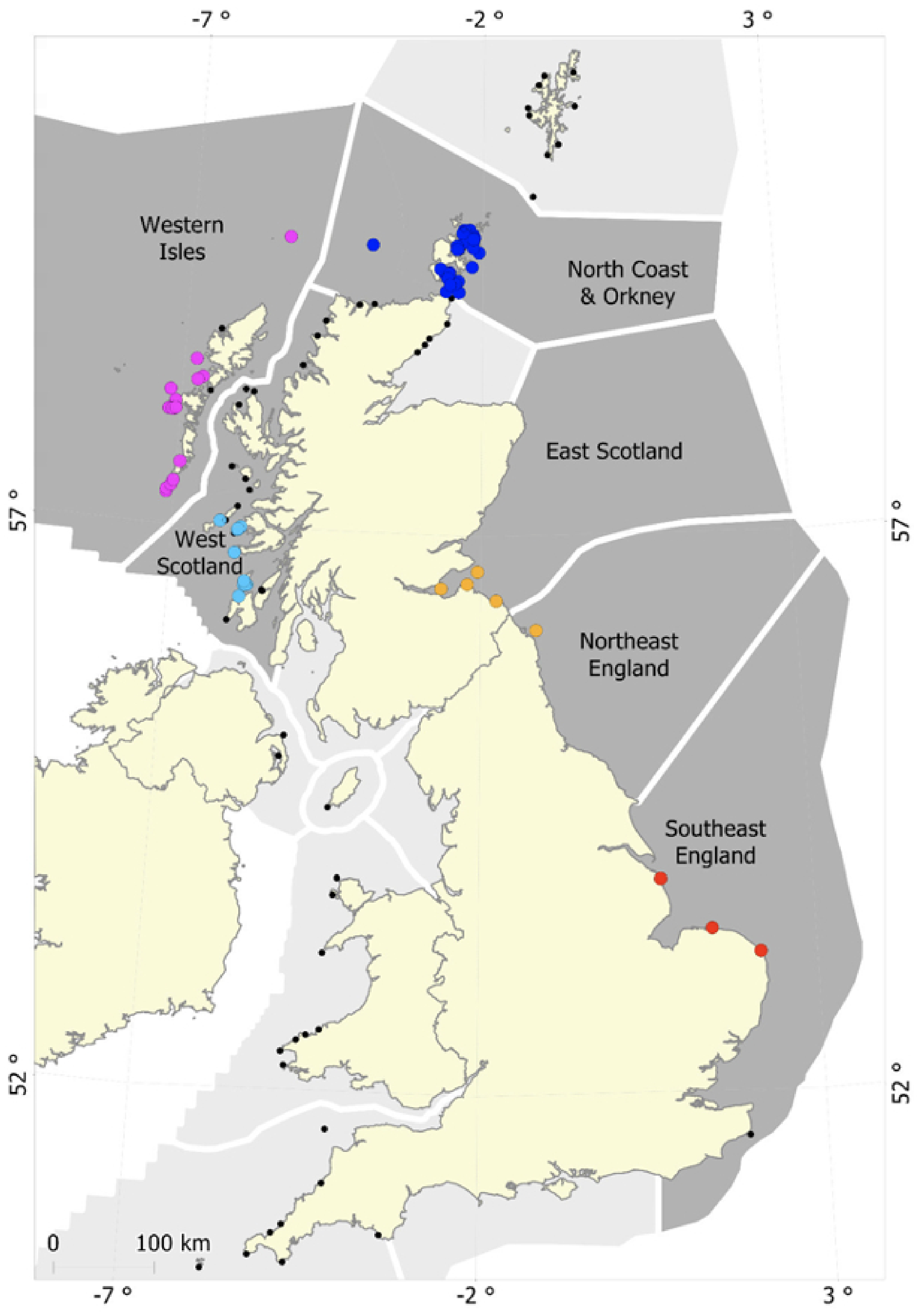


**Figure 1.3:** Map showing the location of grey seal colonies in the UK and Isle of Man. Regularly monitored colonies, which are the ones included in the analysis undertaken in this thesis, are colour coded by region: Inner Hebrides (cyan), Outer Hebrides (pink), Orkney (blue) and North Sea (central – orange, south – red; these were combined in the analysis reported here). Image reproduced from Thomas et al. (2019, Supplementary Materials).

(although this number is now estimated to have been between 2,000 and 4,000, Lambert, 2002). The Act established a closed season for hunting during the breeding period which was extended to the whole year in 1932 and helped reverse the decline of the population. Increasing population numbers led to complaints by fishermen in the 1950s that grey seals damaged fishing gear and reduced fish stock abundances. Eventually, culls were undertaken in the 1960s and 1970s with the goal of reducing the population to levels that would be satisfactory both from a conservation management and from a commercial fishing perspective. These were accompanied by public protests, most noteably a protest led by Greenpeace with much support of the British public in 1978, which eventually led to an abandonment of the cull (Lambert, 2002).

To improve the scientific evidence to support further conservation management efforts, in 1977 the government founded the Seal Mammal Research Unit (SMRU), managed from within the Natural Environment Research Council. The Unit's task was to provide advice to government for managing seal populations, as specified in the 1970 Conservation of Seals Act. It was initially based within in the British Antarctic Survey in Cambridge, but in 1996 moved to a purpose-built facility at the University of St Andrews. The advice is now given by a committee of international experts, the Special Committee on Seals (SCOS) and is informed by scientific information provided by SMRU (Unit, 2019). Early work in Harwood and Prime (1978) used age structures of adults seals, estimated from a sample of shot seals during culls at the Farne Islands to infer the total number of seals from pup counts. The population was then growing nearly exponentially at an observed growth rate in pup numbers of 7% per year, due to recovery from the historical human exploitation. Since populations have a stable age structure under exponential growth (Iannelli and Milner, 2017), it was relatively straightforward to estimate the total population size from the number of pups born each year. However, the population growth slowed noticeably in some of the regions, requiring more sophisticated modelling. A new approach for modelling the seal population was described in Thomas and Harwood (2003). This forms the basis of the model used today, although the model has been updated and modified since then, with an extensive update given in Thomas et al. (2019).

A disadvantage of the modelling approach of Thomas et al. (2019) is that it is computer-intensive to fit. With the current fitting algorithm, it takes around 3 days running using 20 parallel processes to produce the results given there. Even with this high computational effort, some concessions have to be made with approximations that introduce bias into the results (see Section 3.2.1.1). Improving the fitting algorithm is therefore desirable, both to eliminate the bias and to reduce the computational effort. This would also allow the exploration of some enhancements to the model to better reflect the data. In this thesis, we explore various alternatives to the current algorithm.

### 1.1.2 Data

Directly counting the number of adult seals is difficult since they are widely dispersed, except for during the moulting and breeding season, and so monitoring is undertaken by counting pups on breeding colonies—the pups are concentrated in a small number of colonies and are always present because they cannot yet swim. This was done through high resolution aerial surveys at most colonies, supplemented by ground-based counts at a few more accessible and smaller colonies. A picture showing grey seals, including pups, that was taken as part of these surveys is shown in Figure 1.2 and a map of the monitored colonies is shown in Figure 1.3. The breeding period is longer than the stay of an individual pup and so statistical models were used to estimate the total number of pups born from

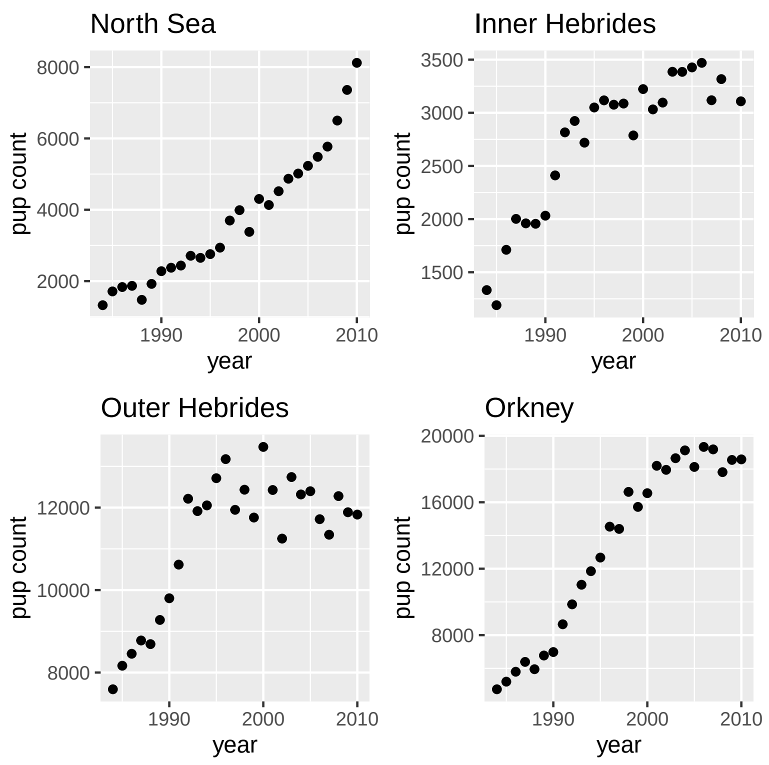


**Figure 1.4:** Regional pup production data from 1984 to 2010 (see Table S1 in Thomas et al., 2019).

repeated aerial and ground surveys of the breeding colonies, using various characteristics of the pups such as body shape and moulting to assess their age (Russell et al., 2019). The total pup production (i.e., number of pups born) was estimated by maximum likelihood for each colony, taking differences in survey methods into account but were aggregated into four regions to facilitate population dynamics modelling: North Sea, Inner Hebrides, Outer Hebrides, and Orkney (see Figure 1.3). While colony-level uncertainty measures are available for the pup estimates in aerially surveyed colonies, these are not independent due to shared parameters within one region, leading to difficulties when deriving an aggregated uncertainty on the regional level. Further, no uncertainty measures are available for the ground-surveyed colonies. We therefore use only the maximum likelihood pup estimates at the regional level for this thesis but treat the observation error variance of these estimates as unknown.

For this thesis, we use the data from the annual surveys from 1984 to 2010, see Figure 1.4 and Table A.1. We note that in 2009, there is a missing value for the pup production estimate for the Inner Hebrides. In this year, there were not enough aerial surveys in this region to produce a reliable pup production estimate. While pup counts have been untertaken in some colonies since the 1950s (Russell et al., 2019), the methods have changed considerably over time and therefore only counts since 1984 are used here. Since 2010 the survey method has changed to conducting only biennial surveys of the colonies. In addition, the camera to take the photos of the colonies has changed with has resulted in a jump in the observed pup numbers and further research is required to accurately determine its cause (Russell et al., 2016). We note that around 10% of breeding occurs in colonies that are not regularly monitored. The results in this thesis correspond only to the population size associated with regions that are regularly monitored.

The second source of information about the number of seals is a single independent estimate of total adult population size. As part of a larger survey targeting harbour seals (Lonergan et al., 2011), the number of hauled-out grey seals was counted in August of 2007 to 2009. The proportion of hauled-out animals was estimated to be 31% by using a sample of seals fitted with telemetry tags. Together, these led to an estimate of the number of total adults, derived by Russell et al. (2016). The uncertainty in the estimate was modelled as a shifted Gamma distribution by Thomas et al. (2019) for which that parameters can be found in Table A.2. Since the survey was conducted in August, before the breeding season, the estimate does not include the pups born in that year. Even though the estimate was derived using survey efforts spanning three years, the total estimate was assigned to the year 2008 to facilitate modelling. The estimate is a sum of all adult seals across all regions, because summer feeding regions are not necessarily close to a female adult's breeding colony (Russell et al., 2013) and because the survey was conducted in August, before the new pups were born. We also note that we assumed no adult mortality and no immigration or emigration between the time of the independent estimate in August and the breeding season, when the number of pups is counted. Both observations are assigned to the same time point, that is, after breeding occurs, which seems a reasonable approximation given the high adult survival rates.

## 1.2 Models

The overarching case study of this thesis is the population dynamics model of the UK grey seal population. This is a non-normal and non-linear state space model. We begin by defining some notation and then outlining the state space model framework in general. Then, we give details on the specific models used in this chapter. We describe the UK grey seal model, which we refer to as the "complete seal model". This is relatively complex and challenging to fit. To simplify the estimation problem and to isolate some of these challenges, we used three different models in addition to the complete seal model. We specify these models and explain how they are related to the complete seal model.

### 1.2.1 Notation

Before formally defining state space models, we set out some notation that is used throughout this thesis. Underlying true states, which can be scalars or vectors containing different components of the state such as population sizes by age, are denoted $x_t$ and observations $y_t$, where $t$ denotes time, a non-negative integer valued index. Indices of the form $i{:}j$ denote the range of integer values between $i$ and $j$, following notation in, e.g., Chopin and Papaspiliopoulos (2020). The notation $x_{0:t}$ then means all states from time 0 to time $t$, so $x_0, ..., x_t$. Parameter vectors are denoted with $\theta$.

For the distributions, we use a slight abuse of notation which is, however, very common in the literature (see, e.g., Chopin and Papaspiliopoulos, 2020, Wills and Schön, 2023, Kantas et al., 2015). Instead of distinguishing each probability density function (pdf) with a different name or different indices, the pdf is identified by its arguments. Model pdfs are generally denoted with $p$, and pdfs that are not part of a model but necessary for proposing values in the context of inference methods are denoted with $q$. For example, the prior pdf of a model is denoted with $p(\theta)$ and its posterior pdf with $p(\theta|y)$. Section

1.3 defines some of the most commonly used densities in this thesis. An exception to this are the state process (or transition) and observation equations in a state space model because of their central importance to the model. These are denoted $f$ and $g$ respectively, following, e.g., Kantas et al. (2015) and Newman et al. (2014a), and defined in detail in the next section. We note that while $p$ and $q$ are probability density functions, we also use these symbols when referring to the corresponding probability distribution. In contexts where the parameter vector is assumed known throughout, e.g., Chapter 2, we drop $\theta$ from the condition in the densities $p(\cdot|\theta)$ for better readability.

A notation that becomes relevant from Chapter 2 on is that of denoting states with a superscript $(i)$ in addition to the subscript $t$ for time, so $x_t^{(i)}$. This is used when describing SMC methods where the simulation of many possible states at time $t$ is necessary—the superscript then indexes the different simulated states.

### 1.2.2 State Space Models

State space models (SSMs) are a framework for modelling two time series evolving in parallel: one describing the state of some unobserved system and the other describing the measurements taken on that system. They have a flexible and intuitive structure and can separate process stochasticity from observation error. Process stochasticity refers to the variation in the true underlying states through time, i.e., year-to-year changes of population sizes, whereas observation error refers to the differences between the hidden state and observed data. The first SSMs were normal dynamic linear models (NDLM), i.e., models where the underlying states and observations can be modelled with linear equations and normal distributions (see Section 4.2.1 for a formal definition). These were used in aerospace engineering and together with the Kalman filter (see Chapter 4) were an important tool for the Apollo missions where they helped with the estimation of a spacecraft's trajectory using imprecise location measurements. In ecology, SSMs began to be used in the 1980s, for example in fisheries (Mendelssohn, 1988). Increase in computing power and advances in Markov chain Monte Carlo (MCMC) methodology, a computational technique for sampling from complex probability distributions (see Chapter 3), in the 1990s extended the possible applications of SSMs by allowing more complexity, such as non-linear and non-Gaussian models. This also made Bayesian parameter inference feasible when models had before been restricted to state inference with fixed parameters or parameter inference within a frequentist framework. Today, sophisticated fitting methods and software allow intricate model formulations which can incorporate parameters that depend on covariates, multiple hierarchical levels, random effects, and complex observations such as capture-recapture data (King, 2012). For an extensive overview of the use of state-space models in ecology, see Auger-Méthé et al. (2021).

Formally, SSMs are models for a time series of state vectors $x_{0:T} = (x_0, ..., x_T)$, with $x_t \in \mathbb{R}^n$ for $t \in \mathbb{N}_0, n \in \mathbb{N}$. The state vectors themselves can take any value in $\mathbb{R}^n$. We also assume the times $t = 1, ..., T$ to be evenly spaced and so have regular time intervals. (This assumption can readily be relaxed, but all models described in this thesis have evenly spaced time points.) The elements of $x_{0:T}$ arise from a first order Markov process. This means that the distribution of the state at time $t+1$ given the state at time $t$ is independent of any previous states, i.e., $p(x_{t+1}|x_{0:t}) = p(x_{t+1}|x_t)$. The transition process can therefore be described through the following pdf, where $\theta$ denotes the vector of model parameters:

$$\begin{aligned} \text{Initial state pdf: } & x_0 && \sim f_0(\cdot|\theta) \\ \text{Transition pdf: } & x_t|x_{0:t-1} && \sim f_t(\cdot|x_{t-1}, \theta) \end{aligned} \tag{1.1}$$

The state vectors $x_t$ are not directly observed, but are related to a series of observations, or measurements $y_{1:T} = (y_1, ..., y_T)$, with $y_t \in \mathbb{R}^m$. Any observation $y_t$ is independent of $(x_0, ..., x_{t-1}, x_{t+1}, ..., x_T)$ conditional on $x_t$ . This relationship is described through the observation pdf

$$\text{Observation pdf: } y_t|x_{0:T} \sim g_t(\cdot|x_t, \theta). \tag{1.2}$$

The process and observation densities can vary throughout the time series which is indicated by the index $t$, i.e., $f_t$ and $g_t$. However, we will omit this index from now on. It is usually clear from the arguments of the pdf which time point is meant. We will also omit writing the parameter $\theta$ as a conditioning variable unless the dependence on a parameter needs to be made explicit. With the transition and observation densities as defined above, we can write the joint pdf as

$$p(x, y) = f(x_0) \prod_{t=1}^{T} f(x_t|x_{t-1}) g(y_t|x_t)$$

which is sometimes called the complete data likelihood and the marginal pdf for the observations as

$$p(y) = p(y_1) \prod_{t=2}^{T} p(y_t|y_{1:t-1}) \tag{1.3}$$

which is sometimes called the observed data likelihood. This way of specifying the model is helpful later on, when various algorithms are discussed.

We use a Bayesian framework for this model, and hence add an additional layer to the transition and the observation process which is the prior pdf for the model parameters $\theta$:

$$\text{Prior pdf: } \theta \sim p(\cdot)$$

In the context of population dynamics, we are interested in the posterior distribution of the parameters which we denote $p(\theta|y_{1:T})$ as well as the distribution of $x_{1:T}|y_{1:T}$. For a more detailed description of state space models, see Doucet et al. (2001), and see Newman et al. (2014a) for the usage of state space models in the context of population dynamics.

### 1.2.3 Complete Seal Model

The model described here is the one that is currently used for the estimation of the UK grey seal population (Thomas, 2021) and extensively described and justified in Thomas et al. (2019). We note that this model is very complex given the available data. It has 10 parameters that need to be estimated, while relying almost exclusively on observations on only one age class out of seven, other than in one year where an estimate of total abundance is available. This leads to a challenging estimation task, as can be seen throughout this thesis. Chapter 5 explores the challenges with fitting this model and suggests some solution for future research.

#### 1.2.3.1 States

The complete seal model has a 28-dimensional state: it contains the number of pups and adult female seals in each of the four surveyed regions North Sea (NS), Inner Hebrides (IH), Outer Hebrides (OH) and Orkneys (Ork). Adult males are not modelled as part of the SSM. For every region the number of seals is modelled according to age. Pups and female seals aged between one and five years are modelled in their own age group, and female seals aged six and above are grouped together. The reason for this is the following. Harwood and Prime (1978) describes two samples of seals that were shot in culls in 1972 and 1975. For these two samples, it was found that 16% of female adult seals had had their first pup aged 5, 45% at age 6 and 39% at age 7 and over. For the purposes of this model, these findings are simplified to assume that only females from age 6 onwards are able to produce a pup. This leads to seven age categories per region and so to 28 categories in total. The state in year $t$ is denoted by a row vector with 28 entries $x_t = (x_{a,r,t})_{a=0:6,r=1:4} = (x_{a=0,r=1,t}, ..., x_{a=6,r=1,t}, x_{a=0,r=2,t}, ..., x_{a=6,r=4,t})$ where $t$ denotes the year, $r$ the region numbered 1 through 4, and $a$ the age group from 0 to 6[1].

#### 1.2.3.2 State Process

The state process from one year to the next consists of survival, age incrementation and birth, in that order. These three sub-processes are described below, with the notation following Newman et al. (2014a).

The survival process is modelled as binomial, with survival probabilities $\phi_{p,r,t}$ for pups dependent on region and year, and $\phi_a$ for adults[2], constant across regions and time. We assume pup survival to be density dependent and use an extended Beverton-Holt function (Beverton and Holt, 1957) to relate pup survival to the pup carrying capacity of the region, the maximum pup survival rate $\phi_{p,\max}$, and the numbers of new-born pups $x_{0,r,t}$:

$$\phi_{p,r,t} = \frac{\phi_{p,\max}}{1 + (\beta_r x_{0,r,t})^\rho} \tag{1.4}$$

The parameter $\beta_r$ determines for which numbers of pups the survival probability starts to decline. The parameter $\rho$ governs the shape of the density dependent pup survival rate. The steepness of the decline in survival probability increases as $\rho$ decreases (Figure 1.5).

The density dependent pup survival leads to a maximum number of pups (and a resulting maximum total number of seals) that can be sustained, also called the carrying capacity $\chi_r$ of the region $r$. The parameter $\beta_r$ is inversely proportional to this carrying capacity. As described in Thomas and Harwood (2005), we can calculate the carrying capacity dependent on the other parameters:

$$\chi_r = \frac{1}{\beta_r}\left(\frac{0.5\alpha\phi_{p,\max}\phi_a^5}{1-\phi_a} - 1\right)^{1/\rho}. \tag{1.5}$$

The proof for this relationship is given in Appendix A.2.1. As we can easily switch between $\beta_r$ and $\chi_r$ with the help of the above transformation, it does not matter mathematically which of the two parametrisations we use in the model formulation. As the interpretation

[1]Occasionally, we use the absolute year of a state or observation instead of the time index $t$. When it is not clear from the context whether the year or the time index is used, we will use *year* in the index instead of $t$.

[2]For the purposes of this model, all animals aged 1 and older are considered to be “adults”.

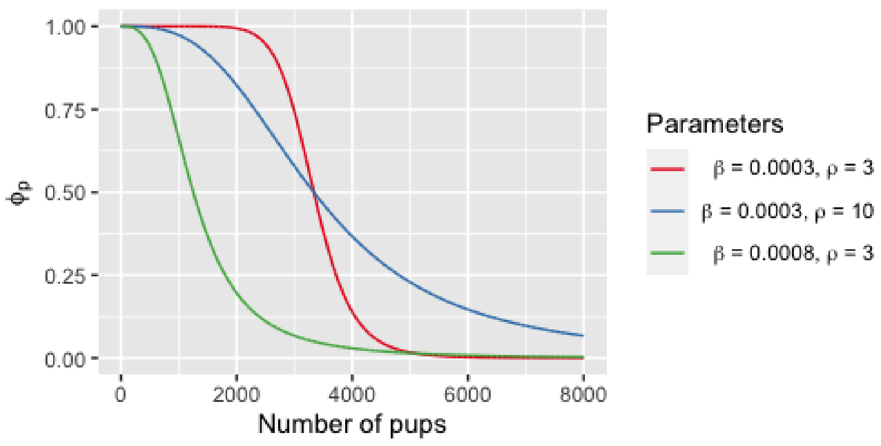


**Figure 1.5:** Beverton-Holt density dependent pup survival for different values of $\beta$ and $\rho$. The maximum pup survival probability $\phi_{p,\max}$ here is 1.

of $\chi_r$ is more intuitive, making it easier to choose a prior distribution, we use this parameter in our model formulation and analysis.

This leads to the following survival process, where $u$ denotes an intermediate state between $x_t$ and $x_{t-1}$ and the index $s$ denotes numbers after survival:

$$u_{t,s} = \begin{pmatrix} u_{0,r=1,t,s} \\ u_{1,r=1,t,s} \\ \vdots \\ u_{6,r=1,t,s} \\ \vdots \\ u_{0,r=4,t,s} \\ \vdots \\ u_{6,r=4,t,s} \end{pmatrix} \sim \begin{pmatrix} \text{Bin}(x_{0,r=1,t}, \phi_{p,r=1,t}) \\ \text{Bin}(x_{1,r=1,t}, \phi_a) \\ \vdots \\ \text{Bin}(x_{6,r=1,t}, \phi_a) \\ \vdots \\ \text{Bin}(x_{0,r=4,t}, \phi_{p,r=4,t}) \\ \vdots \\ \text{Bin}(x_{6,r=4,t}, \phi_a) \end{pmatrix},$$

where the parameter $\phi_a$ is the probability of adult survival (for all individuals older than pups). The age incrementation (numbers after ageing denoted with index $a$) is deterministic but we include the binomial sexing process in the pup age incrementation. The sex ratio at birth is assumed to be 50:50.

$$u_{t,a} = \begin{pmatrix} u_{0,r=1,t,a} \\ u_{1,r=1,t,a} \\ u_{2,r=1,t,a} \\ \vdots \\ u_{6,r=1,t,a} \\ \vdots \\ u_{0,r=4,t,a} \\ \vdots \\ u_{6,r=4,t,a} \end{pmatrix} = \begin{pmatrix} 0 \\ \sim \text{Bin}(u_{0,r=1,t,s}, 0.5) \\ u_{1,r=1,t,s} \\ \vdots \\ u_{6,r=1,t,s} + u_{5,r=1,t,s} \\ \vdots \\ 0 \\ \vdots \\ u_{6,r=4,t,s} + u_{5,r=4,t,s} \end{pmatrix}.$$

Finally, we model the birth process which leads to the new state $x_{t+1}$ where females aged 6 and above are assumed to be able to produce a pup.

$$x_{t+1} \quad = \quad \begin{pmatrix} x_{0,r=1,t} \\ x_{1,r=1,t} \\ \vdots \\ x_{6,r=1,t} \\ \vdots \\ x_{0,r=4,t} \\ \vdots \\ x_{6,r=4,t} \end{pmatrix} = \begin{pmatrix} \sim \text{Bin}(u_{6,r=1,t,a}, \alpha) \\ u_{1,r=1,t,a} \\ \vdots \\ u_{6,r=1,t,a} \\ \vdots \\ \sim \text{Bin}(u_{6,r=4,t,a}, \alpha) \\ \vdots \\ u_{6,r=4,t,a} \end{pmatrix},$$

where $\alpha$ is the probability than an age 6+ female gives birth to a single pup. Adult males are not modelled explicitly in the states. Whenever the total numbers of adults is required, it can be calculated by multiplying the number of female adults with the adult sex ratio parameter denoted $\omega$. We refer to $\omega$ as the sex ratio parameter throughout this thesis but note that more strictly it is the ratio of the total number of adult seals to the number of adult female seals.

We note that for fixed parameter values, the four regions are independent of each other. The model assumes that seals never change region. However, all parameters other than the carrying capacity $\chi_r$ are shared between the four regions.

#### 1.2.3.3 Observation Process: Pup Production Estimates

There are two sources of information about the seal numbers which are assumed to be independent of each other. The first is a yearly estimate of pup production (i.e., numbers of pups born). The observed pup numbers $y_t = (y_{r=1,t}, y_{r=2,t}, y_{r=3,t}, y_{r=4,t})'$ are assumed to be unbiased normally distributed observations of the true pup numbers with a constant coefficient of variation (CV) and precision parameter $\tau$, so

$$y_t = \begin{pmatrix} y_{r=1,t} \\ y_{r=2,t} \\ y_{r=3,t} \\ y_{r=4,t} \end{pmatrix} \sim \begin{pmatrix} N(x_{0,r=1,t}, x^2_{0,r=1,t}/\tau) \\ N(x_{0,r=2,t}, x^2_{0,r=2,t}/\tau) \\ N(x_{0,r=3,t}, x^2_{0,r=3,t}/\tau) \\ N(x_{0,r=4,t}, x^2_{0,r=4,t}/\tau) \end{pmatrix}.$$

We note that strictly $\tau$ is not the precision of the distribution, which is the inverse of the variance and in this case is $\tau/x^2_{0,r,t}$ but rather the inverse of the square of the CV, so $\tau = 1/CV^2 =$. For convenience, we refer to $\tau$ as the precision parameter throughout this thesis but note that to arrive at the true precision of the distribution, $\tau$ needs to be divided by the square of the the mean of the distribution.

For this case study, we use the pup production estimate for the years 1984-2010 as derived by Russell et al. (2019). The 1984 estimate is used as $y_0$ to derive the initial state distribution.

#### 1.2.3.4 Observation Process: Independent Estimate

The independent estimate of total adult population size by Russell et al. (2016) based on the survey in Lonergan et al. (2011) includes male and female adult animals. The number that was estimated therefore requires the sex ratio parameter $\omega$ and is

$$x_{adults,24} = \omega \sum_{r=1}^{4} \sum_{a=1}^{6} x_{a,r,t=24}. \tag{1.6}$$

The uncertainty in the estimate was quantified using a non-parametric bootstrap by Russell et al. (2016) which was modelled as a right-shifted Gamma distribution in Thomas et al. (2019). They used the following notation for the estimate and its distribution:

$$y_{adults,24} \sim \kappa_0 + \mathrm{Ga}(\kappa_1, \kappa_2). \tag{1.7}$$

Here, $\kappa_0$ is the shift parameter, $\kappa_1$ the shape parameter and $\kappa_2$ the scale parameter. These parameters were estimated as $\kappa_0 = 59167.84$, $\kappa_1 = 12.96$ and $\kappa_2 = 2719.38$, leading to an expected value of 94398.51 and a standard deviation of 9788.03 (see Table A.2). However, there are two problems with using this in the state space model framework. Firstly, there needs to be a realisation of the observation process, so a number (or a vector of numbers) that denote the observation that was actually made in that year. With the notation in Equation 1.7, there is no fixed number associated with $y_{adults,24}$, just a distribution. Secondly, the observation density should depend on the underlying state $x_{24}$. This is also not the case here, as $\kappa_0$, $\kappa_1$ and $\kappa_2$ are fixed in advance and independently of the states. These two issues can be solved in two steps. The first step is to change the interpretation of the uncertainty in the estimate. Rather than saying that the estimate is distributed according to Equation 1.7, the distribution quantifies how likely the underlying true state is given the observations that were made in the survey. This introduces the underlying state into the equation and allows us to make a connection between the state and the parameters $\kappa_{0:2}$.

$$x_{adults,24} \sim \kappa_0 + \mathrm{Ga}(\kappa_1, \kappa_2).$$

In a second step, we change the interpretation of the parameters and the state. Instead of treating $x_{adults,24}$ as the random variable and $\kappa_0$, $\kappa_1$ and $\kappa_2$ as fixed parameters, we treat $x_{adults,24}$ as fixed, and $\kappa_0$ as the random variable. This leads to

$$\kappa_0 \sim x_{adults,24} - \mathrm{Ga}(\kappa_1, \kappa_2). \tag{1.8}$$

Therefore, if we set the observation of the independent estimate to $y_{IE,24} := \kappa_0$, and treat $\kappa_1$ and $\kappa_2$ as fixed and known parameters, the independent estimate fits exactly into the SSM framework. Since there is only one independent estimate, we will usually drop the index 24 and denote the independent estimate only by $y_{IE}$.

We note that the above formulation can seem unintuitive and offer the following mathematically equivalent formulation,

$$\kappa_0^* \sim x_{adults,24} - \mathrm{Ga}(\kappa_1, \kappa_2) + \kappa_1\kappa_2 \tag{1.9}$$

where the observation is re-defined as $\kappa_0^* := \kappa_0 + \kappa_1\kappa_2$. Because $\kappa_1\kappa_2$ is the mean of the Gamma distribution, the observation now has a mean of $x_{adults,24}$ and is therefore centred around the true state value, similar to the pup observations. By contrast, the observation error is not distributed according to a normal distribution but according to the mirrored and shifted Gamma distribution $-\mathrm{Ga}(\kappa_1, \kappa_2) + \kappa_1\kappa_2$ which has mean 0. Since $\kappa_1$ and $\kappa_2$ are known, adding this constant to both sides in Equation 1.9 does not change the model other than providing a more intuitive formulation. From a mathematical perspective it unnecessarily complicates the model and we therefore work with the formulation in Equation 1.8 throughout this thesis when implementing models. Whenever observations of adult counts are depicted in a plot, we shift them so that the mean of their distributions is equal to the true count.

Including the independent estimate $y_{IE,24}$ in the state space model in addition to the annual pup production is done by augmenting the observation in year $t = 24$:

$$p(y_{1:T}, y_{IE}) = \left(\prod_{t=1}^{t^*-1} p(y_t|y_{1:t-1})\right) p(y_{t^*}, y_{IE}|y_{1:t^*-1}) \left(\prod_{t=t^*+1}^{T} p(y_t|y_{1:t-1}, y_{IE})\right)$$

This means that whatever the fitting algorithm usually does with the observation $y_t$, it now does with both the pup production observation and the independent estimate in year $t^*$. All later steps in the algorithm incorporate the information provided by the independent estimate. A second option for including the independent estimate is discussed in Chapter 6.

#### 1.2.3.5 Initial state pdf

For the initial distribution of the states in year $t = 0$, we use the first observation, made in 1984, and effectively reverse the observation density to obtain the initial distribution of pup numbers. Thomas et al. (2019) additionally dispersed the values further with a uniform distribution using the dispersion parameter $a$. This was done to ensure that all possible initial values are captured by the initial distribution.

$$\begin{aligned} x_r^* &\sim \mathrm{N}(y_{r,0}, y_{r,0}^2/\tau) \\ x_{0,r,t=0} &\sim \mathrm{Unif}(x_r^*/a, a x_r^*), \end{aligned}$$

where $a$ was set to 1.3. As desired, this increases the variance of the distribution of initial pup states from $\mathrm{Var}(x_r^*) = y_{r,0}^2/\tau$ to

$$\mathrm{Var}(x_{0,r,t=0}) = \frac{1}{3}\left(a^2 + 1 + \frac{1}{a^2}\right) y_{r,0}^2/\tau + \frac{1}{12}\left(a - \frac{1}{a}\right)^2 y_{r,0}^2.$$

It also increases the expected value from $\mathrm{E}(x_r^*) = y_{r,0}$ to

$$\mathrm{E}(x_{0,r,t=0}) = \frac{1}{2}\left(a + \frac{1}{a}\right) y_{r,0}.$$

The proof for this is given in Appendix A.2.2. This might be less welcome as it introduces a bias in the distribution. An alternative solution for future updates of the model could be to further distribute the initial values according to the distribution $\mathrm{Unif}((1-a)x_r^*, (1+a)x_r^*)$, where $a$ could be set to 0.3, so $\mathrm{Unif}(0.7x_r^*, 1.3x_r^*)$.

The initial distribution of adult females of age 1 through 5 is the result of applying the survival process to the initial pup numbers:

$$\begin{aligned} x_{a=1,r,t=0} &\sim \mathrm{Bin}(x_{a=0,r,t=0}, 0.5\phi_{p,r,t=0}) \\ x_{a,r,t=0} &\sim \mathrm{Bin}(x_{a-1,r,t=0}, \phi_{a,r,t=0}), \quad 2 \le a \le 5. \end{aligned}$$

Here, $\phi_{p,r,t=0}$ is calculated as in Equation 1.4, substituting the observed number of pups $y_0$ for the true number of pups. For adult female numbers of age 6+, the breeding process is reversed by drawing the number of non-breeding females from a negative binomial distribution and adding the number of breeding females (which is equal to the number of pups).

$$x_{a=6+,r,t=0} \sim \text{Negbin}(x_{a=0,r,t=0}, \alpha) + x_{a=0,r,t=0}.$$

As the first observation $y_0$ is used to determine the initial distribution of the states, it cannot be used again as a standard observations to determine the filtering distribution. We therefore usually only mean $y_{1:T}$ when referring to the observations.

#### 1.2.3.6 Prior Distributions

The prior distributions for the model are the ones used in Thomas et al. (2019), where an extensive justification is included, and are given in Table 1.1.

| Parameter | Prior distribution | Prior mean (SD) |
|---|---|---|
| Maximum pup survival $\phi_{p,\max}$ | Beta(2.87, 1.78) | 0.62 (0.20) |
| Adult survival $\phi_a$ | 0.8+0.17 × Beta(1.6,1.2) | 0.90 (0.04) |
| Fecundity $\alpha$ | 0.6+0.4 × Beta(2,1.5) | 0.83 (0.09) |
| Density dependence shape $\rho$ | Ga(4,2.5) | 10 (5) |
| NS carrying cap. $\chi_{NS}$ | Ga(4,5000) | 20000 (10000) |
| IH carrying cap. $\chi_{IH}$ | Ga(4,1250) | 5000 (2500) |
| OH carrying cap. $\chi_{OH}$ | Ga(4,3750) | 15000 (7500) |
| Ork carrying cap. $\chi_{Ork}$ | Ga(4,10000) | 40000 (20000) |
| Observation precision $\tau$ | Ga(2.1,66.67) | 140 (96.61) |
| Sex ration $\omega$ | 1.6 + Ga(28.08,3.7E-3) | 1.7 (0.02) |

**Table 1.1:** Prior distributions and summary statistics for the parameters of the complete seal model

While these prior distributions are set independently, we note that the Beverton-Holt density dependence induces a dependence between some of the parameters. Examining Equation 1.5 makes it clear that the term

$$\frac{0.5\alpha\phi_{p,\max}\phi_a^5}{1-\phi_a} - 1$$

needs to be non-negative as this number is exponentiated with $1/\rho$. A simulation of 10,000,000 parameter values from the prior showed that only 66.3% of values satisfied the condition above. Figures 1.6 and 1.7 show the resulting prior distributions and correlations once all invalid combinations have been discarded.

### 1.2.4 Complete Seal Model Without Independent Estimate

The independent estimate contains information about the parameters but also introduces some fitting challenges. In order to isolate these challenges and to be able to assess the exact influence of the independent estimate, we also estimated the posterior for the

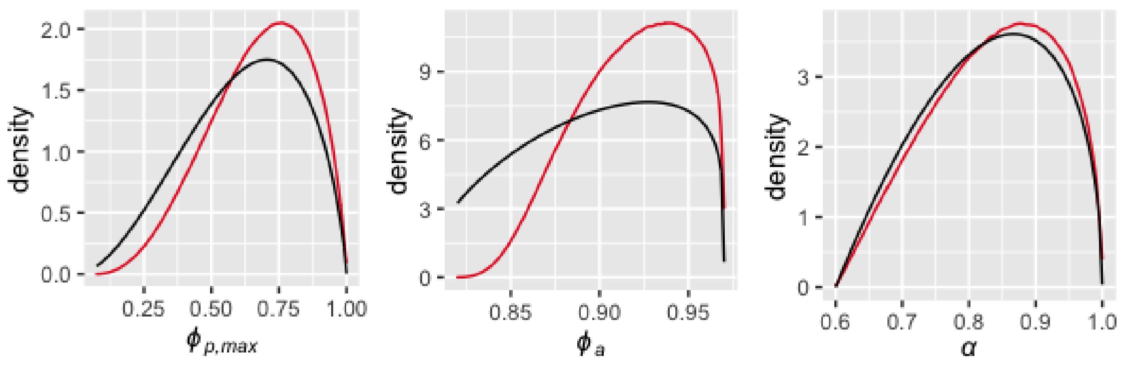


**Figure 1.6:** Prior distributions for $\phi_{p,\max}$, $\phi_a$ and $\alpha$ (red lines) as induced by the Beverton-Holt density dependence. The black lines indicate the original independent prior distributions.

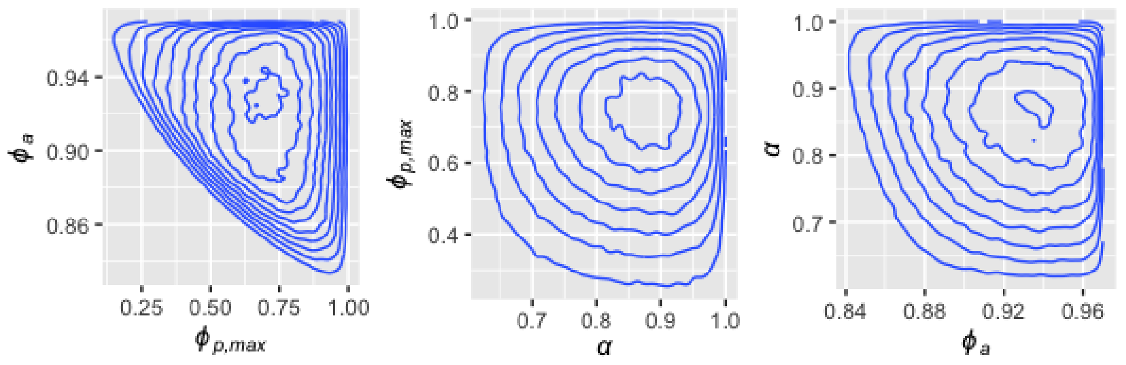


**Figure 1.7:** Paired density plots of the priors for $\phi_{p,\max}$, $\phi_a$ and $\alpha$ as induced by the Beverton-Holt density dependence. The correlation coefficients between the induced prior distributions are $\rho_{\phi_a,\phi_{p,\max}} = -0.312$, $\rho_{\phi_{p,\max},\alpha} = -0.024$ and $\rho_{\alpha,\phi_a} = -0.107$.

complete seal model without the independent estimate, i.e., only using the annual pup production estimates. As the independent estimate is the only source of information about the sex ratio $\omega$, inference for this parameter was omitted in this case. All other aspects of the model remain the same.

### 1.2.5 7-state model

This and the following model were used in the simulation studies. Here, we reduced the model to one region only and therefore had only 7 instead of 28 states. This reduces the number of parameters from 10 (or 9, excluding sex ratio) to 6 because only one instead of 4 carrying capacity parameters need to be estimated. As in the previous Section 1.2.4, only pup production estimates were used as the observation process. Because only one region is modelled, the observation at each time point is a scalar instead of a vector with 4 components. The prior distributions were adopted from the complete seal model. For the carrying capacity parameter, we chose the prior of the Inner Hebrides region.

### 1.2.6 2-state model

This model is the simplest of the 4 models investigated in this thesis. It is closely related to the model described in the previous Section 1.2.5 and has the same 6 parameters but consists of only two age groups (pups and adult breeding females) instead of 7. The survival and age incrementation process is then

$$u_{t,sa} = \begin{pmatrix} u_{0,t,sa} \\ u_{1+,t,sa} \end{pmatrix} \sim \begin{pmatrix} 0 \\ \text{Bin}(x_{0,t-1}, 0.5\phi_{p,t}) + \text{Bin}(x_{1+,t-1}, \phi_a) \end{pmatrix}.$$

After the birth process we obtain the new state

$$x_{t=1} = \begin{pmatrix} x_{0,t+1} \\ x_{1+,t+1} \end{pmatrix} \sim \begin{pmatrix} \text{Bin}(u_{1+,t,sa}, \alpha) \\ u_{1+,t,sa}. \end{pmatrix}.$$

We calculate the density dependent pup survival with the same formula as given in Equation 1.4. Because there are only 2 instead of 7 states, this leads to a different relationship between $\beta$ and $\chi$. By adapting the proof in Appendix A.2.1, we obtain

$$\chi = \frac{1}{\beta}\left(\frac{0.5\alpha\phi_{p,\max}}{1-\phi_a} - 1\right)^{1/\rho}.$$

As in the previous two models, we only use the pup production estimates as the observation process. The priors are the same as in the 7-state model.

## 1.3 Inference for State Space Models

| Name | pdf |
|---|---|
| Filtering | $p(x_t \| y_{1:t})$ |
| Joint filtering | $p(x_{0:t} \| y_{1:t}), t = 0, ..., T-1$ |
| Smoothing | $p(x_t \| y_{1:T}), t = 0, ..., T-1$ |
| Joint smoothing | $p(x_{0:T} \| y_{1:T})$ |
| Prediction | $p(x_{t+1} \| y_{1:t})$ |
| Prior | $p(\theta)$ |
| Posterior | $p(\theta \| y_{1:T})$ |
| Joint posterior | $p(x_{0:T}, \theta \| y_{1:T})$ |
| Likelihood | $L(\theta; y_{1:T}) = p(y_{1:T} \| \theta)$ |
| Joint likelihood | $L(x_{0:T}, \theta; y_{1:T}) = p(x_{0:T}, y_{1:T} \| \theta)$ |

**Table 1.2:** Definition of names for relevant distributions for inference in state space models. Note that the likelihood is not technically a distribution when considered a function of $\theta$ but is featured in this table for convenience. This table is modified from a table in Schön and Lindsten (2017).

When we discuss inference for state space model, several different tasks can be distinguished. First, we might consider inference where the parameter vector is assumed known, and inference for the underlying states is the focus. The so-called *filtering* distribution is the distribution of a state $x_t$ given all observations up to time $t$, so $y_{1:t}$. This is relevant in applications outside of statistical ecology where states need to be estimated live, e.g., in navigation. For reasons related to the computation of the likelihood that are discussed in Chapter 2, the filtering distribution is equally important when estimates of the likelihood are required in a parameter inference context. When data beyond time $t$ are available in the estimation of $x_t$, the distribution of $x_t|y_{1:T}, t, T$ is called the *smoothing* distribution. These distributions can be extended to include a series of states $x_{0:t}$ and are then called the joint filtering and joint smoothing distributions. To emphasise the difference, the distributions of only one state are sometimes called marginal filtering and marginal smoothing distribution (Schön et al., 2018) but we omit this here. The distribution of $x_{t+1}$ given observations until time $t$ is called the *prediction* distribution.

For parameter inference, the prior is denoted with $p(\theta)$ and the posterior with $p(\theta|y_{1:T})$. The likelihood is often denoted with $L(\theta; y_{1:T}) = p(y_{1:T}|\theta)$ although we usually use only the second term in this thesis. When the likelihood of both a parameter $\theta$ and a set of states $x_{0:T}$ is calculated, this is called the joint likelihood.

### 1.3.1 Inference Methods

While SSMs allow flexible and complex model formulations, these can sometimes be difficult to fit, and this is certainly the case for the case study in this thesis. Here, we give a brief overview of the available methods to compute or estimate the distributions mentioned in the previous section. No single method is universally best and rather depends on the model in question, the expertise and time available for tuning the fitting method, and the goal of the fitting task (Fasiolo et al., 2016).

For normal dynamic linear models, a Kalman filter (see Chapter 4 and Kalman, 1960) can be used to calculate the exact state filtering distributions and likelihood values and, with some additional steps, the smoothing distributions. For models with other distributions or non-linear dynamics, extensions to the Kalman filter exist such as the extended Kalman filter (Gordon et al., 1993) and unscented Kalman filter (Wan and van der Merwe, 2001), although these only provide approximations to the likelihood and filtering distributions. Maximum likelihood estimates can be obtained through numerical optimisation. For Bayesian inference, any MCMC method that requires the likelihood of a parameter can be employed (see Section 3.2.2.1 for a brief introduction of MCMC methods).

For more complex model, the likelihood can be approximated in various ways. For low-dimensional states, up to 2-3, the states can be discretised by assigning them to cells in a grid. This discretised model can then be fitted with the tools for Hidden Markov Models (Besbeas and Morgan, 2020). While this might be an option for the 2-state model, the complete seal model has a 28-dimensional state and discretising is not a feasible option.

A second option is the Laplace approximation. Here, the marginal likelihood $p(\theta|y_{1:T}) = \int p(\theta, x_{0:T}|y_{1:T})dx_{0:T}$ is approximated using the second-order Taylor expansion of the integrand to obtain a Gaussian integral. This method has been widely applied in a frequentist setting but also been explored for Bayesian inference (Monnahan and Kristensen, 2018). In this thesis, this method is not investigated although this is an avenue of further research.

Third, and most important in the context of this thesis, are SMC methods (see, e.g., Kantas et al., 2015 for a general review). These provide estimates of the state distributions of interest by iteratively simulating states through importance sampling schemes. They can also generate unbiased estimates of the likelihood. To obtain maximum likelihood estimates (MLEs), stochastic optimisation procedures are required due to the randomness of the likelihood estimates, and that is not covered in this thesis. For Bayesian parameter inference, several methods exist that build on the ideas for state inference, either by expanding the state space with the parameters, or by using the likelihood estimates in an MCMC algorithm. These are the algorithms studied in much of this thesis, namely in Chapters 2 and 3 and they are laid out in detail there.

Instead of calculating, approximating or estimating the likelihood $p(y_{1:T}|\theta)$, Bayesian data augmentation uses the joint likelihood $L(x_{0:T}, \theta; y_{1:T})$ in an MCMC algorithm to generate samples from the joint posterior $p(x_{0:T}, \theta | y_{1:T})$ and then marginalises over the states to obtain samples from the parameter posterior distribution (King, 2011). This method is popular due to its flexibility but was not expected to perform well for the seal model. We used it in Chapter 4 where three different algorithms were compared.

A recent discussion of these methods can be found in Newman et al. (2023).

## 1.4 Software

There is a wide range of software packages available for fitting SSMs which provide one or more fitting methods. Rather than provide a full review of these, we refer to Newman et al. (2023), where a classification of the most commonly used software packages can be found. We also refer to Beraha et al. (2021) for a comparison of `nimble`, `JAGS` and `Stan` for Bayesian inference, where computation times, goodness of fit of the output and tuning requirements are compared across a number of different models.

All analysis in this thesis was done with `R` (R Core Team, 2023). The fitting algorithms used the packages `nimble` (version 1.0.1, de Valpine et al., 2023; see also de Valpine et al., 2017) and `nimbleSMC` (version 0.10.1, NIMBLE Development Team, 2021, see also Michaud et al., 2021). These packages include MCMC algorithms that are translated to compiled `C` code and are therefore faster than implementations of the same algorithms in `R` code. `nimble` was chosen because of its flexibility with implementing complex models and algorithms. This allowed, e.g., the implementation of the Kalman filter in Chapter 4. Where `nimble` was insufficient for implementing the algorithm under consideration, we implemented our own customised algorithm in `R`.

## 1.5 Thesis Outline

The rest of this thesis is structured as follows.

In Chapter 2, we discuss inference methods for the models introduced in Section 1.2 in cases where the parameter is assumed known. The inference tasks are then to generate samples from the state filtering or smoothing distribution and to estimate the likelihood at the parameter value. In the first part of the chapter, we provide a review of SMC methods. We start with importance sampling techniques and give simple ("toy") examples to provide some intuition for the considerations that need to be made when these techniques are used. To apply this technique to state space models, we show how importance sampling techniques can be used sequentially, resulting in a *bootstrap filter*, the basic building block on which the more advanced SMC techniques rely. We then describe several extensions to the bootstrap filter. We discuss the aspects that need to be considered when an extension is used and the theoretical advantages and disadvantages of the available options. A simulation study follows, where we explore the most relevant of these methods by estimating the likelihood for the 2-state model. This simplification is necessary to reduce the computation time to a level that allows the detailed study of 4 different factors while keeping the Monte Carlo error of the results low. We then apply the results of this simulation study to the 7-state and the complete seal model, and only adapt the computational effort for each model. We briefly explore the effect of re-formulating the complete seal model by exploiting the independence of the four regions in a factorised formulation. This is promising and further developed in Chapter 6.

Chapter 3 discusses parameter inference in the context of the seal model. Similar to Chapter 2, we first review the available SMC methods to generate samples from the posterior distribution. These are divided into three groups. The first group consists of methods that augment the unknown state with the parameter and then use SMC methods for state inference, treating the parameters similarly to the states. The second group are methods that build on MCMC algorithms, like data augmentation and the particle marginal Metropolis-Hastings (PMMH) algorithm, which relies on likelihood estimation with SMC and then uses this estimate within an MCMC algorithm. In the third group, we describe the SMC$^2$ algorithm, which combines these two techniques. Following this review, we explore various options for the PMMH in detail by fitting the 2-state model, again for computational reasons. We then adapt the outcomes of this study to each of the increasingly complex models. This yields samples from the posterior distribution for the complete seal model with the real data, albeit requiring an often prohibitively long computing time (11 days running 10 parallel processes). We demonstrate our attempts to fit the seal model with the SMC$^2$ algorithm but the method shows difficulties even for the 2-state model, and did not converge for the complete seal model. The chapter is concluded by analysing the posterior obtained with the PMMH and comparing the results with those previously obtained in Thomas et al. (2019).

The difficulties in Chapter 3 justify the use of an approximation for parameter inference, which is investigated in Chapter 4. We linearise and normalise the seal model and are then able to calculate the exact likelihood of this NDLM approximation with the Kalman filter. This likelihood is used as part of a Metropolis-Hastings algorithm to generate samples from the posterior distribution. To assess the quality of this approximated posterior, we compare it with the posterior distribution produced by a PMMH algorithm and by an MCMC algorithm using data augmentation which both target the true posterior distribution. This comparison is done for the 2-state model in five scenarios to evaluate the closeness of the approximation to the true posterior distribution under various challenging conditions. We then compare the posterior distributions of the three methods, using the complete seal model with the real data. In most of the cases explored in this chapter, the Kalman filter approximation is close to the true posterior while profiting from a much lower runtime. We briefly investigate two ideas for getting the approximation even closer to the true posterior but conclude that neither of these show an improvement.

Rather than discussing more inference methods, in Chapter 5 we analyse the posterior distribution of the 2-state model. Motivated by difficulties in estimation in previous chapters, in particular when comparing posterior means with the true parameter values, we investigate factors affecting identifiability of the models. Building on previous work by Auger-Méthé et al. (2016) for maximum likelihood estimation, we suspected that the size of the observation error and the process stochasticity might be related to these difficulties. First we decrease the observation error, which results in mean parameter estimates that are further away from the true parameter value as the error decreases. We then increase the process stochasticity by placing a random effect on one of the parameters and find that the posterior mean estimates are not strongly affected by these changes but that the estimated correlation effects are. The results also highlight an issue with the identifiability of two of the parameters whose marginal posterior distributions remain relatively wide even when high-precision observations are used for the estimation and who are highly correlated. However, this is not an issue when state estimation is considered. Here, precise observations lead to precise and unbiased state estimates.

In Chapter 6, we pick up the idea of factorising the likelihood to obtain estimates with a lower variance using less computational effort, first explored in Chapter 2. The factorised model formulation cannot not easily be transferred to the complete seal model due to the independent estimate which negates the independence of the four regions. We describe three options for using a modification of this idea that can be used even in the presence of one observation that combines the regions. One of these options is then implemented. We show the effect of this implementation on the variance of the likelihood and see that the required computational effort is reduced by a factor of about 10-30 to obtain the same variation of the estimates, depending on the measure used to assess this. We then generate samples from the posterior distribution and find that the computational effort is reduced by a factor of about 5 to achieve the same effective sample size.

Chapter 7 concludes the thesis and summarises the key findings. We propose how the results obtained might be further developed and give an outlook to further directions of research.

# Chapter 2

# Likelihood Estimation With SMC Methods

## 2.1 Introduction

In this chapter, we study SMC methods that enable state inference and yield estimates of the likelihood when the model is evaluated at a specific parameter value $\theta$. The reasons for studying these methods are twofold. First, gaining knowledge about the hidden states can be a problem in its own right. This might be the case in scenarios where the parameters are known physical constants, for example in Schön et al. (2005), where the flight path for the Swedish fighter aircraft Gripen is estimated, or where parameters have been estimated a priori, as in the analysis of seal movement in Jonsen et al. (2005). Second (and most common in ecology), the model parameters are unknown and estimating them is of interest. Many methods that perform parameter inference rely on state inference methods as one of their building blocks. This is because these methods not only produce an estimate of the distribution of the hidden states for given parameters but also of the likelihood value for these parameters. It is therefore important to ensure that the best possible state inference method has been chosen for a particular parameter inference method. It is predominantly from this perspective that we discuss state inference methods, before we apply the gained insights to parameter inference methods in Chapter 3.

We first (Section 2.2) describe and classify the SMC methods for state inference and likelihood estimation. A subset of these methods are then applied in Section 2.3 to a simplified version of the seal model (the 2-state model described in Section 1.2.6) using simulated data in order to determine which algorithm to use as a building block in the more complex parameter inference methods described in Chapter 3. We use the simplified model to enable us to test the methods thoroughly in feasible computation time. Most results can be generalised to these more complex models but the computational effort needs to be adjusted for the 7-state and the full model by increasing the number of so-called *particles* (Sections 2.4 and 2.5). We also compare in Section 2.5 two different formulations of the full model to study the effect of factorising the model in different blocks. We conclude (Section 2.6) with a discussion of the results and their implications for state and parameter estimation.

For the methods that are described in Section 2.2 but not considered in the simulation study, there are either existing results that can be directly applied to our case, or we discuss the theoretical considerations that rule out the use of these methods for the seal model.

## 2.2 Review of SMC Methods for State Inference and Likelihood Estimation

In this section, we present a survey of several Sequential Monte Carlo (SMC) algorithms for state inference and likelihood estimation. Most of these methods build on the so-called *bootstrap filter* (BF). We therefore start by introducing the bootstrap filter and the technique of importance sampling on which it relies. We then introduce various ways in which elements of this filter can be extended and improved. The result is a methodological toolbox that allows various techniques to be combined in a manner suited to the challenges posed by a particular inference problem.

### 2.2.1 Importance Sampling

Importance sampling is a Monte Carlo procedure that can both estimate expected values expressed as integrals and yield samples from a particular probability distribution. It is a central building block of sequential Monte Carlo methods and we therefore introduce it here briefly, both as a reminder to the reader and to set up the notation that will be used from here on. The description here follows Chapter 8 in Chopin and Papaspiliopoulos (2020) where an extensive discussion of this technique can be found.

Monte Carlo integration is a numerical method to approximate quantities that can be written as an expected value,

$$\mathrm{E}_p(\phi(X)) = \int \phi(x)p(x)dx, \tag{2.1}$$

where $p$ is the so-called *target density* of the random variable $X$. Given an i.i.d. sample of size $N$ from $p$, the quantity

$$\frac{1}{N}\sum_{i=1}^{N}\phi(x_i), \qquad x_i \sim p \tag{2.2}$$

is an unbiased estimator of $\mathrm{E}_p(\phi(X))$ with mean square error

$$\mathrm{E}_p\left(\frac{1}{N}\sum_{i=1}^{N}\phi(x_i) - \mathrm{E}(\phi(X))\right)^2 = \frac{1}{N}\,\mathrm{Var}_p\,\phi(X).$$

Importance sampling (IS) comes into play when generating samples from the target density $p$ is difficult or impossible but generating samples from another density $q$ is feasible and potentially easier and faster. We will call $q$ the proposal density and an essential requirement is that its support includes the support of $p$, or more strictly put, the support of the integrand. In that case, we observe that

$$\begin{aligned}\mathrm{E}_p(\phi(X)) = \int \phi(x)p(x)dx = \int \phi(x)\frac{p(x)}{q(x)}q(x)dx = \\ = \mathrm{E}_q\left(\phi(X)\frac{p(X)}{q(X)}\right).\end{aligned}$$

Since we are able to generate samples from $q$, we can use the Monte Carlo estimator for the expectation in the last term as given in Equation 2.2. Introducing the so-called *importance weight function* $w(x) = p(x)/q(x)$, we obtain the unbiased estimator

$$\hat{E}_p(\phi(X)) = \frac{1}{N}\sum_{i=1}^{N} \phi(x_i)\frac{p(x_i)}{q(x_i)} = \frac{1}{N}\sum_{i=1}^{N} \phi(x_i)w(x_i), \qquad x_i \sim q \tag{2.3}$$

which has a mean square error of

$$\frac{1}{N}\operatorname{Var}_q\left(w(X)\phi(X)\right).$$

A useful measure to quantify the Monte Carlo error of the estimate in an intuitive manner is the effective sample size (ESS, introduced in Kong et al., 1994), defined as

$$\mathrm{ESS}(w(x_i)_{i=1:N}) = \frac{\left(\sum_{i=1}^{N} w(x_i)\right)^2}{\sum_{i=1}^{N} w(x_i)^2}. \tag{2.4}$$

For normalised weights, i.e., if $\sum_{i=1}^{N} w(x_i) = 1$, this can be rewritten as

$$\begin{aligned}\mathrm{ESS}(w(x_i)_{i=1:N}) &= \frac{\left(\sum_{i=1}^{N} w(x_i)\right)^2}{\sum_{i=1}^{N} w(x_i)^2} = \frac{N^2}{N(\mathrm{Var}(w(x_i)) + 1)}\\ &= \frac{N}{\mathrm{Var}(w(x_i)) + 1}.\end{aligned}$$

In the case where $p = q$ and therefore all weights are equal, the ESS equals $N$. At the other extreme, where all but one of the weights are 0, the ESS is 1. The ESS therefore offers an intuitive interpretation to understand the effect of the variance of the weights on the quality of the importance sample.

The estimator in Equation 2.3 is useful in situations where it is impossible to sample from the target density $p$ directly but $p(x)$ can be evaluated for any given $x$. This is called *normalised importance sampling.* We can easily verify that this estimator is unbiased:

$$\begin{aligned}\mathrm{E}_q\left(\frac{1}{N}\sum_{i=1}^{N}\phi(x_i)w(x_i)\right) &= \frac{1}{N}\sum_{i=1}^{N}\mathrm{E}_q\left(\phi(x_i)\frac{p(x_i)}{q(x_i)}\right) &(2.5)\\ &= \int \phi(x_i)\frac{p(x_i)}{q(x_i)}q(x_i)dx_i = \int \phi(x_i)p(x_i)dx_i &(2.6)\\ &= \mathrm{E}_p(\phi(X_i)) &(2.7)\end{aligned}$$

The importance sampling estimator is a useful approximation when evaluating the integral in Equation 2.7 is infeasible but generating samples from the density $q$ with corresponding importance weights $w(x_i) = p(x_i)/q(x_i)$ is possible. In many practical settings, however, the weight function $w = p/q$ can only be evaluated up to a constant. A typical example of this is in Bayesian inference when $p$ is a posterior distribution, for example

$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{\int p(\theta)p(y|\theta)d\theta},$$

for some parameter vector $\theta$ and data vector $y$. Here, it is often easy to evaluate both factors in the numerator for given $\theta$ and $y$ but the normalising constant in the denominator, the marginal density for the data, is an intractable integral. In this case, a variation of importance sampling called *self-normalised IS*, sometimes also referred to as *auto-normalised IS* (e.g., in Chopin and Papaspiliopoulos, 2020), can be used. Instead of using $p$ and $q$ to calculate the importance weight, we use unnormalised versions $p_u$ and $q_u$ of these densities with $p = p_u/Z_p$ and $q = q_u/Z_q$ where $Z_p = \int p_u(x)dx$ and $Z_q = \int q_u(x)dx$. The estimator then becomes (with $1/N$ kept in the fraction for comparability with Equation 2.3)

$$\hat{\mathrm{E}}_p(\phi(X)) = \frac{1/N \sum_{i=1}^{N} \phi(x_i) w_u(x_i)}{1/N \sum_{i=1}^{N} w_u(x_i)}, \qquad x_i \sim q,$$

where

$$w_u(x_i) = \frac{p_u(x)}{q_u(x)}.$$

While self-normalised importance sampling is often the only possible choice in practice, it has the substantial disadvantage that its estimator is biased for any fixed $N$ and is only asymptotically unbiased.

The asymptotic unbiasedness follows from Slutsky's theorem (Slutsky, 1925) since the denominator converges to the normalisation constant $Z_p/Z_q$, and the numerator converges to

$$Z_p/Z_q \, \mathrm{E}_p(\phi(X_i))$$

as $n \to \infty$. It then follows that the ratio of the numerator and the denominator converges to the ratio of their limits. The bias and variance of the estimator are of order $1/N$. For a formal discussion of asymptotic results, see Chopin and Papaspiliopoulos (2020).

We demonstrate the bias of the estimator with an example. We estimate the mean of $p = \mathcal{U}(0, 4)$ through importance sampling with the proposal distribution $q = N(0, 1)$. We note that in practice this is not a good choice for a proposal distribution because it is very unlikely to obtain values close to 4. These values then have large importance weights, leading to a low ESS. For normalised importance sampling (IS), the estimator for the expected value $\mathrm{E}_{\mathcal{U}(0,4)}(X)$ is

$$\hat{E}_{IS}(X) = \frac{1}{N} \sum_{i=1}^{N} x_i \frac{p(x_i)}{q(x_i)}, \qquad X \sim \mathcal{U}(0, 4),$$

where $p$ is the density of the target distribution $\mathcal{U}(0, 4)$ and $q$ is the density of the proposal distribution $\mathcal{N}(0, 1)$. For self-normalised importance sampling (autoIS), we divide by the sum of all weights rather than by $N$:

$$\hat{E}_{autoIS}(X) = \frac{1}{\sum_{i=1}^{N} x_i \frac{p(x_i)}{q(x_i)}} \sum_{i=1}^{N} x_i \frac{p(x_i)}{q(x_i)}, \qquad X \sim \mathcal{U}(0, 4).$$

To demonstrate how the variance and bias (in the case of self-normalised importance sampling) of these estimates decrease with increasing sample size $N$, we show in Figure 2.1 the mean of 100 estimated means for sample sizes $N = 1, ..., 2000$ for both normalised and self-normalised importance sampling. Here, we use the term *particles* for the proposed values in correspondence with the use of this term for the SMC methods introduced in the next section.

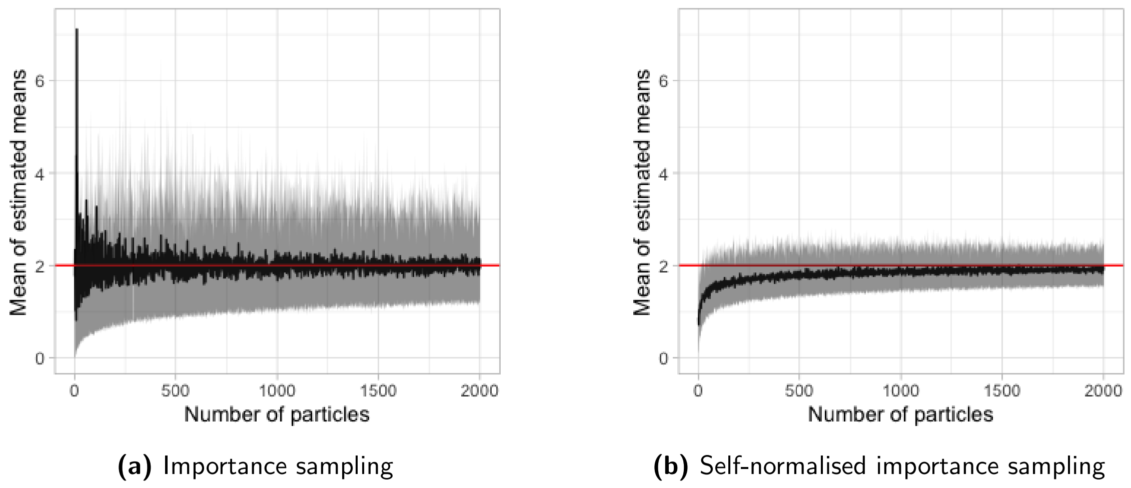


**Figure 2.1:** Means of 100 estimated means of $\mathcal{U}(0,4)$ through importance sampling and self-normalised importance sampling with proposal distribution $\mathcal{N}(0,1)$ for a varying number of particles. The shaded area shows the region between the 10th and 90th percentile.

We note that importance sampling can not only be used to estimate integrals of the form in Equation 2.1 but also to more formally approximate the target density $p(x)$ with

$$\hat{p}_N(x) = \frac{1}{N}\sum_{i=1}^{N} w(x_i)\delta_{x_i}(x), \tag{2.8}$$

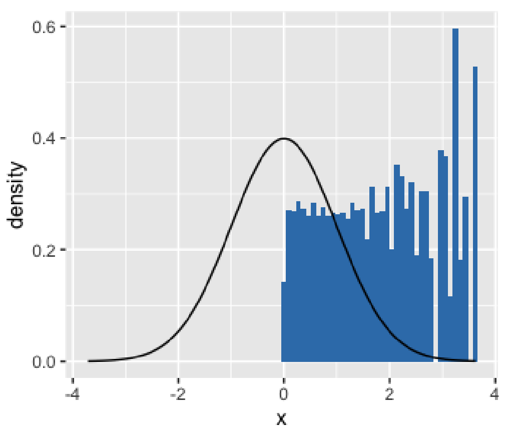
where $\delta_{x_i}(x)$ denotes the Dirac delta mass at $x_i$ (see Doucet and Johansen, 2009) and where

$$w(x_i) = \frac{w_u(x_i)}{\sum_{j=1}^{N} w_u(x_j)}$$

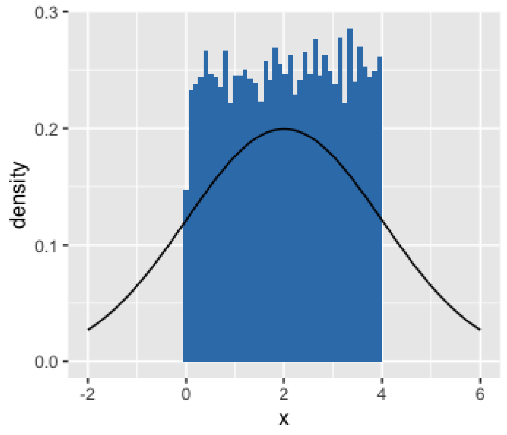
in the self-normalised case.

An important consideration for importance sampling is the choice of proposal distribution. It is generally advisable to choose a proposal distribution $q$ that is similar to $\phi p$ or simply $p$. For normalised importance sampling, the variance of the estimator is minimised if the proposal distribution $q$ is set to $q(x) \propto p(x)|\phi(x)|$ (Proposition 8.2 in Chopin and Papaspiliopoulos, 2020) and a similar theorem is available for self-normalised importance sampling. This theorem is not particularly applicable because the normalisation constant $\mathrm{E}_p(\phi(x))$ cannot be calculated (precisely the reason to use importance sampling in the first place). In addition, we are often interested in a number of different test functions $\phi$ and therefore do not benefit from tuning the proposal distribution to fit only one specific function. In practice, we therefore often try to proposal distributions that are close to the target distribution $p$ which ensures that the importance weights do not vary too much.

A particularly inadequate choice for $q$ is one where the probability mass in the tails $p$ decreases more slowly than it does for $q$. This can lead to very few importance weights with a very high value, resulting in a very high Monte Carlo error variance of the estimator (see Chopin and Papaspiliopoulos, 2020 for the theoretical optimal choice for a proposal distribution and a discussion of its practical implications). We demonstrate this by extending the example above and comparing the effect of using different proposal distributions on the estimation of the mean of the target distribution $\mathcal{U}(0,4)$ through self-normalised



**(a)** Proposal distribution $\mathcal{N}(0, 1)$



**(b)** Proposal distribution $\mathcal{N}(2, 4)$

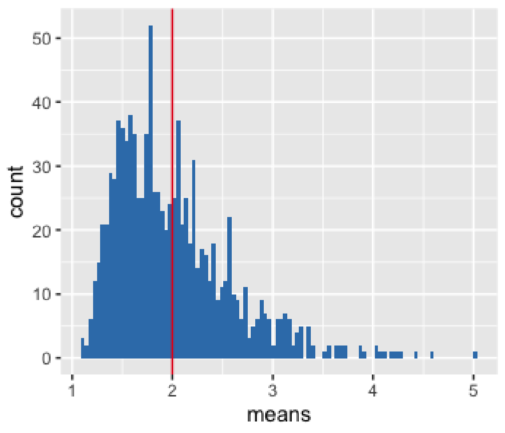
**Figure 2.2:** Histogram of weighted samples of sample size 10000 from $\mathcal{U}(0, 4)$ generated from two different normal distributions as proposal distribution, with the density of their proposal distribution superimposed.

importance sampling. As the first proposal distribution, we use the inadequate choice $\mathcal{N}(0, 1)$ as before, and compare it with the much better suited $\mathcal{N}(2, 4)$. This second proposal distribution has a relatively high density in the interval $[0, 4]$ and therefore does not suffer from having a high variance in the importance weights. Figure 2.2a shows the effect of having weights with a high variance in the importance sample—the sample is not an adequate approximation of the distribution towards the right side of the support. The proposal distribution $\mathcal{N}(2, 4)$ used in Figure 2.2b achieves a much closer approximation with the same sample size. This is reflected in the ESS of the two samples. For Figure 2.2a, it is 165.9, and for Figure 2.2b it is 6625.0. Figure 2.3 shows the effect of using importance samples of different quality on estimating the mean of the target distribution. With a sample size of 5000, the estimated means are very close to the true value of 2 when an adequate proposal distribution is used as in Figure 2.3b but the variance of the estimated mean is much higher with the less suitable proposal distribution in Figure 2.3a.

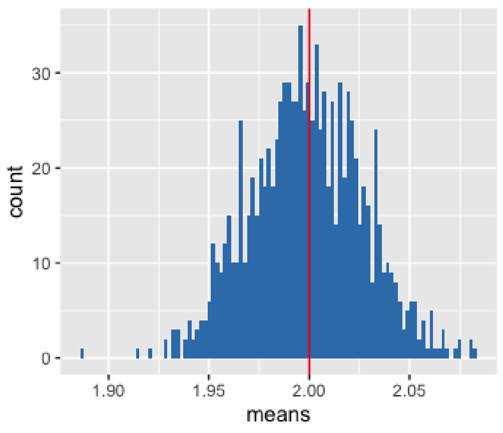
Sometimes it is desirable to have independently sampled values with equal weights, rather than pairs of sampled values and weights $(x_i, w_i)$. We can achieve this by drawing a sample from $\{x_i\}_{i=1^N}$ where the probability to draw a value $x_i$ is proportional to its weight $w_i$ (see Gelman et al., 2013, Chapter 10.4). An overview of methods for generating this sample is given in Section 2.2.5.

### 2.2.2 Bootstrap Filter

We now introduce the first algorithm of a class of methods called *Sequential Monte Carlo* methods. These are based on the idea of importance sampling and are useful for state space models because they exploit the sequential nature of these models to their advantage. The basic building block is the so-called *bootstrap filter* (BF) which was introduced by Gordon et al. (1993). We describe this algorithm in this section and use the following sections to discuss extensions and possible improvements to this filter. As in the definition of the models in Section 1.2, we use $f$ for the transition pdf and $g$ for the observation pdf. Proposal distributions are denoted with $q$, and $p$ is used for all other pdf's.



**(a)** Proposal distribution $\mathcal{N}(0,1)$



**(b)** Proposal distribution $\mathcal{N}(2,4)$

**Figure 2.3:** Histogram of estimated means of $\mathcal{U}(0,4)$ through self-normalised importance sampling with 5000 particles for two different proposal distribution. Note the very different x-axis scales on the two plots.

The bootstrap filter is an algorithm that can be applied to a state space model to produce a weighted sample of filtered states, so a sample from the distribution $p(x_t|y_{1:t})$. The algorithm works by using importance sampling to sequentially produce these estimates. From the model densities, it sequentially simulates a set of possible states, called *particles* and focusses its computational effort only on those particles that match the observations well. We note that because of this use of particles, SMC methods are sometimes also referred to as *particle methods* and filtering algorithms in particular as *particle filters*.

After an initialisation step to generate a sample $\{x_0^{(i)}\}_{i=1,\dots,N}$[1] from $f_0(x_0)$, we choose $p(x_1|y_1)$ as the first target distribution. Rewriting this density gives

$$\begin{aligned} p(x_1|y_1) \propto p(x_1, y_1) &= \int p(x_0, x_1, y_1) dx_0 \\ &= \int g(y_1|x_1) f(x_1|x_0) f_0(x_0) dx_0. \end{aligned}$$

To estimate this integral we choose $q(x_1) = p(x_1) = \int p(x_1, x_0)dx_0 = \int f(x_1|x_0)f_0(x_0)dx_0$ as the proposal distribution. It is straightforward to simulate from this distribution by sampling $X_1^{(i)} \sim f(X_1|x_0^{(i)})$ where we use the sample $\{x_0^{(i)}\}_{i=1}^N$ generated from $f_0(x_0)$ in the initialisation step. The weights are then

$$\begin{aligned} w_1^{(i)} &\propto \frac{p(x_1^{(i)}|y_1, x_0^{(i)})}{p(x_1^{(i)}|x_0^{(i)})} \propto \frac{g(y_1|x_1^{(i)})f(x_1^{(i)}|x_0^{(i)})}{f(x_1^{(i)}|x_0^{(i)})} \\ &= g(y_1|x_1^{(i)}). \end{aligned}$$

The convenience of this choice of proposal distribution now becomes apparent: after cancelling, the only term left in the importance weight function is $g(y_1|x_1^{(i)})$. Evaluating $f(x_1^{(i)}|x_0^{(i)})$ is not necessary: it is enough to be able to simulate from this distribution. If necessary, we can now turn this weighted sample into independently sampled values of

[1]Note that the index to count the sampled values is now a superscript to allow the index for time to remain a subscript.

equal weights by resampling the particles with probability proportional to the importance weights and setting the weights to 1. In the most basic version of the bootstrap filter, this is done at every iteration, but it is also possible to continue with a weighted sample and resample at a later iteration. In that case, the weights are carried over to the next iteration, and multiplied by the new importance weights. While resampling introduces additional Monte Carlo error, it has the advantage of focusing the computational effort in the areas of interest, i.e., the particles with higher density, and eliminates very unlikely particles. The aspects to consider for this decision are discussed in Section 2.2.6, as well as the different options for the resampling scheme in Section 2.2.5, like multinomial sampling where each particle is chosen with probability proportional to its weight.

The steps above generate a weighted sample from the filtering distribution $p(x_1|y_1)$, starting with a sample from $f_0(x_0)$. Following the same steps again, we can generate a sample from the filtering distribution $p(x_2|y_{1:2})$, using the sample from $p(x_1|y_1)$ as a starting point. Iterating this process allows a simulation of a sample for all filtering distributions $p(x_t|y_{1:t})$. Saving the so-called *ancestor particles* or paths of each particle, so $x_{0:T}^{(i)}$, results in samples from the smoothing distribution $p(x_t|y_{1:T})$. When instead the filtering distributions $p(x_t|y_{1:t})$ are required, only the current state is resampled, not the prior states. The bootstrap filter algorithm (with smoothed state trajectories) is summarised in Algorithm 1. In practice, the weights and likelihoods are computed on the log-scale to avoid underflow.

**Algorithm 1** Bootstrap Filter

```
$t \leftarrow 0$                                                          ▷ Initialisation
for $i \leftarrow 1, ..., N$ do
    sample $X_0^{(i)} \sim f(x_0)$
    $w_0^{(i)} \leftarrow 1$                                            ▷ Set importance weights
end for
for $t \leftarrow 1, ..., T$ do
    if resampling condition met then
        Resample state trajectories $(\bar{X}_{0:t-1}^{(i)})$ with probabilities proportional to
        $\{w_{t-1}^{(i)}\}_{i=1...N}$
        $w_{t-1}^{(i)} \leftarrow 1$ for all $i = 1, ..., N$                ▷ Set weights
    else
        $(\bar{X}_{0:t-1}^{(i)}) \leftarrow (X_{0:t-1}^{(i)})$ for all $i = 1, ..., N$
    end if
    for $i \leftarrow 1, ..., N$ do
        Sample $X_t^{(i)} \sim f(x_t|\bar{X}_{t-1}^{(i)})$                   ▷ Forward propagation
        Set $X_{0:t}^{(i)} \leftarrow (\bar{X}_{0:t-1}^{(i)}, X_t^{(i)})$
        Compute $w_t^{(i)} \leftarrow w_{t-1}^{(i)} g(y_t|X_t^{(i)})$          ▷ Weighting
    end for
end for
return a set of weighted state trajectories $\{X_{0:T}^{(i)}, w_t^{(i)}\}_{i=1...N}$
return an unbiased estimator of $p(y_{1:T})$, $\hat{p}(y_{1:T}) = \prod_{t=1}^{T} \frac{1}{N} \sum_{i=1}^{N} w_t^{(i)}$
```

We note that the samples produced by the bootstrap filter are the product of a sophisticated version of self-normalised importance sampling; hence the estimate of the distribution is only asymptotically unbiased and should therefore be treated with caution for low sample sizes $N$. We can, however, establish a central limit theorem for the estimates produced by the bootstrap filter.

**Theorem 1.** *Consider any test function $\phi : \mathcal{X} \to \mathbb{R}$ and its filtered expectation $I_t(\phi) = \mathrm{E}(\phi(X_t)|y_{1:t})$. For $x_t^{(i)}$ and $w_t^{(i)}$ generated by a standard bootstrap filter with multinomial resampling at every time step, as $N \to \infty$*

$$\sqrt{N}\left(\sum_{i=1}^{N} w_t\phi\left(x_t^{(i)}\right) - I_t(\phi)\right) \xrightarrow{d} \mathcal{N}\left(0, V_t(\phi)\right)$$

*with*

$$\begin{aligned} V_t(\phi) &= \sum_{k=0}^{t} \int \frac{p(x_k|y_{1:t})^2}{p(x_k|y_{1:(k+1)})} \left(I_{k,t}(\phi|x_k) - I_t(\phi)\right)^2 dx_k \\ I_{k,t}(\phi|x_k) &= \mathrm{E}\left(\phi(X_t)|y_{(k+1):t,x_k}\right) \end{aligned}$$

*Proof.* See Chopin (2004). □

### 2.2.3 Bootstrap Filter as Likelihood Estimator

Estimating the filtering distributions is only one use of the bootstrap filter. Another is to provide an estimate of the likelihood $p(y_{1:T})$ or $p(y_{1:T}|\theta)$ if we do not treat the model parameters $\theta$ as fixed. The likelihood can be rewritten as

$$p(y_{1:T}) = \prod_{t=1}^{T} p(y_t|y_{1:t-1}). \tag{2.9}$$

Examining the factors on the right-hand side, we see that

$$\begin{aligned} p(y_t|y_{1:t-1}) &= \int p(y_t, x_t|y_{1:t-1})dx_t = \int g(y_t|x_t)p(x_t|y_{1:t-1})dx_t \\ &\approx \frac{1}{N}\sum_{i=1}^{N} w_t^{(i)} = \hat{p}(y_t|y_{1:t-1}). \end{aligned} \tag{2.10}$$

We note that the weights are required in their unnormalised form as the last step is simply using the Monte Carlo estimator introduced in Equation 2.2 where we exploit the fact that the bootstrap filter produces a (potentially weighted) sample from the prediction distribution $p(x_t|y_{1:t-1})$ in the propagation step. Multiplying these estimates results in an estimate for the likelihood:

$$\begin{aligned} \hat{p}(y_{1:T}) &= \hat{p}(y_1)\prod_{t=2}^{T} \hat{p}(y_t|y_{1:t-1}) \\ &= \frac{1}{N^T}\prod_{j=1}^{T}\sum_{i=1}^{N} w_t^{(i)}. \end{aligned} \tag{2.11}$$

Contrary to the estimate of the filtering distribution, and perhaps suprisingly, this estimate is an unbiased estimate of the likelihood, even for the more advanced particle filters that will be discussed below. A formal proof is given in Del Moral (2004) (Theorem 7.4.2) (see Pitt et al., 2012 for a more accessible version of the same proof). Here, we show a

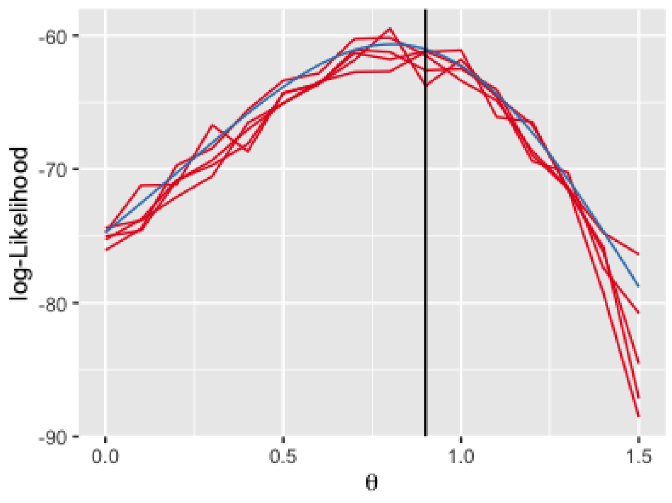


**Figure 2.4:** Likelihood estimates $\hat{p}(y_{1:t}|\theta)$ for the toy example given in Equations 2.12 with true parameter $\theta = 0.9$ (indicated by the black vertical line). The likelihood was estimated with a bootstrap filter (BF, red lines) 5 times at each parameter value with 100 particles each. The true likelihood was calculated with a Kalman filter (KF, blue line).

simplified proof sketch for an SSM with $T = 1$ and using only the basic bootstrap filter as given in Algorithm 1:

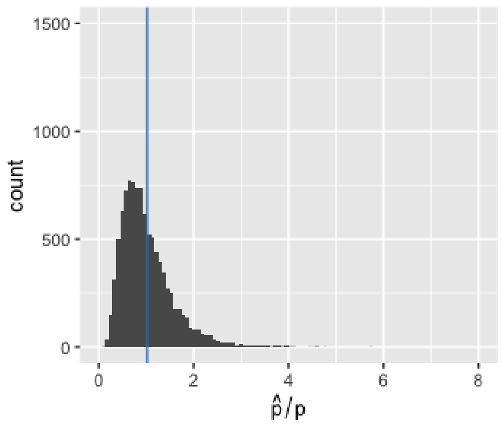
$$\begin{aligned}
\mathrm{E}\left(\hat{p}(y_1)\right) &= \mathrm{E}\left(\frac{1}{N}\sum_{i=1}^{N} w_1^{(i)}\right) \\
&= \frac{1}{N}\sum_{i=1}^{N} \mathrm{E}\left(g(y_1|X_1)\right) \\
&= \int_{\mathcal{X}_0}\int_{\mathcal{X}_1} g(y_1|x_1)f(x_1|x_0)f(x_0)dx_1dx_0 \\
&= \int_{\mathcal{X}_1} g(y_1|x_1)\int_{\mathcal{X}_0} f(x_1,x_0)dx_0dx_1 \\
&= \int_{\mathcal{X}_1} g(y_1|x_1)p(x_1)dx_1 \\
&= p(y_1)
\end{aligned}$$

The relative variance of the likelihood estimate $\hat{p}(y_{1:t}|\theta)/p(y_{1:t}|\theta)$ can be bounded with $C_\theta t/N$ where $C_\theta$ is a constant dependent on the model and the parameter $\theta$ (Whiteley, 2013). This means that while the variance grows with $t$, this can be balanced by increasing the number of particles $N$ linearly with $t$.

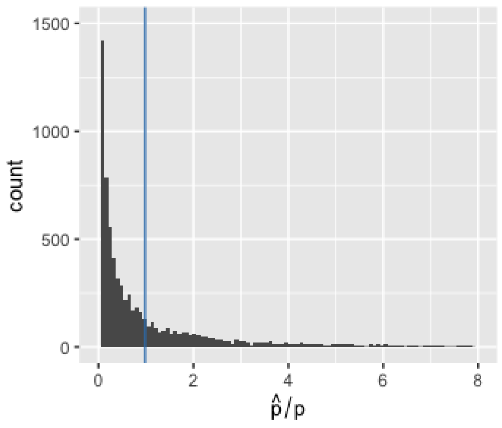
To illustrate the properties of this likelihood estimator, we make use of the following toy example:

$$\begin{aligned}
\text{Transition pdf:} \qquad & x_{t+1} = \theta x_t + \mathcal{N}(0,1) \\
\text{Observation pdf:} \qquad & y_t = x_t + \mathcal{N}(0,1)
\end{aligned} \tag{2.12}$$

Observations from this model were simulated with $\theta = 0.9$ for $T = 30$ observations. As this model is linear and Gaussian, the likelihood can be calculated analytically with the



**(a)** $\theta = 0.9$. The mean of $\hat{p}(y_{1:t}|\theta)/p(y_{1:t}|\theta)$ is 1.001513.



**(b)** $\theta = 1.5$, The mean of $\hat{p}(y_{1:t}|\theta)/p(y_{1:t}|\theta)$ is 1.001043.

**Figure 2.5:** Histogram of 10,000 estimates $\hat{p}(y_{1:t}|\theta)/p(y_{1:t}|\theta)$ for two values of $\theta$. The estimates $\hat{p}$ were produced by a bootstrap filter with 100 particles. The mean of the estimates is indicated with a blue vertical line (🟦).

Kalman filter (see Chapter 4, where this algorithm is studied in detail) which serves as the point of comparison. In Figure 2.4, we can see that the likelihood esimate by the bootstrap filter is quite close to the true likelihood, especially in proximity to the true parameter value. However, the plot already hints at a potential problem. When the likelihood decreases, as it does here for high values of $\theta$, the estimate of the bootstrap filter is less accurate and, in 4 out of the 5 simulation replicates shown, underestimates the true likelihood. This is not due to a bias but rather to the skewness of the distribution of the likelihood estimates which is discussed in the next paragraph.

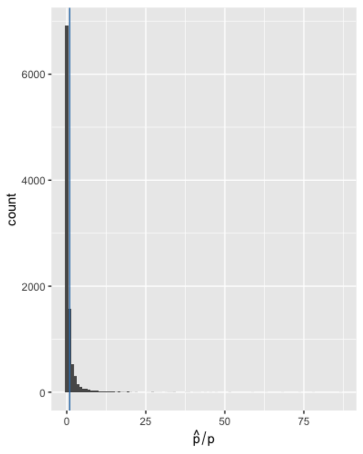
With this example, we also demonstrate that the likelihood estimates produced by the bootstrap filter are unbiased. At two values of $\theta$, the true value 0.9 and 1.5, the likelihood was estimated 10,000 times with a bootstrap filter with 100 particles. Once again, the true likelihood was calculated with the Kalman filter. Figure 2.5 illustrates the unbiasedness of the likelihood estimator, as the mean of the ratio $\hat{p}(y_{1:t}|\theta)/p(y_{1:t}|\theta)$ is very close to 1: 1.001513 at $\theta = 0.9$ and 1.001043 at $\theta = 1.5$. However, it also shows that the estimates exhibit some skewness that increases when $\theta = 1.5$, where the true likelihood is much lower than for $\theta = 0.9$. In Figure 2.6, we illustrate how the skewness decreases as the number of particles increases.

The following sections discuss how the bootstrap filter can be extended and improved.

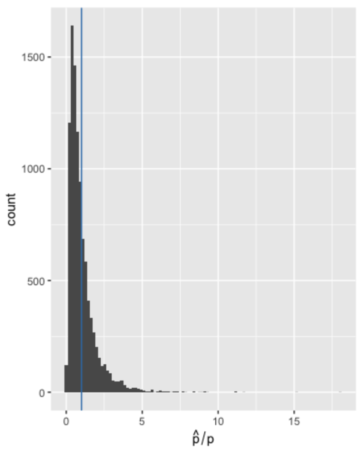
### 2.2.4 Number of Particles

As seen in the previous examples in Figures 2.5 and 2.6, an important decision when using the bootstrap filter is how to choose the sample size, or number of particles, $N$. This depends on many factors and ultimately means balancing computation time with the size of the Monte Carlo error of the filtering distribution and the likelihood estimate.

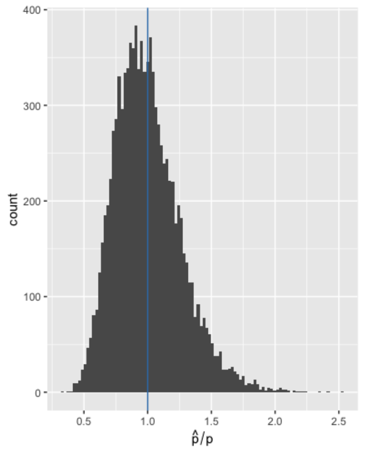
This choice also depends heavily on the purpose of the filter. If it only needs to be run once, for example to produce state estimates for fixed parameters, the number of particles can be increased more liberally. When the likelihood estimate of the filter is used as part of an MCMC algorithm, the number of particles needs to be carefully balanced



**(a)** $N = 20$



**(b)** $N = 100$



**(c)** $N = 1000$

**Figure 2.6:** Histograms of 10,000 likelihood estimates $\hat{p}(y_{1:t}|\theta)/p(y_{1:t}|\theta)$ for $\theta = 1.5$ for numbers of particles $N$. The mean of the estimates is indicated with a blue vertical line (🟦). Note the difference in x-axis scale.

with the number of iterations of the MCMC chain (see Chapter 3). Under some strong assumptions including linearity and normality of the model Sherlock et al. (2015), Doucet et al. (2015) and Golightly et al. (2019) give guidelines on the variance of the log-likelihood estimate, when this estimate is used in a PMCMC scheme. These guidelines recommend values between 2 and 3.3 for the log-likelihood variance at a central value of the posterior distribution, depending on the exact, often strict, assumptions. The result in Sherlock et al. (2015) even assumes that the variance is independent of the parameter value at which the likelihood is evaluated which is decidedly not the case with any of the models studied here. These number can therefore only provide a very rough orientation when the number of particles are chosen in more complex models such as the seal model.

### 2.2.5 Resampling Method

The resampling step in the bootstrap filter introduces randomness and therefore increases the variance of the likelihood estimate $\hat{p}(y_{1:t})$. We can, however, use sampling techniques that reduce the increase in variance. The only requirement for the resampling method is that the expected number of copies stays the same, that is, proportional to its resampling weight (Schön et al., 2018). Various different resampling schemes and their properties are discussed in detail in Douc et al. (2005) and in Murray et al. (2016) which also provides an excellent intuitive visualisation of the different methods in their Figure 1. We present a selection of those most relevant to the seal model case study here. In the following section, we assume that the weights $(w_1, ..., w_N)$ are normalised, i.e., $\sum_{i=1}^{N} w_i = 1$.

**Multinomial** Multinomial resampling is the obvious choice for this step, and arguably the easiest to implement. Because of its simplicity, it is often used as a placeholder for more sophisticated methods in the literature. For each index value, we independently pick from the particles with probability proportional to their weights $(w_1, ..., w_N)$. We can do this by associating each index $k$ with the interval $I_k = \left(\sum_{j=1}^{k-1} w_j, \sum_{j=1}^{k} w_j\right)$ with normalised weights ($\sum_{j=1}^{N} w_j = 1$) and using the convention $\sum_{j=1}^{0} w_j = 0$.We then sample $N$ independent uniform variables $U_i \sim \mathcal{U}(0, 1)$ and select for each the index $k$ with $U_i \in I_k$.

**Stratified** Instead of sampling the indices independently from $\mathcal{U}(0,1)$, we sample $U_i \sim \mathcal{U}(\frac{i-1}{N}, \frac{i}{N})$. As before, we select the index $k$ with $U_i \in I_k$. This enforces a more uniform selection of the particle indices.

**Systematic** Taking the scheme above one step further, we can space the $U_i$ evenly. We sample $U_1 \sim \mathcal{U}(0, 1/N)$ and set $U_i = U_1 + (i-1)/N$.

**Residual** This method uses the expected number of copies for each index, which is $Nw_i$. Each index $i$ is copied $C_i = \lfloor Nw_i \rfloor + \bar{C}_i$ times, where the first term denotes the rounded down expected value. The terms $\bar{C}_1, ..., \bar{C}_N$ are distributed according to $Mult(N - R; \bar{w}_1, ..., \bar{w}_N)$, with $R = \sum_{i=1}^{N} \lfloor Nw_i \rfloor$ and

$$\bar{w}_i = \frac{Nw_i - \lfloor Nw_i \rfloor}{N - R}.$$

In Douc et al. (2005), it is stated that stratified, systematic and residual resampling lead to comparable results but that multinomial resampling should be avoided due to its increased Monte Carlo variance.

### 2.2.6 Resampling Schedule

In the bootstrap filter as described in Algorithm 1, one choice that needs to be made is whether to resample the particles at every iteration. If not, some criterion needs to be chosen to determine at each iteration whether resampling occurs. Before examining the options for this, we discuss heuristically the advantages and disadvantages of resampling. Each resampling step introduces additional Monte Carlo variance, which should usually be limited as much as possible. Skipping the resampling step could also be helpful if there are outliers or other unexpected behaviour in the observations $y_{1:T}$. Rather than immediately discarding a particle that does not fit the observation, skipping the resampling step allows the particle to continue on. If the observations exhibit an unexpected behaviour, this particle might then display a state trajectory that fits the observations quite well overall compared to other particles that were closer to the observations earlier on in the time series.

On the other hand, not resampling means that computation time is wasted on particles that do not fit the observations at all and have low weight. More formally this means that the variance of the importance weights is high and that the ESS is low. An extreme case of this is *sequential importance sampling*, where no resampling takes place but instead the weights are multiplied at every step. While this is theoretically correct in that it produces an asymptotically unbiased estimate of the filtered state distribution and an unbiased estimate of the likelihood, these estimates usually have a very large Monte Carlo error because of the low ESS (see Doucet et al., 2000 for a discussion).

Another consideration when determining the resampling schedule is the potential computing time reduction achieved by resampling less often. In one example in Chopin and Papaspiliopoulos (2020) (Chapter 10.5.1) a reduction of up to 30% computation time was achieved by resampling parsimoniously.

We now introduce a few common choices when determining when and how often to perform a resampling step.

**Always** Resampling at every step is the easiest to implement and requires no additional tuning by the user. This is what is done in a standard bootstrap filter procedure. In the literature, this is often used as a placeholder for more sophisticated schedules.

**Never** As mentioned above, it is possible to never resample the particles and instead build up the weights by multiplying the new weight at every step (sequential importance sampling). In practice, this leads to large variance in the weights and a low ESS, also referred to as *particle degeneracy*, and is not recommended.

**Deterministic schedule** Liu (2001) suggests choosing a fixed interval $t_0$ and hence to sample at times $t_0, 2t_0, 3t_0, \ldots$. In this case, the resampling interval $t_0$ needs to be tuned to the particular inference problem.

**Dynamic schedule** Alternatively, Doucet et al. (2000) uses the ESS (see Equation 2.4) as a measure for degeneracy. One can then set a threshold $\mathrm{ESS}_{\mathrm{min}}$ (for example $0.5N$) and resample whenever $\mathrm{ESS} < \mathrm{ESS}_{\mathrm{min}}$. Liu (2001) even suggests using a series of time-dependent thresholds $\mathrm{ESS}_{\mathrm{min}}(t)$. Convergence results using a fixed ESS-based threshold can be found in Moral et al. (2012).

It is often recommended to use the ESS at each time step as a dynamic criterion for resampling. Thresholds of $0.5N$ (e.g., Zhou et al., 2016 and Doucet and Johansen, 2009) or $0.9N$ (Finke et al., 2019) are commonly used but this value depends on the specific inference problem.

### 2.2.7 Auxiliary Particle Filter

An extension of the bootstrap filter is the *auxiliary particle filter* (APF, Pitt and Shephard, 1999). The algorithm is given in Algorithm 2, with the notation closely following that of Schön et al. (2018). The APF introduces two ways to improve the bootstrap filter.

First, rather than simulating the particles by sampling from the initial state density $f_0(x_0)$ and the process density $f(x_t|x_{t-1})$, it allows the choice of proposal densities $q(x_0)$ and $q(x_t|x_{t-1}, y_t)$. Importantly, the proposal density for $x_t$ can depend on the new observation $y_t$ and so guide the particles in the right direction rather than blindly simulating forward. The weights then become

$$w_i \propto \frac{p(x_t^{(i)}|y_t, x_{t-1}^{(i)})}{q(x_t^{(i)}|y_t, x_{t-1}^{(i)})} \propto \frac{g(y_t|x_t^{(i)})f(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|y_t, x_{t-1}^{(i)})}.$$

This shows the drawback of introducing a different proposal density $q$. Here, the factor $f(x_t^{(i)}|x_{t-1}^{(i)})$ does not conveniently cancel as in the bootstrap filter and therefore the ability to evaluate the process density is required. With the goal of reducing the Monte Carlo error of our estimate and therefore having a low variance in the weights, the theoretical optimal choice for a proposal distribution is

$$q(x_t|x_{t-1}, y_t) = p(x_t^{(i)}|y_t, x_{t-1}^{(i)}).$$

In this case, the weights cancel to $w_i = 1$ and the variance of the weights is therefore 0. An important class of state-space models where this is possible is the class of normal dynamic linear state-space models (NDLM). However, for this class there are also other deterministic inference algorithms available, for example the Kalman filter (see Section

4.2.2), and we do not need to rely on Monte Carlo methods. For more complicated models, we can almost never sample from $p(x_t^{(i)}|y_t, x_{t-1}^{(i)})$. We can, however, attempt to design proposal distributions that are close to $p(x_t^{(i)}|y_t, x_{t-1}^{(i)})$.

The second place where the APF introduces some flexibility is the option of adjusting the resampling weight with an adjustment multiplier. Rather than resampling with probability proportional to $w_t^{(i)}$, we calculate the adjustment multiplier $v_t^{(i)} = v(x_{t-1}^{(i)}, y_{1:t})$ and resample with probability proportional to $w_t^{(i)} v_t^{(i)}$. As with the proposal distribution $q(x_t|x_{t-1}, y_t)$, this allows us to incorporate knowledge about $y_t$ when generating the particles $x_t^{(i)}$.

The importance weights in the APF ensure that the particle filter delivers a weighted sample of state trajectories with the desired target distribution $p(x_{0:t}|y_{1:t})$. The weights compensate for the fact that $f(x_t|x_{t-1})$ was not used as the proposal density and $g(y_t|x_t)$ not as the resampling weight. In contrast to the bootstrap filter, they are therefore not set to 1.

**Algorithm 2** Auxiliary Particle Filter

```
  $t \leftarrow 0$                                                    ▷ Initialisation
  for $i \leftarrow 1, ..., N$ do
      sample $X_0^{(i)} \sim q(x_0)$
      $w_0^{(i)} \leftarrow \frac{f_0(X_0^{(i)})}{q(X_0^{(i)})}$                        ▷ Set importance weights
  end for
  for $t \leftarrow 1, ..., T$ do
      for $i \leftarrow 1, ..., N$ do
          $v_{t-1}^{(i)} \leftarrow v(X_{t-1}^{(i)}, y_t)$                  ▷ Adjustment multiplier
          resample state trajectories $(\bar{X}_{0:t-1}^{(i)})$ with probabilities proportional to $\{w_{t-1}^{(i)} v_{t-1}^{(i)}\}_{i=1}^N$
          sample $X_t^{(i)} \sim q(x_t|\bar{X}_{t-1}^{(i)}, y_t)$               ▷ Forward propagation
          $X_{0:t}^{(i)} \leftarrow (\bar{X}_{0:t-1}^{(i)}, X_t^{(i)})$
          $w_{t-1}^{(i)} \leftarrow \frac{g(y_t|X_t^{(i)}) f(X_t^{(i)}|X_{t-1}^{(i)})}{q(X_t^{(i)}|X_{t-1}^{(i)}, y_t) v_{t-1}^{(i)}}$          ▷ Set weights
      end for
  end for
  return a set of weighted state trajectories $\{X_{0:T}^{(i)}, w_t^{(i)}\}_{i=1}^N$
  return an unbiased estimator of $p(y_{1:T})$ :
          $\hat{p}(y_{1:T}) = \left(\frac{1}{N}\sum_{i=1}^N w_0^{(i)}\right)\left(\prod_{t=1}^T \frac{1}{N}\sum_{i=1}^N w_t^{(i)} v_{t-1}^{(i)}\right)$
```

As with the bootstrap filter, the APF also produces an unbiased estimate of the likelihood $p(y_{1:T})$ in addition to the weighted state trajectories, although this estimate needs to be adjusted to reflect the use of different resampling weights. At time $t$, the algorithm delivers an unbiased estimator of $p(y_t|y_{1:t-1})$, namely

$$\hat{p}(y_t|y_{1:t-1}) = \frac{1}{N}\sum_{i=1}^N w_t^{(i)} v_{t-1}^{(i)} = \frac{1}{N}\sum_{i=1}^N \frac{g(y_t|X_t^{(i)}) f(X_t^{(i)}|X_{t-1}^{(i)})}{q(X_t^{(i)}|X_{t-1}^{(i)}, y_t)}.$$

The factor $v_{t-1}^{(i)}$ cancels and we therefore need to define the adjustment multiplier only up to a constant of proportionality.

We note that, for ease of notation, resampling occurs at every step in Algorithm 2 but it is easily possible to introduce more flexible resampling schedules as in Algorithm 1.

In the following sections, we discuss different options for the proposal distribution both for simulating the initial particles $x_0^{(i)}$ and for propagating the particles forward to $x_t^{(i)}$, and for the adjustment multiplier.

#### 2.2.7.1 APF: Initialization

We now discuss some options for the initial state proposal density $q(x_0)$.

**Using the initial state density** The most simple choice is to sample from the initial state pdf $f(x_0)$ as in the bootstrap filter. This has the benefit that evaluation of $f_0(x_0)$ is not necessary, as this value cancels in the calculation of $w_0$ and leads to an importance weight $w_0$ of 1.

**Diversifying the initial states** To ensure a greater diversity among the initial sample, we can choose $q(x_0)$ to have a greater variance than $f_0(x_0)$. This sometimes leads to a better ESS in the first iterations of the algorithm in cases where the initial state density of the model does not match the data. We can do this by flattening the initial state density and for example use $q(x_0) \propto (f_0(x_0))^\rho$, where $\rho \in (0, 1)$. When the complete seal model was fitted in Thomas et al. (2019), the diversity was increased by first sampling $\tilde{x} \sim f_0(x_0)$ and then $X_0 \sim \mathcal{U}(\tilde{x}/1.3, 1.3\tilde{x})$. More generally, we can replace the factor 1.3 with any multiplier $c > 1$ or use other more sophisticated distributions to increase diversity.

#### 2.2.7.2 APF: Forward propagation

The options for the proposal distribution in the propagation step are similar to the initialization step, but we can now also utilize the observation $y_t$.

**Using the transition density** As in the bootstrap filter, we can use $q = f$ and so sample from the transition density. This is the only choice that avoids evaluation of $f$ and therefore the only available option when evaluation of $f$ is impossible.

**The optimal proposal** If available in closed form, we can use

$$q(x_t|x_{t-1}, y_t) = p(x_t|x_{t-1}, y_t)$$

and so sample from the distribution of $x_t$ conditional on $x_t$ and $y_t$. This is desirable because it minimizes the variance of the weights and therefore maximizes the ESS (see Chopin and Papaspiliopoulos (2020), Theorem 10.1). In most cases, this is not possible, but staying close to this is desirable.

**Kalman filter** For linear Gaussian SSMs, the Kalman filter can calculate the exact density $p(x_t|x_{t-1}, y_t)$. For non-linear non-Gaussian SSMs, variants of the Kalman filter exist (e.g., the Unscented Kalman filter, see Section 4.5.1) that give approximations to this density. For the forward propagation, we can sample from this exact or approximated density. This is described in detail in Wan and van der Merwe (2001).

#### 2.2.7.3 APF:Adjustment multiplier

The adjustment multiplier $v_{t-1}^{(i)} = v(x_{t-1}, y_t)$ allows us to incorporate information about the next observation $y_t$ when resampling. A few possible choices for this multiplier are given here.

**No adjustment** The simplest option here is to set $v_{t-1} = 1$, as is the case in the bootstrap filter.

**The optimal adjustment** Pitt and Shephard (1999) recommend using

$$v(x_{t-1}, y_t) = p(y_t|x_{t-1}) = \int g(y_t|x_t) f(x_t|x_{t-1}) dx_t. \tag{2.13}$$

In most practical scenarios this density is not tractable and therefore cannot be used. However, the theoretical implications of this choice are interesting to study and have been discussed in Poyiadjis et al. (2005). Since this is the optimal choice, any $v$ that approximates this can be a good possible candidate for the adjustment multiplier.

**Expected value** Instead of integrating over all possible states $x_t$ in Equation 2.13, we can simplify the problem and only consider the expected value $\mathbb{E}_f(X_t|x_{t-1})$. Liu and West (2001) use this approach and set $v(x_{t-1}, y_t) = g(y_t|\mathbb{E}_f(X_t|x_{t-1}))$. This tends to be much easier calculate than the optimal adjustment and is often a very helpful approximation. However, it depends a lot on the process density and would not work for e.g., a bimodal process density.

**Tempering the weights** Liu (2001) suggests to resample according to weights $w_i^\rho$ with $\rho \in (0, 1)$. In our notation this corresponds to the adjustment multiplier $v_{t-1}^{(i)} = w_i^{\rho-1}$. This strategy allows us to balance the need for a diverse sample with the need to keep only promising samples via $\rho$. A typical value for $\rho$ is $1/2$ and Thomas et al. (2019) used $\rho = 1/4$.

### 2.2.8 Smoothing

Often, we are not only interested in the filtering density $p(x_t|y_{1:t})$ but rather in the joint smoothing density $p(x_{1:T}|y_{1:T})$ or marginal smoothing density $p(x_t|y_{1:T})$. In the bootstrap filter and auxiliary particle filter, an estimate of the smoothing density is available simply by using the trajectories $\{X_{1:T}^{(i)}\}_{i=1}^N$ that are part of the output of the algorithm. However, the resampling step leads to very few unique states at early points of the time series. Therefore, the approximation suffers from particle degeneracy and has a high Monte Carlo variance. There are several methods to try and reduce this variance.

One is the *forward filter/backward simulator* suggested by Lindsten and Schön (2013). This algorithm starts with a regular particle filter algorithm but saves all states that are generated at each iteration. Then it generates new trajectories by starting with the last states $\{X_T^{(i)}\}_{i=1}^N$ and sampling a new ancestor $X_{T-1}^{(j)}$ particle from all states that were generated by the particle filter at time $T-1$ (whether subsequently resampled or not). The importance weight for this sampling step is a product of the weight that was assigned to the state when the particle filter was first run, and the transition density from $f(X_T^{(i)}|X_{T-1}^{(j)})$.

Harvey (1990) describes *Forward Backward Smoothing* as a method to obtain the smoothing density $p(x_t|y_{1:T})$. There, it is described with respect to the Kalman filter but it can be applied to other particle filter algorithms as well. The idea is to run the particle filter forwards to time $t$ to obtain $p(x_t|y_{1:t})$, but to also reverse the whole time series and run the filter backwards from $T$ to $t$ to obtain $p(x_t|y_{t:T})$. This is only possible under certain conditions, the most obvious being that we need to be able to reverse the transition pdf.

Another approach to this problem are fixed-lag approximations, first suggested in this context by Kitagawa and Sato (2001). The underlying idea is that $p(x_{0:t}|y_{0:T}) \approx p(x_{0:t}|y_{0:n+L})$ where $L$ is a large enough positive integer, that is, that new observations do not bring a lot of new information about states that are more than $L$ time steps in the past. For the particle filter, this can be implemented by only resampling the last $L$ components of the trajectory rather than all of it. The difficulty is tuning the lag $L$ with respect to the underlying model.

A similar concept lies behind a popular extension to the auxiliary particle filter, the *Resample-Move* algorithm (Gilks and Berzuini, 2001). This algorithm moves the last $L$ states by sampling from a Markov kernel $K$ of invariant distribution $p(x_{1:t}|y_{1:t})$ but updating only the last $L$ iterations. As the fixed-lag approximation in the previous paragraph, it suffers from the same difficulty of correctly tuning the lag $L$.

We note that none of these techniques are necessary for the likelihood estimate, as we can see from Equations 2.9 and 2.10 that only the filtering distributions are necessary. However, we will return to these techniques when state inference is required.

## 2.3 Simulation Study

For the seal model, the parameter values are unknown and state inference with fixed parameter values is thus not in itself the main goal. However, many of the more complex SMC algorithms for parameter estimation contain steps that require state inference methods for fixed parameters. Often, these methods are necessary to compute an estimate of the likelihood $p(y_{1:T})$ (or $p(y_{1:T}|\theta)$ if we make the dependence on a parameter $\theta$ explicit). In order to optimize this building block of the more complex algorithms, we need to determine which of the SMC methods introduced in the previous section are most useful in the particular case of the seal model. We isolate the likelihood estimation problem and conduct a simulation study to determine which variant of a particle filter should be used for this problem.

### 2.3.1 Simulation Methods

For this study, we wanted to be able to run any algorithm many times in order to reduce the Monte Carlo error and to increase our certainty in the conclusions we draw. As the complete seal model takes a relatively long time to fit, a simplified version of the model was chosen for the majority of the simulations: the 2-state model as introduced in Section 1.2.6. We used this simplified model as a baseline model to determine which of the algorithms above produce the most precise likelihood estimate (Section 2.3.2). The outcomes could then be used as a starting point when the algorithm was adapted for the more complex models (Section 2.4 and 2.5).

The baseline algorithm used to estimate the likelihood for each simulated dataset in the 2-state model was a simple bootstrap filter (see Algorithm 1). We then tuned the algorithm to obtain the best possible likelihood estimate and identified which of the algorithm variants introduced in Section 2.2 could be applied to the seal model. While many of the variants and tuning parameters can be applied at the same time, here only one factor was studied at a time instead of a complete factorial study. In total we undertook four simulations to study the effect of four factors: (1) the number of particles, (2) resampling using an ESS-dependent threshold, (3a) APF with an expected value adjustment multiplier, and (3b) APF with tempering of the weights. Each simulation build on the results of the previous and uses the optimal setting for the factor studied there.

For each factor, we simulated a time series of $T = 50$ observations and studied the effect of varying that factor with three parameters vectors: the one that was used to create the observations, and two that are further away and have a much lower likelihood given the simulated data. To be useful in the context of parameter inference, any likelihood estimation algorithm has to be able to reliably produce estimates not just for parameters with a high true likelihood but also for parameters with lower true likelihood.

The true parameter values used to simulate the observations were chosen such that the observations show a similar behaviour to those from the complete seal model, and show the population growth both in its exponential growth phase and once carrying capacity is reached. The incorrect parameter values were chosen in a subjective manner to be values in a region with non-negligible posterior weight but also not close to the true value. The values of the three parameter vectors can be seen in Table 2.1. To determine the Monte Carlo variance of the algorithms, each state inference algorithm was run multiple times. For Factor 1, 100 likelihood estimates were produced for each number of particles and each of the three parameter vectors. For the remaining three factors, the runtime was decreased due to the lower number of particles and therefore, 1000 likelihood estimates were produced for each setting to determine the Monte Carlo variance.

To ensure that any observed effects were not just due to the specific observations, we repeated each simulation with a second set of observations, created in the same way as described above. Here, we only show the results of the first set of observations unless they differ greatly from the second set. The results of the second set of observations can be found in Appendix B. The two sets of simulated observation can be seen in Figure 2.7.

| Parameter | True parameter | False parameter 1 | False parameter 2 |
|---|---|---|---|
| maximum pup survival $\phi_{p,\max}$ | 0.48 | 0.30 | 0.55 |
| adult survival $\phi_a$ | 0.9 | 0.95 | 0.85 |
| fecundity $\alpha$ | 0.8 | 0.9 | 0.75 |
| Carrying capacity $\chi$ | 2500 | 3200 | 2300 |
| dens dep. $\rho$ | 6 | 8 | 5 |
| CV precision $\tau$ | 100 | 120 | 90 |

**Table 2.1:** Parameter vectors used in the simulation study

In order to compare different algorithms, we need a criterion to determine which of the algorithm is preferable for our purposes. The main reason for using state inference algorithms is that they provide us with an unbiased estimate of the likelihood of the ob-

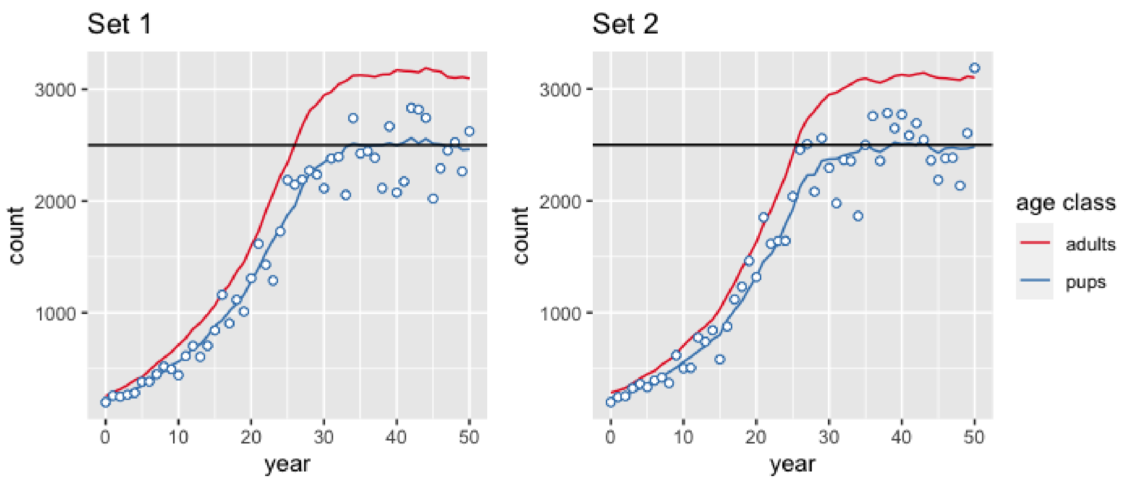


**Figure 2.7:** Two sets of pup and adult numbers used in the simulation study, generated using the 2-state model with the true parameter in Table 2.1. Observations of pup numbers are displayed as blue circles and the pup carrying capacity is indicated by a black horizontal line.

servations $p(y|\theta)$. Hence, we want to reduce the Monte Carlo variance of that estimate. Because the variance is scale-dependent in our case, we chose the coefficient of variation (CV) (standard deviation divided by mean) as the criterion to determine which methods work best. We do not know the true mean and therefore use the estimated mean to calculate the CV. This criterion has been used for similar studies in the literature, e.g., by White et al. (2016) in their simulation study to find the optimal number of particles. A second justification for choosing the CV as criterion is as follows. In many parameter inference methods, we use likelihood ratios to determine which parameter vector works best. For example, in the PMMH (Particle Marginal Metropolis Hastings) algorithm (Algorithm 6), the acceptance probability contains this ratio as a factor. Assuming we have two parameters $\theta_1$ and $\theta_2$, we calculate how much the ratio changes if we overestimate $L(\theta_1)$ by one standard deviation, so $\hat{L}(\theta_1) = L(\theta_1) + sd$, where $sd$ is the Monte Carlo standard deviation of the likelihood estimate produced by a particle filter. This is

$$\frac{\hat{L}(\theta_1)/L(\theta_2)}{L(\theta_1)/L(\theta_2)} = \frac{\hat{L}(\theta_1)}{L(\theta_1)} = \frac{L(\theta_1) + sd}{L(\theta_1)} = 1 + cv,$$

where $cv$ is the coefficient of variation of the likelihood estimator. We are therefore interested in keeping the CV as low as possible, and used this as the criterion. We also examined the variance of the log-likelihood and aimed to keep this at least as low as 2 for the true parameter. However, this threshold was only used as a guideline because the assumptions used in the theoretical derivation of the various thresholds discussed in Section 2.2.4 are not met by the seal model.

Now that the general set-up of the study has been described, we discuss the four studied factors. A summary can be found in Table 2.2.

First (Factor 1), the effect of the number of particles on the likelihood estimate was studied, from 1-30,000. The tested values were 1, 3, 10, 30, 100, etc., because 3 is approximately halfway between 1 and 10 on the log-scale and we wanted to cover multiple orders of magnitude for $N$ in the study. We also used the mean of the 100 estimates with $N = 30,000$ particles as a close estimate of the true likelihood of the three parameter vectors. Finally, the computational cost of running the algorithm with an increasing number of

| | Studied Factor | Partices $N$ | ESS threshold $\text{ESS}_{\text{min}}/N$ | Adjustment multiplier |
|---|---|---|---|---|
| Factor 1 | Number of particles | 1-30,000 | fix at 80% | - |
| Factor 2 | Resampling Schedule | fix at 100 | 0-100% | - |
| Factor 3a | Adj. multiplier | fix at 100 | fix at 80% | expected value |
| Factor 3b | Adj. multiplier | fix at 100 | fix at 100% | tempering, exponent 1/16-2 |

**Table 2.2:** Choices for each of the four simulations

particles $N$ was considered. While the computational complexity of the bootstrap filter is $\mathcal{O}(N)$ (Chopin and Papaspiliopoulos, 2020), for small numbers of particles the overhead of running the algorithm becomes a larger part of the overall runtime and the algorithm is therefore less efficient. The consequence of this was studied by recording the runtime of the bootstrap filter per particle for each of the selected numbers of particles.

After a reasonable number of particles had been determined, we then investigated letting the resampling step depend on the ESS, testing thresholds of $\text{ESS}_{\text{min}}/N$ from 0 to 100% (Factor 2). For each of these values, the particles were only resampled when their ESS fell below the threshold. A threshold of 1 corresponds to a standard bootstrap filter where resampling happens at every step and a threshold of 0 corresponds to sequential importance sampling with no resampling.

Next, we turned towards the auxiliary particle filter and the approach of changing the resampling weights in two ways. First, we examined the effect of 'looking ahead" when resampling the particles (Factor 3a). Instead of using only the current importance weight when resampling a particle, we considered the expected value of the state of that particle at the next time step. In the 2-state model this expected value $\mathbb{E}(x_{t+1}|x_t)$ was straightforward to calculate, as was the likelihood $g(y_{t+1}|\mathbb{E}(x_{t+1}|x_t))$. This likelihood was multiplied with the importance weight to obtain the resampling weight. After the resampling step, we divided by this factor to revert back to a correct importance weight. Adjusting the resampling weight in this way is not implemented in `nimble` and therefore bespoke code was written in `R`. To assess a potential increase in computing time due to this change, we tracked the computing times for the BF in R code and in compiled nimble code, and for the APF only in R code.

Second, the resampling weights were tempered by exponentiating the weights with values $\rho = 2, \sqrt{2}, 1, \sqrt{2}/2, 1/2, ..., 1/16$ (Factor 3b). Using values of $\rho < 1$ dampens the effect of the weights, and results in particles with low weights being discarded less quickly, keeping the set of particles more diverse and allowing for potentially unexpected developments of the observations in future time steps. For completeness' sake, the opposite situation was also examined. Using $\rho > 1$ leads to a resampling step which disproportionately favours particles that match the observation well. Previously, Thomas and Harwood (2008) used $\rho = 1/4$ and another typical value is $\rho = 1/2$.

The simulations were implemented in `R`. For the investigations of the number of particles and the ESS threshold, we used the `R` package `nimbleSMC` (NIMBLE Development Team, 2021). Changing the resampling weights is not implemented in `nimble` yet, and so bespoke code was written for the algorithms that adjust these weights by using the expected value or by tempering them. These simulations do not cover all of the methods mentioned in Section 2.2.

In the remainder of this subsection we give a justification for each of the omitted methods.

**Resampling Method** We did not include a trial of these methods in the simulation study in Section 2.3 but instead relied on Douc et al. (2005). Code used in this thesis that is written by the author will usually use residual resampling but this is not necessarily the same for other software packages. As the differences between the resampling methods, other than the high variance for multinomial resampling, are minor (Douc et al., 2005), we usually use multinomial resampling as a placeholder in any description of the algorithms and omit any details on which exact method is used.

**Resampling Schedule** For the resampling schedule, using a deterministic schedule has a similar effect as a dynamic schedule only that it does not incorporate any information on particle diversity at any given time step and this is therefore not studied.

**APF: Initialisation** In the initialization step, the diversifying step employed by Thomas et al. (2019) is not necessary for the 2-state model, as the initialisation is more straightforward and does not require sequentially simulating 6 age groups.

**APF: Forward propagation** Any improvement on the forward propagation step in the APF requires that we are able to evaluate the process density $f$. In the seal model, this requires a relatively lengthy computation. The calculation of the probability of the new number of pups is straightforward as it is only the result of the binomial distribution of the number of births by the number of adults at the previous time step:

$$f(x_{t,0}|x_{t-1}) = \text{Bin}(x_{t,0}|x_{t-1,1}, \alpha).$$

However, calculating the probability of the new number of adults is more complicated, as this number is the sum of the number of survived seals that were already adults at the previous time step and the number of survived female pups. We therefore need to sum over all possible numbers of survived female pups, denoted by $u_{t,s}$ as in Section 1.2.3.2:

$$\begin{aligned} f(x_{t,1}|x_{t-1}) &= \sum_{u_{t,s}=0}^{x_{t-1,0}} f(x_{t,1}, u_{t,s}|x_{t-1}) = \sum_{u_{t,s}=0}^{x_{t-1,0}} f(x_{t,1}|u_{t,s}, x_{t-1}) f(u_{t,s}|x_{t-1}) \\ &= \sum_{u_{t,s}=0}^{x_{t-1,0}} \text{Bin}(x_{t,1} - u_{t,s}|x_{t-1,1}, \phi_a)\text{Bin}(u_{t,s}|x_{t-1,0}, 0.5\phi_{p,t-1}). \end{aligned}$$

The number of seals in the data and our simulations are often in the 100s or 1000s and therefore so is the number of evaluations of the density of the binomial distribution. This is prohibitive when this sum needs to be evaluated many times as would be the case in the particle filter. Thus, no changes to the forward propagation distribution were made as using the process density $f$ is the only choice that avoids its evaluation because the term cancels in the computation of the importance weights (see Step 2(d) in Algorithm 2). We note that we briefly explored using the forward propagation step of the Unscented Kalman filter which is described in Section 4.5.1. However, we could not detect an improvement in the performance of the filter and this change increased the computation time considerably. That approach is therefore not further pursued here.

**Smoothing algorithms** We also did not investigate any smoothing algorithms in this simulation study, as we are only interested in changes in the algorithm that affect the likelihood estimate and not the sampled state trajectories. Whenever samples from the smoothing density are required, we use the naïve approach (as implemented in `nimble`) of simply saving the ancestors of a particle and producing an estimate of the smoothing density with these ancestor particles as the first choice (Michaud et al., 2021). This approach can lead to poor estimates when $t$ is much smaller than $T$. When this approach is not sufficient, we use the forward filter/backward simulator algorithm. For fixed parameters, generating samples from the smoothing density in our model is relatively simple, and this algorithm meets our requirements without much further tuning.

### 2.3.2 Simulation Results and Interpretation

We describe the results for the 2-state model obtained with the first dataset. Since the results obtained with the second simulated dataset (see Figure 2.7) showed no notable difference, these are reported only in Appendix B.1.

#### 2.3.2.1 Factor 1: Number of Particles

The CV of the likelihood and the variance of the log-likelihood estimates can be seen in Table 2.3 and in Figure 2.8. The log-likelihood of the true parameter was estimated to be -320.4. For the first false parameter it was -382.1 and for the second false parameter it was -362.3.

| | True parameter | | False parameter 1 | | False parameter 2 | |
|---|---|---|---|---|---|---|
| $N$ | CV(L) | Var(logL) | CV(L) | Var(log-L) | CV(L) | Var(log-L) |
| 1 | 3.49E+00 | 6.72E+03 | 6.67E+00 | 1.71E+04 | 1.00E+01 | 1.85E+06 |
| 3 | 1.83E+00 | 4.39E+01 | 4.65E+00 | 5.23E+02 | 1.00E+01 | 4.12E+04 |
| 10 | 1.26E+00 | 1.15E+00 | 2.01E+00 | 7.67E+00 | 6.99E+00 | 1.44E+03 |
| 30 | 5.40E-01 | 3.13E-01 | 1.18E+00 | 1.56E+00 | 9.99E+00 | 4.05E+02 |
| 100 | 3.04E-01 | 9.54E-02 | 7.13E-01 | 4.64E-01 | 7.30E+00 | 1.33E+02 |
| 300 | 1.49E-01 | 2.21E-02 | 4.57E-01 | 2.19E-01 | 6.51E+00 | 6.02E+01 |
| 1000 | 1.03E-01 | 1.13E-02 | 2.65E-01 | 6.52E-02 | 9.30E+00 | 2.30E+01 |
| 3000 | 5.39E-02 | 2.91E-03 | 1.27E-01 | 1.55E-02 | 6.07E+00 | 1.29E+01 |
| 10000 | 3.13E-02 | 9.85E-04 | 6.94E-02 | 4.86E-03 | 5.23E+00 | 5.07E+00 |
| 30000 | 1.65E-02 | 2.73E-04 | 3.72E-02 | 1.41E-03 | 2.48E+00 | 3.12E+00 |
| log-L | -320.39 | | -382.09 | | -362.32 | |

**Table 2.3:** Measures of variation of 100 likelihood estimates computed by a standard bootstrap filter with varying numbers of particles $N$. The likelihood was estimated at the true parameter value and at two false parameter values (defined in Table 2.1). The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 30,000$ particles.

As expected, the Monte Carlo variance of the estimate decreased as the number of particles increases. However, this variance was drastically different for the three parameters. It was lowest for the true parameter, which had the highest log-likelihood. The variance of the log-likelihood estimates of the first false parameter was about 4 to 10 times higher than for the true parameter. For the second false parameter, this factor ranged from 275 to almost 12,000. Two things are interesting to note here. First, the likelihood of the second

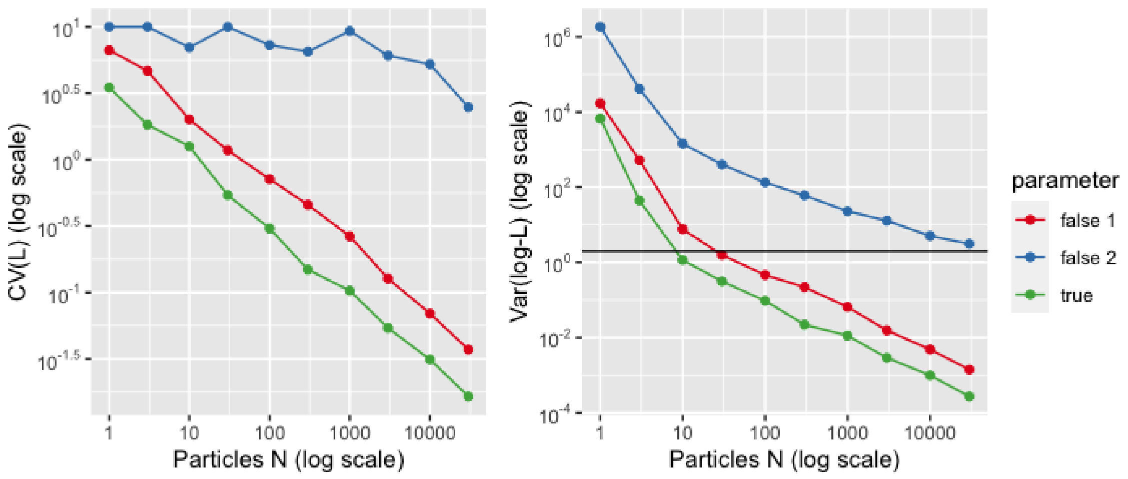


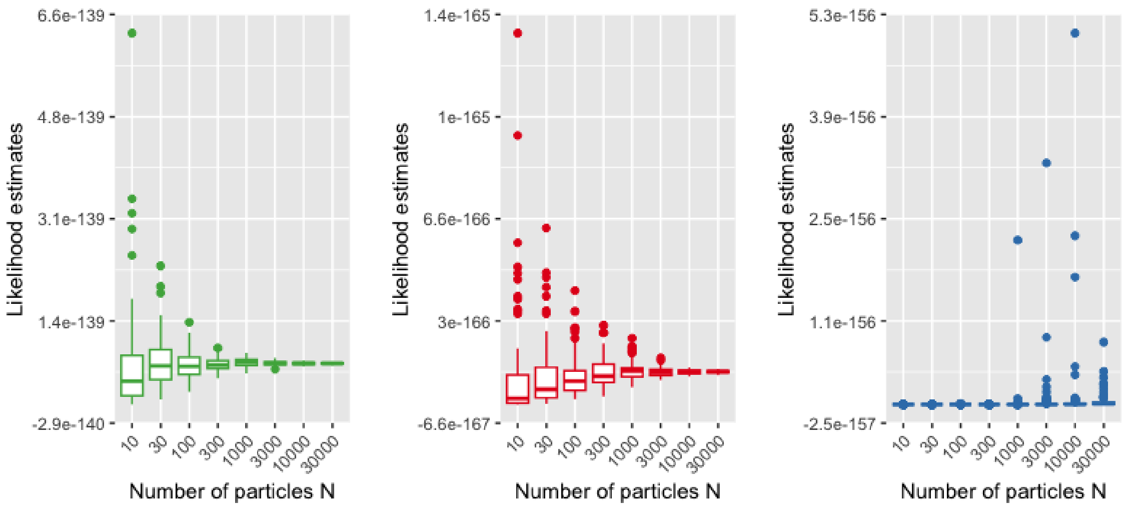
**Figure 2.8:** Measures of variation of 100 likelihood estimates of three different parameter values in the 2-state model computed by a standard bootstrap filter with varying numbers of particles $N$. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates.

false parameter was higher than for the first false parameter. The increase in variance was therefore not owed to a decrease in likelihood. Second, as can be seen in Figure 2.8 (blue line), the variance of the log-likelihood estimates for the second false parameter decreased with increasing particles much more slowly than for the other two parameters—indeed, the CV hardly decreased as the number of particles increased. This is likely due to the mean likelihood estimate being much lower than the true likelihood when only 100 estimates with low numbers of particles are used, see Figure 2.9c.

Looking at the distribution of the likelihood estimates in Figure 2.9, we noticed that the estimates were right-skewed for the true parameter and the first false parameter and showed the same behaviour as in the toy example in Figure 2.3a. Because of the theoretical unbiasedness of the algorithm, this means that there were many estimates that very slightly underestimated the likelihood and a few that overestimated it by a lot. From Figure 2.9a to 2.9b, this skewness became more pronounced as the variance of the estimates increased. Figure 2.9c seems to illustrate a different behaviour. However, we can assume that for this parameter, where the variance is much larger, the skewness is also much larger and very high values are possible but occur only rarely. It is therefore unsurprising that for low numbers of particles none of these large values appeared when only 100 estimates were computed. This changed once the number of particles reaches 1000.

When the computation time was considered, the increasing efficiency of the filter can be seen in Figure 2.10. The computation time per particle decreased as the number of particles in the filter increased from 1 to 300 and then stayed roughly the same for the particle filters with 300 to 30,000 particles. This indicates that for these larger particle filters the overhead of the algorithm is negligible for the total computation time.

These results were used to determine the number of particles to use in subsequent simulations. One criterion was to aim for a log-likelihood variance of 2 or less at a central value of the posterior distribution. This threshold was passed with 10 particles for the true parameter (Figure 2.8), which arguably can be considered a "central" value of the



**(a)** True parameter value **(b)** False parameter value 1 **(c)** False parameter value 2

**Figure 2.9:** Boxplots of 100 likelihood estimates of three different parameter values in the 2-state model for varying numbers of particles $N$.

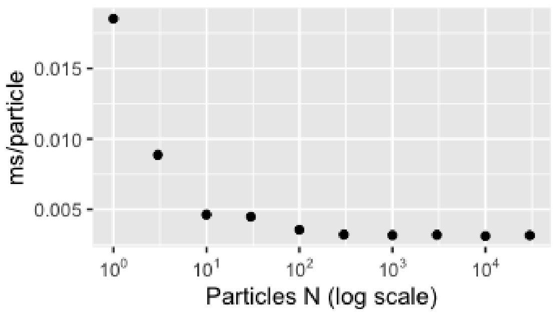


**Figure 2.10:** Bootstrap filter runtime per particle for varying number of particles $N$.

posterior distribution. For the first false parameter, this threshold was passed with 30 particles whereas it was never passed for the second false parameter even with 30,000 particles. Because the likelihood estimation seemed to suffer from very high variance for the false parameters, we opted for a higher number of particles than the guideline of a log-likelihood variance of 2 suggests. We also wanted to avoid too much time spent on the overhead of the algorithm, as would be the case with only 10 particles. The somewhat heuristic choice moving forward was therefore $N = 100$ particles for the 2-state model in this chapter. This was a compromise that kept the running time reasonably low (e.g., 0.349 sec for 100 likelihood estimates with $N = 100$ versus 32.1 sec with $N = 10,000$) but had to make the concession that the likelihood estimates for some parameters like the second false parameter suffer from severe Monte Carlo error.

#### 2.3.2.2 Factor 2: Resampling Schedule

Table 2.4 and Figure 2.11 show the variation in the likelihood estimates when the number of resampling steps was decreased depending on the ESS of the weighed particles for the true parameter and the first false parameter. The CV was higher at either extreme of the ESS threshold and lower but similar for a wide range of values from approximately 0.5 to 0.9. Note that a threshold of 0, i.e., an algorithm without resampling, seems quite

| | True parameter | | False parameter 1 | | False parameter 2 | |
|---|---|---|---|---|---|---|
| $\text{ESS}_{\min}/N$ | CV(L) | Var(logL) | CV(L) | Var(log-L) | CV(L) | Var(log-L) |
| 0 | 7.05E-01 | 3.24E-01 | 3.11E+00 | 2.71E+00 | 3.16E+01 | 9.49E+03 |
| 0.1 | 4.03E-01 | 1.66E-01 | 9.32E-01 | 6.78E-01 | 1.36E+01 | 3.06E+02 |
| 0.2 | 3.57E-01 | 1.27E-01 | 7.82E-01 | 4.80E-01 | 2.80E+01 | 2.36E+02 |
| 0.3 | 3.18E-01 | 1.05E-01 | 6.87E-01 | 4.25E-01 | 2.55E+01 | 1.86E+02 |
| 0.4 | 3.17E-01 | 1.01E-01 | 7.41E-01 | 4.25E-01 | 2.79E+01 | 1.50E+02 |
| 0.5 | 2.97E-01 | 8.83E-02 | 7.01E-01 | 3.85E-01 | 3.00E+01 | 1.25E+02 |
| 0.6 | 3.01E-01 | 9.35E-02 | 7.19E-01 | 4.04E-01 | 2.22E+01 | 1.31E+02 |
| 0.7 | 3.11E-01 | 1.02E-01 | 7.59E-01 | 4.93E-01 | 2.15E+01 | 1.35E+02 |
| 0.8 | 3.03E-01 | 9.71E-02 | 7.97E-01 | 5.57E-01 | 1.62E+01 | 1.25E+02 |
| 0.9 | 3.15E-01 | 9.64E-02 | 8.79E-01 | 5.15E-01 | 2.18E+01 | 1.11E+02 |
| 1 | 3.61E-01 | 1.34E-01 | 1.06E+00 | 8.26E-01 | 2.80E+01 | 1.30E+02 |

**Table 2.4:** Measures of variation of 1000 likelihood estimates computed by a bootstrap filter with varying ESS thresholds $\text{ESS}_{\min}/N$ for resampling. The likelihood was estimated at the true parameter value and at two false parameter values for the 2-state model. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates.

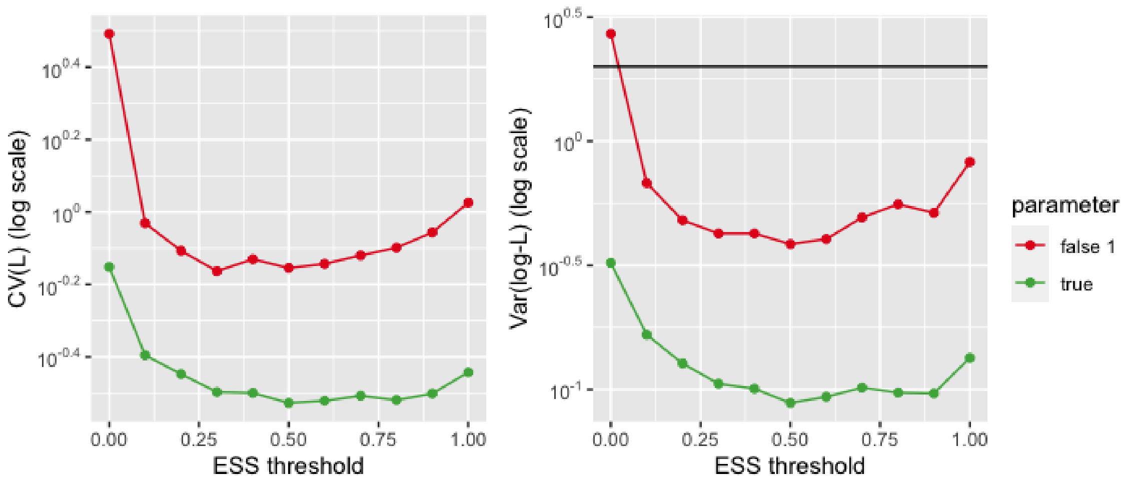


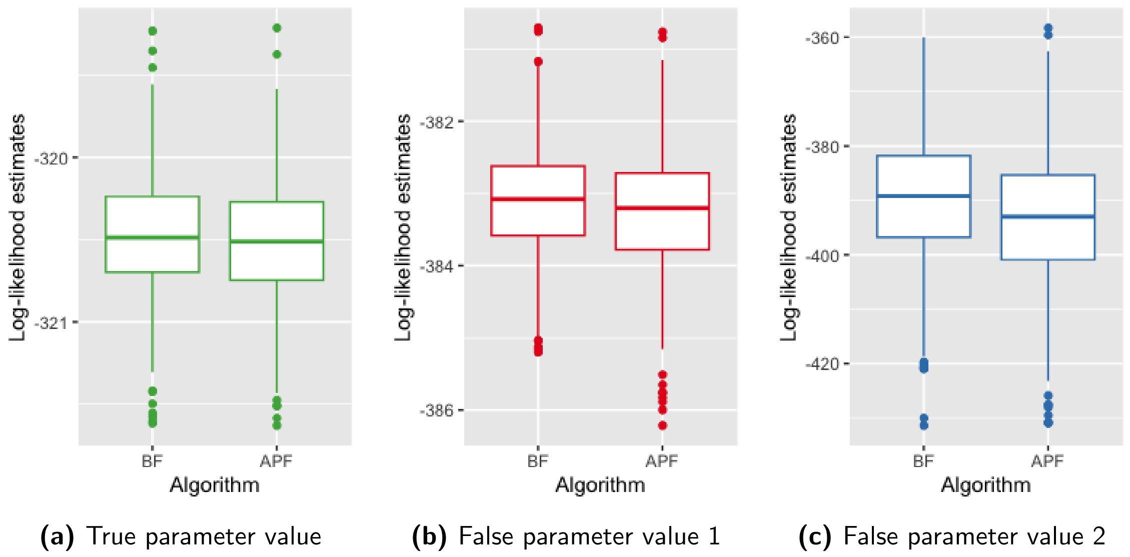
**Figure 2.11:** Measures of variation of 1000 likelihood estimates of two parameters in the 2-state model computed by a bootstrap filter with varying ESS thresholds $\text{ESS}_{\min}/N$ for resampling. False parameter 2 is not included in the plot because the values are too large and show no discernible trends. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates.

disadvantageous, whereas a threshold of 1, i.e., always resampling only showed a slightly worse performance than the values between 0.5 and 0.9. This means that neither extreme of the choice of threshold was ideal and instead a value in the middle seemed more favourable. The flatness of the CV in the area between 0.5 and 0.9 indicates that all of these values seemed to achieve a relatively low variance for the two parameter vectors. As predicted from the results in the previous section, the variance of the log-likelihood estimates for the second false parameter was very high with only $N = 100$ particles, namely three orders of magnitudes larger than for the true parameter, and showed no discernible pattern. The values for this parameter were therefore excluded from Figure 2.11, so that the nuances are visible for the other two parameters. We note that lower values for the resampling threshold led to slightly less computation time because the resampling step occurred less frequently. From the range of values that we found to work well, we chose 0.8 as the ESS threshold $\mathrm{ESS}_{\min}/N$ moving forward.

#### 2.3.2.3 Factor 3a: Adjustment Multiplier—Expected Value

| | True parameter | | False parameter 1 | | False parameter 2 | |
|---|---|---|---|---|---|---|
| Alg | CV(L) | Var(logL) | CV(L) | Var(log-L) | CV(L) | Var(log-L) |
| BF | 3.65E-01 | 1.24E-01 | 8.11E-01 | 4.98E-01 | 2.31E+01 | 1.21E+02 |
| APF | 3.72E-01 | 1.29E-01 | 8.43E-01 | 6.14E-01 | 2.54E+01 | 1.35E+02 |

**Table 2.5:** Measures of variation of 1000 likelihood estimates of three different parameter values in the 2-state model computed by a bootstrap filter (BF) and by an auxiliary particle filter (APF) using the expected value in the adjustment multiplier. The likelihood was estimated at the true parameter value and at two false parameter values. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates.



**(a)** True parameter value **(b)** False parameter value 1 **(c)** False parameter value 2

**Figure 2.12:** Boxplots of 1000 log-likelihood estimates of three different parameter values in the 2-state model comparing a bootstrap filter (BF) and an auxiliary particle filter (APF) using the expected value in the adjustment multiplier.

Table 2.5 and Figure 2.12 show that while the performance of the two algorithms was similar, the APF seemed to exhibit no improvement at all. In fact, the bootstrap filter showed a slightly smaller Monte Carlo variance when compared to the auxiliary particle filter for all three parameters. This might be explained by the relatively small process stochasticity in the model due it consisting of binomial distribution with a high number of trials. For example, with a fecundity of $\alpha = 0.8$ in the true parameter vector and 3000 adult females (just below carrying capacity in the simulated data), the CV of the number of pups is 0.91%. This is a lot lower than the CV of 10% for the observations. The expected value might therefore not be necessary to guide the particles in the right direction. In cases where the data contain outliers, this technique might be more helpful.

It is worth mentioning that `nimble` is about twice as fast as standard `R` code, excluding the compilation time necessary for `nimble`. Calculating 1000 estimates with 100 particles each and an ESS threshold of 0.8 took 2.98 seconds with `nimble` and 5.24 seconds in standard `R` code. Including the information about the expected value in the adjustment multiplier increased this time to 5.87 seconds. We note that NIMBLE code requires building and compiling of the model and corresponding bootstrap filter which is not necessary with R code. This took 32.62 seconds though it is only necessary to execute this once. Because the APF with an expected value adjustment multiplier showed no improvement but increased computing time by a factor of almost 2, we do not use the expected value adjustment multiplier in the algorithm in the remainder of this thesis.

#### 2.3.2.4 Factor 3b: Adjustment Multiplier—Tempering of Weights

| | True parameter | | False parameter 1 | | False parameter 2 | |
|---|---|---|---|---|---|---|
| Exp. | CV(L) | Var(logL) | CV(L) | Var(log-L) | CV(L) | Var(log-L) |
| 2 | 2.45E+00 | 4.34E+01 | 7.98E+00 | 2.00E+02 | 3.16E+01 | 2.99E+04 |
| $\sqrt{2}$ | 5.33E-01 | 6.28E-01 | 1.09E+00 | 1.59E+00 | 2.61E+01 | 2.22E+02 |
| 1 | 2.56E-01 | 6.46E-02 | 1.06E+00 | 5.11E-01 | 2.98E+01 | 1.28E+02 |
| $\sqrt{2}/2$ | 2.93E-01 | 8.84E-02 | 7.79E-01 | 4.89E-01 | 2.93E+01 | 1.74E+02 |
| 1/2 | 3.24E-01 | 1.03E-01 | 8.33E-01 | 6.15E-01 | 3.13E+01 | 2.49E+02 |
| $\sqrt{2}/4$ | 3.83E-01 | 1.53E-01 | 9.65E-01 | 8.89E-01 | 3.04E+01 | 3.85E+02 |
| 1/4 | 4.83E-01 | 2.21E-01 | 1.31E+00 | 1.31E+00 | 2.81E+01 | 5.61E+02 |
| $\sqrt{2}/8$ | 6.08E-01 | 3.68E-01 | 1.78E+00 | 2.10E+00 | 2.46E+01 | 8.83E+02 |
| 1/8 | 1.17E+00 | 6.87E-01 | 3.49E+00 | 3.37E+00 | 3.05E+01 | 1.28E+03 |
| $\sqrt{2}/16$ | 2.02E+00 | 1.08E+00 | 3.41E+00 | 5.07E+00 | 3.16E+01 | 1.82E+03 |
| 1/16 | 1.99E+00 | 1.69E+00 | 5.82E+00 | 7.54E+00 | 2.89E+01 | 2.57E+03 |

**Table 2.6:** Measures of variation of 1000 likelihood estimates computed by an auxiliary particle filter using tempered weights with varying exponents in the resampling step. The likelihood was estimated for the 2-state model at the true parameter value and at two false parameter values. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates.

The results in Table 2.6 and Figures 2.13 and 2.14 show that using an exponent of 1 (for the true and the second false parameter) or one only slighty smaller at $\sqrt{2}/2$ (for the first false parameter) led to the smallest Monte Carlo variance in the likelihood estimates. Smaller exponents seemed to slowly increase the variance whereas exponents above 1 led to a drastic increase. As in the previous section, we suspect that smaller exponents would only be beneficial in the case of outliers in the data. The boxplots in Figure 2.14 seem to illustrate a bias in the estimates when the exponent is 2. However, the mean of the likelihood estimates for the true parameter for this exponent is -320.31—close to the -320.39 determined in Section 2.3.2.1 and within the values of the means across all the exponents which range from -320.52 to -319.78. The distribution seems biased due to the log-scale and because it shows medians rather than means. Since both 1 and a slightly smaller value are reasonable choices from a performance view when all three parameters are considered, we proceeded with an exponent of 1 and therefore eliminated the need for an adjustment multiplier. This allowed the continued use of `nimble`, thereby profiting from the fast methods already implemented in this software package.

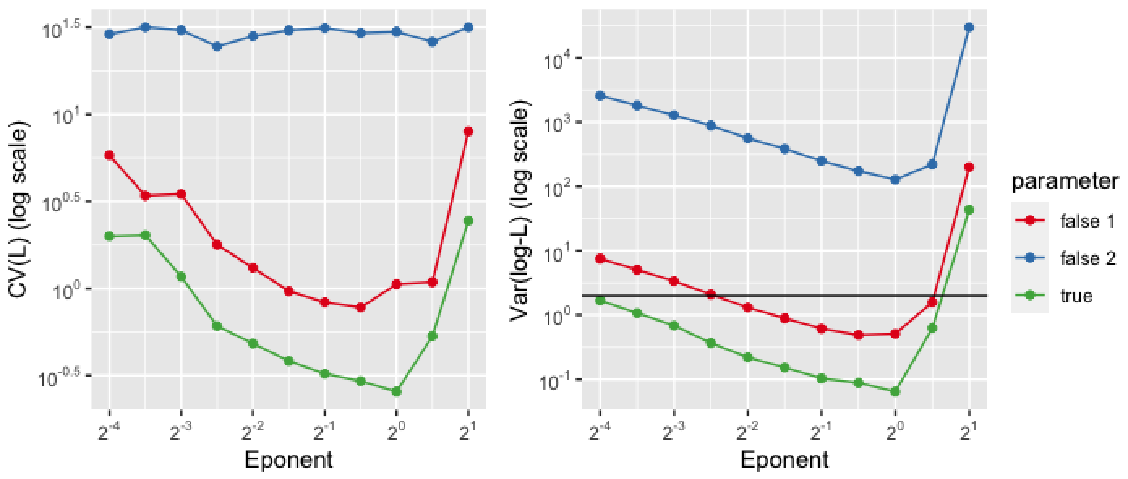
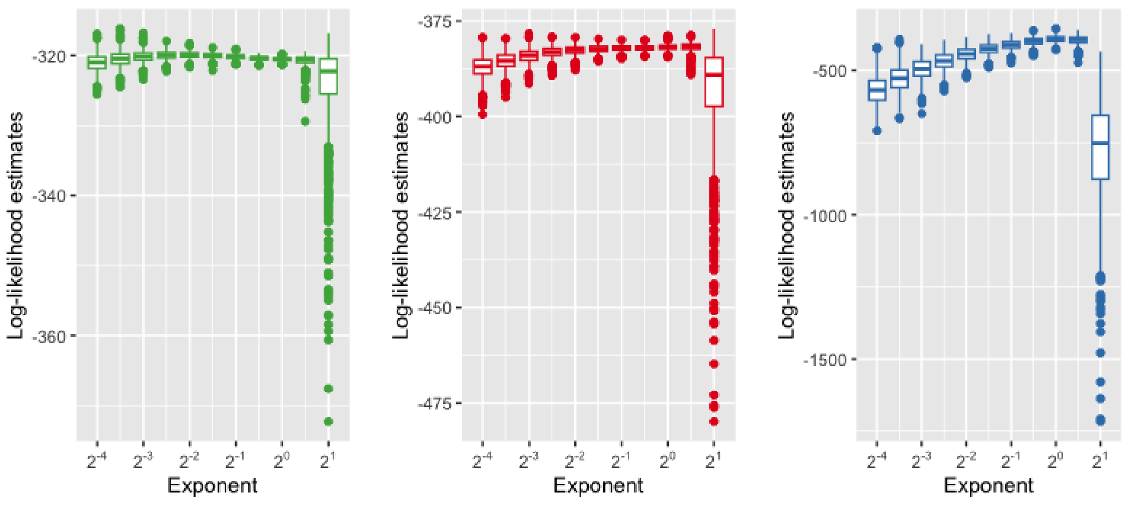


**Figure 2.13:** Measures of variation of 1000 likelihood estimates of three different parameter values in the 2-state model computed by an auxiliary particle filter using tempered weights with varying exponents in the resampling step. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates.



**(a)** True parameter value **(b)** False parameter value 1 **(c)** False parameter value 2

**Figure 2.14:** Boxplots of 1000 log-likelihood estimates computed by an auxiliary particle filter using tempered weights with varying exponents in the resampling step, for three different parameter values in the 2-state model.

## 2.4 Application to the 7-State Model

In this Section, we describe how we adapted the algorithm for the 7-state model, again with simulated data, using the outcomes for the 2-state model as a starting point.

### 2.4.1 Methods

Regarding the results for the ESS threshold and the effect of introducing an adjustment multiplier, we transferred the results obtained with the 2-state model to the 7-state model which adds 5 age classes between pups and pup-producing adults in comparison to the 2-state model (see also Section 1.2.5). However, the number of particles needed to be adapted to this specific model and was varied from 3 to 30,000 particles. For each number of particles, 100 estimates of the likelihood were generated to measure the Monte Carlo variance of these estimates by calculating the CV of the likelihood and the variance of the log-likelihood. As before, a value of 2 for the log-likelihood variance was used as a guideline when choosing the number of particles.

The parameter for simulating the data was the mean of the posterior distribution in Thomas et al. (2019), using the carrying capacity and starting observation of the Inner Hebrides region (see Table 2.7). We also reduced the number of annual observations to $T = 26$, which is the number of observations for the complete seal model. Figure 2.15 shows the number of pups and the number of the adult females aged six years and older in the simulated data. The effect of reaching the carrying capacity is clearly visible in the data. On the other hand, the period of near-exponential growth is missing in the simulated data even for the early years and the effect of approaching the carrying capacity noticeable from the start of the times series.

We estimated the likelihood at two further sets of parameter values besides the true parameter. The first false parameter was set to the prior mean in Thomas et al. (2019) which is quite far away from the true parameter. Nevertheless, as this is a central value in the prior distribution, it is important to be able to reliably evaluate its likelihood. The second false parameter was chosen to have a likelihood between the other two parameters. All three parameter values can be found in Table 2.7.

| Parameter | True parameter | False parameter 1 | False parameter 2 |
|---|---|---|---|
| maximum pup survival $\phi_{p,\max}$ | 0.48 | 0.62 | 0.45 |
| adult survival $\phi_a$ | 0.95 | 0.9 | 0.93 |
| fecundity $\alpha$ | 0.9 | 0.83 | 0.92 |
| Carrying capacity $\chi$ | 3110 | 5000 | 2900 |
| dens dep. $\rho$ | 5.95 | 10 | 5 |
| CV precision $\tau$ | 112 | 140 | 90 |

**Table 2.7:** Parameter vectors used in the simulation study for the 7-state model

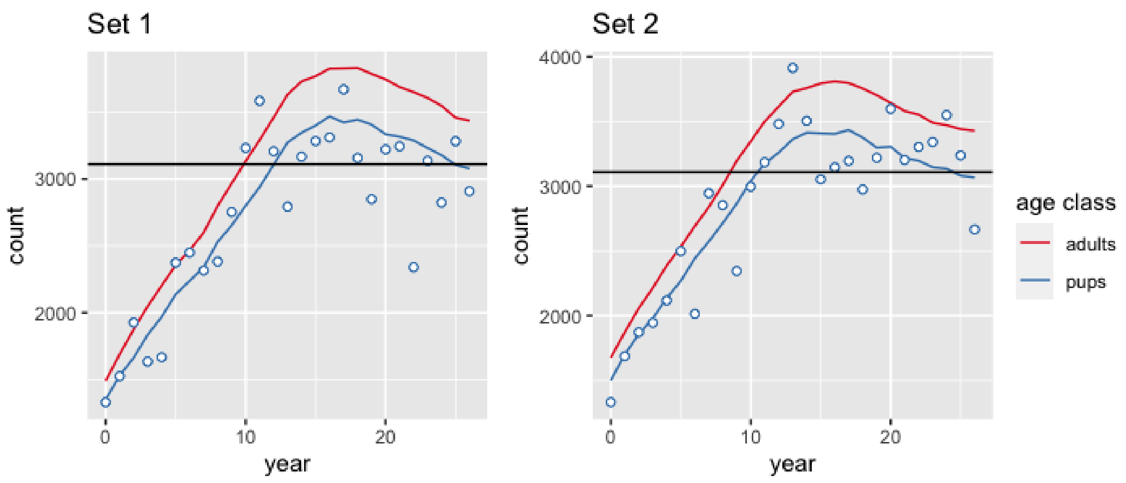


**Figure 2.15:** Two sets of simulated counts from the 7-state model. Only numbers of pups and adult females aged at least 6 years are shown. Observations of pup numbers are shown as blue circles and the pup carrying capacity is indicated with a black horizontal line.

### 2.4.2 Results and Interpretation

The estimated log-likelihood of the true parameter was -185.92. The first false parameter was far away from the true parameter which was reflected by the much lower log-likelihood of -230.97. For the second false parameter, the likelihood was estimated to be -197.90.

| | True parameter | | False parameter 1 | | False parameter 2 | |
|---|---|---|---|---|---|---|
| $N$ | CV(L) | Var(logL) | CV(L) | Var(log-L) | CV(L) | Var(log-L) |
| 3 | 1.51E+00 | 3.35E+01 | 9.90E+00 | 4.67E+03 | 9.48E+00 | 3.92E+02 |
| 10 | 8.22E-01 | 3.06E+00 | 1.00E+01 | 8.87E+02 | 6.60E+00 | 7.19E+01 |
| 30 | 5.24E-01 | 3.62E-01 | 8.90E+00 | 2.38E+02 | 4.64E+00 | 1.81E+01 |
| 100 | 3.36E-01 | 1.26E-01 | 9.01E+00 | 8.31E+01 | 5.26E+00 | 8.30E+00 |
| 300 | 1.69E-01 | 3.01E-02 | 7.39E+00 | 4.36E+01 | 1.83E+00 | 3.20E+00 |
| 1000 | 9.67E-02 | 9.34E-03 | 8.95E+00 | 2.04E+01 | 1.91E+00 | 1.66E+00 |
| 3000 | 4.84E-02 | 2.37E-03 | 3.53E+00 | 9.94E+00 | 1.88E+00 | 9.47E-01 |
| 10000 | 3.02E-02 | 9.23E-04 | 7.10E+00 | 6.57E+00 | 6.15E-01 | 3.10E-01 |
| 30000 | 1.63E-02 | 2.64E-04 | 4.88E+00 | 3.19E+00 | 4.39E-01 | 1.62E-01 |
| log-L | -185.92 | | -230.97 | | -197.90 | |

**Table 2.8:** Measures of variation of 100 likelihood estimates of the 7-state model computed by a standard bootstrap filter with varying numbers of particles $N$. The likelihood was estimated at the true parameter value and at two false parameter values. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 30,000$ particles.

The effect of increasing the number of particles was similar to the effect seen with the 2-state model (see Table 2.8 and Figures 2.16 and 2.17). The CV of the estimate decreased as the number of particles increased, and the estimates for the true parameter showed a much lower variance than for the other two parameters. For the true parameter, 30 particles were sufficient to achieve a variance of the log-likelihood of less than 2, whereas 1000 were necessary for the second false parameter, and even 30,000 were not enough for

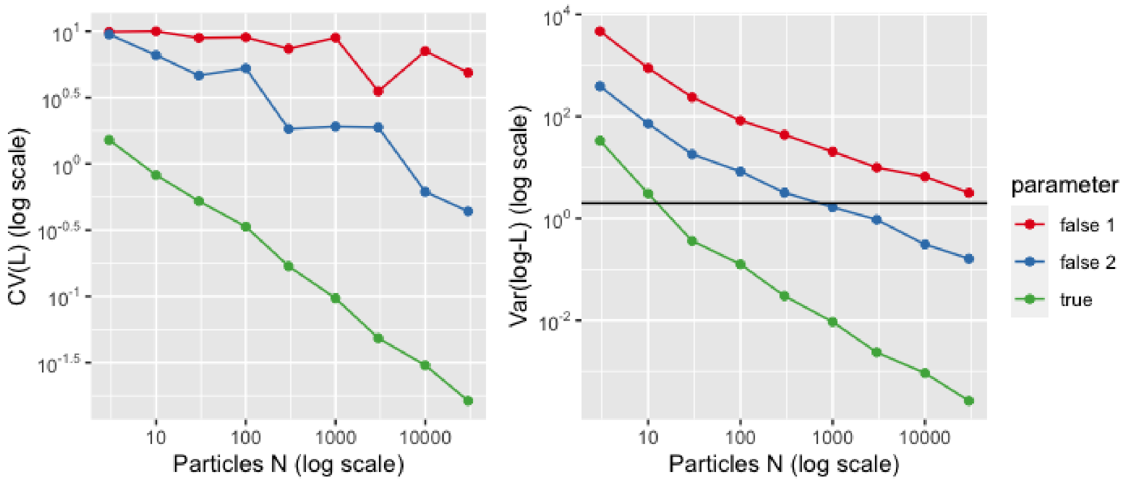


**Figure 2.16:** Measures of variation of 100 likelihood estimates of the 7-state model computed by a standard bootstrap filter with varying numbers of particles $N$. The first figure gives the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates.

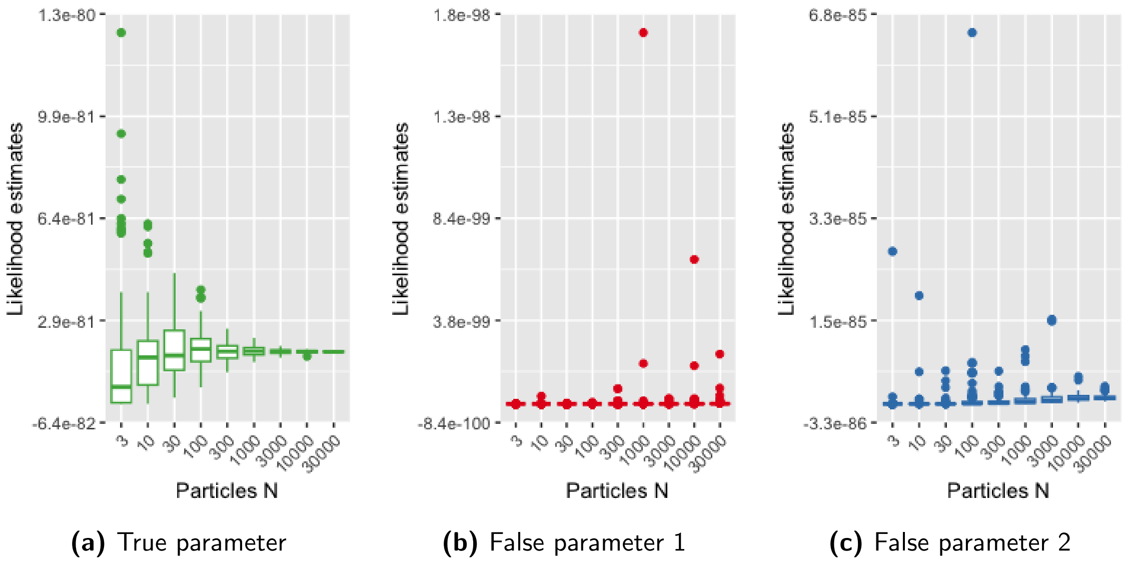


**Figure 2.17:** Boxplots of 100 likelihood estimates of the 7-state model for varying numbers of particles $N$ at three different parameter values.

the first false parameter. The distribution of the likelihood estimates of the true parameter in Figure 2.17a showed the same right-skewed behaviour as in the 2-state case. For the two false parameters in Figures 2.17b and 2.17c, the few outliers indicated a right-skewed distribution too.

As the Monte Carlo variance for the true parameter is slightly higher in the 7-state model as in the 2-state model, the number of particles should reflect this. Therefore, a reasonable choice could be 300 to account for the increased complexity of this model. However, we adapt this number in Chapter 3 depending on the purpose of the filter.

## 2.5 Application to the Complete Seal Model

Finally, the outcomes of the simulation study for the 2-state model were applied to the complete seal model without independent estimate, which extends the 7-state model by simultaneously tracking 4 regions with independent populations. We used both simulated and real data here. For the simulated data, we compared two ways of formulating the model and its effect on the likelihood estimation.

### 2.5.1 Methods

As with the 7-state model in the previous section, we relied on the results obtained with the 2-state model for all of the settings of the algorithm other than the number of particles. This was varied from 10 to 30,000 particles and the likelihood estimated 100 times for each number. In the previous sections, it became apparent that while the likelihood estimation of the false parameters suffered from higher Monte Carlo variance, the overall trends remained the same. We therefore studied the effect of the number of particles only by estimating the likelihood of the true parameter. This is also sufficient when the guideline of a log-likelihood variance below 2 (Section 2.3.2.1) is considered because this guidelines refers to the likelihood estimates at a central parameter value and not at values at the extremes.

Instead of studying the behaviour of the estimator for different parameter values when the number of particles are varied, we focussed on another aspect of the complete model. For this, we only consider the slightly simplified version of the complete model, where there is no independent estimate of the total number of adult seals. The independent estimate is the only observation that links the four regions with each other. Omitting this estimate therefore results in the independence of the four regions from each other, given the joint parameter, and the likelihood as calculated in Equations 2.9 and 2.10 can be factorised into four components from each of the regions.

$$\begin{aligned}
p(y_{t,r=1:4}|y_{1:t-1,r=1:4}) &= \int p(y_{t,r=1:4}, x_t|y_{1:t-1,r=1:4})dx_t \qquad (2.14)\\
&= \int g(y_{t,r=1:4}|x_t)p(x_t|y_{1:t-1,r=1:4})dx_t = \\
&= \int \prod_{r=1}^{4} g(y_{t,r}|x_{t,r}) \prod_{r=1}^{4} p(x_{t,r}|y_{1:t-1,r})dx_t \\
&= \prod_{r=1}^{4} \int g(y_{t,r}|x_{t,r})p(x_{t,r}|y_{1:t-1,r})dx_{t,r} \\
&= \prod_{r=1}^{4} p(y_{t,r}|y_{1:t-1,r}) \approx \prod_{r=1}^{4} \frac{1}{N}\sum_{i=1}^{N} w_{t,r}^{(i)} \qquad (2.15)
\end{aligned}$$

The likelihood for each of the four regions can therefore be estimated separately by independent particle filters. For the overall likelihood, these estimates are simply multiplied. The unbiasedness of the total estimate is preserved because of the independence of the four estimates.

$$\mathrm{E}\left(\prod_{r=1}^{4} \frac{1}{N}\sum_{i=1}^{N} w_{t,r}^{(i)}\right) = \prod_{r=1}^{4} \mathrm{E}\left(\frac{1}{N}\sum_{i=1}^{N} w_{t,r}^{(i)}\right) = \prod_{r=1}^{4} p(y_{1:T,r})$$

This factorisation of the likelihood might improve the estimation. In the joint particle filter, each particle consists of 28 components, each of which must be a relatively good match for the observation in order to be resampled. This means that particles that might contain reasonable simulated values of the states in one region are still discarded because of the simulated values in another regions. More particles might therefore be necessary to obtain the same ESS. We compared the variance of the log-likelihood estimates when they were estimated jointly for all four regions with the estimates that were produced by multiplying the estimates of the four independent regions. This was done by generating 100 estimates with both the joint and the factorised model formulation.

We note that while these deliberations about the model formulation might lead to improvements in the computational cost of the particle filter, these improvements cannot directly be transferred to the complete seal model with independent estimate. In that model, the regions are dependent via the independent estimate of the total number of adult seals in all regions. Chapter 6 discusses some options how this idea can be modified to include the independent estimate.

To further simplify things, we omitted the overdispersion of the initial states for the simulated data when comparing the joint and the factorised model formulation. The purpose of the overdispersion in the model was to allow a wider range of possible starting conditions because the initial distribution might not quite reflect the real data correctly and therefore produce a too narrow range of initial states. As the simplified full model was studied only using data that were simulated from that model, this was not a concern here.

As the last step and the ultimate objective of this chapter, we tuned the number of particles for the real model—including the independent estimate, the overdispersion of the initial states, and missing data in one region in one of the years. Here, we used the real data, rather than simulated data. The parameter value at which the likelihood was estimated was the mean of the posterior distribution as given by Thomas et al. (2019).

### 2.5.2 Results and Interpretation

| | Joint | | Factorised | | Real data, joint | |
|---|---|---|---|---|---|---|
| $N$ | CV(L) | Var(logL) | CV(L) | Var(log-L) | CV(L) | Var(log-L) |
| 10 | 4.98E+00 | 1.09E+02 | 1.64E+00 | 2.09E+01 | 7.00E+00 | 2.57E+03 |
| 30 | 3.07E+00 | 3.18E+01 | 1.05E+00 | 1.19E+00 | 5.44E+00 | 8.75E+02 |
| 100 | 1.71E+00 | 5.77E+00 | 6.33E-01 | 2.96E-01 | 4.68E+00 | 3.49E+02 |
| 300 | 1.05E+00 | 1.72E+00 | 3.42E-01 | 1.07E-01 | 7.82E+00 | 1.11E+02 |
| 1000 | 5.67E-01 | 4.51E-01 | 2.05E-01 | 4.04E-02 | 2.39E+00 | 2.37E+01 |
| 3000 | 3.84E-01 | 1.75E-01 | 9.30E-02 | 9.25E-03 | 1.74E+00 | 8.21E+00 |
| 10000 | 2.24E-01 | 5.47E-02 | 5.34E-02 | 2.79E-03 | 1.60E+00 | 2.17E+00 |
| 30000 | 1.30E-01 | 1.79E-02 | 3.27E-02 | 1.09E-03 | 9.70E-01 | 9.31E-01 |
| log-L | -807.31 | | -807.69 | | -806.34 | |

**Table 2.9:** Measures of variation of 100 likelihood estimates computed by a particle filter with varying numbers of particles $N$. The likelihood was estimated at the true parameter value for two different formulations of the simplified full model, using simulated data, and at the posterior mean for the real full model using real data. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 30,000$ particles.

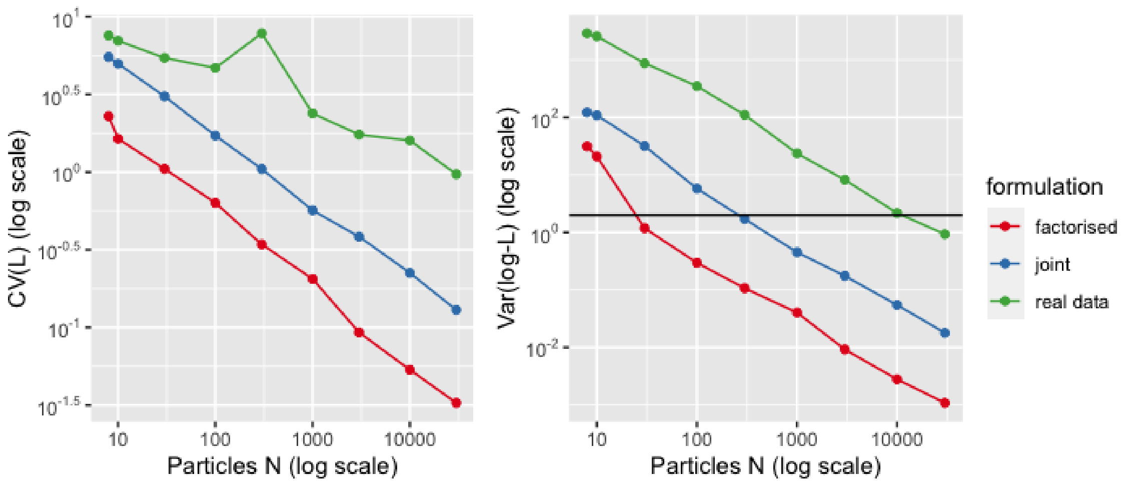


**Figure 2.18:** Measures of variation of 100 likelihood estimates computed by a standard bootstrap filter with varying numbers of particles $N$. The likelihood was estimated at the true parameter value for the joint formulation and the factorised formulation of the simplified full model, using simulated data, and at the posterior mean for the real full model using real data. The first figure gives the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates.

Figure 2.18 and Table 2.9 show the effect of the number of particles on the variance of the likelihood estimate. The general relationship between the CV of the likelihood estimate or the variance of the log-likelihood estimate and the number of the particles was the same as in the previous sections. Focusing first on the effect of using a factorised approach to estimate the likelihood, we noticed that there was a large decrease of the variance when the factorised estimate was used. The variance of the log-likelihood when using the factorised estimate was less than 2 even when only 30 particles were used. To achieve the same low variance for the joint estimate, 300 particles were necessary. This factor 10 in computational effort remained roughly the same for other values of variance of the log-likelihood.

The variance of the log-likelihood estimate for the real data showed the same trend but was overall larger than the variance for the estimates from the simulated data. To achieve a variance of the log-likelihood estimate of less than 2, more than 10,000 particles were necessary. This is unsurprising and corresponds to the findings in the previous sections. There, we saw that estimating the likelihood at the false parameter values led to a higher variance of the estimate. Here, the parameter was chosen at a central point of the posterior, but using real instead of simulated data inevitably resulted in a data that fit the model less well.

Finally, we investigated the variance of the weights in the complete seal model with the real data at each iteration of the bootstrap filter, using the joint formulation. In Figure 2.19, the CV of the weights at each time $t$ is shown for three runs of a bootstrap filter with $N = 1000$ particles each. We note that the CV tended to decrease as time $t$ increased. This indicates that more particles were projected forward to high-density areas and thus might point to the fact that the data more closely matched the model. The first 5 iterations seemed particularly problematic as the weights had a CV larger than 1 for all 3 runs for these iterations. A reason for this might be that the initialisation of the model with density $f_0$ is not appropriate for the data and that many particles were therefore starting in low-density areas. Other than this general trend in time, no clear outliers or other anomalies could be detected that might explain the large variance in the likelihood estimates.

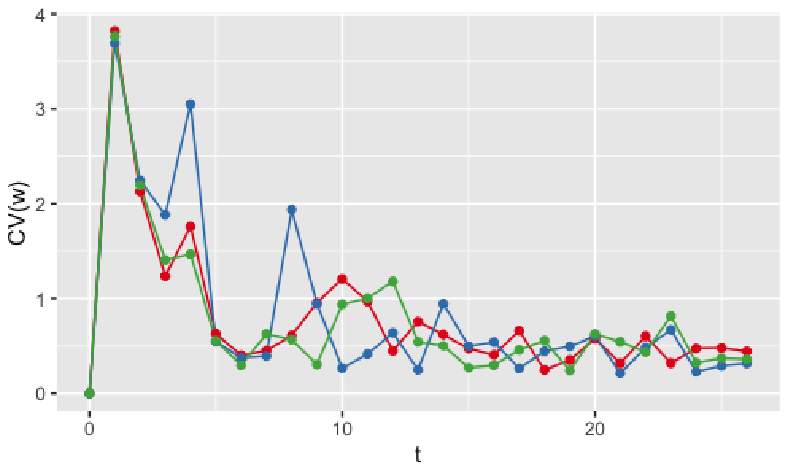


**Figure 2.19:** Effective sample sizes of the particles at each iteration from 3 runs of a bootstrap filter with $N = 1000$ particles, run with the complete seal model using real data with the independent estimate, with the parameter set to the posterior mean.

## 2.6 Discussion and Conclusion

In this chapter, we introduced particle filter algorithms and showed how they can be used to estimate the likelihood in state-space models. We also described some of the available techniques to improve the performance of these algorithms, both when estimating the underlying states and the likelihood for a fixed parameter. While there is a large repertoire of these techniques in the literature, many are not available for the seal model because they require the evaluation of the process density $f(x_t|x_{t-1})$ or require a Gaussian or linear model. Of the methods that can be applied to the seal model, none greatly improved the performance of the algorithm. Most of the methods did either not reduce the Monte Carlo variance of the likelihood estimate at all, or only did so at a much higher computational cost.

When using the ESS as a measure of particle degeneracy to reduce the number of resampling steps in the particle filter, the results of the simulation study agreed with the literature. In the rest of this thesis, a threshold of 0.8 will be used as this results in a slightly lower Monte Carlo variance compared to resampling at every iteration or resampling less frequently.

Using an adjustment multiplier to change the resampling weights proved less useful. When the expected value of $x_t|x_{t-1}$ was used to calculate how well the state $x_{t-1}$ in a particle fits the observation at the next time step $y_t$, no decrease could be seen in the variance of the likelihood estimate, while the computation time rose slightly. When the resampling weights were flattened by exponentiating them with a value less than one, no large effect could be observed for exponents between 1/2 and 1, with 1 resulting in the lowest variance for two of the three examined sets of parameter values. Smaller exponents, and exponents above 1, resulted in a much higher Monte Carlo variance. Therefore, neither of these two options to adjust the resampling weights was used for the seal model.

The number of particles used in the algorithm had the most direct effect on the variance of the likelihood estimate. As this number comes at a linearly increasing computational cost, it is important to choose it carefully depending on the context. This could be state inference or likelihood estimation for a fixed parameter, or parameter inference. When parameter estimation is the ultimate goal and the likelihood estimate is used as part of an MCMC chain, some guidelines exist in the literature, e.g., aiming for a variance of the log-likelihood of less than 2 at a central parameter value (Golightly et al., 2019). The trade-offs associated with the number of particles in this scenario will be explored in more detail in the next chapter.

We found a large difference in the performance of the algorithms when estimating the likelihood at different parameters across the different numbers of particles. While for all three parameters the general downward trend of the variance of the log-likelihood estimates was similar, the Monte Carlo error differed considerably between them. For certain parameters the required number of particles was much higher than for other parameters to achieve the same likelihood variance. While in our simulation study the variance was always lowest for the true parameter value, there was no direct relation between the value of the likelihood and the variance of its estimates. In fact, the second false parameter had a much higher log-likelihood of -362.1 compared to the first with -382.1 but its estimates showed a much higher variance throughout the simulation study. It is however hard to draw a firm conclusion based on the results of only three sets of parameter values. We note that even 30,000 particles were not sufficient to achieve a variance of the log-likelihood of less than 2.

The only change that resulted in a large reduction of the likelihood estimate variance was the factorisation of the likelihood estimate into four separate regions. This change is not readily available for the complete seal model as the regions are not completely independent. We devote Chapter 6 to an investigation of how modifications of this idea might still be implemented for the complete seal model.

In summary, most methods described in Section 2.2 are either not applicable to the seal model or failed to improve the likelihood estimate variance. Resampling less frequently while taking the ESS into account was moderately successful at improving the estimate but not enough to manage the large variances for some of the parameters used in the study. How to handle these large variances will be a key point when studying parameter inference methods in Chapter 3.

# Chapter 3

# Parameter Inference

## 3.1 Introduction

In this chapter, we discuss methods for parameter inference in state-space models. Compared to the algorithms for state inference in the previous chapter, which are relatively straightforward and tend to work well, the situation for parameter inference is much more complex.

In alignment with the structure of the previous chapter, we first give an overview of several state-of-the-art parameter inference methods and discuss their advantages and drawbacks. The algorithms are divided into two categories. In the first, the parameters are treated as if they were a part of the state vector and SMC algorithms are applied to the augmented states. In the second, standard MCMC algorithms are used as a building block of more complex algorithms that rely on the likelihood $p(y_{1:T}|\theta)$. Lastly, we discuss the SMC$^2$ algorithm which combines these two approaches. Algorithms for maximum likelihood parameter estimation are only briefly mentioned, as the seal model requires a Bayesian approach which therefore is the focus of this chapter. We refer to Kantas et al. (2015) for a comprehensive review of sequential particle methods both within a Bayesian and in a maximum likelihood framework.

After this review of methods, we conduct several simulation studies that highlight different aspects of some of the methods. The aim of these studies is to learn how to perform parameter inference for the seal model and to understand the potential challenges with this task. We also attempt to apply the gained knowledge to the complete seal model to make parameter inference with the real data.

## 3.2 Review of SMC Methods for Parameter Inference

### 3.2.1 Augmenting the unknown states with the parameters

The idea behind the algorithms in this section is to treat the unknown parameters as if they were unknown states. The vector of the unknown states is augmented with the parameter vector and becomes $(x_t, \theta_t)$. With this framework, the state inference algorithms detailed in Chapter 2 can be applied directly to this augmented state vector. In the initialisation step, $\theta_0^i$ is sampled from the prior distribution $p(\theta)$. As the parameter vector remains constant over time, the forward propagation is simply the identity function, so $f(\theta_t|\theta_{t-1}) = \theta_{t-1}$. For the forward propagation of the states in any particle, the parameter values in that particle are used in the model densities, so $x_t^i \sim f(x_t|\bar{X}_{t-1}^i, \theta_{t-1}^i)$. The same happens when the weight for a particle is calculated with $w_t^i \leftarrow w_{t-1}^i g(y_t|X_t^i, \theta_t^i)$. The algorithm is given in Algorithm 3. It returns a set of weighted particles containing state trajectories and parameter values that approximate the density $p(x_{0:T}, \theta|y_{1:T})$. To obtain a sample from the posterior distribution $p(\theta|y_{1:T})$, one can simply marginalise over the states, that is, use only the parameter value of each weighted particle and discard the states.

**Algorithm 3** Bootstrap filter with augmented state

```
for $i \leftarrow 1, ..., N$ do                                              ▷ Initialisation
    sample $\theta_0^i \sim p(\theta)$
    sample $X_0^i \sim f(x_0|\theta^i)$
    $w_0^i \leftarrow 1$                                                      ▷ Set importance weights
end for
for $t \leftarrow 1, ..., T$ do
    for $i \leftarrow 1, ..., N$ do
        if resampling condition met then
            resample state trajectories and parameters $(\tilde{X}_{0:t-1}^i, \tilde{\theta}_{t-1}^i)$ with probabilities proportional to $\{w_{t-1}^i\}_{i=1...N}$
            $w_{t-1}^i \leftarrow 1$                                          ▷ Set weights
        end if
        Set $\theta_t^i \leftarrow \tilde{\theta}_{t-1}^i$
        Sample $X_t^i \sim f(x_t|\tilde{X}_{t-1}^i, \theta_t^i)$                ▷ Forward propagation
        Set $X_{0:t}^i \leftarrow (\tilde{X}_{0:t-1}^i, X_t^i)$
        Compute $w_t^i \leftarrow w_{t-1}^i g(y_t|X_t^i, \theta_t^i)$            ▷ Weighting
    end for
end for
return a set of weighted particles containing state trajectories and parameters $\{X_{0:T}^i, \theta_T^i, w_T^i\}_{i=1...N}$
```

While this approach easily allows the transfer of the state inference methods from Chapter 2 to the parameter inference problem, this method leads to severe particle degeneracy. The states naturally vary throughout the time series, and therefore diversity is introduced into the sample with every forward propagation. This is not the case for the parameters which, at least in the models considered here, stay constant through time. The parameter space is only explored in the initialisation step, when parameter values are sampled from the prior. At every iteration of the algorithm, parameter values are discarded when the particles are resampled. As the parameters do not evolve over time, no diversity is introduced into the sample as the algorithm progresses. Instead, the number of unique parameter values decreases at each iteration.

The severity of this effect can be seen in the following example, where we use the toy example first introduced in Chapter 2 in Equation 2.12. The parameter $\theta$ is now treated as unknown and has a normal prior:

$$\begin{aligned} \text{Prior:} \quad & \theta && \sim \mathcal{N}(1,1) \\ \text{Transition pdf:} \quad & x_{t+1} && = \theta x_t + \mathcal{N}(0,1) \\ \text{Observation pdf:} \quad & y_t && = x_t + \mathcal{N}(0,1). \end{aligned} \tag{3.1}$$

As in Chapter 2, we simulate a series of $T = 30$ observations with $\theta = 0.9$. Applying Algorithm 3 to estimate the posterior distribution of $\theta$ clearly illustrates the particle

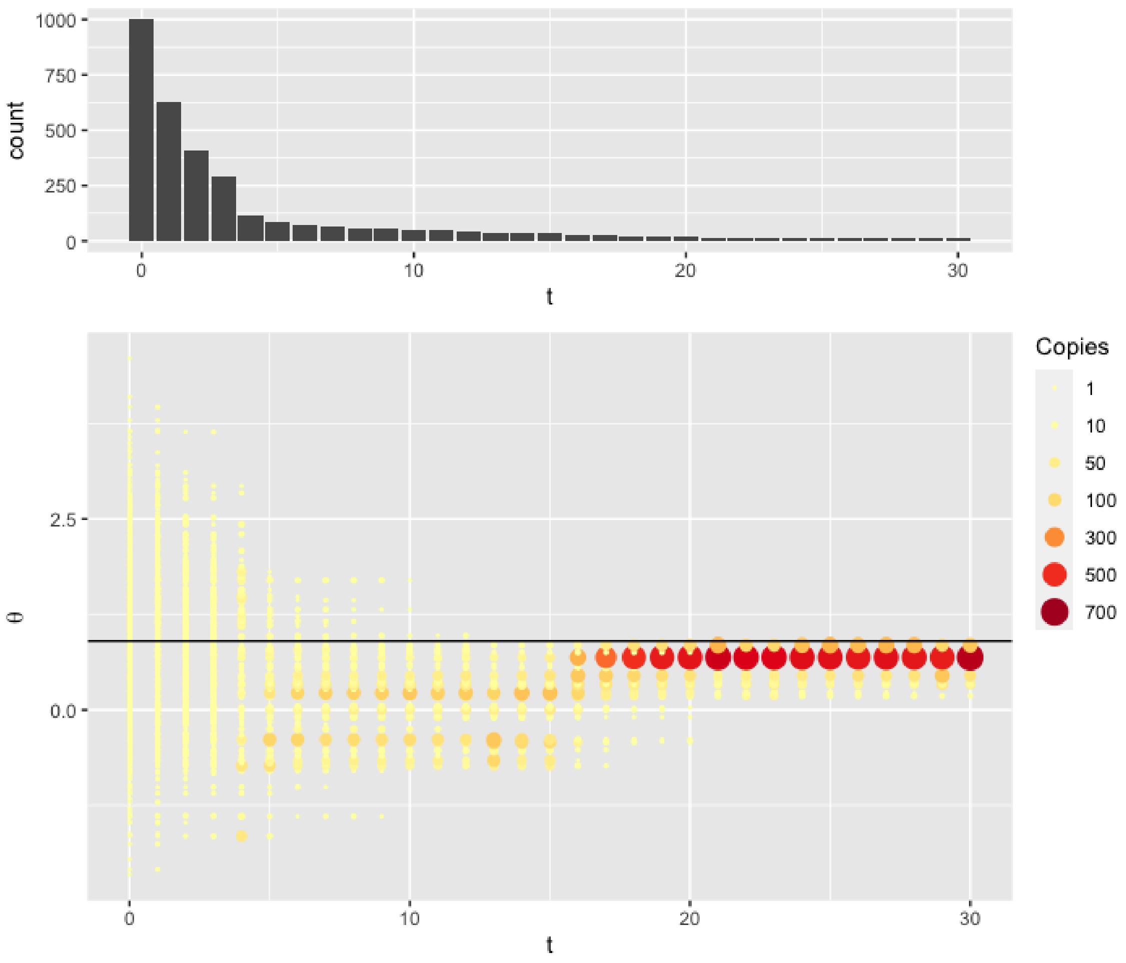


**Figure 3.1:** Decrease of unique parameter values in the bootstrap filter with augmented state algorithm. Observations were simulated from the model in Equation 3.1 with $\theta = 0.9$ (black horizontal line) and $T = 30$. The top plot shows the number of unique parameter values over time. The bottom plot shows the values of the remaining parameter values, with the size and color indicating how many copies of this values are present in the sample.

degeneracy problem. After 30 iterations of the algorithm, the initial 1000 unique parameter values have reduced to 10 unique values (see Figure 3.1). Because of the large disparity in the weights between these values, the effective sample size is even smaller at only 2.21. Using such a small sample size leads to a large Monte Carlo error in parameter estimates. While it is possible to increase the number of particles for toy examples like this, the computational cost quickly becomes too high and make this approach infeasible for more complicated models.

Several strategies exist to alleviate this problem. We notice that the particle degeneracy does not pose a problem for the state inference. When two particles are assigned the same state value $x_t$ in the resampling step, new diversity is introduced in the forward propagation step, as this usually contains a random part. Therefore, while these two particles might originate from the same state, they differ once they pass the forward propagation step. This idea can be applied to the parameters by introducing an artificial evolution of the parameter values over time. Kitagawa (1998) proposed adding a normally distributed noise with variance $W_t$ to the parameter value in the forward propagation step:

$$\theta_t = \mathcal{N}(\theta_{t-1}, W_t).$$

We note that jittering the parameter values according to a normal distribution is only appropriate when these values are on the real number line. When this is not the case, the values need first to be transformed to the real line, then moved, and then back-transformed. There are several choices for these transformations, two of which are given in Table 3.1. These logarithmic and logistic transformations are common in the literature (see, e.g., Chapter 1.8 in Gelman et al., 2013) and are used in this thesis whenever a transformation is necessary.

| Domain $A$ | $r : A \to \mathbb{R}$ | $r^{-1} : \mathbb{R} \to A$ |
|---|---|---|
| $x \in [a, \infty)$ | $r(x) = \log(x - a)$ | $r^{-1}(y) = \exp(y) + a$ |
| $x \in (a, b)$ | $r(x) = \log\left(\frac{x-a}{b-x}\right)$ | $r^{-1}(y) = \frac{b\exp(y)+a}{\exp(y)+1}$ |

**Table 3.1:** Transformations from different domains to $\mathbb{R}$ and back

While moving the parameter values by adding a normal error prevents degeneracy of the parameter sample, it artificially inflates its variance. In Liu and West (2001), this problem is mitigated by using a kernel smoothing step instead of simply adding random noise to each parameter value at each iteration of the algorithm. After the resampling step, the parameters are moved according to this kernel which is chosen such that the expected mean and variance of the parameter sample stays the same. For this, the weighted mean $\bar{\theta}_t$ and variance-covariance matrix $V_t$ of the parameters are calculated. In the forward propagation step of the algorithm, the parameter value of each particle is then moved according to the following process:

$$\theta^i_{t+1} = \lambda\theta^i_t + (1-\lambda)\zeta_t, \qquad \zeta_t \sim \mathcal{N}\left(\bar{\theta}_t, \frac{1+\lambda}{1-\lambda}V_t\right), 0 \le \lambda \le 1. \tag{3.2}$$

This move corrects for the over-dispersion of the parameters by pushing the new parameter value towards the mean. Combining this move with the resampling step shows that the new parameter values in the particles now come from the normal mixture distribution

$$p(\theta_{t+1}|\theta^{1:N}_t) = \sum_{i=1}^{N} w_i p_{\mathcal{N}}\left(\theta_{t+1}|\lambda\theta^i_t + (1-\lambda)\bar{\theta}_t, (1+\lambda)(1-\lambda)V_t\right),$$

where $p_{\mathcal{N}}(x|\mu, \Sigma)$ denotes the density of a multivariate normal distribution with mean $\mu$ and covariance matrix $\Sigma$ evaluated at $x$. Note that the change of the factor in the variance component arises from the factor $1-\lambda$ in Equation 3.2. A short calculation shows that this distribution has the same expected value and variance as the mean and sample variance of the particles before the move.

$$\begin{aligned}
\mathrm{E}(\theta_{t+1}) &= \sum_{i=1}^{N} w_i\left(\lambda\theta^i_t + (1-\lambda)\bar{\theta}_t\right) = \lambda\sum_{i=1}^{N} w_i\theta^i_t + (1-\lambda)\bar{\theta}_t\sum_{i=1}^{N} w_i \\
&= \lambda\bar{\theta}_t + (1-\lambda)\bar{\theta}_t = \bar{\theta}_t \\
\mathrm{Var}(\theta_{t+1}) &= \mathrm{Var}(\mathrm{E}(\theta_{t+1}|\theta^i_t)) + \mathrm{E}(\mathrm{Var}(\theta_{t+1}|\theta^i_t)) = \\
&\quad \mathrm{Var}(\lambda\theta^i_t + (1-\lambda)\bar{\theta})) + \mathrm{E}((1+\lambda)(1-\lambda)V_t) \\
&= \lambda^2 V_t + (1-\lambda^2)V_t = V_t
\end{aligned}$$

This kernel fixes the problem of the over-dispersion in Algorithm 3. However, the corresponding states are not moved, which means that the augmented state particles are no longer a sample from the joint distribution $p(x_t, \theta_t|y_{1:t})$. This can lead to considerable bias (see Chopin et al., 2013 for a numerical demonstration). To keep this bias small, the tuning parameter $\lambda$ is usually kept close to 1. Liu and West (2001) recommend a value of $\lambda$ between 0.974 and 0.995 while Thomas et al. (2019) even use $\lambda = 0.99997$.

Another technique employed by Liu and West (2001) in their algorithm is the use of an adjustment multiplier as explained in Section 2.2.7.3. They incorporate not only the expected value of $x^i_{t+1}|x^i_t$ but also the expected value of the parameter after moving it, which is $\lambda\theta^i_t + (1-\lambda)\bar{\theta}_t$. Some examples where this algorithm is used in practice are pedestrian navigation combining indoor maps and smartphone sensors (Yu et al., 2017) and the prediction of remaining battery life of lithium-ion batteries (Liu et al., 2011). It is also the basis of the algorithm that is currently used for inference for the Grey Seal model. The details of that exact algorithm are given in Paragraph 3.2.1.1 at the end of this section. In spite of its continued usage, de Valpine et al. (2022) note that the performance of the Liu-West-Algorithm is often poor compared to other more recent SMC algorithms.

We briefly describe two further algorithms that are based on the idea of augmenting the state with the parameter. The first is called *iterated filtering* (abbreviated IF2) and was proposed by Ionides et al. (2015). Similarly to the Liu-West-Algorithm, the parameters are moved by a kernel at every iteration. However, here the particle filter is run several times: the parameter input of each particle filter is the output of the previous one and the variance of the parameter kernel is decreased at every iteration. Under certain conditions, this leads to the convergence of the parameters to the maximum likelihood estimate. The second is the *data cloning* method by Lele et al. (2007) which relies on the same principle of running a particle filter iteratively to find parameter estimates. Here, the parameters are only moved with a Metropolis-Hastings kernel after each run of the particle filter rather than also at each iteration within it. Cloning the data here refers to re-using the same data multiple times for multiple simulated state trajectories $x^i_{0:T}|\theta$. Multiplying the likelihoods $p(y_{1:T}|x^i_{0:T}, \theta)$ for the acceptance ratio in the Metropolis-Hastings kernel (see Section 3.2.2.1) leads to samples from an exaggerated posterior, that is, the mode of the posterior becomes a more extreme peak. With enough copies of the data, this eventually leads to parameter samples that approximate the maximum likelihood estimate. Because we are mainly interested in Bayesian inference for the seal model, both of these algorithms for maximum likelihood inference are not further considered in this thesis.

#### 3.2.1.1 Case Study: Complete Seal Model

Here, we describe the algorithm that has been used for inference on the seal model until now. The data and some minor tuning choices in the algorithm are updated every one to two years. We use here the algorithm and data as described in Thomas et al. (2019). The algorithm is based on the Liu-West algorithm and is run 400 times with $N = 1,000,000$ each to provide sufficient Monte Carlo accuracy. It therefore includes a few further techniques to handle the numerical issues that arise when running the algorithm on a large scale.

The first of these is a technique called *rejection control*, which was introduced by Liu (2001). Here it is used both in the initialisation and in the final step of the algorithm. The idea behind this is to reduce the computational effort spent on particles with particularly low weight while only reducing the effective sample size slightly. An arbitrary weight limit

**Algorithm 4** Seal algorithm based on the Liu-West-Algorithm

```
choose tuning parameter $0 \leq \lambda \leq 1$ for kernel smoothing (Thomas et al., 2019 use $\lambda = 0.99997$)
while less than $N$ particles have been retained do
    $t \leftarrow 0$                                                    ▷ Initialisation
    for $i \leftarrow 1, ..., N$ do
        sample $\theta_0^i \sim p(\theta)$
        sample $x_0^i \sim f(x_0|\theta_0^i)$
        sample $x_1^i \sim f(x_1|x_0^i, \theta_0^i)$                    ▷ Look-ahead
        set weights $w_1^{i*} \leftarrow g(y_1|x_1^i, \theta_0^i)$       ▷ Weights for rejection control
    end for
    set $w_c \leftarrow 1/N \sum_{i=1}^N w_1^{i*}$
    retain each particle with probability $r_1^i = \min\left(w_1^{i*}/w_c, 1\right)$   ▷ Initial rejecton control
    update weights of retained particles $w_1^i \leftarrow w_1^{i*}/r_i$
end while
keep first $N$ of the retained particles, re-index with $1, ..., N$
for $t \leftarrow 1, ..., T$ do
    for $i \leftarrow 1, ..., N$ do
        transform parameters to real line $\tau_{t-1}^i \leftarrow h(\theta_{t-1}^i)$ with $h$ as in Table 3.1
    end for
    calculate weighted transformed parameter mean $\bar{\tau}_{t-1}$   ▷ Prepare kernel smoothing
    calculate weighted covariance matrix $V_{t-1}$
    for $i \leftarrow 1, ..., N$ do
        calculate expected value $x_t^{i*} \leftarrow \mathrm{E}(x_t|x_{t-1}^i)$
        calculate weights $w_t^{i*} \leftarrow w_{t-1}^i g(y_t|x_t^{i*})$
        calculate tempered resampling weights $q_t^i \leftarrow \sqrt[4]{w_t^{i*}}$
    end for
    resample $N$ particles with probability proportional to $\{q_t^i\}_{i=1...N}$ (residual resampling)
    re-index with $1, ..., N$
    for $i \leftarrow 1, ..., N$ do
        sample $\tau_t^{i*} \leftarrow \lambda\tau_{t-1}^i + (1-\lambda)\zeta_i$, where $\zeta_i \sim \mathcal{N}\left(\bar{\tau}_{t-1}, \frac{1+\lambda}{1-\lambda} V_{t-1}\right)$   ▷ Kernel smoothing
        set $\theta_t^i \leftarrow h^{-1}(\tau_t^{i*})$
        sample $x_t^i \sim f(x_t|x_{t-1}^i, \theta_t^i)$                 ▷ Project forward
        update weights $w_t^i \leftarrow \frac{w_{t-1}^i g(y_t|x_t^i, \theta_t^i)}{q_t^i}$
    end for
end for
for $i \leftarrow 1, ..., N$ do                                         ▷ Final rejection control
    standardize weights $v_T^i \leftarrow w_T^i \frac{\sum_j^N w_T^j}{N}$
end for
retain each particle with probability $r_T^i = \min\left(v_T^i/v_c, 1\right)$, where $v_c = 100$
update weights of retained particles $w_T^i \leftarrow v_T^i/r_T^i$
return a set of weighted particles containing state trajectories and parameters $\{x_{0:T}^i, \theta_T^i, w_T^i\}_{i=1...N}$
```

$w_c$ is set—here, the mean weight is chosen in the initialisation and 100 times the mean weight in the final step. All particles with a weight above this limit are automatically retained but particles with lower weights are discarded with probability $1 - w_i/w_c$. After this step, the weights of the resampled particles are adjusted by setting them to $w_c$ while the weights of the particles that were automatically retained due to their weight higher than $w_c$ remain unchanged. With these adjusted weights, the remaining particles after rejection control still represent the target distribution. Using this rejection control step in the initialisation means that only relatively promising parameter and state pairings are used to start the particle filter. In the final step it is done to reduce computer memory and to make post-processing feasible. After the particle filter has been run 400 times, an additional rejection control step with a rejection limit of 1,000 times the mean weight is undertaken before the remaining particles are combined. This further reduces the required computer memory while keeping a high effective sample size.

Another step that is necessary to avoid numerical underflow is the standardisation of the weights at each iteration so that their mean is approximately 1. As the divisor for the standardisation needs to be the same across all 400 runs of the particle filter, the divisors were pre-computed in a pilot run.

As implemented by Thomas et al. (2019), the algorithm with these settings took approximately 3 days to run, using 20 parallel processes. After final rejection control, the number of unique ancestor particles was 1669, from 400 million unique particles at initialisation. While we are not aware of a method for estimating the ESS for this algorithm, this number of unique ancestor particles can be treated as a lower bound of the ESS. The independent estimate was only incorporated at the very end by reweighting the particles. Calculating the ESS of this weighted sample, again via the conservative method of counting the number unique ancestor particles, led to an ESS of 478.

We initially experimented with improving the performance of this algorithm by improving the quality of the initial parameter sample. For this, we generated an approximated sample from the posterior distribution with the *unscented Kalman filter* (Wan and van der Merwe, 2001). This sample was then used instead of sampling parameters from the prior, and the weights adjusted to reflect this change. The idea was that by starting with parameters that were already relatively close to the posterior distribution, the particle degeneracy problem could be mitigated. This was however not the case and the technique did not show any clear improvements. One reason for this is that there is a lot of random variability in the forward propagation of the states. Therefore, even parameter values from high-density areas of the posterior distribution can get discarded in the resampling step because their state values do not match the observations. In addition, using the unscented Kalman filter requires a lot of tuning and careful implementation even for small changes to the model or data. We therefore did not further pursue this approach.

### 3.2.2 Particle MCMC and Variants

The algorithms in this section rely on simulating parameter values with *Markov chain Monte Carlo* (MCMC). This powerful statistical technique stands in contrast to direct sampling techniques like rejection sampling or importance sampling which was introduced in Section 2.2.1. Rather than the direct sampling of values, the idea behind MCMC is to construct a Markov chain whose stationary distribution is the target distribution. Then a sequence of values of this chain is simulated which approximate the target distribution. One key advantage of MCMC over other sampling methods is its ability to handle high-dimensional and complex distributions, where other techniques may be computationally intractable or may not be possible at all. For this reason, it is an important technique for inference for state-space models.

We first describe one of the most widely used MCMC methods, the *Metropolis-Hastings algorithm*, in detail. Then we discuss two techniques that rely on MCMC methods for parameter inference, namely data augmentation and the particle marginal Metropolis-Hastings (PMMH) algorithm. While the focus lies on the latter, we describe the first due to its popularity and use in well-known software packages like JAGS. We also discuss the difference between the PMMH algorithm and a similar algorithm called the *Monte Carlo within Metropolis* (MCWM) algorithm which differs in the way the likelihood estimate is treated.

#### 3.2.2.1 MCMC

Markov Chain Monte Carlo methods rely on constructing a sequence of random variables $\theta_0, \theta_1, \theta_2 ...$ that have the Markov property. This means that the distribution of $\theta_{m+1}$ given $\theta_m$ is independent of all previous values $\theta_0, ..., \theta_{m-1}$. To obtain samples from some target distribution, e.g., a posterior distribution given some data $p(\theta|y)$[1], the Markov process needs to have this target distribution as a stationary distribution. This means that we construct transition kernels $T_m(\theta_m|\theta_{m-1})$ such that if $\theta_{m-1}$ is a random sample from the target distribution, so is $\theta_m$. Then, under the additional conditions of aperiodicity and irreducibility (see, e.g., Chapter 11.2 in Gelman et al., 2013), the values are drawn from a distribution that more closely matches the target distribution with each iteration of the chain. For more details on the concepts on which MCMC builds, see Chapter 11 and 12 in Gelman et al. (2013).

The key challenge is the construction of the transition kernels $T_m$. Two standard and widely used methods are the Gibbs sampler and the Metropolis-Hastings algorithm. Many of the more advanced algorithms rely on these two methods as their building blocks. We therefore briefly describe these before moving on to algorithms that use them for inference for state-space models.

**Gibbs Sampler** When the parameter $\theta$ is multidimensional, the components of the parameter vector can be updated sequentially. Each component or sub-vector of $\theta$ is sampled conditionally on the other components. In many simple models with standard distributions, it is then possible to directly sample from the conditional target distribution $p(\theta_i|(\theta_1, ..., \theta_{i-1}, \theta_{i+1}, ..., \theta_d), y)$, while cycling through the components $\{\theta_i\}_{i=1,...,d}$. Even when direct sampling is not possible for all of the components, more involved methods need then only be applied to the components that require it.

**Metropolis-Hastings** The *Metropolis-Hastings* (MH) algorithm, introduced by Hastings (1970) who extended the method proposed by Metropolis et al. (1953), is a method to generate samples from the posterior distribution $p(\theta|y)$ when one can evaluate the prior density $p(\theta)$ and the likelihood $p(y|\theta)$ for any given parameter vector $\theta$ but direct conditional sampling of the parameter components as described above is impossible or for other reasons undesirable. This is achieved by proposing a new parameter value $\theta^*$, given the previous value in the chain, $\theta_{m-1}$, by sampling from a proposal distribution $q(\theta^*|\theta_{m-1})$.

[1]We use $y$ in this section to mean any type of data including observations in a state-space model, as MCMC methods can be used for many different types of models and not only time series.

Then an acceptance probability is calculated for this proposed value:

$$\alpha(\theta_{m-1}, \theta^*) = \min\left(\frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta_{m-1})p(\theta_{m-1})}\frac{q(\theta_{m-1}|\theta^*)}{q(\theta^*|\theta_{m-1})}, 1\right).$$

With probability $\alpha(\theta_{m-1}, \theta^*)$, the next value of the chain $\theta_m$ is set to the proposed value $\theta^*$, otherwise it is set to $\theta_{m-1}$. The first ratio in the acceptance probability compares how much higher the posterior density of the proposed value is compared to the previous value of the chain. This corresponds to the intuition that values with a larger posterior should be accepted with higher probability. The second ratio takes the proposal distribution $q$ into account and compensates for the fact that some values might more easily be proposed by the chain. The entire process is given in Algorithm 5.

**Algorithm 5** Metropolis-Hastings Algorithm

```
set $\theta_0$                                                        ▷ Initialisation
for $m \leftarrow 1, ..., M$ do
    sample $\theta^* \sim q(\cdot|\theta_{m-1})$                       ▷ Generate proposal value
    $\alpha(\theta_{m-1}, \theta^*) \leftarrow \min\left(\frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta_{m-1})p(\theta_{m-1})}\frac{q(\theta_{m-1}|\theta^*)}{q(\theta^*|\theta_{m-1})}, 1\right)$   ▷ Acceptance probability
    sample $u_m \sim \mathcal{U}(0, 1)$
    if $u_m < \alpha(\theta_{m-1}, \theta^*)$ then
        $\theta_m \leftarrow \theta^*$
    else
        $\theta_m \leftarrow \theta_{m-1}$
    end if
end for
```

There exist many different special cases of the Metropolis-Hastings algorithm. One is the *Metropolis algorithm* (Metropolis et al., 1953) which precedes the MH algorithm. Here, the proposal distribution for a new value is symmetric, i.e., $q(\theta^*|\theta_{m-1}) = q(\theta_{m-1}|\theta^*)$. This leads to a simplification of the acceptance probability as the second ratio cancels. In this case, $q(\theta^*|\theta_{m-1})$ is often chosen to be a random walk with $\theta^* = \theta_{m-1} + \epsilon, \epsilon \sim f$ (*random walk Metropolis*). A common choice for $f$ is a normal distribution. Similarly to the Gibbs sampler, not all components of the parameter need to be updated simultaneously but can be updated sequentially. It is even possible to combine different methods to update different components, e.g., use direct Gibbs sampling where possible and a MH kernel otherwise.

One of the key challenges of the Metropolis-Hastings algorithm is the design of a "good" proposal distribution whereby the chain can efficiently explore the parameter space of the posterior distribution. Even for a normal proposal distribution in a random walk Metropolis algorithm, the difficulty in tuning the proposal can be appreciated: if the variance parameter of the proposal distribution is too small, successive values of the chain are highly correlated and it takes many iterations of the chain for the exploration of the parameter space. If it is too high, often a value will be proposed that lies far outside of the plausible range of the posterior distribution and will likely be rejected. Much research has been undertaken to address this and we only mention two examples here. The *adaptive Metropolis algorithm* (Haario et al., 2001) varies the variance of the proposal distribution based on the history of the chain and thereby achieves a desirable acceptance ratio. A more advanced technique is *Hamiltonian Monte Carlo* (HMC) (see, e.g., Neal, 2012) which uses Hamiltonian dynamics to propose distant jumps that still result in proposed values with high posterior density and therefore reduces the autocorrelation of the samples in the

Markov chain. However, it requires the calculation of the slope of the target distribution which is a difficult task in a state-space model. It also tends to perform well in situations where the shape of the posterior is such that the Markov chain struggles to explore the space of the distribution when a simpler proposal distribution such as a random walk is used. As this is not the case in the seal model, we do not explore this technique in this thesis. We refer to Buchholz et al. (2021) for some recent research on combining SMC methods with HMC.

**Diagnostic Tools for MCMC algorithms** We mention two concepts here to assess the quality of a chain of values produced by an MCMC algorithm. The first is diagnosing whether a chain has approximately converged, that is, is no longer strongly dependent on its starting conditions and is producing samples from the target distribution. The second is to quantify how efficient the algorithm is in producing samples from the target distribution once converged.

To assess whether convergence has been achieved, we use the *potential scale reduction factor* $\hat{R}$ proposed by Gelman and Rubin (1992) (extended by Brooks and Gelman, 1998 for the multivariate case) which uses an analysis of variance technique to compare the posterior estimates from (at least) two chains with overdispersed starting values. The idea is to estimate the variance of the posterior distribution both by computing the mean of the variances of each chain (after discarding some initial values as burn-in), and by computing the overall variance of all chains combined. If the chains have converged, both values are unbiased estimates of the posterior variance. Otherwise, the first method will generally underestimate the variance as the chains have not yet explored the entire range of the posterior distribution, and the second method will overestimate the variance, as the starting values were overdispersed. The convergence diagnostic for a one-dimensional parameter for $m$ chains with $n$ iterations each is

$$\hat{R} = \sqrt{\frac{(d+3)\hat{V}}{(d+1)W}}, \tag{3.3}$$

with

$$\begin{aligned} \hat{V} &= \hat{\sigma}^2 + \frac{B}{mn}, \\ d &= \frac{2\hat{V}^2}{\mathrm{Var}(\hat{V})} \end{aligned}$$

$W$ is the mean of the empirical variances within each chain, $B$ is $n$ times the empirical variance of the means of the chains and $\hat{\sigma}^2$ the empirical variance from all chains together. This measure and variants for the multivariate case can be calculated in `R` with the `coda` package (Plummer et al., 2006). Values close to 1 indicate covergence.

In addition, trace plots of two chains with different starting values are a useful tool to understand when two chains might have converged. These are line graphs with the iteration of the Markov chain on the x-axis and the value of one of the components of the parameter vector on the y-axis (e.g., Figure 3.6). When two chains that started at different values have converged, the trace plots shows lines that cover the same area of the parameter space. This visual aid also helps with determining the burn-in period of the chain, that is, the number of iterations at the start of the chain before it has converged. The samples from this period are discarded for any analysis of the posterior distribution.

It is important to note we can generally not prove convergence but these methods can identify some cases where the chains have not converged.

The Monte Carlo accuracy of the samples produced by a converged MCMC chain can be assessed using a measure for the effective sample size. In Equation 2.4 in Chapter 2, such a measure was given for importance samples, where a high variance of the weights leads to a decreased effective sample size. For MCMC samples, the individual parameter values in a chain are often highly autocorrelated. This means that a sequence of sampled values is usually much less informative than an independent sample of the same size would be. This concept can be quantified by incorporating the autocorrelation of the sequence into the measure for the effective sample size. The idea is to calculate the Monte Carlo variance when estimating $\mathrm{E}(\theta|y)$ which would be $1/n\,\mathrm{Var}(\theta|y)$ for $n$ independently sampled values. The ESS is then the size of a sample of i.i.d. values that would lead to the same Monte Carlo variance of the estimate of $\mathrm{E}(\theta|y)$ as when estimated the autocorrelated values of the MCMC chain. For one variable, the effective sample size is defined as

$$\mathrm{ESS}_{MCMC} = \frac{n}{1 + \sum_{t=1}^{\infty} \rho_t}, \tag{3.4}$$

where $\rho_t$ is the lag-$t$ autocorrelation of the values of the MCMC chain (Gelman et al., 2013). For the multivariate case, Vats et al. (2019) generalise this measure to incorporate the multivariate dependence structure. Since the individual parameter values in a chain are usually highly autocorrelated, the ESS is typically much lower than the number of iterations and depends on how well the chain has mixed. Good mixing here means that the Markov chain efficiently explores the parameter space of the posterior distribution. We use the `coda` package (Plummer et al., 2006) to estimate the ESS of the chains for the one-dimensional case, and the `mcmcse` package (Flegal et al., 2021) for the multivariate case.

#### 3.2.2.2 Data augmentation

The MCMC methods described above all rely on the computation of the data likelihood $p(y|\theta)$. However, other than for a small number of special cases (see, e.g., the example in Equation 2.12 in Chapter 2), this likelihood is intractable for state-space models where the true states are unknown. One popular method to deal with this problem is *data augmentation* (Tanner and Wong, 1987). Here, the parameter $\theta$ is augmented with the underlying unknown states $x_{0:T}$. This leads to a straightforward calculation of the likelihood $\mathcal{L}(\theta, x_{0:T}; y_{1:T}) = p(y_{1:T}|\theta, x_{0:T})$. Standard MCMC methods can then be used to obtain a sample from the joint posterior distribution $p(\theta, x_{0:T}|y_{1:T})$. To obtain samples from the parameter posterior distribution, only the marginal distribution of $\theta$ is considered: $\int p(x_{1:T}, \theta|y_{0:T})dx_{1:T} = p(\theta|y_{0:T})$. In practice, this means that the samples states are discarded (see p. 293 in Gelman et al., 2013). These types of algorithms can take much longer to converge as the parameter space under consideration is larger but have the advantage of being more general and not requiring the specific structure of a state space model. The joint posterior distribution, while often high-dimensional, can be treated like any other posterior distribution and therefore opens up the inference problem to the whole range of available MCMC methods. This is the approach to inference taken in software like JAGS (Plummer, 2003) and OpenBUGS (Spiegelhalter et al., 2003) when an SSM is modelled. However, it often suffers from performance problems if the states and the parameters are highly correlated (Borowska and King, 2023). We do not explore data augmentation in this chapter, as it shows no potential for the complete seal model due to the high correlation between the parameters and the components of the state. However, it is used as one of three methods for the simpler 2-state model in Chapter 4, where we see that even for some cases for the 2-state model it does not converge within a feasible time period, e.g., in Section 4.3.3.3.

#### 3.2.2.3 Particle MCMC

As described above, not being able to calculate the likelihood $p(y|\theta)$ prevents us from directly using a Metropolis-Hastings algorithm with target distribution $p(\theta|y)$. However, as established in Section 2.2.3, we can use a particle filter algorithm to obtain an unbiased estimate of this likelihood. This leads us to the question of whether this estimate can be used to replace the exact likelihood. This approach was first proposed by Fernández-Villaverde and Rubio-Ramírez (2007) but only in Andrieu et al. (2010) was it shown that, remarkably, the target distribution indeed remains the same. Since then, it has been applied in many areas such as systems biology (Golightly and Wilkinson, 2011), natural language processing (Dubbin and Blunsom, 2012), finance (Pitt et al., 2012) and ecology, e.g., for population modelling (Knape and de Valpine, 2012, White et al., 2016) and range expansion (Osada et al., 2019).

The idea follows from the more general theory of pseudo-marginal samplers (see, e.g., Andrieu and Roberts, 2009 for a discussion of their properties). There, we assume that we can construct a random variable $Z \sim p(z|\theta)$ and a function $L(\theta, z)$ such that $\mathrm{E}(L(\theta, Z)) = p(y|\theta)$. If the new proposal $z^*$ only depends on $\theta^*$ and is independent of the previous value of $z$, the MH algorithm can than be used to sample from the augmented space $\Theta \times \mathcal{Z}$, replacing the likelihood term in the acceptance ratio with $L(\theta, z)$. Even though this approximation is used, it can be shown that the marginal distribution for $\theta$ of the invariant distribution of the Markov chain is the posterior distribution $p(\theta|y)$ (see Proposition 16.1 in Chopin and Papaspiliopoulos, 2020 for a proof). Andrieu and Roberts (2009) discuss the properties of this sampler in detail. In particular, they compare the pseudo-marginal sampler with a possibly more intuitive Monte Carlo approximation to the MH algorithm, where the likelihood $p(\theta|y)$ is newly estimated at each iteration by sampling a new value for $z$, rather than augmenting the space with $z$. This second approach does not admit $p(\theta|y)$ as the invariant distribution. However, if the likelihood estimate $L(\theta, Z)$ is the result of importance sampling with sample size $N$, under mild assumptions, the invariant distribution converges to the posterior distribution $p(\theta|y)$ as $N \to \infty$. We further discuss the notion of re-estimating the likelihood in Section 3.2.2.4.

In the Particle Marginal Metropolis Hastings algorithm (PMMH) as proposed by Andrieu et al. (2010), a particle filter is run at each iteration of the MH algorithm. The random variable $Z$ is then the collection of particles from the particle filter, and the likelihood estimate $L(\theta, Z)$ is the one computed by the filter. One step of the PMMH is given in Algorithm 6. Here, we obtain samples from the joint posterior distribution $p(x_{0:T}, \theta|y_{0:T})$ but if the interest lies only in the posterior distribution of the parameter $\theta$, the steps for sampling a smoothed state trajectory $x_{0:T}$ can be omitted as this is not required in the acceptance probability.

As with all MCMC algorithms, the PMMH requires a number of tuning choices that affect the performance of the algorithm. First of all, we can choose any particle filter variant to calculate the unbiased likelihood estimate. A crucial choice is balancing the number of particles $N$ of the particle filter with the number of iterations $M$ of the MCMC sampler. Using strong assumptions, a number of theoretical results exist that recommend

**Algorithm 6** Particle Marginal Metropolis-Hastings (One Step)

```
Require: $\theta_{m-1}, X_{0:T}(m-1), \hat{p}_{m-1}(y_{1:T}|\theta_{m-1})$
  sample $\theta^* \sim q(\cdot|\theta_{m-1})$                          ▷ Generate proposal value
  run particle filter to generate $\hat{p}^*(y_{1:T}|\theta^*)$ and to sample $X^*_{0:T}$
  $\alpha(\theta_{m-1}, \theta^*) \leftarrow \min\left(\frac{\hat{p}(y|\theta^*)p(\theta^*)}{\hat{p}_{m-1}(y|\theta_{m-1})p(\theta_{m-1})}\frac{q(\theta_{m-1}|\theta^*)}{q(\theta^*|\theta_{m-1})}, 1\right)$   ▷ Acceptance probability
  sample $u_m \sim \mathcal{U}(0,1)$
  if $u_m < \alpha(\theta_{m-1}, \theta^*)$ then
      $\theta_m \leftarrow \theta^*$, $X_{0:T}(m) \leftarrow X^*_{0:T}$, $\hat{p}_m(y_{1:T}|\theta_m) \leftarrow \hat{p}_{m-1}(y_{1:T}|\theta_{m-1})$
  else
      $\theta_m \leftarrow \theta_{m-1}$, $X_{0:T}(m) \leftarrow X_{0:T}(m-1)$, $\hat{p}_m(y_{1:T}|\theta_m) \leftarrow \hat{p}^*(y_{1:T}|\theta^*)$
  end if
```

selecting $N$ to achieve a certain standard deviation of the log-likelihood estimates. For example, it is recommended to aim for a value of around 0.9 for the standard deviation of the log-likelihood estimates at a central value of the posterior distribution if the proposal distribution for $\theta$ is the perfect proposal $p(\theta|y_{1:T})$, i.e., the posterior distribution itself (Pitt et al., 2012). In the case where the parameter posterior distribution can be factorised into $d$ independent and identically distributed components, this value becomes 1.8 as $d \to \infty$ (Sherlock et al., 2015). As none of these assumptions hold for the seal model, simulation studies in Sections 3.3.1 and 3.3.2 are dedicated to establishing the optimal number of particles for the different versions of the seal model.

We also need to be careful when selecting a proposal distribution for the MH algorithm. While selection of the optimal proposal distribution for standard MH algorithms is a well-researched topic (see, e.g., Chapter 12 in Gelman et al., 2013), these results cannot be directly applied to the PMMH algorithm. The basic strategy for tuning the proposal distribution in a standard MH algorithm is often to aim for a specific acceptance ratio of the proposed values. It was shown by Gelman et al. (1997) that, under some strong assumptions, this optimal ratio is 44% for a one-dimensional parameter and converges to 23.4% as the dimension of the parameter space tends to infinity. Under weaker assumptions, a popular rule of thumb is often to aim for an acceptance ratio between 0.2 and 0.4 (Chopin and Papaspiliopoulos, 2020, Chapter 15). However, this is not a helpful strategy for the PMMH, as the acceptance probability is not only affected by the proposal distribution but also by the variance of the likelihood estimate. Often, the chain will reject new proposals in favour of a parameter value with a particularly high likelihood estimate for many iterations, even though the newly proposed values have a high posterior density and would be accepted with a high probability if the likelihoods could be calculated exactly. Adapting the scale of the proposal distribution is therefore not helpful for increasing the acceptance ratio because the cause is the high variance of the likelihood estimate rather than a proposal distribution with a too large variance.

We demonstrate this with the 2-state model. The number of particles is set at 3 to show the consequence of a likelihood estimate with a high variance. The proposal distribution $q$ is multivariate normal, where the proposal covariance $\Sigma_q$ is proportional to the posterior covariance $\Sigma_{\theta|y}$ obtained from a pilot run, so $\Sigma_q = h\Sigma_{\theta|y}, h > 0$. We attempt to tune $h$ such that the acceptance ratio is close to the recommended 0.234. As can be seen in Figure 3.2 and Table 3.2, the acceptance rate increases as $h$ decreases but never reaches the target value of 0.234. Decreasing the scaling factor even further to $h = 0.01$ resulted in the chains not converging. Using only the acceptance ratio as a criterion, these results

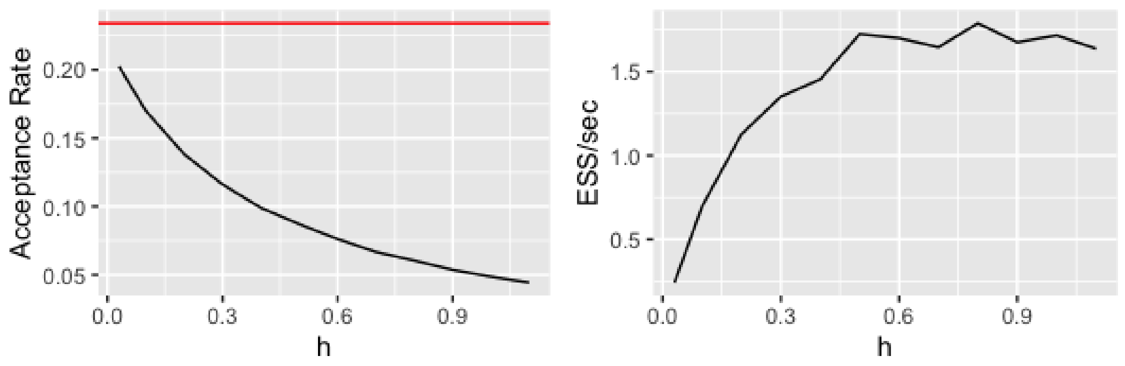


**Figure 3.2:** The acceptance rate and effective sample size per second when running the PMMH for the 2-state model with $N = 3$ particles and $M = 10,000$ iterations while varying the scale $h$ of the covariance of the proposal distribution.

might suggest to choose $h$ as low as possible while the chains still converge. However, Figure 3.2 shows the opposite effect for the effective sample size per runtime. As the ultimate goal is to create a sample with a high effective sample size, this means that the acceptance ratio is not a useful measure when constructing an effective PMMH sampler. For the remainder of this chapter, ESS per runtime will be used as the measure to evaluate algorithm efficiency.

| $h$ | Runtime (sec) | $\hat{R}$ | Acceptance Rate | ESS | ESS/sec |
|---|---|---|---|---|---|
| 0.01 | 9425.93 | 6.27 | 0.11 | - | - |
| 0.03 | 8456.20 | 1.01 | 0.20 | 2057.37 | 0.24 |
| 0.10 | 8244.27 | 1.00 | 0.17 | 5741.19 | 0.70 |
| 0.20 | 8286.23 | 1.00 | 0.14 | 9339.82 | 1.13 |
| 0.30 | 7746.05 | 1.00 | 0.12 | 10466.38 | 1.35 |
| 0.40 | 7563.62 | 1.00 | 0.10 | 10994.95 | 1.45 |
| 0.50 | 7437.15 | 1.00 | 0.09 | 12816.09 | 1.72 |
| 0.60 | 7252.47 | 1.00 | 0.08 | 12321.35 | 1.70 |
| 0.70 | 7077.63 | 1.00 | 0.07 | 11642.44 | 1.64 |
| 0.80 | 6930.96 | 1.00 | 0.06 | 12390.02 | 1.79 |
| 0.90 | 6761.22 | 1.00 | 0.05 | 11320.21 | 1.67 |
| 1.00 | 6434.97 | 1.00 | 0.05 | 11040.03 | 1.72 |
| 1.10 | 6454.60 | 1.00 | 0.04 | 10560.55 | 1.64 |

**Table 3.2:** Results from running the PMMH for the 2-state model with $N = 3$ particles and $M = 10,000$ iterations while varying the scale $h$ of the proposal distribution. For $h = 0.01$, the chains did not converge.

In addition to the PMMH, Andrieu et al. (2010) also describe a similar alternative called the *particle Gibbs* sampler. Here, the algorithm samples iteratively from $p(\theta|y_{1:T}, x_{1:T})$ and $p(x_{1:T}|\theta, y_{1:T})$. The algorithm uses SMC to sample from the latter, but requires a model that enables sampling from the former. As this is not easily possible in the seal model, this sampler is not further discussed here.

A recently developed modification of the PMMH algorithm is the so called *correlated pseudo-marginal algorithm* by Deligiannidis and Doucet (2018). Here, the likelihood ratio in the Markov chain is computed with two correlated likelihood estimates. This is achieved by correlating the particles in the particle filter that estimates the likelihood of the newly proposed value with the ones of the current parameter value. This reduces the relative variance of the likelihood ratio and improves the mixing of the Markov chain. This technique has successfully been used in various applications, e.g., Golightly et al. (2019) and Persson et al. (2022). Applying it here goes beyond the scope of this thesis but is a promising direction for further research.

#### 3.2.2.4 Likelihood Re-estimation (MCMW)

In the previous section, it became clear that occasional very high likelihood estimates negatively affect the mixing of the Markov chains. These can occur occasionally due to the right skew of the distribution of the likelihood estimates, as was seen in, e.g., Figure 2.9 and prevent the acceptance of a newly proposed parameter for many iterations in a row.

A naïve approach to improve mixing could be to simply re-estimate the likelihood at every iteration. This algorithm is called *Monte Carlo within Metropolis* (MCWM) in Andrieu and Roberts (2009) where a detailed comparison between MCMW and PMMH is given. While the likelihood re-estimation might improve the mixing of the Markov chains, the invariant distribution of the Markov chain is then no longer necessarily the posterior distribution of the parameters. Formally, the Markov chain in the PMMH produces samples from the extended parameter space which contains both the parameters and the state particles. The estimate of the posterior distribution is then the result of marginalising over the particles and only considering the parameter values at each iteration.

To better understand this and the effect of using an unbiased likelihood estimate in place of the exact likelihood in the PMMH, we study both algorithms and a standard MH algorithm with a toy example. Inspiration for the construction of $L(\theta, z)$ is taken from Example 16.4 in Chopin and Papaspiliopoulos (2020), p.301, but we substantially extend the example to show both the effect of introducing an auxiliary variable $Z$ on the mixing of the chain, and the effect of re-estimating $Z$. We consider a model with a single data point $y \sim \text{Ber}(\theta)$ and place a discrete prior on $\theta$: $\mathbb{P}(\theta = 0.2) = \mathbb{P}(\theta = 0.8) = 0.5$. Suppose we observe $y = 1$. A short calculation shows that the posterior distribution is then

$$\begin{aligned}\mathbb{P}(\theta = 0.2|y = 1) &= \frac{\mathbb{P}(y = 1|\theta = 0.2)\mathbb{P}(\theta = 0.2)}{\mathbb{P}(y = 1|\theta = 0.2)\mathbb{P}(\theta = 0.2) + \mathbb{P}(y = 1|\theta = 0.8)\mathbb{P}(\theta = 0.8)} = 0.2 \\ \mathbb{P}(\theta = 0.8|y = 1) &= 0.8.\end{aligned}$$

While in this case, the posterior can easily be calculated, we now compare the performance of a standard MH algorithm with the PMMH and the MCWM by arbitrarily introducing a random variable $Z$ to replace the likelihood term in the acceptance ratio with an unbiased estimate. With $c \in (0, 1]$, we sample $Z \sim \text{Ber}(c)$ and set the likelihood estimate to

$$L(\theta, z) = \begin{cases} \text{if } \theta = 0.8: & \frac{z}{c}\mathbb{P}(y|\theta) \\ \text{if } \theta = 0.2: & \mathbb{P}(y|\theta). \end{cases}$$

The expected value of $L(\theta, Z)$ is indeed $\mathbb{P}(y|\theta)$ for both values of $\theta$. As a proposal distribution for $\theta$, we choose $q(\theta^*|\theta) = 1 - \theta$, and so deterministically always propose the other possible value of $\theta$. With these simple choices and $y = 1$, the acceptance probability for the MH algorithm becomes

$$\alpha_{MH}(\theta, \theta*) = \min\left(\frac{\mathbb{P}(y|\theta^*)}{\mathbb{P}(y|\theta)}, 1\right) = \begin{cases} 1 & \text{if } \theta^* = 0.8 \\ 0.25 & \text{if } \theta^* = 0.2. \end{cases}$$

Running a standard MH algorithm with these settings clearly produces samples from the correct target distribution, as Figure 3.3a together with the estimation of $\mathbb{P}(\theta = 0.2|y = 1)$ as 0.202 confirms.

For the PMMH, the acceptance probability is

$$\alpha_{MH}\left((\theta, z), (\theta^*, z^*)\right) = \min\left(\frac{L(\theta^*, z^*)}{L(\theta, z)}, 1\right) = \begin{cases} \min\left(4z^*/c, 1\right) = z^* & \text{if } \theta^* = 0.8 \\ 0.25c & \text{if } \theta^* = 0.2. \end{cases}$$

The random variable $z$ in the second case can be omitted because $\theta = 0.8$ could only ever have been accepted as a value in the chain for $z = 1$. Comparing these acceptance probabilities with the ones for the MH algorithm above, we see that including the auxiliary variable $z$ reduces the acceptance probability in both directions by the factor $c$. It is immediately clear that, while the pseudo-marginal MH has the same marginal invariant distribution for $\theta$ as the MH algorithm for any $c \in (0, 1]$, the mixing of the chain becomes poorer the smaller we set $c$. This is also shown in Figure 3.3b.

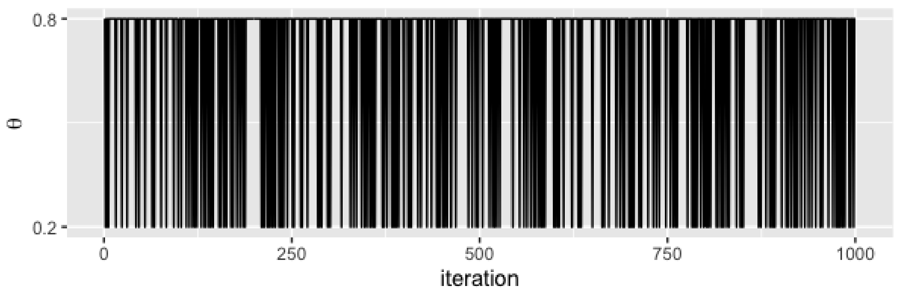
Lastly, we study the effect of re-estimating the likelihood for both parameters at each iteration, so using the MCWM algorithm.

$$\begin{aligned}\alpha_{MH}\left(\theta, \theta^*\right) &= \min\left(\frac{L(\theta^*, z^*)}{L(\theta, z)}, 1\right) \\ &= \begin{cases} \min\left(4z^*/c, 1\right) = z^* & \text{if } \theta^* = 0.8 \\ \min\left(0.25c/z, 1\right) = \left\{\begin{matrix} 0.25c \text{ if } z = 1 \\ 1 \text{ if } z = 0 \end{matrix}\right\} = 0.25c^2 + 1 - c & \text{if } \theta^* = 0.2. \end{cases}\end{aligned}$$

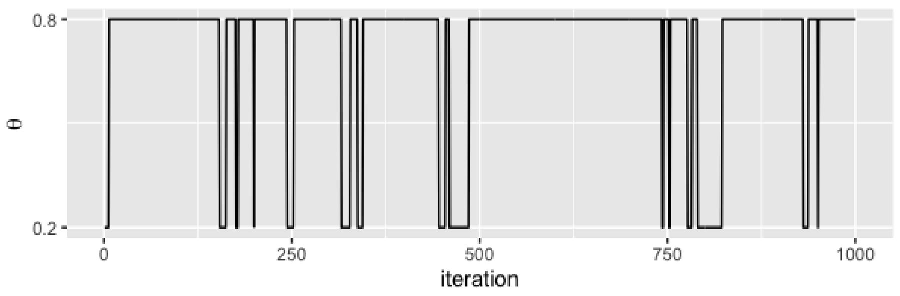
In the second case, for $z = 0$, we use the convention that dividing by 0 gives $\infty$ and the minimum of both values is then 1. In comparison to the PMMH above, the probability to switching from $\theta = 0.8$ to $\theta^* = 0.2$ is higher than it should be. For example, with $c = 0.1$, the transition probability for switching from 0.8 to 0.2 becomes 0.9025 rather than the correct 0.025 as in the PMMH. Figure 3.3c shows that while the acceptance rate is much increased compared to Figure 3.3b (acceptance rate 18.7% vs. 3.5%), this leads to a wrong estimate of $\mathbb{P}(\theta = 0.2|y = 1)$ which is here estimated to be 0.903 instead of the correct 0.2. The lower the value of $c$ and therefore the higher the variance of the unbiased likelihood estimate $Z$, the higher is the bias of the estimate of the posterior distribution.

Even though it introduces a bias in the estimation of the posterior distribution, re-estimation is occasionally done to overcome poor mixing of the PMMH. In Kattwinkel and Reichert (2017), a compromise to improve mixing while keeping the bias small is suggested whereby the likelihood is only re-estimated when the Markov chain has not accepted a new parameter value for 20 iterations. They state that the bias introduced by this measure is likely to be small since re-estimation only occurred in 0.38% of all iterations.

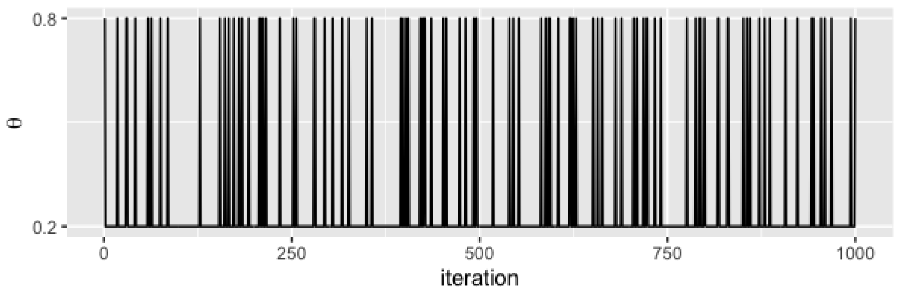
We tested this approach with the toy example with $c = 0.1$, increasing the number of iterations to $M = 100,000$. Here, re-estimating only after 20 rejections of the newly proposed value led to an estimate of $\mathbb{P}(\theta = 0.2|y = 1) = 0.386$ where the likelihood was re-estimated in 2.63% of all iterations (counting only re-estimates when $\theta = 0.8$ since the likelihood is constant for $\theta = 0.2$). A more extreme case was tested by decreasing $c$ to 0.004, which led to re-estimation in 0.342 % of iterations, similar to the percentage given in Kattwinkel and Reichert (2017). These settings led to an estimate of $\mathbb{P}(\theta = 0.2|y = 1) = 0.931$ rather than the true probability 0.2. This shows that even if re-estimation occurs rarely, the bias can be significant compared to the PMMH algorithm. We do not



**(a)** Standard MH algorithm. Estimating $\mathbb{P}(\theta = 0.2|y = 1)$ with this sample gives 0.202.



**(b)** Pseudo-marginal MH algorithm. Estimating $\mathbb{P}(\theta = 0.2|y = 1)$ with this sample gives 0.133.



**(c)** Pseudo-marginal MH with re-estimation of the likelihood. Estimating $\mathbb{P}(\theta = 0.2|y = 1)$ with this sample gives 0.903.

**Figure 3.3:** Trace plots for estimating $p(\theta|y = 1)$ with three different MCMC methods. The chains have length $M = 1000$. The probability of sampling $z = 1$ is set to $c = 0.1$. The correct posterior probability for $\mathbb{P}(\theta = 0.2|y = 1)$ is $0.2$.

pursue the approach of only occasionally re-estimating further for the seal model but note that this is an avenue for further research.

### 3.2.3 SMC$^2$

An algorithm that combines the ideas of the Liu-West algorithm and Particle MCMC is the so-called *SMC$^2$* by Chopin et al. (2013). As in the Liu-West algorithm, $N_\theta$ parameter values are first sampled from the prior and then resampled at every iteration $t$ (or if some degeneracy criterion is fulfilled, see Section 2.2.6) and moved by a transition kernel to maintain diversity in the sample. However, in contrast to the Liu-West algorithm, each parameter value is attached to its own particle filter with $N_x$ particles instead of just one particle. The resampling weight of a parameter is then not the importance weight of a single particle but rather the likelihood as estimated by the particle filter associated with that parameter, similar to particle MCMC algorithms. The second important distinction is that whenever a parameter is newly sampled from a kernel, the states of the associated particles are re-generated from the start of the time-series to the current iteration $t$. This prevents the bias of the Liu-West algorithm, where states and weights are not updated together with the parameter. The steps of the SMC$^2$ are given in Algorithm 7, where the details of each particle filter attached to a parameter $\theta_m$ are omitted. We note that all techniques mentioned in Chapter 2 can be used for the particle filter associated with each parameter value. We also only give the algorithm with the PMMH kernel but other choices are possible (see 18.2.2 in Chopin and Papaspiliopoulos, 2020). This is the recommended choice and helps reduce the memory cost of the algorithm.

**Algorithm 7** SMC$^2$ algorithm

```
for $m \leftarrow 1, ..., N_\theta$ do
    sample $\theta^m \sim p(\theta)$ and set $w^m \leftarrow 1$
    initialise PF with particles $x_0^{(1:N_x,m)}$ and weights $w_0^{(1:N_x,m)}$
end for
for $t \leftarrow 1, ..., T$ do
    for $m \leftarrow 1, ..., N_\theta$ do
        perform iteration $t$ of the PF, generating particles $x_t^{(1:N_x,m)}$ and weights $w_{t,\theta}(x_t^{i,m})$
(see Algorithm 2)
        Compute $\hat{p}(y_t|y_{t-1}, \theta^m) = \frac{1}{N_x}\sum_{i=1}^{N_x} w_{t,\theta}(x_t^{i,m})$
        update importance weights $w^m \leftarrow w^m \hat{p}(y_t|y_{t-1}, \theta^m)$
        if resampling condition met then
            resample parameters $\tilde{\theta}^m$) with probabilities proportional to $\{w^m\}_{m=1...N_\theta}$
            sample $\tilde{\theta}^m \sim q(\cdot|\theta^m)$                                ▷ PMMH kernel
            run new PF for $\tilde{\theta}^m$ and compute $\hat{p}(y_{1:t}|\tilde{\theta}^m)$
            calculate MH acceptance probability $\alpha$ (see Algorithms 5 and 6),
            With probability $\alpha$, set $\theta^m \leftarrow \tilde{\theta}^m$ (and replace the associated PF)
            $w^m \leftarrow 1$                                                  ▷ Set weights
        end if
    end for
end for
return a set of parameter values and weights $\{\theta^m, w^m\}_{m=1...N_\theta}$
```

As with the PMMH, finding the right balance between the number of parameters $N_\theta$ and the size of each particle filter $N_x$ can be challenging. It is even possible to dynamically increase $N_x$ as the algorithm progresses. One reason for this is that, as explained in Section 2.2.3, the relative variance of the likelihood estimate for each parameter can be bounded by $C_\theta t/N_x$ which justifies the linear increase of $N_x$ with $t$. When and how exactly this increase is appropriate has not yet been fully answered, although some work on this question has been done by Chopin et al. (2015). One suggestion in Chopin et al. (2013) is to double the number of particles $N_x$ when the acceptance rate of the PMMH kernel falls below a certain threshold, e.g., 10%.

Other than optimising $N_x$ and $N_\theta$, there are many more options to tune this algorithm. For the PMMH kernel, the type and scale of the proposal distribution can be chosen. For example, the covariance of the proposal can be adapted to be proportional to the empirical covariance of the current parameter sample at every time step, similarly to the distribution in Equation 3.2. The resampling criterion often means monitoring the ESS of the parameter particles, and setting a threshold, e.g., $0.5N_\theta$, which triggers the resampling step.

Assessing the Monte Carlo error of the generated sample is less straightforward than it is for the PMMH as the target distribution changes throughout the algorithm. At every iteration $t$, a weighted sample from $p(\theta|y_{1:t})$ is produced. This means that it is difficult to assess the quality of the sample, as a large change in the sample at the very last time step could be due to Monte Carlo variance but also represent a true large difference between $p(\theta|y_{1:(T-1)})$ and $p(\theta|y_{1:T})$. Finding a measure for the effective sample size is equally challenging. Using the formula for the ESS of weighted samples as in Equation 2.4 over-estimates the amount of information in the sample. For example, right after the resampling step, the importance weights of all parameter particles are set to 1. If all proposed new parameter values were accepted and therefore the parameter sample consisted of $N_\theta$ unique values, the ESS when calculated as in Equation 2.4 was $N_\theta$. However, this number does not show that many of these parameter particles might be highly correlated, e.g., if they all originated from the same ancestor parameter particle and had only been moved through a kernel with a small variance. On the other hand, a very conservative way to quantify the diversity of the parameter particles measure is to monitor the ancestor particle of each parameter particle and calculate how many different ancestors are represented in the final sample. These two approaches give a lower and upper limit of the true ESS but this does generally not lead to a useful measure.

Another difficulty with this algorithm is the required memory size which is $\mathcal{O}(N_\theta N_x)$ when the PMMH kernel is used and can even be $T\mathcal{O}(N_\theta N_x)$ for other less parsimonious kernels. One suggestion given in Chapter 18.3 in Chopin and Papaspiliopoulos (2020) is to repeat runs of the algorithm with a lower number for $N_\theta$. This reduces the memory cost and allows parallelisation to reduce the overall runtime.

## 3.3 Empirical Investigations of SMC Methods for Parameter Inference

In this section, we implement some of the methods described above and study different aspects of them. The four different models first introduced in Section 1.2 are used for this. As a reminder, these are the 2-state model with 6 parameters, the 7-state model with 6 parameters, the complete seal model without independent estimate of total adult abundance, which has 9 parameters, and the complete seal model with independent estimate, which has 10 parameters. For the last, we use the real data from 1984 to 2010, whereas for the first three, we use simulated data. The goal is to find an algorithm that produces samples from the posterior distribution for the complete seal model in feasible computing time.

We initially explore various strategies to adapt the PMMH algorithm for the different versions of the seal model. First, we study the effect of the choice of proposal distribution and the size of its variance on the mixing of the chain. We also study how the number of particles interact with the scale of the proposal distribution. As in Chapter 2, these computing-intensive explorations are undertaken only with the 2-state model. In the next step, the findings about the proposal distribution are transferred for the other more complex models while we adapt the optimal number of particles as necessary. We also discuss the effect of modifying the PMMH algorithm by re-estimating the likelihood at each iteration.

Lastly, we illustrate some features of the SMC$^2$ algorithm by applying it to the 2-state model but also show that this algorithm does not work with the complete seal model.

### 3.3.1 PMMH: Tuning the Proposal Distribution

Here, we study how the choice of proposal distribution in the MH kernel in combination with the number of particles affects the mixing of the chain. The key things we explore are the number of particles $N$ to estimate the likelihood in relation to the number of MCMC iterations $M$, the scale of the variance of the proposal distribution, and a possible transformation of the parameters to improve proposals.

We propose new parameter values jointly with a multivariate normal distribution but do so first using the untransformed parameters, even though their support is only a subset of the real line. After studying the effect of the scale of the covariance of this proposal distribution in combination with the number of particles, we repeat the same analysis when proposing on the transformed parameter space as in Table 3.1. We use the 2-state model for these investigations which allows us to study various settings of the algorithm and their interactions in feasible computing time. The same simulated time series as in Chapter 2 was used for the observations $y_{1:T}$.

#### 3.3.1.1 Untransformed parameters

The first choice that needs to be made when running the PMMH is deciding which particle filter to use for estimating the likelihood for any given parameter vector. For this, we refer back to the results of the simulation study in Chapter 2 and use the filter that was shown to produce the likelihood estimates with the lowest variance. This means that we do not use any adjustment multiplier when resampling the particles or a function to forward-project the particles other than the model transition density. However, we do adapt the frequency with which the particles are resampled and only resample when the ESS of the particles falls below the threshold if $0.8N$. As we do not expect these features of the algorithm to affect the PMMH in different ways depending on the exact version of the model or the MH algorithm, they will be kept as just described for the remainder of this chapter.

Next, a proposal distribution for the MH kernel needs to be designed. We use a multivariate normal distribution where the covariance is proportional to the covariance of the posterior distribution. The decision to jointly propose all 6 parameters is driven by two factors. Firstly, evaluating the likelihood of a proposed value is expensive, as a particle filter needs to be run every time and proposing each parameter separately increases this

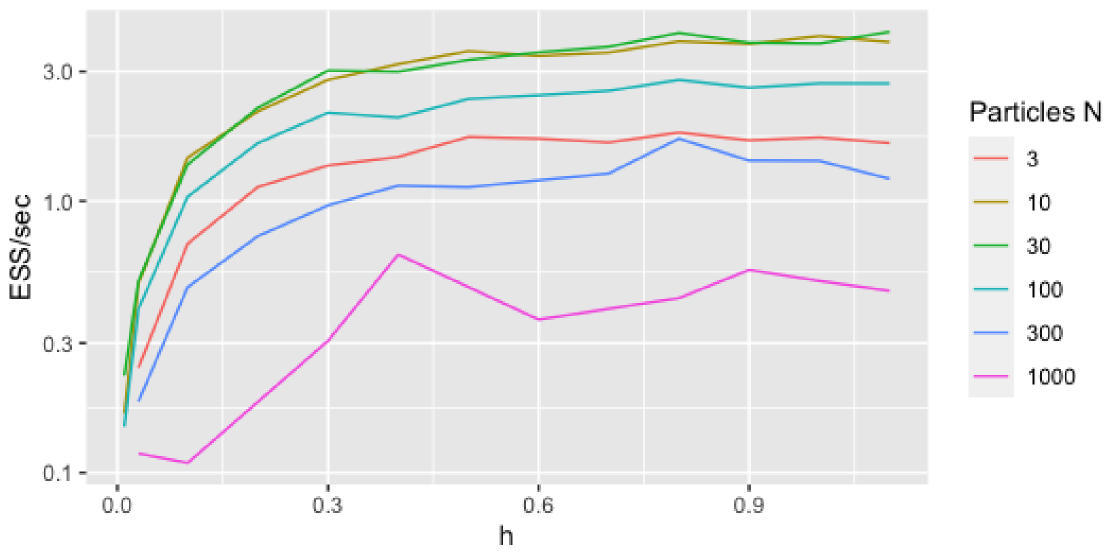


**Figure 3.4:** The effective sample size per second when running the PMMH for the 2-state model while varying the number of particles $N$ the scale $h$ of the covariance of the proposal distribution.

cost by a factor of 6. Secondly, some of the parameters are highly correlated. Using the covariance from the posterior distribution from a pilot run allows us to use this correlation when making joint proposals.

To identify optimal combinations of the scaling factor $h$ and the number of particles $N$ that lead to the highest effective sample size per runtime for the 2-state model, we now systematically study the effect of varying both these parameters for the 2-state model, and any interactions between the two. We vary the scaling factor $h$ between 0.01 and 1.1, and the number of particles between 3 and 1000. The number of iterations of the MCMC is inversely proportional to the number of particles so that the computation time is similar for each run of the algorithm. As the overhead of the MCMC takes a significant proportion of the overall runtime for low number of particles, the runtime for $N = 3$ and 10 particles was substantially higher.

Table C.1 and Figure 3.4 show the results when running the algorithm with these different settings. To assess whether the Markov chain had converged and determining an appropriate burn-in period, we used a combination of trace plots from two chains and an estimate of the multivariate potential scale reduction factor $\hat{R}$ (see Section 3.2.2.1). A high effective sample size per runtime was achieved with values of $h$ between 0.8 and 1.1 for all numbers of particles $N$ between 3 and 300, and the two highest values were generated with the combination of $N = 30$ and $h = 0.8$ respectively $h = 1.1$. Both $N = 10$ and 30 for the number of particles produce a high effective sample size. There seems to be no notable interaction between $h$ and $N$. Setting $h$ to 0.8 is the best choice for all numbers of particles other than $N = 1000$, although there the overall effective sample size is so low that the results need to be cautiously interpreted due to Monte Carlo error. Equally, when keeping $h$ constant, $N = 10$ and 30 is the best choice for all studied values of $h$.

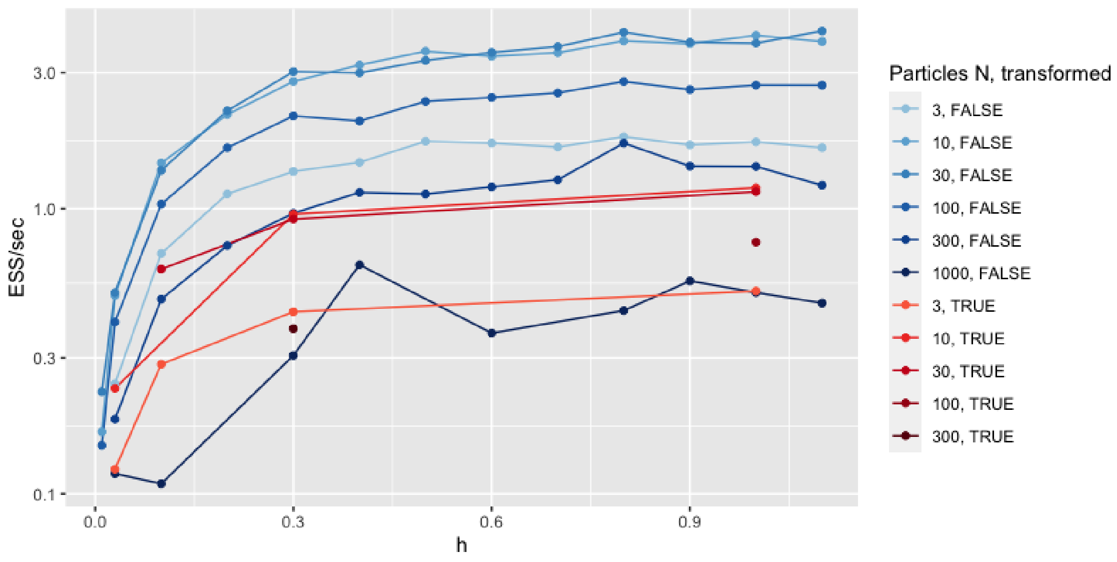


**Figure 3.5:** ESS/sec for the 2-state model using the PMMH algorithm with two proposal distributions. The number of particles $N$ and the covariance scale $h$ of the proposal distribution were varied. Blue lines represent results with a multivariate normal distribution on the original parameter scale, while red lines show results after transforming parameters to the real line. Missing values indicate non-convergence.

#### 3.3.1.2 Transformed parameter values

We now repeat the study above but propose new parameter values on the transformed scale. That means that we apply a bijective function $r : A \to \mathbb{R}, \theta \mapsto r(\theta)$, to the parameter as specified in Table 3.1. The proposal distribution is then multivariate normal on the transformed scale:, so

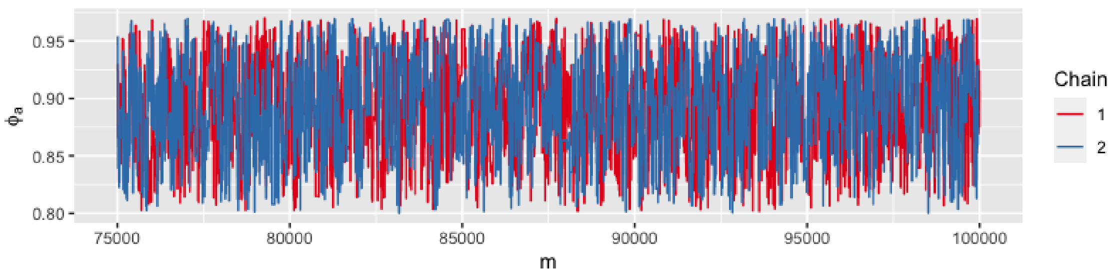
$$r(\theta^*) \sim \mathcal{N}(r(\theta), h\Sigma_{r(\theta)|y}),$$

where $\Sigma_{r(\theta)|y}$ is the covariance of the transformed posterior distribution, calculated from a pilot run. This has the advantage that the proposal distribution only generates parameter values from the support of their prior distribution. The disadvantage is that the proposal distribution is no longer symmetric on the untransformed scale.

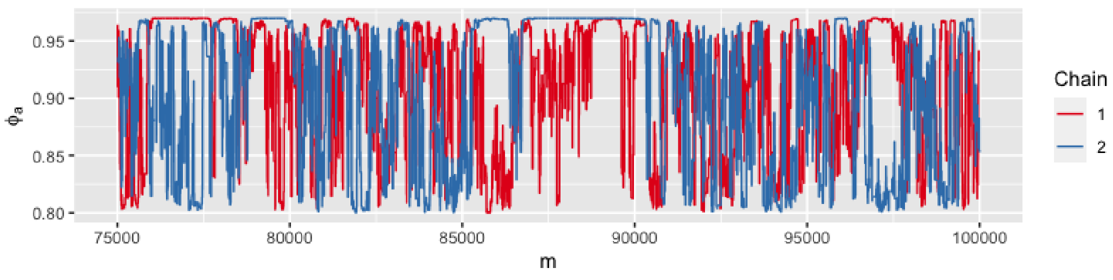
As before, we vary both the scaling factor $h$ and the number of particles $N$ while keeping the computational effort roughly constant by keeping $NM$ constant. Again, the ESS per runtime is used to measure the efficiency of the algorithm and trace plots and the potential scale reduction factor $\hat{R}$ are used to assess the convergence of the Markov chains for each setting.

The ESS per runtime was computed for the 2-state model using the PMMH algorithm with two different proposal distributions. The number of particles ($N$) and the covariance scale ($h$) of the proposal distribution were varied. Blue lines represent results with a multivariate normal distribution on the original parameter scale, while red lines show results after transforming parameters to the real line. Missing values indicate non-convergence.

The results in Table C.2 show that the chains failed to converge a lot more frequently than in the untransformed case. In Figure 3.5, we see that where the chains converged, the trends are similar when using the transformed proposal distribution compared to using



**(a)** Proposals on the untransformed parameter space.



**(b)** Proposals on the transformed parameter space.

**Figure 3.6:** Trace plots of the last 25% of iterations from running 2 chains with the PMMH for the parameter $\phi_a$, with $h = 1$ and $N = 30$.

the untransformed proposal distribution. As before, using $N = 10$ or 30 particles yields the highest ESS per runtime for all values of $h$, and setting the scaling factor $h = 1$ compared to the values of $h = 0.3$ or lower is the best choice for all values of $N$. Again, there appears to be no interaction between the two variables. Most notably though, using a proposal on the transformed parameter space shows a clear decrease in the quality of the sample when compared to proposals on the untransformed space. Comparing the highest values of ESS per second for $h = 1$, which was produced with $N = 10$ particles in both cases, the untransformed proposal leads to a value that is 3.4 times higher than with the transformed proposal. The trace plots in Figure 3.6 give some indication for the reason of the bad performance when proposing on the transformed scale. Because of the asymmetry of the proposal distribution, the chain often samples values close to the limits of the support of the prior distribution which leads to poor mixing.

In conclusion, this proposal distribution does not lead to better results than simply proposing values with a multivariate normal distribution, even though in that case, sometimes parameter values with prior density 0 are proposed. Rejection of these parameter values is quick as it does not require the evaluation of the likelihood. We acknowledge that many other possibilities for the proposal distribution exist. However, since the multivariate normal distribution generates satisfactory results with the right choices of $h$ and $N$, this proposal distribution will be kept for the more complex models.

#### 3.3.1.3 Conclusion

To maximise the algorithm's efficiency for the 2-state model, we propose new parameter values on the untransformed scale and set $h = 0.8$ and $N = 30$. Running the algorithm with these settings for around 15 minutes already produces a reasonable estimate of the posterior distribution, which can be seen in Figure 3.7a, even though a longer runtime would increase the Monte Carlo accuracy. We note the high correlation in Figure 3.7b between the two survival parameters $\phi_{p,\max}$ and $\phi_a$. This is investigated in more detail in Chapter 5. Moving on to the more complex models, we utilise the these findings and continue to use a multivariate normal proposal distribution on the untransformed scale of the parameters.

### 3.3.2 PMMH: Adapting the Number of Particles For Each Model

In this section, we attempted to generate samples from the posterior distributions of three increasingly complex models—the 7-state model, the complete seal model, and the complete seal model with independent estimate—by using the PMMH algorithm. As in the section above, we explored the relationship between the number of particles $N$ when estimating the likelihood and the number of iterations $M$ of the MCMC, while keeping the computation time roughly constant. We learned in Chapter 2 that for a fixed number of particles $N$ the variance of the likelihood estimate varies greatly between the different models. As this variance strongly affects the mixing of the Markov chain, the number of particles needs to be adapted for each individual model. We first did this for the 7-state model using simulated data, where we estimated all 6 model parameters simultaneously. Then, we used the complete seal model but excluded the independent estimate of the total size of the adult population, again using simulated data. Lastly, we ran the algorithm using the real data and the complete seal model including the independent estimate and any other modifications (see Section 1.2).

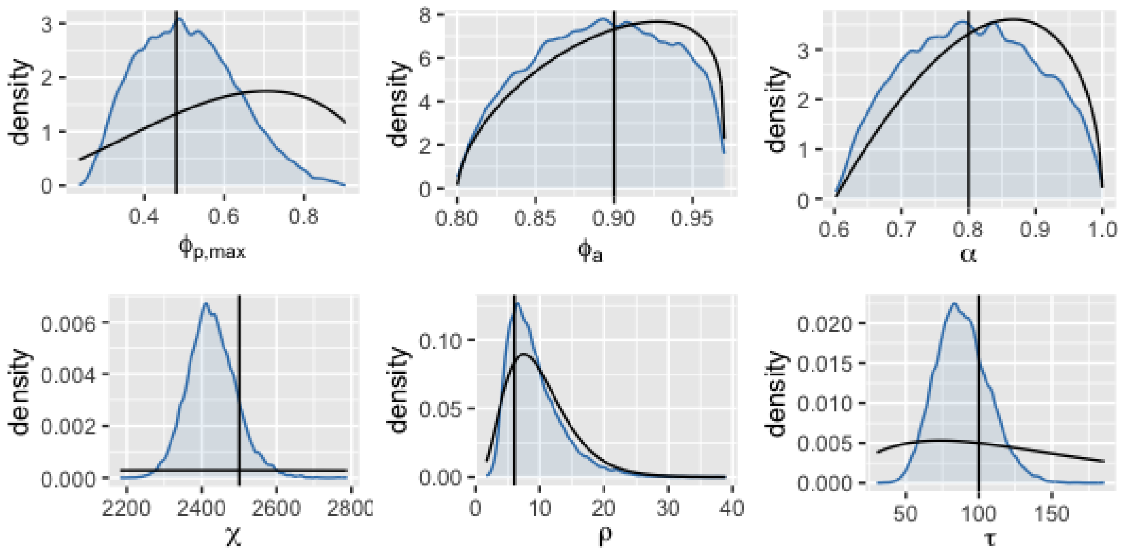
#### 3.3.2.1 7-State Model

To find the ideal algorithm settings for the PMMH algorithm when generating samples from the posterior distribution for the 7-state model, we used a similar approach as for the 2-state model. We varied both the number of particles $N$ and the scaling factor $h$ while adapting the number of iterations of the Markov chain to keep the computation time relatively constant. For the proposal distribution, a multivariate normal distribution was used with a covariance proportional to the covariance of the posterior distribution as calculated in a pilot run.

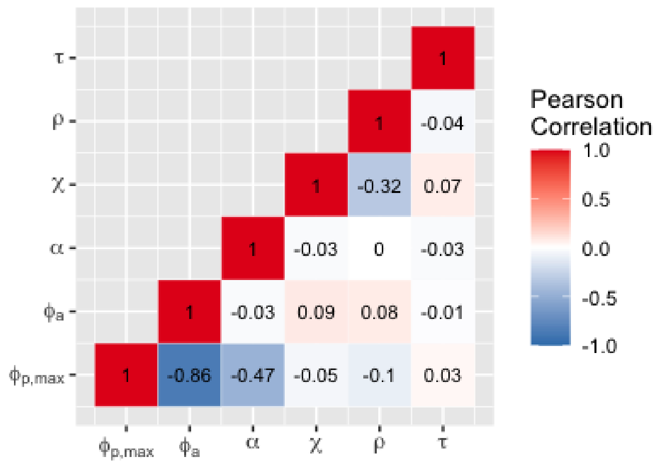
The results in Figure 3.8 and Table C.3 show similar trends to the results for the 2-state model. Again, the best choice in terms of achieving the highest ESS per runtime is to set $h = 0.8$ although the results are comparable for $h = 0.6$ and 1.0. For the number of particles, $N = 10$ resulted in the highest ESS per runtime for all examined values of $h$ but the samples produced with $N = 3$ and $N = 30$ were close in quality across all values of $h$. As before, no interaction between the two factors could be observed. Figure 3.9 shows the resulting posterior distribution as estimated with the best combination of $N$ and $h$. The correlation between $\phi_{p,\max}$ and $\phi_a$, while still high, is smaller than for the 2-state model.

#### 3.3.2.2 Complete Seal Model Without Independent Estimate

Having found the best settings for the 7-state model in one region, we now extend the model to all four regions and attempt to produce samples from the posterior distribution of the complete seal model without the independent estimate (see Section 1.2.4) using simulated data. Because of the significant increase in computation time for this model, it is no longer feasible to do a grid search for both the optimal $h$ and $N$. Instead, we rely on the results from the previous section and set $h = 0.8$. Firstly, this was the best value across all tested numbers of particles for both the 2-state and the 7-state model. Secondly, varying $h$ slightly only had a small effect on the ESS per runtime. For example, for the



**(a)** Plot of the marginal posterior densities for all 6 parameters. The black vertical line indicates the true parameter value with which the data was simulated. The black density line indicates the prior distribution.



**(b)** Posterior correlation coefficients of the 6 parameters in the 2-state model.

**Figure 3.7:** Results generated by running the PMMH for the 2-state model with 2 chains with $M = 100,000$ iterations each. The scaling factor $h$ was set to 0.8 and the number of particles was $N = 30$ for estimating the likelihood. The runtime was $945.6$s$\approx 15.76$min, resulting in an effective sample size of 3932.0.

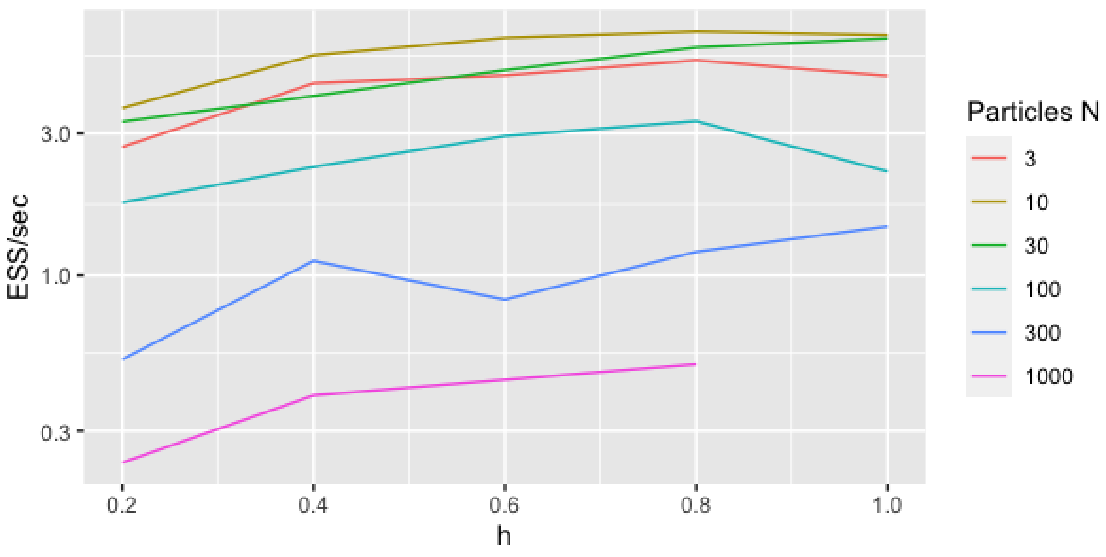


**Figure 3.8:** The effective sample size per second when running the PMMH for the 7-state model while varying the number of particles $N$ the scale $h$ of the covariance of the proposal distribution.

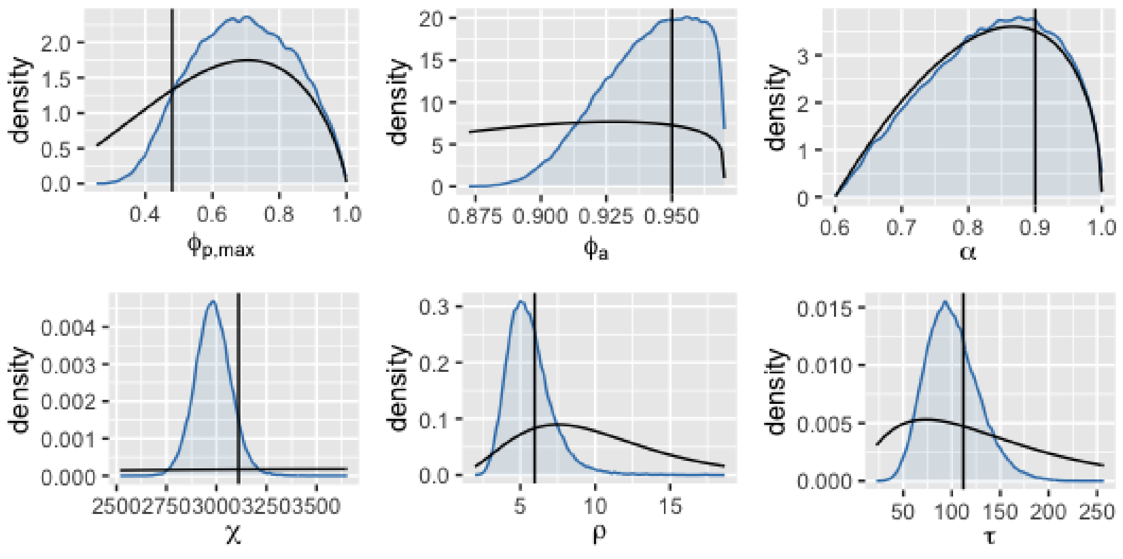
7-state model with $N = 10$ particles, the highest value of ESS/sec was 6.59 for $h = 0.8$ but was still 6.28 for $h = 0.6$ and 6.41 for $h = 1.0$. We therefore expect that setting $h = 0.8$ for the complete seal model without independent estimate is an adequate choice and provides the same insights as another potentially slightly better value might.

With constant $h = 0.8$, we vary the numbers of particles while keeping the runtime relatively constant, which generally means keeping $NM$ constant. However, since the computational overhead of the MCMC takes a significant proportion of the overall runtime for low number of particles, the number of iterations $M$ was reduced for $N = 3$, 10 and 30 particles.

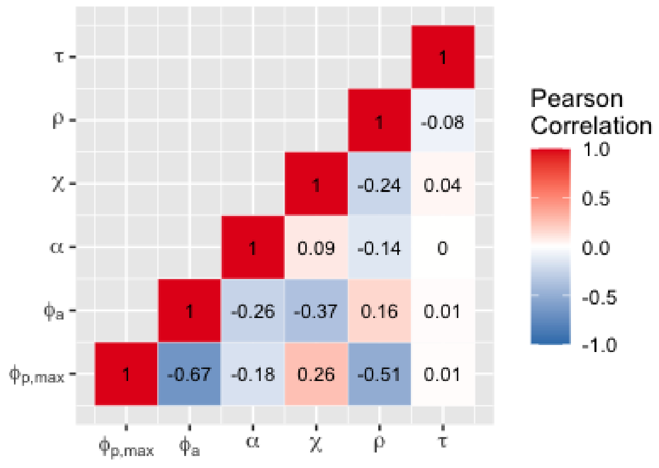
| $N$ | $M$ | $h$ | Runtime (sec) | $\hat{R}$ | Acceptance Rate | ESS | ESS/sec |
|---|---|---|---|---|---|---|---|
| 3 | 1000000 | 0.80 | 13233.33 | 1.33 | 0.0010 | - | - |
| 10 | 750000 | 0.80 | 12594.55 | 1.04 | 0.0066 | 396.42 | 0.031 |
| 30 | 500000 | 0.80 | 11891.46 | 1.01 | 0.0163 | 921.66 | 0.078 |
| 100 | 300000 | 0.80 | 13920.59 | 1.01 | 0.0410 | 1389.41 | 0.100 |
| 300 | 100000 | 0.80 | 10718.82 | 1.04 | 0.0711 | 718.83 | 0.067 |

**Table 3.3:** Results from running the PMMH for the complete seal model without independent estimate with relatively constant runtime, while varying the number of particles $N$. The proposal distribution is a multivariate normal distribution on the transformed parameter space with a covariance $h = 0.8$ times the posterior covariance. The ESS for $N = 3$ was not calculated due to its high $\hat{R}$ value.

Table 3.3 and Figure 3.10 show that setting the number of particles to $N = 100$ resulted in the highest ESS per runtime. For $N = 3$ the Markov chains did not converge. In comparison with the 7-state model which models only one region, extending the model to include all 4 regions led to a large decrease in ESS per runtime: for the 7-state model, the highest value is 6.59/sec whereas here the highest value is 0.1 ESS/sec, which corresponds to almost 66 times more computation time to achieve the same ESS. Figure 3.11a shows



**(a)** Density plots of the marginal posterior densities for all 6 parameters. The black vertical line indicates the true parameter value with which the data were simulated. The black density line indicates the prior distribution.



|  | $\phi_{p,max}$ | $\phi_a$ | $\alpha$ | $\chi$ | $\rho$ | $\tau$ |
|---|---|---|---|---|---|---|
| $\tau$ |  |  |  |  |  | 1 |
| $\rho$ |  |  |  |  | 1 | -0.08 |
| $\chi$ |  |  |  | 1 | -0.24 | 0.04 |
| $\alpha$ |  |  | 1 | 0.09 | -0.14 | 0 |
| $\phi_a$ |  | 1 | -0.26 | -0.37 | 0.16 | 0.01 |
| $\phi_{p,max}$ | 1 | -0.67 | -0.18 | 0.26 | -0.51 | 0.01 |

**(b)** Posterior correlation coefficients of the 6 parameters in the 7-state model.

**Figure 3.9:** Results generated by running the PMMH for the 7-state model with 2 chains with $M = 300,000$ iterations each. The scaling factor was set to $h = 0.8$ and the number of particles was $N = 10$. The runtime was $1982.53$s$\approx 33.0$min, resulting in an effective sample size of 13073.27.

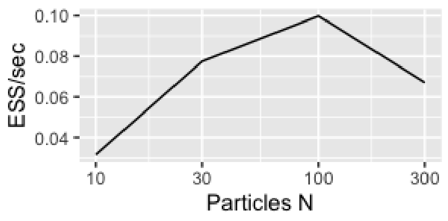


**Figure 3.10:** Relationship between the number of particles $N$ and ESS per runtime when sampling from the posterior distribution of the complete seal model without independent estimate. The proposal distribution is a multivariate normal distribution on the transformed parameter space with a covariance $h = 0.8$ times the posterior covariance.

that with an ESS of 1389.41 which was achieved after 3.87h runtime with $N = 100$, the density as estimated by the produced sample still exhibits some Monte Carlo variance. However, the plots also indicate that should more accurate estimates be required, more precise estimates could probably be produced with a ten times higher runtime. This is still feasible if only these exact settings are of interest but rules out any more complex analysis or simulations. For this model, Figure 3.11b shows a few high correlations which could affect the convergence speed of the Markov chain, in particular between $\phi_{p,\max}$ and $\phi_a$ and between $\rho$ and the two survival probability parameters.

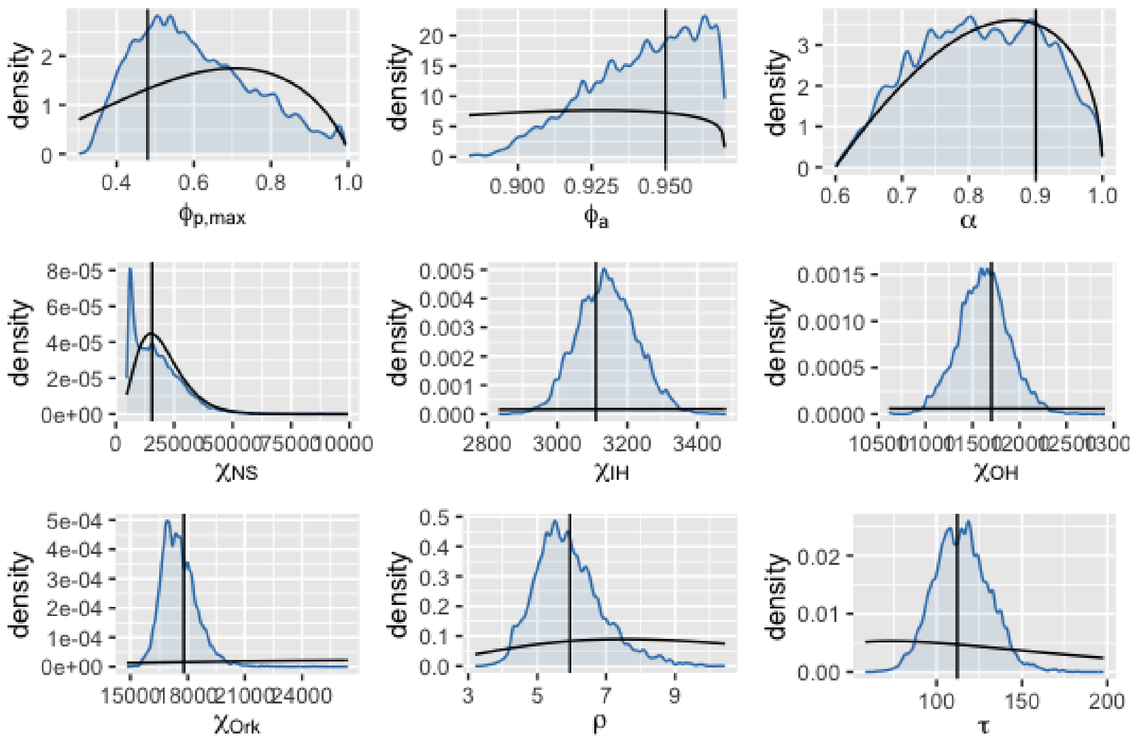
#### 3.3.2.3 Complete Seal Model With Independent Estimate

After the previous investigations with simulated data using simpler models, we attempt to generate samples from the complete seal model with independent estimate, using the real data (see Section 1.2.3). The proposal distribution is a multivariate normal distribution with a covariance proportional to an estimate of the covariance of the posterior distribution. This estimate was provided by Calliste Fagard-Jenkin, who worked with fitting the same model with a similar algorithm using the computing power of graphics processing units (Fagard-Jenkin, 2024). The scaling factor for the proposal covariance was $h = 0.8$. Five different values for the number of particles between $N = 3$ and $N = 30,000$ were used to estimate the likelihood within the PMMH. Again, we attempted to keep the runtime roughly constant for the different runs by adapting the number of iterations $M$.

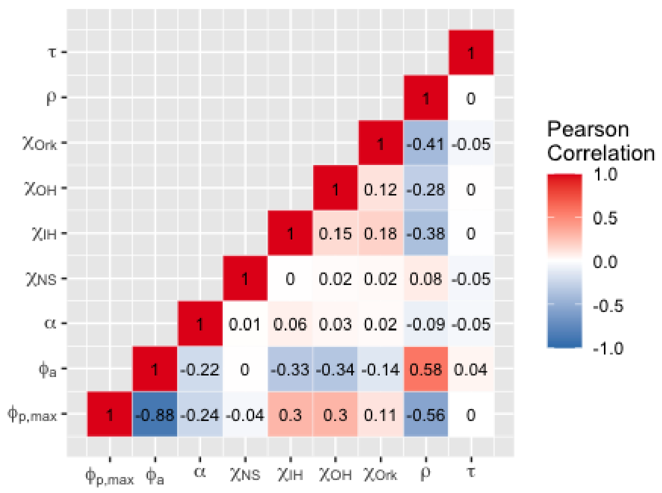
| $N$ | $M$ | $h$ | Runtime (sec) | $\hat{R}$ | Acceptance Rate | ESS | ESS/sec |
|---:|---:|---:|---:|---:|---:|---:|---:|
| 3 | 10000000 | 0.80 | 174641.06 | 1.14 | 0.0002 | 316.25 | 0.0018 |
| 30 | 4000000 | 0.80 | 107079.94 | 2.10 | 0.0002 | 95.48 | 0.0009 |
| 300 | 1000000 | 0.80 | 97852.61 | 1.54 | 0.0013 | 193.06 | 0.0020 |
| 3000 | 100000 | 0.80 | 92222.71 | 1.18 | 0.0090 | 279.15 | 0.0030 |
| 30000 | 10000 | 0.80 | 162685.43 | 28.23 | 0.0256 | - | - |

**Table 3.4:** Results from running the PMMH for the complete seal model with independent estimate and with the real data with relatively constant runtime, while varying the number of particles $N$. The proposal distribution is a multivariate normal distribution on the transformed parameter space with a covariance $h = 0.8$ times the posterior covariance. The ESS for $N = 30000$ was not calculated due to its high $\hat{R}$ value.

The results of the five runs can be seen in Table 3.4. According to the $\hat{R}$ value, none of the Markov chains seemed to converge in the given runtime, which varied between 25.6 and 48.5 hours for the different runs. Examining the trace plots confirmed that the mixing of



**(a)** Density plots of the marginal posterior densities for all 9 parameters. The black vertical line indicates the true parameter value with which the data were simulated. The black density line indicates the prior distribution.



**(b)** Posterior correlation coefficients of the 9 parameters.

**Figure 3.11:** Results generated by running the PMMH for the complete seal model without independent estimate with 2 chains with $M = 300,000$ iterations each. The scaling factor was set to $h = 0.8$ and the number of particles was $N = 100$. The runtime was $13920.59$s$\approx 3.87$h, resulting in an effective sample size of 1389.41.

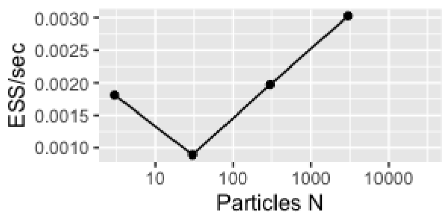
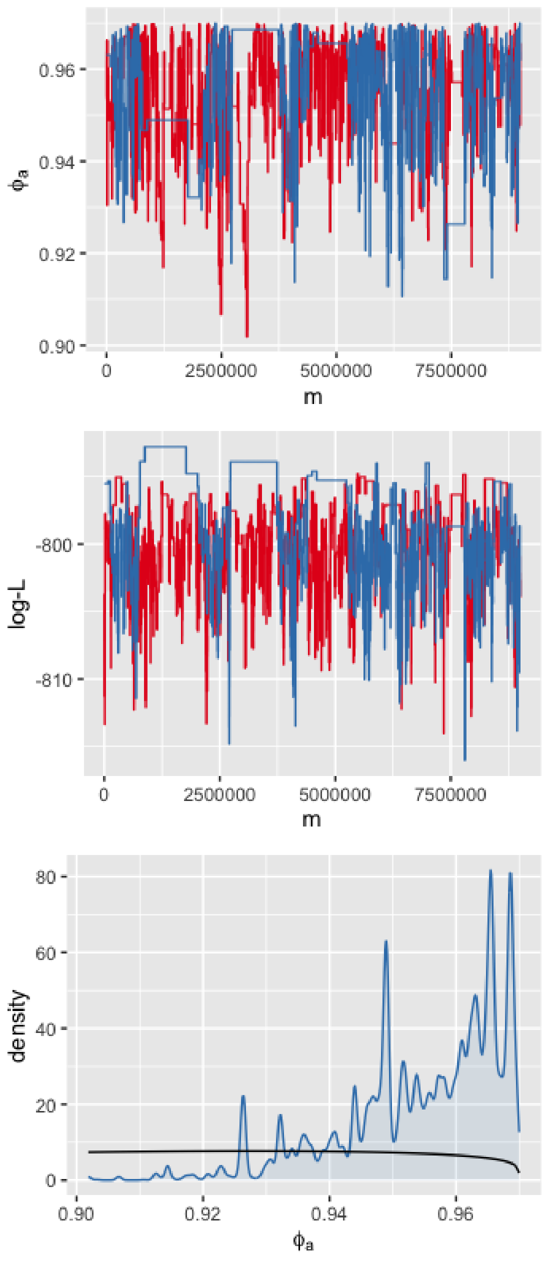


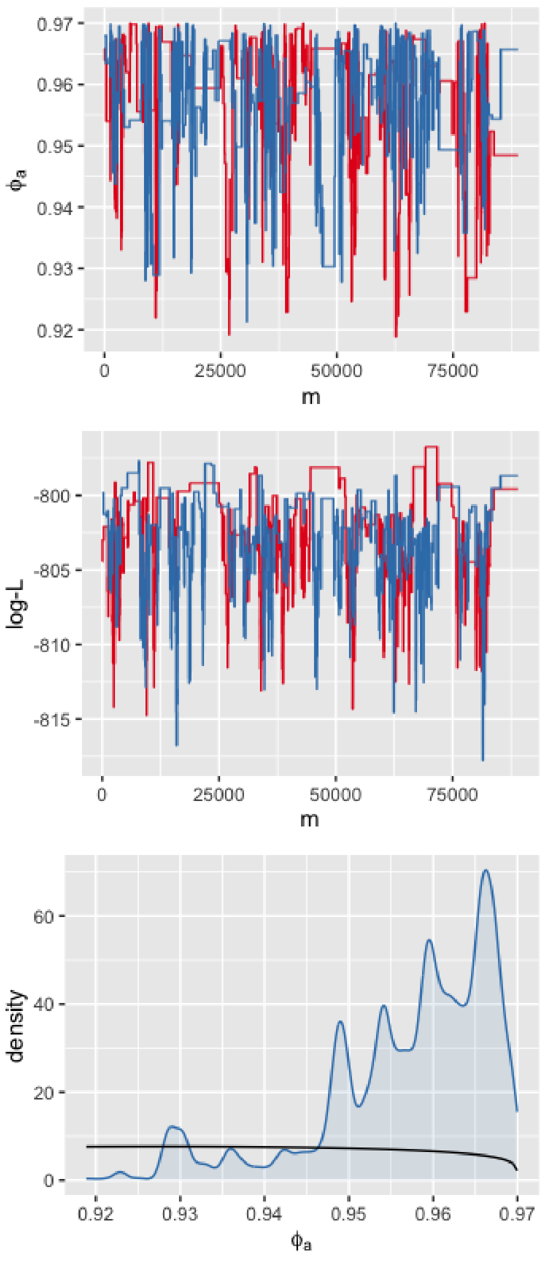
**Figure 3.12:** Relationship between the number of particles $N$ and ESS per runtime when sampling from the posterior distribution of the complete seal model with independent estimate using the real data. The proposal distribution is a multivariate normal distribution on the transformed parameter space with a covariance $h = 0.8$ times the posterior covariance.

the chains was not fully satisfactory for any of the runs. For $N = 30,000$, the chains did not converge at all and showed no tendency to do so even after $M = 10,000$ iterations. However, we still attempted to find a reasonable burn-in value for the other runs, to gain insights from analysing the resulting samples. Calculating the ESS per runtime, we see in Figure 3.12 that the highest value was achieved for $N = 3,000$. This also corresponds to one of the two lowest values of $\hat{R}$ which makes this the best choice for the value of $N$ even though the results need to be treated with great caution because of the poor mixing of the chains. Interestingly, and contrary to the results in the previous sections, using $N = 3$ led to similar results in terms of $\hat{R}$. The ESS per runtime was lower but still in the same order of magnitude.

To understand the reason behind the poor mixing of the Markov chains, we study the trace and density plots for the two cases that seemed the closest to generating samples from the posterior distribution, that is, for $N = 3$ and $N = 3000$. Figure 3.13 shows the trace plot and estimated density for the parameter $\phi_a$, as well as a plot of the log-likelihood estimates for both cases. Often, the chain remains with one parameter value for many iterations in a row, rather than accepting the newly proposed value. In the run with $N = 3$, the two chains have 19 periods were they do not accept the new parameter value for more than 100,000 iterations in a row, and one chain even exhibits a period of 1,009,270 iterations with the same parameter value. As a result, the number of unique parameter values in the two chains was only 3142. For the $N = 3000$ case, the same behaviour can be observed, though much less extreme. Here, the longest run with one parameter value is $6,247$ iterations and there are 34 runs of over 1000 iterations, resulting in 1451 unique parameter values in the two chains. Looking at the corresponding trace plots of the log-likelihood estimates, we see that the long runs with the same parameter value correspond to very high likelihood estimates. In Section 2.3.2.1, we found that the distribution of the likelihood estimate is often right-skewed. This means that very high estimates can occasionally occur which lead to a very low acceptance ratio in the MH acceptance ratio for the newly proposed value in the next iteration. These long runs of the same parameter value clearly impact the posterior distribution estimate. In the plot of the estimated posterior density, there are pronounced peaks which correspond to the parameter values that had extremely long runs. For example, in the $N = 3$ case the three parameter values with the longest runs are at the $\phi_a$ values of 0.969, 0.949, and 0.966, which correspond to the three most pronounced peaks in the density plot. The situation is similar for $N = 3000$ though less extreme. The density plots for the other parameters can be found in Figures C.1 and C.2 in the appendix and show a similar behaviour.



**(a)** $N = 3$



**(b)** $N = 3,000$

**Figure 3.13:** Trace plots and density plots from the sample generated from the PMMH with two different values of $N$, with two chains each. The first plot shows the trace plot of the parameter for adult survival, $\phi_a$, and the second shows the log-likelihood estimate at each iteration. The third plot shows a plot of the estimated posterior density.

#### 3.3.2.4 Conclusion

In summary, we found the settings in Table 3.5 to produce the highest ESS per runtime among the tested number of particles $N$ and scaling factors $h$. While increasing $M$ increases the ESS linearly after the burn-in period, it also leads to a linear increase in runtime. This choice therefore depends on the available computing power and desired effective sample size. We give the value of $M$ in the table that was used to produce the reported ESS.

| Model | Data | $N$ | $h$ | $M$ | Runtime (sec) | ESS | ESS/sec |
|---|---|---|---|---|---|---|---|
| 2-state | simulated | 30 | 0.8 | 100000 | 945.56 | 3931.97 | 4.16 |
| 7-state | simulated | 10 | 0.8 | 300000 | 1982.53 | 13073.27 | 6.59 |
| Complete without independent estimate | simulated | 100 | 0.8 | 300000 | 13920.59 | 1389.41 | 0.100 |
| Complete with independent estimate | real | 3000 | 0.8 | 100000 | 92222.71 | 279.15 | 0.0030 |

**Table 3.5:** Settings that led to the highest observed ESS per runtime for each of the four models when running a PMMH with a multivariate normal proposal distribution with covariance $h\Sigma_{\theta|y_{1:T}}$. $N$ stands for the number of particles to estimate the likelihood, $h$ is the scaling factor of the proposal covariance matrix and $M$ stands for the number of iterations of the MCMC.

### 3.3.3 PMMH: With Re-estimation

In Section 3.2.2.4, we studied on a toy example how re-estimating the likelihood at every iteration affects the target distribution of the Markov chain and saw that this is no longer the posterior distribution. However, it became clear in Section 3.3.2.3 that occasional high likelihood estimates severely disrupt the mixing of the Markov chains such that convergence requires an often prohibitively long runtime. It is therefore worth investigating what the effect of re-estimating the likelihood in the seal model is. This means firstly confirming whether re-estimating has the hypothesized effect of improving the mixing of the Markov chain, and secondly quantifying the size of the resulting bias. If re-estimation leads to a significantly reduced computation time while only introducing a small bias in the posterior distribution estimate, this technique could provide a useful way of quickly generating samples from an approximation to the true posterior distribution.

To asses this, we first study how much the estimated posterior distributions for the 2-state model differ depending on whether likelihood is re-estimated or not, and whether the ESS per runtime increases. We then study how the mixing improves for the more complex complete seal model with independent estimate when the likelihood is re-estimated.

#### 3.3.3.1 2-State Model

We use the algorithm settings from Section 3.3.1 and run the PMMH four times, twice with re-estimating the likelihood at every iteration, and twice without. With this, we are able to better determine whether a difference in distribution comes only from Monte Carlo error or whether it is systematic.

Table 3.6 shows the result from the four runs. In all four runs, the chains appeared to have converged according to the $\hat{R}$ value, which was confirmed by the trace plots. As expected, the runtime for the algorithm with re-estimation was higher, since at each iteration two

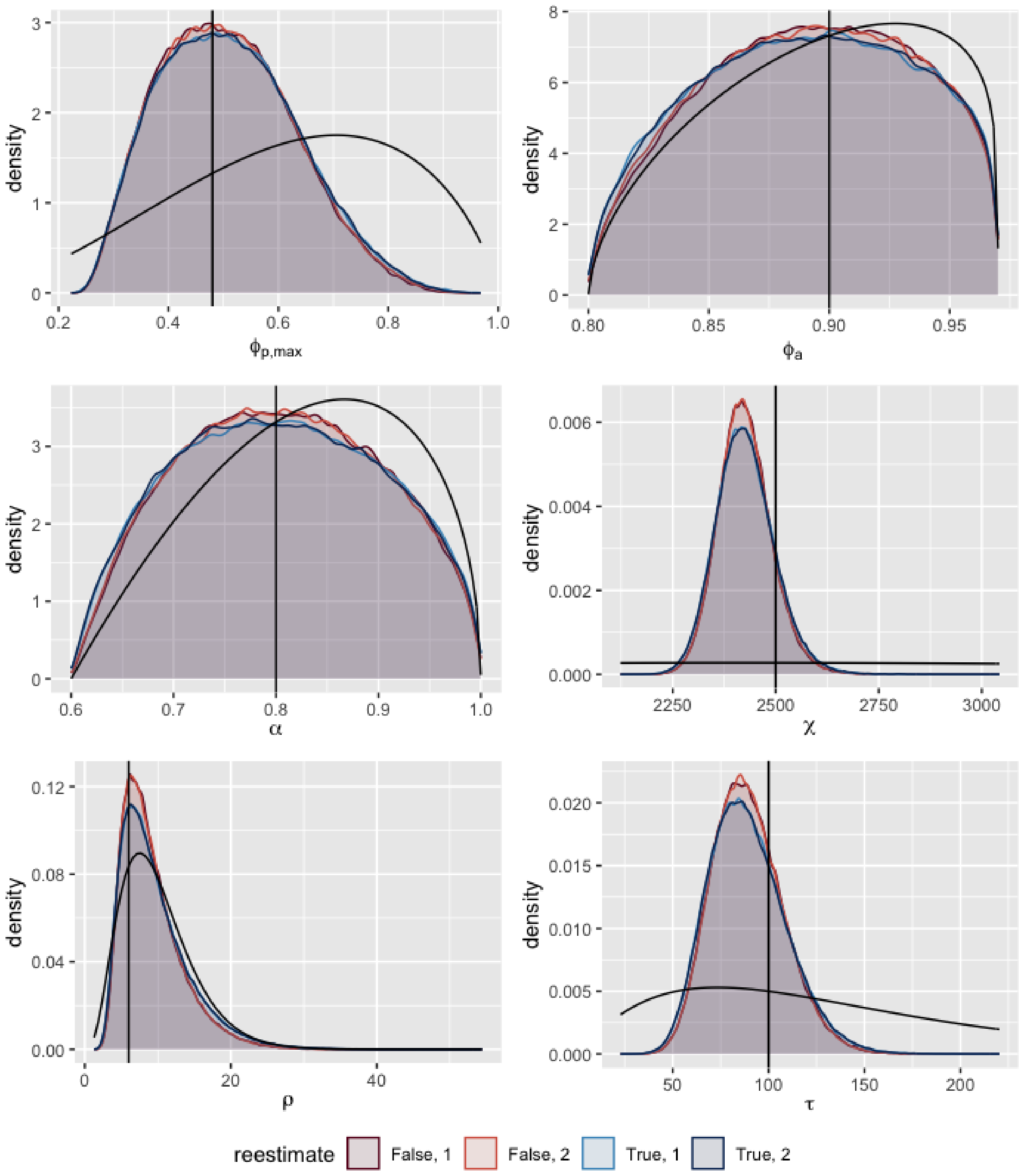


**Figure 3.14:** Plot of the four marginal posterior densities for all 6 parameters, as estimated by running a PMMH four times. The two red lines show the estimated posterior density when the likelihood is not re-estimated, the two blue lines show the same when the likelihood is re-estimated at every iteration. The black vertical line indicates the true parameter value with which the data were simulated. The black density line indicates the prior distribution.

| Re-estimation | Runtime (sec) | $\hat{R}$ | Acceptance Rate | ESS | ESS/sec |
|---|---|---|---|---|---|
| TRUE | 15197.32 | 1.00 | 0.19 | 39764.26 | 2.62 |
| TRUE | 15019.52 | 1.00 | 0.19 | 41004.55 | 2.73 |
| FALSE | 9509.79 | 1.00 | 0.16 | 39713.58 | 4.18 |
| FALSE | 9524.42 | 1.00 | 0.16 | 38400.20 | 4.03 |

**Table 3.6:** Results from running the PMMH for the 2-state model, comparing the effect of re-estimating the likelihood term in the MH acceptance rate. The algorithms settings were the ones that proved the best choices in Section 3.3.1, so $h = 0.8, N = 30, M = 1,000,000$. Each of the two choices was repeated to determine whether any differences were only due to Monte Carlo error.

| | $\phi_p$ | $\phi_a$ | $\alpha$ | $\chi$ | $\rho$ | $\tau$ |
|---|---|---|---|---|---|---|
| Re-estimated | 99.32 | 99.44 | 99.29 | 99.22 | 99.27 | 99.22 |
| Regular PMMH | 99.11 | 99.20 | 99.17 | 99.19 | 99.14 | 99.23 |
| Comparison | 98.22 | 98.00 | 97.96 | 95.52 | 94.92 | 95.60 |

**Table 3.7:** Percentages of overlapped area of the marginal densities for each parameter, measuring the shared area for both the two runs with re-estimation, the two runs without re-estimation, and finally the shared area when these two versions are compared.

likelihood values needed to be estimated rather than just the one of the newly proposed parameter value. We see that the ESS is indeed higher when the likelihood is re-estimated but, importantly, this advantage is lost when the runtime is taken into account. Indeed, when ESS per runtime is used as a criterion, the correct PMMH without re-estimation is about 1.5 times faster.

In Figure 3.14 the posterior densities as estimated in the four runs are shown. While the two distribution estimates where the likelihood has been re-estimated are close to the two unbiased distribution estimates, there is a clear difference. This difference is quantified in Table 3.7 with the overlapping index by Pastore and Calcagnì (2019) which estimates the shared area between two densities. For each parameter, the marginal densities when estimated with the same technique share at least 99.11% of their area. When the two techniques are compared, the shared area is only between 94.92% and 98.22% which confirms the differences observed in Figure 3.14.

#### 3.3.3.2 Complete Seal Model With Independent Estimate

We also study the effect of re-estimating the likelihood for the complete seal model with independent estimate, using the real data. Even though the previous section showed that re-estimation leads to a bias in the estimate of the posterior distribution, this bias was relatively small. If we find the same for the complete seal model, while seeing a significant reduction in computational cost, this approximation could be useful tool.

Since $N = 3$ and $N = 3,000$ led to the most promising results in Section 3.3.2.3, we assessed the effect of re-estimating for both of these settings. Table 3.8 shows as summary of the four runs. As with the 2-state model, the runtime increased when the likelihood was re-estimated. For the $N = 3,000$ case, the runtime almost doubled, which reflects the fact that twice as many likelihood estimates need to be computed when re-estimating. As expected, the issue with rejecting the newly proposed parameter value disappeared when re-estimating. The longest run without accepting a proposal was 258 iterations for the

$N = 3$ case (compared to 1,009,270 iterations when not re-estimating) and 168 for the $N = 3,000$ case (compared to 6247 before).

| $N$ | $M$ | Re-estimation | Runtime (sec) | $\hat{R}$ | Acc. Rate | ESS | ESS/sec |
|---|---|---|---|---|---|---|---|
| 3 | 10,000,000 | TRUE | 271,402.60 | 5.83 | 0.2504 | | |
| 3 | 10,000,000 | FALSE | 174,641.06 | 1.14 | 0.0002 | 316.25 | 0.0018 |
| 3,000 | 100,000 | TRUE | 184,087.35 | 1.01 | 0.1842 | 1664.07 | 0.0090 |
| 3,000 | 100,000 | FALSE | 92,222.71 | 1.18 | 0.0090 | 279.15 | 0.0030 |

**Table 3.8:** Results from running the PMMH for the complete seal model with independent estimate, comparing the effect of re-estimating the likelihood term in the MH acceptance rate. The algorithms settings were the ones that led to the two best results in Section 3.3.2.3, so $h = 0.8$, $N = 3$ and $N = 3,000$ with $M$ adapted to ensure comparable computation times.

The $\hat{R}$ value shows that the run with $N = 3$ when re-estimating did not converge which was also confirmed by inspecting the corresponding trace plots. We therefore now focus only on the $N = 3,000$ case. Here, there was a clear improvement in the $\hat{R}$ value as well as the trace plots and the two chains clearly converged. The ESS increased almost by a factor of 6 which, taking the runtime into account, led to an increase in ESS per unit time by a factor of 3.

| $\phi_p$ | $\phi_a$ | $\alpha$ | $\chi_{NS}$ | $\chi_{IH}$ | $\chi_{OH}$ | $\chi_{Ork}$ | $\rho$ | $\tau$ | $\omega$ |
|---|---|---|---|---|---|---|---|---|---|
| 66.99 | 73.01 | 83.76 | 72.13 | 68.60 | 72.16 | 67.38 | 72.94 | 45.09 | 72.49 |

**Table 3.9:** Percentages of overlapped area of the marginal densities for each parameter, measuring the shared area between the run when the likelihood was re-estimated at every iteration and the run when this was not done. The number of particles was $N = 3,000$.

Having confirmed that the first condition for a useful approximation is fulfilled—namely, the increase in ESS per runtime and the convergence of chains—we turn our attention to the second crucial aspect: the size of the bias. We assess whether the obtained sample can be considered a useful approximation by examining marginal density plots and calculating the overlapping percentages of these densities. To reduce the Monte Carlo error in this comparison as much as possible, we used a sample with an 11 times higher number of iterations, that is, $M = 1,100,000$ for the run with no re-estimation (see Section 3.4). The results of this comparison can be seen in Table 3.9 and Figure 3.15. While a higher ESS for the unbiased sample would be desirable to obtain a smoother estimate, it seems highly probable that not all of the differences between the two density estimates can be attributed to Monte Carlo error. For example, for the parameter $\tau$, the two algorithms estimate very different means (118.6 and 148.9). For three of the carrying capacity parameters $\chi$, the variance of the posterior is over-estimated in the approximation. For $\omega$, we expect no new information about the parameter from the available data which the unbiased density estimate confirms, whereas the approximation clearly deviates from the prior.

In conclusion, the bias that is introduced when re-estimating the likelihood in the PMMH is too great to consider this approach a viable approximation for estimating the posterior. While the bias was much smaller for the 2-state model, there was no improvement in ESS per unit time, and the unbiased version of the algorithm already provides very good results. Re-estimating the likelihood is therefore not a good solution for estimating the posterior distribution in the seal model.

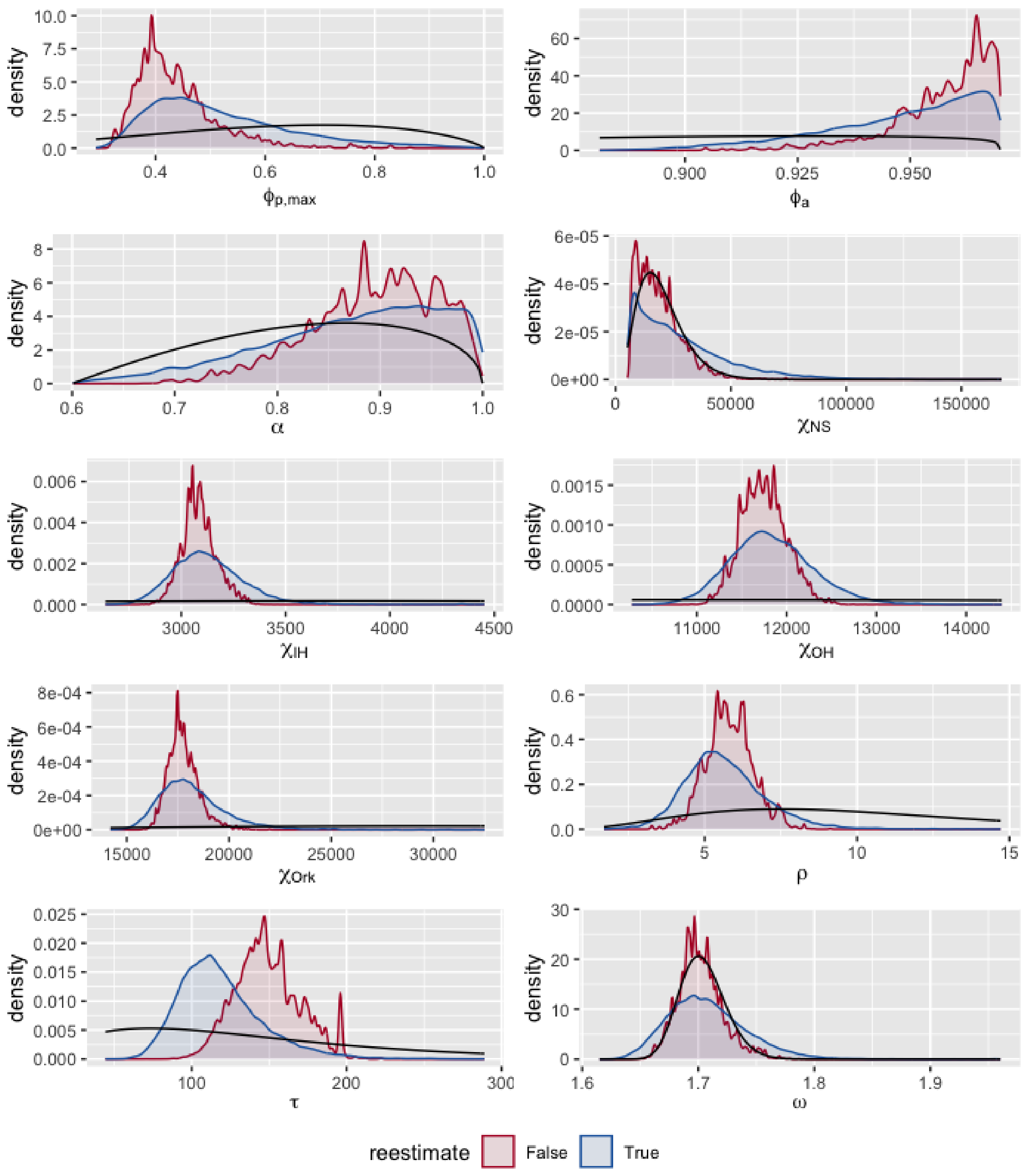


**Figure 3.15:** Plot of the two marginal posterior densities for all 10 parameters, as estimated by running a PMMH. The red lines shows the estimated posterior densities when the likelihood is not re-estimated, the blue lines show the same when the likelihood is re-estimated at every iteration. The black density line indicates the prior distribution.

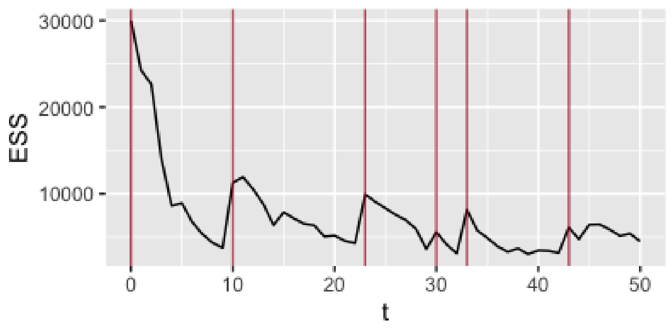


**Figure 3.16:** ESS of the parameter particles through the iterations $t$ of the SMC$^2$. The total number of parameter particles was $N_\theta = 30,000$ and a resampling step was triggered when the ESS fell below $3,000$. The red lines indicate the iterations where resampling ocurred.

### 3.3.4 SMC$^2$ for the Seal Model

Here, we apply the SMC$^2$ algorithm described in Section 3.2.3 to the 2-state model using simulated data, and then to the complete seal model with independent estimate using the real data.

#### 3.3.4.1 2-State Model

To estimate the posterior distribution for the 2-state model, we chose $N_x = 100$ and $N_\theta = 30,000$. This led to a runtime of 14.70 minutes which is comparable to the runtime of 15.76 minutes for generating the samples in Figure 3.7 with the PMMH. As the criterion for resampling the parameter values, we chose a threshold of an ESS of $0.1N_\theta$ for the parameter particles, where the ESS was calculated as in Equation 2.4. We had initially chosen an ESS-threshold of $0.5N_\theta$ following the example in Chopin and Papaspiliopoulos (2020) which led to a resampling step at almost every iteration and consequently high runtimes and no insights in how the decrease in ESS evolved through the time series. Therefore, we decreased the threshold to $0.1N_\theta$. In the resampling step, the proposal distribution was a multivariate normal distribution with a covariance matrix of $0.1\Sigma_{\theta|y_{1:T}}$, where $\Sigma_{\theta|y_{1:T}}$ is the covariance matrix of the posterior distribution calculated from a pilot run. The scale of the covariance matrix was an ad-hoc choice to move the parameter particles somewhat but simultaneously benefit from the diversity of the sample. In Figure 3.16, the ESS of the parameter particles through time can be seen. Even after the parameter particles are resampled, indicated by the red vertical lines, the ESS does not reach $N_\theta = 30,000$. This is because in the PMMH kernel, not all of the proposed parameter values are accepted and so, even though the weights of all parameter particles are 1, many of these parameter particles share the same parameter value.

Figure 3.17 shows the evolution of the estimated posterior distribution from $p(\theta)$ to $p(\theta|y_{1:T})$ in the SMC$^2$. The effect of adding more data on the posterior distribution can be seen most clearly for the carrying capacity parameter $\chi$. Looking at the simulated data in Figure 2.7, we see that the population grows almost exponentially until it approaches carrying capacity at around year $t = 30$. Only then is significant information about the carrying capacity contained in the data because the pup survival probability decreases (see Equation 1.4 and Figure 1.5). This is clearly visible in the plot of its interquartile range through time, where almost no change from the prior can be observed until $t = 30$, when a sudden decrease in interquartile range and change of the median happens.

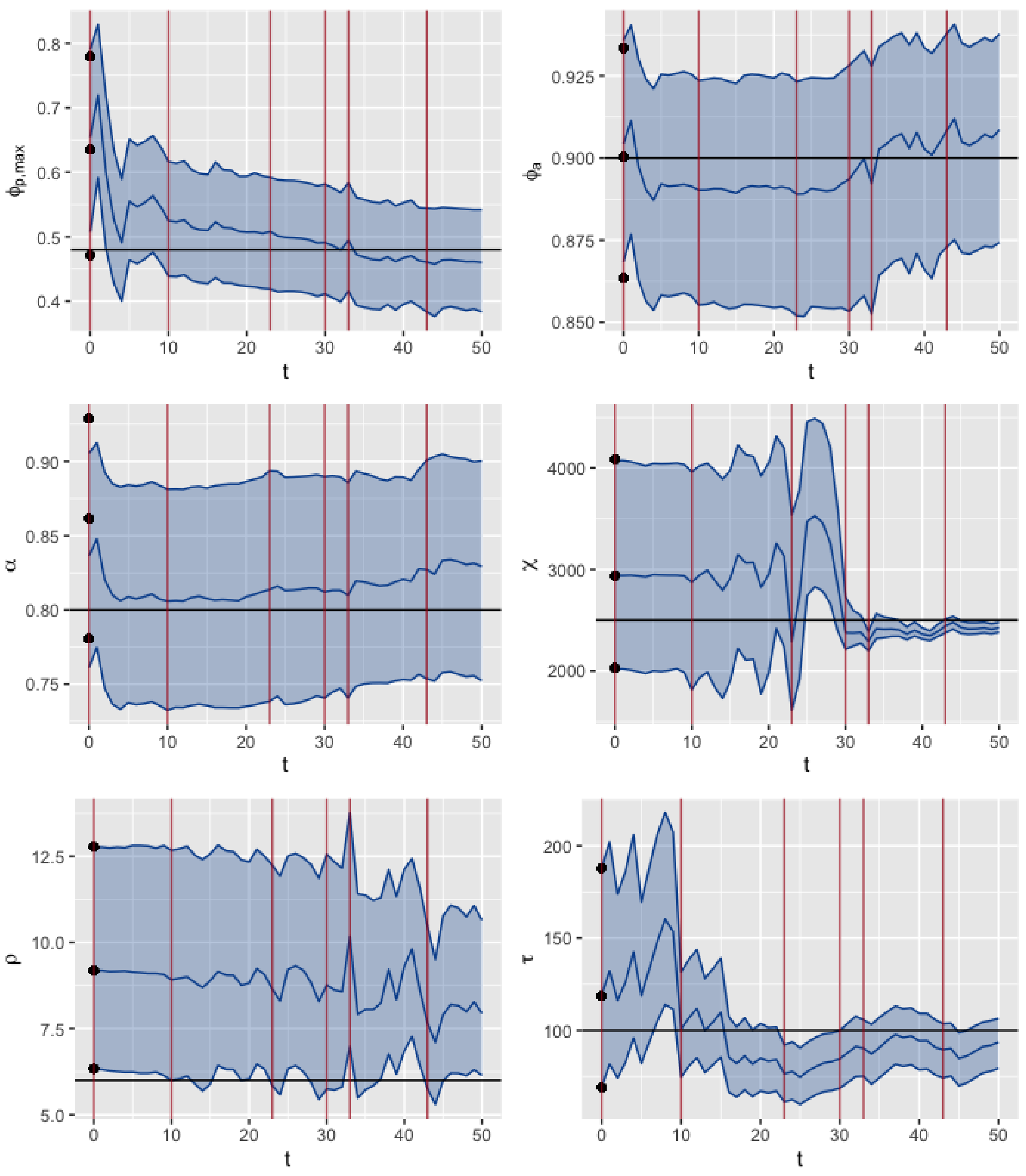


**Figure 3.17:** Inter-quartile range and median of the distributions $p(\theta|y_{1:t})$ through time $t$ for all six parameters of the 2-state model, as estimated by the SMC$^2$ algorithm. The black dots represent the inter-quartile range and median of the prior distribution. The vertical red lines indicate the iterations where a resampling step occurred. The horizontal line indicates the true parameter value that was used to simulate the data.

As discussed, assessing the Monte Carlo accuracy of the generated sample is difficult. We therefore estimated the posterior distribution 12 times with the above settings and compared the results. The runtime for each of the runs ranged from 11.2 minutes to 26.4 minutes. The large difference originates from the fact that the amount of resampling steps can differ between runs, and these resampling steps contribute most to the overall runtime. Figure 3.18 shows the estimated densities for each of the runs. Clearly, there are some differences between the estimates. The 12 densities have a pairwise overlapping percentage between 90.3% and 99.3%, with a mean of 95.6%.

Because there is no reliable measure for the effective sample size, we instead compared these numbers with the overlapping percentages for the posterior densities when estimated 12 times with the PMMH with the settings as in Figure 3.7. The resulting runtimes for the PMMH ranged from 16.2 to 17.2 minutes. The computational effort for the two algorithms is therefore comparable. The range and the mean of the pairwise computed overlapping percentages for each parameter can be seen in Table 3.10. The similarity between two estimates of the posterior density is much higher for the PMMH, such that the lowest overlapping percentage for the PMMH is higher than the highest overlapping percentage for the SMC$^2$ across all parameters. This indicates that the Monte Carlo error in the output of the SMC$^2$ is higher than for the output PMMH.

| | | $\phi_p$ | $\phi_a$ | $\alpha$ | $\chi$ | $\rho$ | $\tau$ |
|---|---|---|---|---|---|---|---|
| SMC$^2$ | lowest | 75.49 | 85.20 | 78.81 | 80.40 | 82.64 | 86.61 |
| | mean | 87.99 | 90.79 | 87.90 | 89.58 | 90.70 | 92.04 |
| | highest | 95.09 | 97.19 | 95.69 | 95.46 | 96.04 | 96.44 |
| PMMH | lowest | 97.90 | 97.86 | 97.66 | 98.04 | 97.96 | 97.83 |
| | mean | 98.41 | 98.50 | 98.31 | 98.44 | 98.41 | 98.48 |
| | highest | 98.85 | 99.00 | 98.85 | 98.89 | 98.82 | 98.95 |

**Table 3.10:** Range and mean of percentages of overlapped area of the marginal densities for each parameter from 12 different runs each for the SMC$^2$ and the PMMH algorithm.

#### 3.3.4.2 Complete Seal Model With Independent Estimate

To run the SMC$^2$ for the complete seal model with independent estimate, using the real data, we chose $N_x = 3,000$ particles which corresponds to the best number of particles determined for the PMMH in Section 3.3.2.3. As for the 2-state model, we chose an ESS of $0.1N_\theta$ as the resampling threshold for the parameter particles, and the proposal distribution was again a multivariate normal distribution with a covariance of 0.1 times the posterior covariance matrix $\Sigma_{\theta|y_{1:T}}$, calculated in a pilot run. An immediate difficulty with these setting was that the SMC$^2$ requires a lot of computer memory because $N_\theta$ particle filters with $N_x$ particles are run simultaneously. We therefore had to reduce the number of parameter particles to $N_\theta = 1,000$. While it might be possible to work around this issue by saving some of the particles to disk while they are not required, we instead chose to circumvent this problem by running the SMC$^2$ 20 times with the settings described above. The runtime for a single one of these runs was between 51.8 and 155.6 minutes, resulting in a total runtime of 40.82 hours.

In Figure 3.19 we show the evolving posterior distribution as $t$ increases and more observations $y_t$ are incorporated. Similarly to the equivalent figure for the 2-state model, the information gained by adding the later observations can be seen. We also note that for the time periods where the posterior is undergoing large changes, more resampling steps are

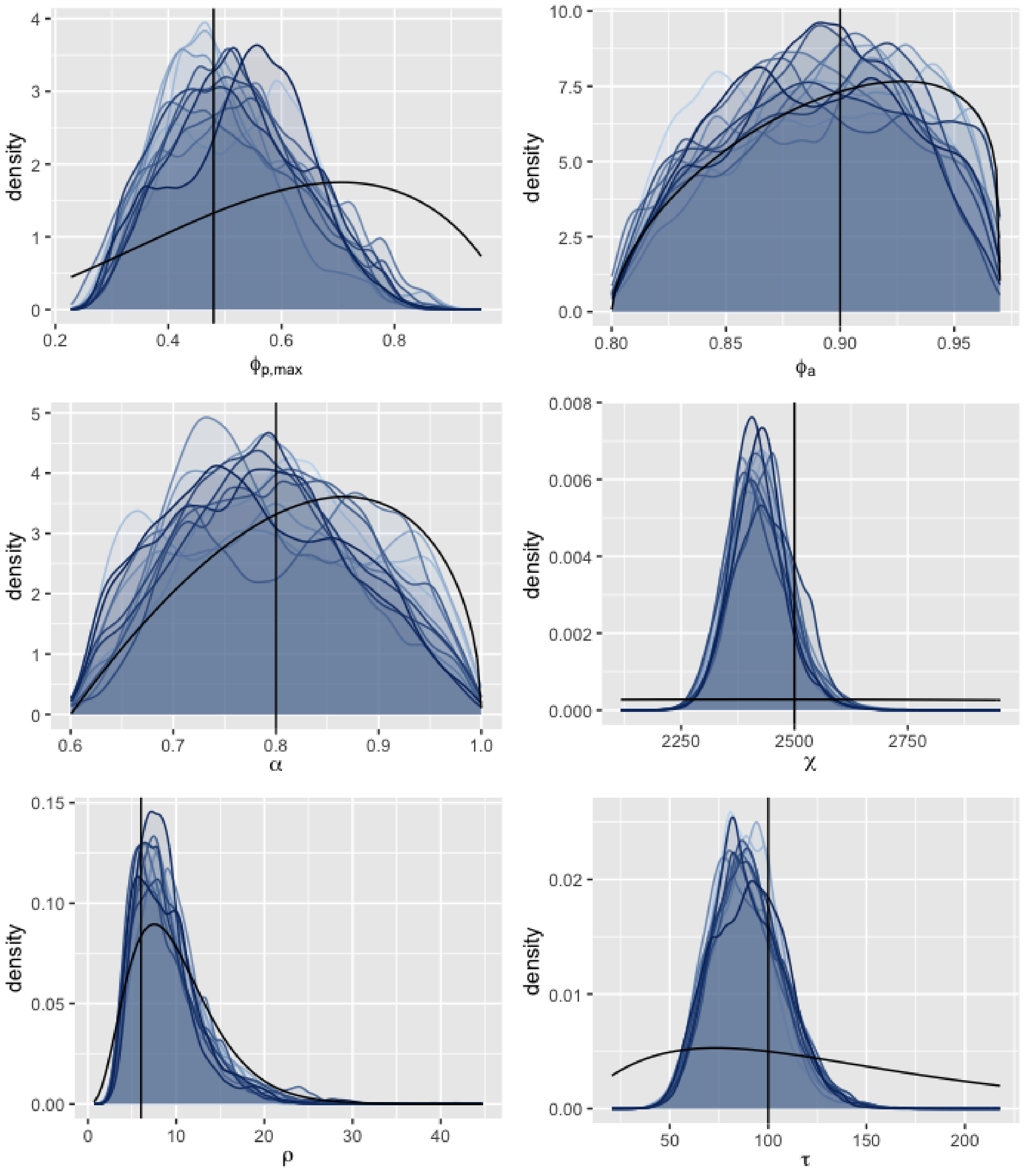


**Figure 3.18:** Estimated densities of all six parameters in the 2-state model from 12 runs of the SMC$^2$ algorithm in blue. The black vertical line shows the parameter values used to simulate the data. The black density line indicates the prior distribution.

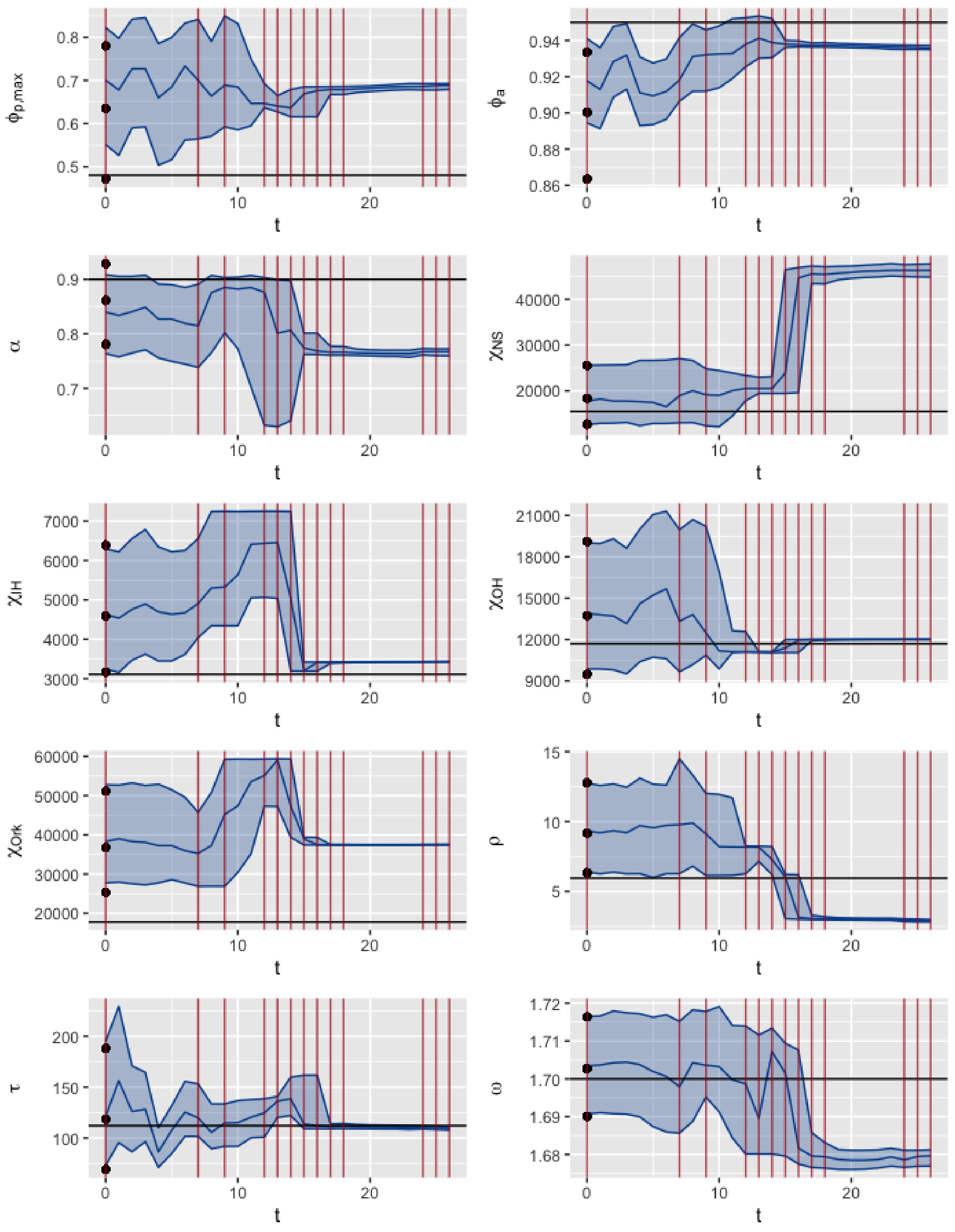
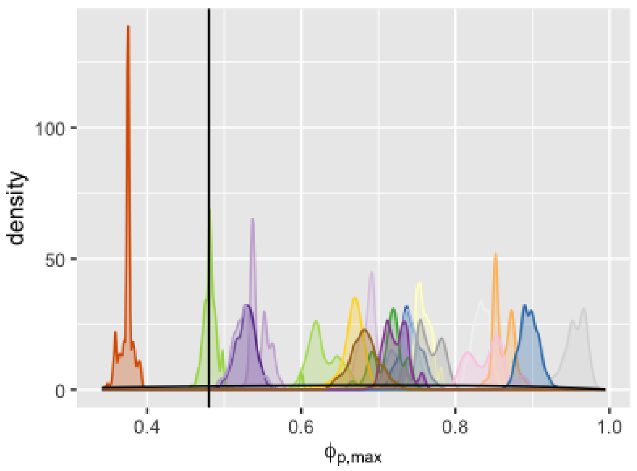
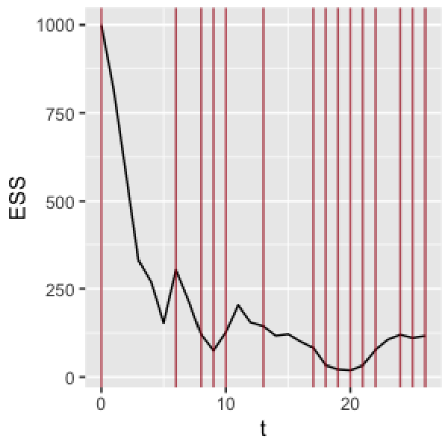


**Figure 3.19:** Inter-quartile range and median of the distributions $p(\theta|y_{1:t})$ through time $t$ from one of the 20 runs for all ten parameters of the 28-state model with independent estimate, using the real data, as estimated by the SMC$^2$ algorithm. The black dots represent the inter-quartile range and median of the prior distribution. The vertical red lines indicate the iterations where a resampling step occurred. The horizontal line indicates the posterior mean estimated in Thomas et al. (2019).





**(a)** Densities of the estimated marginal posterior distribution of $\phi_{p,\max}$ from each of the 20 runs of the SMC$^2$ algorithm. The black vertical line indicates the true parameter value used for the simulation, the black density line indicates the prior distribution.

**(b)** ESS of the parameter particles through the iterations $t$ of the SMC$^2$ for the first of the 20 runs. The red lines indicate the iterations where a resampling ocurred.

**Figure 3.20:** Results from running the SMC$^2$ 20 times for the complete seal model. For each of the runs the number of parameter particles was $N_\theta = 1,000$ and the number of state particles for each parameter was $N_x = 3,000$. The resampling step for the parameter particles was triggered when the ESS fell below 100.

occurring. We also note that the variance at time $T = 26$ is much smaller than we would expect. For example, we know from Section 3.4 that the variance of the marginal posterior distribution for $\chi_{NS}$ remains large even when all observations are included because the number of seals in the North Sea does not seem to have reached carrying capacity yet. Here however, the interquartile range is only 3210, whereas our analysis of the output by the PMMH showed an interquartile range of 11877.

As discussed above, quantifying the ESS for the SMC$^2$ is difficult. Instead we compare the output from the 20 separate runs. Figure 3.20a shows the marginal posterior densities for the parameter $\phi_{p,\max}$. The results for the other parameters are similar. It is apparent that the 20 runs have not converged to a common target distribution. Instead it seems that the variance of the likelihood estimate results in unreliable parameter estimates. This might be due to the occasional very low ESS of the parameter particles. For example, in the first run the lowest value for the ESS was 19.40 (see Figure 3.20b). It seems like the particle diversity cannot be regained after such a low value. Because the algorithm has not converged we cannot further use these results.

It is possible that better results could be achieved by exploring all possibilities for tuning this algorithm. For example, the proposal distribution could be adapted and the resampling threshold for the parameters could be changed. We can also study the effect of increasing the number of particles $N_x$ dynamically throughout the run of the algorithm, and study in detail the optimal balance between $N_x$ and $N_\theta$ (see, e.g., Chopin and Papaspiliopoulos, 2020). However, our results here show severe problems and do not point towards an obvious solution to address them. We also quickly run into issues with computer memory which prohibits simply scaling up the computational effort of the algorithm. In addition, any improvements are difficult to diagnose as no good measure of Monte Carlo accuracy is available. For these reasons, we did not explore this algorithm further for the seal model.

## 3.4 Analysis of Posterior Distribution

From the studies in the previous section, we found that the best choice for the complete seal model with independent estimate was using the PMMH with $N = 3,000$ particles. To produce a final estimate of the posterior distribution, we increased the number of iterations from $M = 100,000$ to 10 parallel process running 2 chains with $M = 1,000,000$ iterations each. The initial values of the Markov chains were sampled from posterior samples obtained in the previous section and hence no burn-in period was set. For inference, the samples from the 10 processes were combined. The runtime for the 10 processes was between 266.5 and 283.7 hours, resulting in an overall multivariate ESS of 4798.861. A summary of the 10 processes is given in Table 3.11.

| $N$ | $M$ | $h$ | Runtime (sec) | $\hat{R}$ | ESS | ESS/sec |
|---|---|---|---|---|---|---|
| 3,000 | 1,000,000 | 0.8 | 1,008,101 | 1.20 | 455.2 | 0.00045 |
| 3,000 | 1,000,000 | 0.8 | 1,009,357 | 1.05 | 540.2 | 0.00054 |
| 3,000 | 1,000,000 | 0.8 | 1,008,960 | 1.05 | 789.3 | 0.00078 |
| 3,000 | 1,000,000 | 0.8 | 986,841 | 1.11 | 523.8 | 0.00053 |
| 3,000 | 1,000,000 | 0.8 | 1,000,573 | 1.08 | 576.2 | 0.00058 |
| 3,000 | 1,000,000 | 0.8 | 1,006,974 | 1.06 | 721.0 | 0.00072 |
| 3,000 | 1,000,000 | 0.8 | 1,021,296 | 1.08 | 485.2 | 0.00048 |
| 3,000 | 1,000,000 | 0.8 | 986,580 | 1.05 | 844.8 | 0.00086 |
| 3,000 | 1,000,000 | 0.8 | 959,564 | 1.05 | 967.3 | 0.00101 |
| 3,000 | 1,000,000 | 0.8 | 1,001,565 | 1.05 | 571.7 | 0.00057 |

**Table 3.11:** Results from running the PMMH for the complete seal model with independent estimate and with the real data, using $N = 3,000$ particles and 20 chains with $M = 1,000,000$ iterations each. The algorithm settings are as determined in Section 3.3.2.3.

#### 3.4.0.1 Marginal distributions

To analyse the marginal posterior distributions of the 10 parameters, we examine the marginal densities in Figure 3.21 and the summary statistics in Table 3.12. We also compare the posterior means and standard deviations with the ones given in Thomas et al. (2019), where the modified version of the Liu-West algorithm described in Algorithm 4 was used.

We note that for most of the parameters, the prior distribution have a great influence on the marginal prior distributions. The overlap is larger than 28% for all parameters other than the three carrying capacities $\chi_{IH}$, $\chi_{OH}$ and $\chi_{Ork}$. For $\alpha$, $\chi_{NS}$ and $\omega$ the overlap is particularly high. This is unsurprising in the case of $\chi_{NS}$ as it is apparent from the observed counts that the population in this region is still growing almost unrestrictedly and therefore very little information about the carrying capacity can be gained from the data. For the sex ratio $\omega$, this is also expected. This parameter is necessary to include the independent estimate of all adult seal (including males) in the model. Only when the prior on $\omega$ is very wide is any learning on it expected but here with a narrow prior there is no further information about it in the data. For the fecundity $\alpha$, the change in the posterior distribution from the prior seems due only to the independent

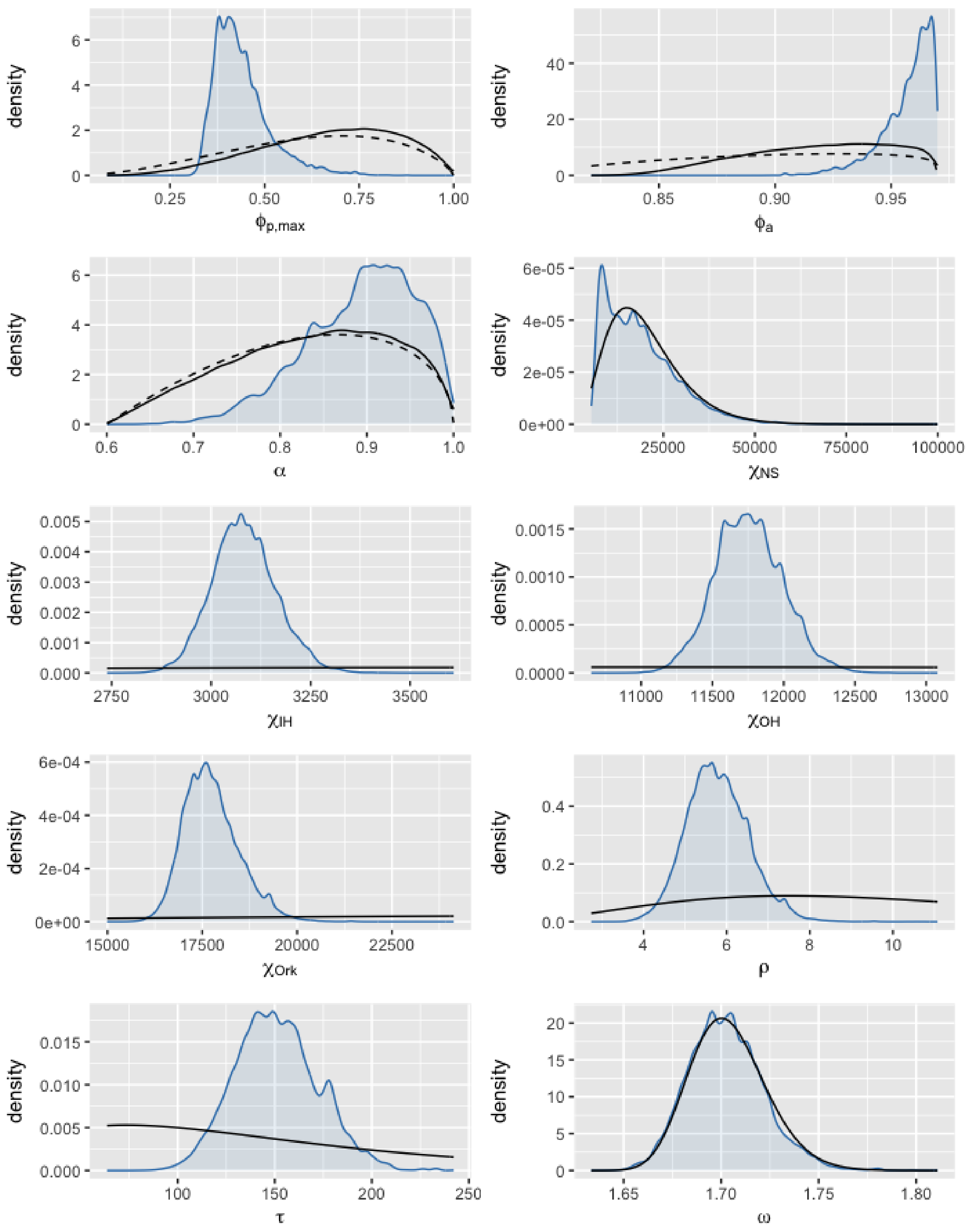


**Figure 3.21:** Density plots of the marginal posterior densities for all 10 parameters, generated by running the PMMH for the complete seal model with independent estimate with $N = 3,000$ particles and 20 chains with $M = 1,000,000$ iterations each. The scaling factor for the covariance of the proposal distribution was set to $h = 0.8$. The black density line indicates the prior distribution. For $\phi_{p,\max}, \phi_a$ and $\alpha$, the dashed line indicates the theoretical marginal prior and the solid line indicates the prior induced by the Beverton-Holt density dependence.

| Parameter | Prior<br>Mean (SD) | Posterior<br>Mean (SD) | Prev. posterior<br>Mean (SD) | Prior-post.<br>overlap | Previous<br>overlap |
|---|---|---|---|---|---|
| $\phi_{p,max}$ | 0.671 (0.179) | 0.439 (0.073) | 0.48 (0.09) | 29.9% | 40% |
| $\phi_a$ | 0.919 (0.031) | 0.957 (0.011) | 0.95 (0.01) | 36.6% | 35% |
| $\alpha$ | 0.835 (0.092) | 0.893 (0.063) | 0.90 (0.06) | 70.7% | 70% |
| $\chi_{NS}$ | 20,000 (10000) | 18,800 (9870) | 15,500 (8210) | 87.7% | 77% |
| $\chi_{IH}$ | 5,000 (2500) | 3,080 (80.5) | 3,110 (173) | 8.2% | 11% |
| $\chi_{OH}$ | 15,000 (7500) | 11,800 (238) | 11,700 (535) | 8.5% | 14% |
| $\chi_{Ork}$ | 40,000 (20,000) | 17,700 (734) | 17,800 (1680) | 7.0% | 9% |
| $\rho$ | 10 (5) | 5.78 (0.746) | 5.95 (1.73) | 28.2% | 50% |
| $\tau$ | 140 (96.61) | 151 (21.4) | 112 (34.60) | 34.6% | 49% |
| $\omega$ | 1.70 (0.02) | 1.70 (0.019) | 1.70 (0.02) | 95.1% | 99% |

**Table 3.12:** Mean (with standard deviation in parentheses) for the marginal prior and posterior distributions of the 10 parameters in the complete seal model. For the parameters $\phi_{p,\max}, \phi_a$ and $\alpha$, the induced prior was used to calculate the prior summary statistics and overlap percentages. The previous posterior statistics are the ones given in Thomas et al. (2019) using Algorithm 4.

estimate. Comparing the marginal density with the one in Figure 3.11a, we see that when no independent estimate is included in the model, there is almost no change from the prior. This points to identifiability issues with this parameter when only the pup production estimates are used.

Comparing our results with the ones in Thomas et al. (2019), we note that many of the results are similar, but some differences should be highlighted. The posterior means are relatively similar except for $\phi_{p,max}$, $\chi_{NS}$ and $\tau$. For the maximum pup survival probability $\phi_{p,max}$, we estimated a lower value of 0.439 whereas in Thomas et al. (2019), the mean value was 0.48. For the carrying capacity in the North Sea region, Thomas et al. (2019) estimated a posterior mean of 15500; we calculated a mean estimate of 18800. While these differences are large, they are still within one standard deviation of the posteriors. This is not the case for the observation precision $\tau$, where we estimated a much larger mean of 151 with an SD of 21.4 while Thomas et al. (2019) obtained a mean of 112 (with an SD of 34.60).

When comparing the standard deviations and the prior-posterior overlap, we notice more differences. For the three carrying capacities $\chi_{IH}$, $\chi_{OH}$ and $\chi_{Ork}$, our estimated standard deviations are less than half those estimated by Thomas et al. (2019). This also leads to a smaller prior-posterior overlap. A similar phenomenon can be observed with the shape parameter $\rho$, where our estimated standard deviation is less than half and the prior-posterior overlap decreases from 50% to 28.2% compared to the results in Thomas et al. (2019). For the observation precision, not only is the mean different as discussed above but the standard deviation is also smaller when the posterior is estimated with the PMMH, decreasing from 34.6 to 21.4. Again, this leads to a decrease in prior-posterior overlap from 49% to 34.6%.

#### 3.4.0.2 Joint distributions

We also examine the joint distributions. Figure 3.22 shows the correlation coefficients, complemented by scatter plots in Figure C.4 in the appendix. The largest posterior correla-

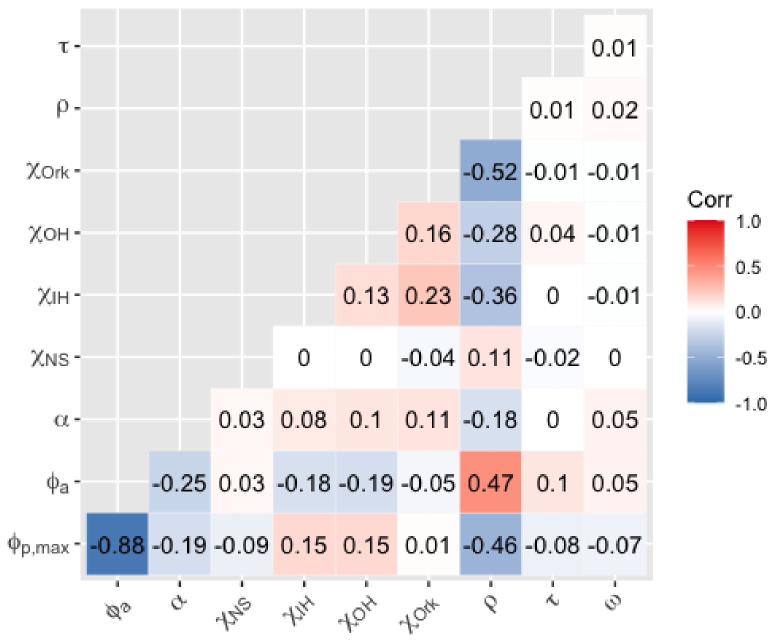


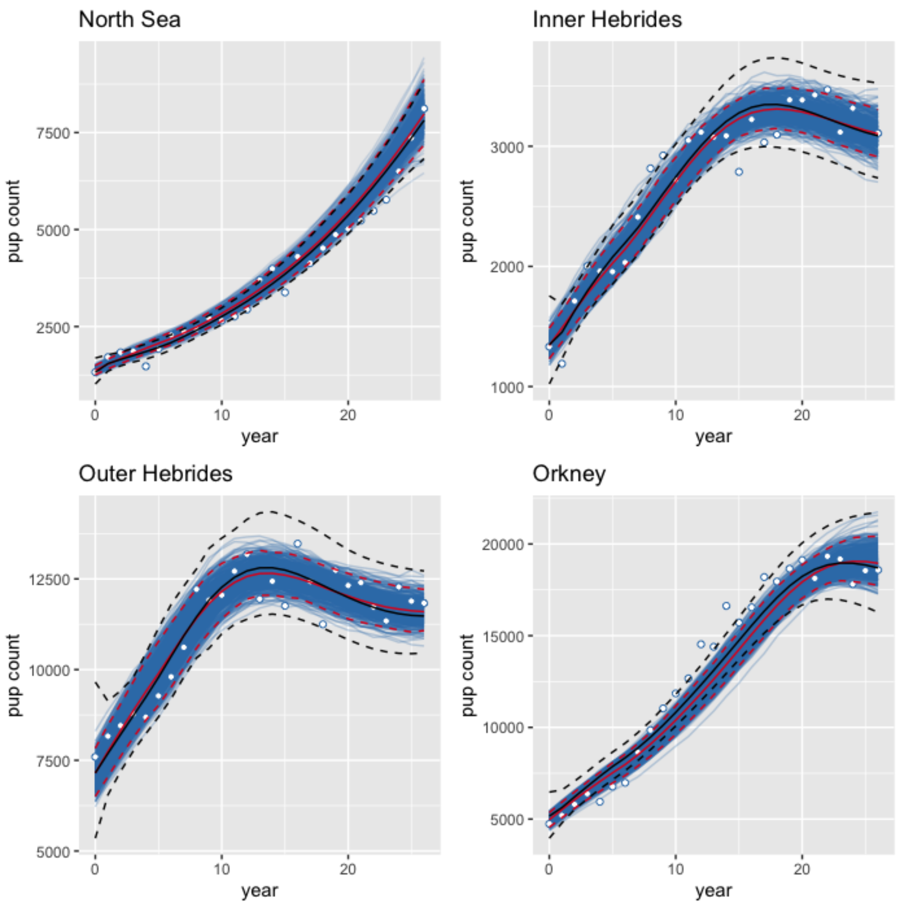
**Figure 3.22:** Posterior correlation coefficients of the 10 parameters of the complete seal model, generated from running the PMMH with $N = 3,000$ particles and 20 chains with $M = 1,000,000$ iterations each.

tion exists between the two survival probabilities $\phi_{p,max}$ and $\phi_a$ with a negative correlation of -0.88. We also note the relatively high correlation that the density dependence shape parameter $\rho$ has with several of the other parameters, e.g., the three carrying capacities $\chi_{IH}$, $\chi_{OH}$ and $\chi_{Ork}$, and the two survival probabilities $\phi_{p,max}$ and $\phi_a$. The shape parameter $\rho$ governs the relationship between the carrying capacity and demographic parameters and the annual pup survival probability $\phi_{p,t}$. Comparing these correlation coefficients with the ones in Thomas et al. (2019), we see the same large correlation is estimated between $\phi_{p,max}$ and $\phi_a$. For $\rho$, not all correlations are as strong as they were observed here, for example with $\chi_{OH}$ (0.0408 versus -0.28 in our results) and with $\phi_a$ (0.198 versus 0.47 in our results). This could be related to the differently estimated standard deviations of $\rho$. Otherwise, the estimates correlation coefficients are relatively similar.

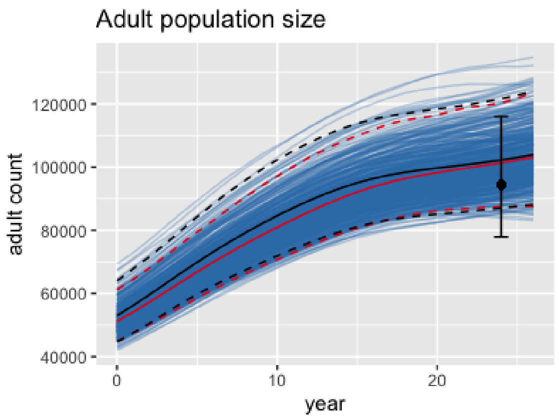
### 3.4.0.3 State estimates

To produce samples from the smoothed posterior state distribution $p(x_{0:26}|y_{1:26})$, we sampled $\theta_i$ for $i = 1, ..., 500$ from the posterior distribution and then ran bootstrap filters with `nimble` with 1000 particles each to generate smoothed state estimates from $p(x_{0:26}|y_{1:26}, \theta_i)$. We then randomly selected one of the 1000 state trajectories from each bootstrap filter.

Figure 3.23 show the resulting smoothed posterior state estimates and 95% credible intervals (red solid and dashed lines). The population in the North Sea region is estimated to be growing nearly exponentially which corresponds to the posterior distribution of $\chi_{NS}$.



**(a)** Estimate of pup counts in all 4 regions. The white circles show the observed pup counts.



**(b)** Estimate of total adult count across all 4 regions. The black point and vertical errorbar show the independent estimate of the total adult population size with 95% credible interval of the estimate.

**Figure 3.23:** Smoothed posterior state estimates of pup counts in all 4 regions and of total adult population size, produced by the PMMH. The red dashed lines show the 95% credible interval and the red solid line shows the mean. The black dashed and solid lines show the same with the estimates from Thomas et al. (2019).

There is little information about this parameter in the data except that it must be bigger than the most recent count. Given that according to the prior, $\chi_{NS}$ is larger than 8267.6 (the posterior mean of the most recent pup numbers in the North Sea region) with probability 0.914, the posterior is almost unchanged from the prior. In the other regions, a carrying capacity seems to have been reached. As discussed in Thomas et al. (2019), the model cannot capture some of the short-term trends in pup numbers. As an example, the smoothed posterior mean estimate lies below all observed pup count in Orkney from year 8 to year 20, and similarly for the North Sea from year 17 to 25.

Comparing the results obtained with the PMMH (in red) with the ones obtained with the Liu-West algorithm in Thomas et al. (2019) (in black), we note that the mean estimates are similar for both regional pup numbers as well as total adult numbers across time. However, the width of the credible intervals differ noticeably for the pup estimates in all but the North Sea region. We suspect a number of reasons for these differences. First, the standard deviations of the marginal posterior distributions for the three carrying capacity parameters for the Inner Hebrides, Outer Hebrides and Orkney, as well as the density dependence shape parameter $\rho$ are about twice as large in Thomas et al. (2019) as they are in the results obtained with the PMMH (see Table 3.12). These parameters directly affect the pup survival probability (see Equations 1.4 and 1.5) and might cause the larger credible intervals. In a preliminary investigation, we also found that decreasing the tuning parameter $\lambda$ from 0.99997 to 0.7, thereby increasing the bias, increased the differences between the credible intervals even further (pers. comm. by Len Thomas).

## 3.5 Conclusion

This chapter has shown that fitting the complete seal model is a challenging task. Having studied the PMMH in detail and explored the available options to choose the best settings, we found the choices that allow the Markov chains to converge and produce samples from the posterior distribution. Using a relatively large covariance for the random walk proposals ($h = 0.8$) in the PMMH kernel was shown to lead to the highest ESS per runtime. We also found that in many cases, a small number of particles can be chosen which then allows us to run a larger number of iterations of the Markov chain in the same runtime as a smaller number of iterations with a larger number of particles. For the 2-state and 7-state models, both $N = 10$ and $N = 30$ were good choices. For the complete seal model without the independent estimate and with simulated data, the optimal number increased to $N = 100$. Only when using the complete seal model with independent estimated using the real data, did the best number of particles increase to $N = 3000$.

However, to obtain a sufficiently high ESS, the algorithm needs to run for a very long time (11 days in 10 parallel processes for a multivariate ESS of 4799). We identified the main reason for this, which is that the Markov chain rejects new proposals for many iterations in a row, when it has obtained a very high likelihood estimate of the current value of the chain. While the approximation where the likelihood is re-estimated at every iteration reduced this behaviour and increased the ESS per runtime by a factor of 3, we found that the bias of the approximation was too large to be considered useful.

Whether this long runtime is practicable depends on the specific context of the application. For example, if there is access to a multi-core machine and the algorithm only needs to be run once, running it is still feasible. This might be the case for the annual or biennial update of the model inference when new data become available. However, the runtime prohibits other uses. Systematically studying any changes to the algorithm, model or data is nearly impossible because observing the effect of a single change requires so much computing power. The long runtime also means that increasing the complexity of the model is not possible when using this algorithm. It might for example be desirable to add a random effect on one of the demographic parameters or to include new model parameters that allow movement between the regions.

Further directions of improvement for this algorithm are implementing the correlated particle filter (Deligiannidis and Doucet, 2018) to reduce the relative variance in the likelihood ratio. Another direction is to study more options for the proposal distribution, although as mentioned above, any explorations with the complete seal model are computationally expensive. For example, we could investigate using block updates only for the most correlated parameters, e.g., $\phi_{p,\max}$ and $\phi_a$, which would allow separate tuning of the scaling factor $h$ for each parameter or parameter block.

The second algorithm that was studied here is the SMC$^2$ algorithm. We did not find a way for this algorithm to work for the complete seal model. Even for the 2-state model, the algorithm provided considerably worse results than the PMMH in the same runtime. We acknowledge that more options can be explored to improve the outcome but based the obtained results we do not think that this algorithm is a viable option for the seal model.

Using the PMMH with the settings that were found to produce the best results, we were able to produce a sample from the posterior distribution with a sufficiently high ESS (4799). Analysing this posterior distribution and comparing the results with the ones obtained by Thomas et al. (2019) using a modified Liu-West algorithm, we found that many of the results were similar. However, there were some significant differences in particular with the variance of the posterior distribution and the prior-posterior overlap. In addition, some of the mean estimates showed clear differences. Assessing the ESS of the output of the modified Liu-West algorithm is difficult and it is therefore hard to diagnose how much of these differences is due to the bias of the Liu-West algorithm and how much due to Monte Carlo error in either of the algorithms. However, with a multivariate ESS of 4799, our results are relatively reliable and the significant differences between the two algorithms should lead to a re-evalutation of the usage of the Liu-West algorithm in the future.

In conclusion, with the PMMH we have found a working algorithm for the complete seal model when enough computing power is available. This allows us to produce unbiased samples with a sufficiently low Monte Carlo error for any further inference and for comparison with other methods. However, for many use cases, a less computationally intensive approach is necessary. Therefore, in the next chapter, we explore the use of a much faster approximation.

Chapter 4

# Using the Kalman Filter to Approximate the Likelihood in Complex State-Space Models

## 4.1 Introduction

In the previous two chapters, it has become apparent that fitting the complete seal model is a challenging task. While the various Sequential Monte Carlo (SMC) methods are designed to deal with complex non-linear non-Gaussian state space models (SSMs) and target the exact posterior distribution of the parameters in the complete seal model, they failed to converge unless an often infeasible amount of computational power was used. Even with state-of-the-art techniques, these long runtimes are required for reliable estimation. This chapter pursues a different approach. Instead of a slow algorithm that targets the exact posterior distribution, we use a much faster algorithm to target an approximation of the posterior distribution. This algorithm is the Kalman filter, which is fast and deterministic but not originally designed for these types of models. Instead, it assumes a normal linear dynamic model (NDLM, see Chapter 4 in West and Harrison, 1997). In this chapter, we approximate the seal model with an NDLM and utilise the Kalman filter to calculate the likelihood to this approximated model. We then focus on the development and application of the Kalman filter within Metropolis-Hastings (KFMH) algorithm, and discuss the ease of implementation, the speed advantages and the robustness of the approximation. To understand the strengths and weaknesses of this approximation, the KFMH is first applied in a simulation study, using the simplified 2-state model described in Section 1.2.6, with a variety of scenarios each representing a different challenge. Two algorithms that target the true posterior distribution are used as baselines to assess the performance of the approximation. After the simulation study, we apply the algorithm to the complete seal model. Here, estimates are compared with those produced by the PMMH algorithm as described in Algorithm 6.

In Section 4.2, we introduce the inference methods for this chapter. First, the class of normal dynamic linear SSMs is defined. A description of the Kalman filter, which calculates the filtered state distributions and the likelihood for these models, follows. We then explain how the linearisation of the full seal model was derived and describe the three parameter estimation algorithms that are compared in the simulation study. We also discuss various criteria to assess the performance of the linear approximation. In

Section 4.3, the simulation study is detailed: first, the design of five different scenarios that represent different challenges in the model fitting process, and then the results. Section 4.4 shows the results obtained in implementing the method on the complete seal model using real data. In Section 4.5 we briefly describe two ideas to improve the Kalman filter approximation and summarise the results of initial investigation where these were applied to the seal model. The results are discussed and set into the wider context in Sections 4.6.

## 4.2 Inference Methods

### 4.2.1 Normal Dynamic Linear Models

An important sub-category of SSMs are normal dynamic linear models (NDLMs). This class of models is categorised by having normally distributed transition and observation processes with fixed variances and an expected value that is a linear function of the conditioning variable, e.g., $x_{t-1}$ in Equation 1.1 and $x_t$ in Equation 1.2 (see, e.g., Chopin and Papaspiliopoulos, 2020). An NDLM can be written as

$$\begin{aligned} x_0 &= c_0 + \eta_0 \\ x_t|x_{0:t-1} &= A_t x_{t-1} + c_t + \eta_t \\ y_t|x_{0:t}, y_{1:t-1} &= B_t x_t + d_t + \epsilon_t, \end{aligned} \tag{4.1}$$

where $\eta_t$ and $\epsilon_t$ are independent Gaussian random variables with $\eta_t \sim \mathrm{N}(0, Q_t)$ and $\epsilon_t \sim \mathrm{N}(0, R_t)$. As in the general definition of an SSM, the state $x_t$ is independent of states $x_{0:t-2}$ given the previous state $x_{t-1}$, and the observation $y_t$ is independent of $x_{0:t-1}$ and $y_{1:t-1}$, given the current state $x_t$. We note that while $A_t, B_t, c_t, d_t, Q_t$ and $R_t$ may be dependent on $t$, they are independent of the underlying state $x_t$. This special class of models is extremely useful, as the filtering distribution $p(x_t|y_{1:t})$ and the observation prediction $p(y_t|y_{1:t-1})$ for these models are normal distributions and can be calculated exactly with the Kalman filter as described in the following section. The Kalman filter can also compute the exact likelihood $p(y_{1:T}|\theta)$ which greatly facilitates parameter inference.

### 4.2.2 Kalman Filter

The Kalman Filter is a deterministic state estimation algorithm that was first developed by Hungarian-American engineer Rudolf Emil Kálmán (Kalman, 1960). In its early history, it played an important role for the trajectory estimation for spacecraft navigation in the Apollo program and has since become a fundamental tool in many fields, including engineering, finance and robotics (see Grewal and Andrews, 2010 for a historical overview). The algorithm calculates filtered point estimates ($\hat{x}_t$) and error variances ($P_t$) for the states $x_t$ in an NDLM, given the observations $y_{1:t}$, an initial state distribution $p(x_0)$, and parameters $\theta$ (Harvey, 1990). Because the model is linear and normal, the distribution of $x_t|y_{1:t}$ is normal as well, and obtaining the mean and variance is therefore sufficient to know the entire distribution.

The Kalman filter is given in Algorithm 8 for the NDLM defined in Equation 4.1. It can be divided into two stages—the prediction stage and the updating stage. During the prediction stage, we make a prediction for $x_t$ given the observations $y_{1:t-1}$, denoted $\hat{x}_{t|t-1}$. In the updating stage of the algorithm, the prediction is updated by including the new observation $y_t$. The estimate for $x_t$ given the observations up to $y_t$, denoted $\hat{x}_t$, is a weighted average between the predicted value $\hat{x}_{t|t-1}$ and the value suggested by the observation $y_t$. How much weight each of these values is given depends on the uncertainty associated with both values (Harvey, 1990).

At the prediction stage for time $t$, we calculate a prediction for the state $x_t$ and the observation $y_t$ as well as an estimate for the covariance matrix of the prediction error. This is straightforward in an NDLM:

$$\begin{aligned}\hat{x}_{t|t-1} &= E(x_t|y_{1:t-1}) \\ P_{t|t-1} &= Var(x_t - \hat{x}_{t|t-1}) \\ \hat{y}_{t|t-1} &= E(y_t|y_{1:t-1}).\end{aligned} \tag{4.2}$$

In the updating stage of the algorithm, we calculate a weighted average between the predicted state $\hat{x}_{t|t-1}$ and the state indicated by the observation $y_t$. The weight is given by the *Kalman gain* matrix $K_t$ which depends on the error covariance matrix of the predicted observation $F_t$ and the cross covariance matrix between the prediction error for the state and for the observation. For convenience, we also introduce the variable $v_t$, often referred to as *innovation* (Durbin and Koopman, 2012), to describe the difference between the actual and the predicted observation, $y_t$ and $\hat{y}_{t|t-1}$ respectively:

$$\begin{aligned}v_t &= y_t - \hat{y}_{t|t-1} \\ F_t &\simeq Var(v_t) \\ G_t &\simeq Cov(x_t - \hat{x}_{t|t-1}, v_t) \\ K_t &= G_t F_t^{-1}.\end{aligned} \tag{4.3}$$

If $F_t$ is a singular matrix, a pseudoinverse can be used. With the gain matrix $K_t$, the updated estimates for the state $x_t$ and error covariance matrix can be calculated as follows:

$$\begin{aligned}\hat{x}_t &= \hat{x}_{t|t-1} + K_t v_t \\ P_t &= P_{t|t-1} - G_t F_t^{-1} G_t' = P_{t|t-1} - K_t F_t K_t'.\end{aligned}$$

The formula for $\hat{x}_t$ has an intuitive interpretation: if the variance of $v_t$ is large, for example because the measurements are highly inaccurate, very little weight is given to $y_t$. The same happens if the covariance between the state and the measurement is small. On the other hand, if the state and the measurement are highly correlated, and the measurement error variance is small, more weight is given to the measurements. We note that to initialise the filter at $t = 1$ the same formulae are used where we set $y_{1:0} := ()$ as the empty tuple.

When the estimate and associated covariance matrix are calculated in this way, $\hat{x}_t$ is the expected value of $x_t|y_{1:t}$, and $P_t$ is the error covariance matrix of the estimation error $x_t - \hat{x}_t$. This process therefore returns the best estimate for the state vector given the observations in the sense that it is the minimum mean square error estimator. In particular, it is better than just taking the state suggested by its measurement $y_t$. The following lemma summarises the properties of this approach.

**Lemma 1.** *If the quantities as given above can be calculated exactly and all errors are Gaussian, the following properties hold.*

1. *The conditional distribution of $x_t|y_{1:t}$ is multivariate normal with mean $\hat{x}_t$ and covariance matrix $P_t$.*
2. *It follows that $\hat{x}_t$ is the minimum mean square estimator for $x_t$ given $y_{1:t}$.*

*Proof.* See Moore and Anderson (1979), p.23-31. □

In addition to the filtered distributions, the Kalman filter also computes the likelihood $l_t = p(y_{1:t}|\theta)$ of the observations for fixed parameter values. It is important to note that unlike the particle filters in Chapter 2, the Kalman filter does not estimate this likelihood but calculates it exactly. It is therefore straightforward to use it as a building block in more complex algorithms that require the likelihood of a parameter, e.g., for any MCMC algorithm that requires an evaluation of the likelihood, like the MH algorithm.

**Algorithm 8** Kalman Filter

```
$t \leftarrow 0$                                                          ▷ Initialisation
$\hat{x}_0 \leftarrow \mathrm{E}(x_0)$
$P_0 \leftarrow \mathrm{Var}(x_0)$
$l_0 \leftarrow 0$                                                        ▷ Log-likelihood
for $t \leftarrow 1, ..., T$ do
    $\hat{x}_{t|t-1} \leftarrow A_t \hat{x}_{t-1} + c_t$                   ▷ Prediction
    $P_{t|t-1} \leftarrow A_t P_{t-1} A_t' + Q_t$
    $\hat{y}_{t|t-1} \leftarrow B_t \hat{x}_{t|t-1} + d_t$                 ▷ Aid variables
    $v_t \leftarrow y_t - \hat{y}_{t|t-1}$
    $F_t \leftarrow B_t P_{t|t-1} B_t' + R_t$
    $G_t \leftarrow P_{t|t-1} Z_t'$
    $K_t \leftarrow G_t F_t^{-1}$
    $\hat{x}_t \leftarrow \hat{x}_{t|t-1} + K_t v_t$                       ▷ Update
    $P_t \leftarrow P_{t|t-1} - G_t F_t^{-1} G_t'$
    $l_t \leftarrow l_{t-1} - \frac{n}{2}\log 2\pi - \frac{1}{2}\log\det F_t - \frac{1}{2} v_t^T F_t^{-1} v_t$   ▷ Log-likelihood
end for
return mean and variance of the distribution of the filtered states, $\hat{x}_t$ and $P_t$;
log-likelihood $\log(p(y_{1:T})) = l_T$
```

Several extensions of the filter have been proposed to allow for more complicated models (Lefebvre et al., 2004 gives a brief summary). For example, the extended Kalman filter (EKF, Gordon et al., 1993) uses Taylor expansion to approximate non-linear functions within the model and the unscented Kalman filter (UKF, Wan and van der Merwe, 2001) approximates all model distributions with a discrete distribution. The UKF is briefly explored as an alternative to the Kalman filter in Section 4.5.1.

### 4.2.3 Linearisation and Normalisation of the Seal Model

The seal model is neither Gaussian nor linear. However, we can construct an NDLM that matches the seal model as closely as possible by attempting to match the first two moments. Since the linearisation process is the same for all four regions, we describe it only for one region and omit the index $r$ for better readability, which corresponds to the 7-state model described in Section 1.2.5. The steps to linearise the 2-state model, which is used in the simulation study in Section 4.3, are not described here. They consist of similar but fewer calculations.

**Initialisation** Linearising and normalizing the initialisation is done by calculating the first and second moment of the distribution of $x_0$ conditional on $y_0$ (see Section 1.2.3.5) and defining a normal distribution with these moments. This is relatively straightforward but requires lengthy calculations as the covariance matrix has dimension 7 × 7 with all non-zero entries. An outline of these calculations can be found in Appendix D.1. The mean of the initial state is

$$\mathrm{E}(x_0) = y_0 \begin{pmatrix} 1 \\ 0.5\phi_{p,0} \\ 0.5\phi_{p,0}\phi_a \\ 0.5\phi_{p,0}\phi_a^2 \\ 0.5\phi_{p,0}\phi_a^3 \\ 0.5\phi_{p,0}\phi_a^4 \\ 1/\alpha \end{pmatrix}$$

and the variances are

$$\begin{aligned} \mathrm{Var}(x_{0,0}) &= y_0^2/\tau \\ \mathrm{Var}(x_{1,0}) &= 0.5\phi_{p,0}y_0 + (0.5\phi_{p,0})^2(y_0^2/\tau - y_0) \\ \mathrm{Var}(x_{2,0}) &= 0.5\phi_{p,0}\phi_a y_0 + (0.5\phi_{p,0})^2\phi_a^2(y_0^2/\tau - y_0) \\ \mathrm{Var}(x_{3,0}) &= 0.5\phi_{p,0}\phi_a^2 y_0 + (0.5\phi_{p,0})^2\phi_a^4(y_0^2/\tau - y_0) \\ \mathrm{Var}(x_{4,0}) &= 0.5\phi_{p,0}\phi_a^3 y_0 + (0.5\phi_{p,0})^2\phi_a^6(y_0^2/\tau - y_0) \\ \mathrm{Var}(x_{5,0}) &= 0.5\phi_{p,0}\phi_a^4 y_0 + (0.5\phi_{p,0})^2\phi_a^8(y_0^2/\tau - y_0) \\ \mathrm{Var}(x_{6+,0}) &= \frac{(1-\alpha)y_0 + y_0^2/\tau}{\alpha^2}. \end{aligned}$$

The covariances are given by

$$
\begin{aligned}
\mathrm{Cov}(x_{0,0}, x_{1,0}) &= 0.5\phi_{p,0} y_0^2/\tau \\
\mathrm{Cov}(x_{1,0}, x_{2,0}) &= 0.5\phi_{p,0}\phi_a y_0 + (0.5\phi_{p,0})^2 \phi_a (y_0^2/\tau - y_0) \\
\mathrm{Cov}(x_{2,0}, x_{3,0}) &= 0.5\phi_{p,0}\phi_a^2 y_0 + (0.5\phi_{p,0})^2 \phi_a^3 (y_0^2/\tau - y_0) \\
\mathrm{Cov}(x_{3,0}, x_{4,0}) &= 0.5\phi_{p,0}\phi_a^3 y_0 + (0.5\phi_{p,0})^2 \phi_a^5 (y_0^2/\tau - y_0) \\
\mathrm{Cov}(x_{4,0}, x_{5,0}) &= 0.5\phi_{p,0}\phi_a^4 y_0 + (0.5\phi_{p,0})^2 \phi_a^7 (y_0^2/\tau - y_0) \\
\mathrm{Cov}(x_{5,0}, x_{6+,0}) &= \frac{0.5\phi_{p,0}\phi_a^4 y_0^2/\tau}{\alpha} \\
\mathrm{Cov}(x_{6+,0}, x_{0,0}) &= \frac{y_0^2/\tau}{\alpha} \\
\mathrm{Cov}(x_{0,0}, x_{2,0}) &= 0.5\phi_{p,0}\phi_a y_0^2/\tau \\
\mathrm{Cov}(x_{1,0}, x_{3,0}) &= 0.5\phi_{p,0}\phi_a^2 y_0 + (0.5\phi_{p,0})^2 \phi_a^2 (y_0^2/\tau - y_0) \\
\mathrm{Cov}(x_{2,0}, x_{4,0}) &= 0.5\phi_{p,0}\phi_a^3 y_0 + (0.5\phi_{p,0})^2 \phi_a^4 (y_0^2/\tau - y_0) \\
\mathrm{Cov}(x_{3,0}, x_{5,0}) &= 0.5\phi_{p,0}\phi_a^4 y_0 + (0.5\phi_{p,0})^2 \phi_a^6 (y_0^2/\tau - y_0) \\
\mathrm{Cov}(x_{4,0}, x_{6+,0}) &= \frac{0.5\phi_{p,0}\phi_a^3 y_0^2/\tau}{\alpha} \\
\mathrm{Cov}(x_{5,0}, x_{0,0}) &= 0.5\phi_{p,0}\phi_a^4 y_0^2/\tau \\
\mathrm{Cov}(x_{6+,0}, x_{1,0}) &= \frac{0.5\phi_{p,0} y_0^2/\tau}{\alpha} \\
\mathrm{Cov}(x_{3,0}, x_{0,0}) &= 0.5\phi_{p,0}\phi_a^2 y_0^2/\tau \\
\mathrm{Cov}(x_{4,0}, x_{1,0}) &= 0.5\phi_{p,0}\phi_a^3 y_0 + (0.5\phi_{p,0})^2 \phi_a^3 (y_0^2/\tau - y_0) \\
\mathrm{Cov}(x_{5,0}, x_{2,0}) &= 0.5\phi_{p,0}\phi_a^4 y_0 + (0.5\phi_{p,0})^2 \phi_a^5 (y_0^2/\tau - y_0) \\
\mathrm{Cov}(x_{6+,0}, x_{3,0}) &= \frac{0.5\phi_{p,0}\phi_a^2 y_0^2/\tau}{\alpha} \\
\mathrm{Cov}(x_{0,0}, x_{4,0}) &= 0.5\phi_{p,0}\phi_a^3 y_0^2/\tau \\
\mathrm{Cov}(x_{1,0}, x_{5,0}) &= 0.5\phi_{p,0}\phi_a^4 y_0 + (0.5\phi_{p,0})^2 \phi_a^4 (y_0^2/\tau - y_0) \\
\mathrm{Cov}(x_{2,0}, x_{6+,0}) &= \frac{0.5\phi_{p,0}\phi_a y_0^2/\tau}{\alpha}
\end{aligned}
$$

This fits into the NDLM framework as defined in Equation 4.1 by setting $c_0 = \mathrm{E}(x_0)$ and by filling the entries of the covariance matrix $Q_0$ with the covariances calculated above.

In these calculations, we omitted the additional dispersion factor $a$ described in Section 1.2.3. When a dispersion factor is required as in the complete seal model in Section 4.4, it can be incorporated by setting the expected value of $x_{0,0}$ to $\frac{1}{2}\left(a + \frac{1}{a}\right) y_0$ instead of $y_0$. Its variance $y_0^2/\tau$ needs to be replaced with $\frac{1}{3}\left(a^2 + 1 + \frac{1}{a^2}\right) y_{r,0}^2/\tau + \frac{1}{12}\left(a - \frac{1}{a}\right)^2 y_{r,0}^2$, and the calculations for the other states need to be adjusted accordingly.

**Transition Process** Linearising and normalizing the transition process is done similarly to the initialisation by calculating the first and second moment of the distribution $x_t|x_{t-1}$.

The calculations are given in Appendix D.2. The expected value is

$$\mathrm{E}(x_t|x_{t-1}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \alpha\phi_a & \alpha\phi_a \\ 0.5\phi_{p,t-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \phi_a & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \phi_a & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \phi_a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \phi_a & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \phi_a & \phi_a \end{pmatrix} x_{t-1}$$

and the variances and non-zero covariances are

$$\begin{aligned} \mathrm{Var}(x_{0,t}|x_{t-1}) &= \phi_a\alpha(1-\phi_a\alpha)(x_{5,t-1}+x_{6+,t-1}) \\ \mathrm{Var}(x_{1,t}|x_{t-1}) &= 0.5\phi_{p,t-1}(1-0.5\phi_{p,t-1})x_{0,t-1} \\ \mathrm{Var}(x_{2,t}|x_{t-1}) &= \phi_a(1-\phi_a)x_{1,t-1} \\ \mathrm{Var}(x_{3,t}|x_{t-1}) &= \phi_a(1-\phi_a)x_{2,t-1} \\ \mathrm{Var}(x_{4,t}|x_{t-1}) &= \phi_a(1-\phi_a)x_{3,t-1} \\ \mathrm{Var}(x_{5,t}|x_{t-1}) &= \phi_a(1-\phi_a)x_{4,t-1} \\ \mathrm{Var}(x_{6+,t}|x_{t-1}) &= \phi_a(1-\phi_a)(x_{5,t-1}+x_{6+,t-1}) \\ \mathrm{Cov}(x_{0,t},x_{6+,t}|x_{t-1}) &= \alpha\phi_a(1-\phi_a)(x_{5,t-1}+x_{6+,t-1}). \end{aligned}$$

All other covariances are 0, as the values are independent given $x_{t-1}$. Fitting this into the NDLM framework in Equation 4.1, we set $A_t$ as the matrix given in the calculation of the expected value and fill the entries of $Q_t$ with the variances and covariances given above, replacing any value of $x_{t-1}$ with its expected value. Note that this introduces an additional approximation to the model: in order for the Kalman filter to produce exact results, the covariance matrices need to be independent of the state $x_t$. However, since the linearised model is an approximation in any case, this step can be justified.

**Observation Process: Pup Production** The observation density of the pup production estimate in the original non-linearised seal model (of Sections 1.2.3 to 1.2.6) is

$$y_t|x_t \sim \mathrm{N}(x_{0,t}, x_{0,t}^2/\tau). \tag{4.4}$$

The fixed term in this distribution already fits the NDLM framework given in Equation 4.1, as $x_{0,t}$ is a linear function of $x_t$. The error term of the distribution, however, poses a challenge. While the expected value of the observation $y_t|x_t$ can be (and almost always is) dependent on $x_t$, the variance of $y_t|x_t$ needs to be independent of $x_t$ in general. Otherwise, it is no longer guaranteed that the model distributions such as $p(y_t|y_{t-1})$ and $p(x_t|y_{1:t})$ are normal, as is shown in Lemma 3, and the Kalman filter is therefore no longer able to provide the true filtering distributions. Contrary to the previous paragraph, where the same problem arose for the process density, here we can alleviate this by integrating out $x_t$ for the calculation of the variance. This correction improves the calculation of the filtering and prediction densities with the Kalman filter.

To determine how this problem can be solved, and to find a version of the model that works with the Kalman filter, we need to examine how the Kalman filter uses the distribution of $y_t|x_t$ (see Section 4.2.2). The Kalman filter enables exact calculation of the likelihood $p(y_{1:T})$ by using the factorisation in Equation 2.9 and sequentially calculating the factors

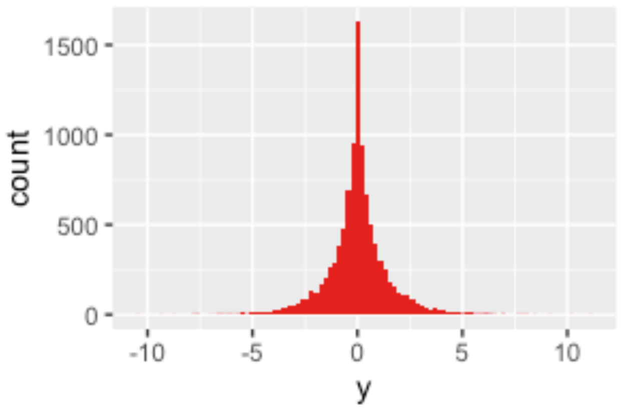


**Figure 4.1:** Histogram of 10,000 samples of $Y$ where $Y|X \sim \mathsf{N}(X, X^2)$ and $X \sim \mathsf{N}(0,1)$

$p(y_t|y_{1:t-1})$. It does this by calculating the intermediate density $p(x_t|y_{1:t-1})$ and using this to integrate out the state $x_t$, so

$$p(y_t|y_{1:t-1}) = \int p(y_t, x_t|y_{1:t-1})dx_t = \int g(y_t|x_t)p(x_t|y_{1:t-1})dx_t.$$

In an NDLM as defined in Equation 4.1, these densities are all normal (for a proof, see p.165 in Harvey, 1990). The distribution of $y_t|y_{1:t-1}$ is therefore also normal and can simply be determined by its expected value and variance. Omitting the index $t$ and the dependence on $y_{1:t-1}$ for both $Y$ and $X$, this result is given in the following lemma.

**Lemma 2.** *Let $X$ be a random variable with $X \sim N(\mu, \sigma_X^2)$, and $Y|X$ be conditionally normally distributed with $Y|X \sim N(X, \sigma_y^2)$.*

1. *The unconditional distribution of $Y$ is normal.*
2. *The unconditional expected value of $Y$ is $\mathrm{E}(Y) = \mu$.*
3. *The unconditional variance of $Y$ is $\mathrm{Var}(Y) = \sigma_X^2 + \sigma_Y^2$.*

*Proof.* See Example 5c on p. 268 in Ross (2011). □

Unfortunately, with the observation density as in Equation 4.4, the distribution of $y_t|y_{1:t-1}$ is not normal, even if $p(x_t|y_{1:t-1})$ is. We show this in the following lemma and it can also be seen in Figure 4.1.

**Lemma 3.** *Let $X$ be a random variable with $X \sim N(\mu, \sigma^2)$, and $Y|X$ be conditionally normally distributed with $Y|X \sim N(X, X^2/\tau)$.*

1. *The unconditional expected value of $Y$ is $\mathrm{E}(Y) = \mu$.*
2. *The unconditional variance of $Y$ is $\mathrm{Var}(Y) = \sigma^2\left(1 + \frac{1}{\tau}\right) + \frac{\mu^2}{\tau}$.*
3. *The unconditional distribution of $Y$ is not normal.*

*Proof.* The proof is given in Appendix D.3. □

To be able to apply the Kalman filter to the model, we need an observation density with a variance that is independent of $x_t$ given $y_{1:t-1}$. For ease of notation, we define

$\mu := \mathrm{E}(x_{0,t}|y_{1:t-1})$ and $\sigma^2 := \mathrm{Var}(x_{0,t}|y_{1:t-1})$. Ideally we want to pick a density that will result in the same unconditional mean and variance as in Lemma 3. This is achieved by replacing the variance term with the expected value of $x^2$:

$$\mathrm{Var}(y_t|x_t) = \mathrm{E}(x_{0,t}^2)/\tau = (\mu^2 + \sigma^2)/\tau.$$

This results in the same correct mean as above and in the correct variance too:

$$\begin{aligned}\mathrm{Var}(y_t|y_{1:t-1}) &= \mathrm{E}(\mathrm{Var}(y_t|x_t, y_{1:t-1})) + \mathrm{Var}(\mathrm{E}(y_t|x_t, y_{1:t-1})) \\ &= \mathrm{E}((\mu^2 + \sigma^2)/\tau|y_{1:t-1}) + \mathrm{Var}(x_{0,t}|y_{1:t-1}) \\ &= (\mu^2 + \sigma^2)/\tau + \sigma^2.\end{aligned}$$

We therefore linearise and normalise the observation density of the true model[1] $y_t|x_t \sim \mathrm{N}(x_{0,t}, x_{0,t}^2/\tau)$ with

$$y_t|x_t \sim \mathrm{N}\left(x_{0,t}, \left((\mathrm{E}(x_{0,t}|y_{1:t-1}))^2 + \mathrm{Var}(x_{0,t}|y_{1:t-1})\right)/\tau\right)$$

**Observation Process: Independent Estimate** Approximating the observation density of the independent estimate of all (male and female) seals aged 1 and above across all regions in 2008 ($t = 24$) with a normal distribution is the most straightforward of the three approximations. As before, we calculate the expected value and variance of the distribution and replace it with a normal distribution with the same mean and variance. The observation density of the independent estimate in the non-linearised seal model is

$$y_{IE,24} \sim x_{1:6+,24} - \mathrm{Ga}(\kappa_1, \kappa_2).$$

The expected value and variance are therefore

$$\begin{aligned}\mathrm{E}(y_{IE,24}|x_{t=24}) &= x_{1:6+,24} - \kappa_1\kappa_2 \\ \mathrm{Var}(y_{IE,24}) &= \kappa_1\kappa_2^2\end{aligned}$$

The problem that arises in the observation density of the pup production estimates is absent here because the variance of the observation is independent of the state $x$. The sum of all adult seals in all regions is a linear combination of the state vector and so already fits into the NDLM framework. The distribution of the independent estimate in the NDLM approximation of the model is therefore

$$y_{IE,24}|x_{24} = \omega \sum_{r=1}^{4}\sum_{a=1}^{6} x_{a,r,24} - \kappa_1\kappa_2 + \epsilon_{IE,24},$$

where

$$\epsilon_{IE,24} \sim \mathrm{N}(0, \kappa_1\kappa_2^2).$$

[1]Throughout this chapter, “true model” refers to any of the models described in Section 1.2 in contrast to the linearised approximation.

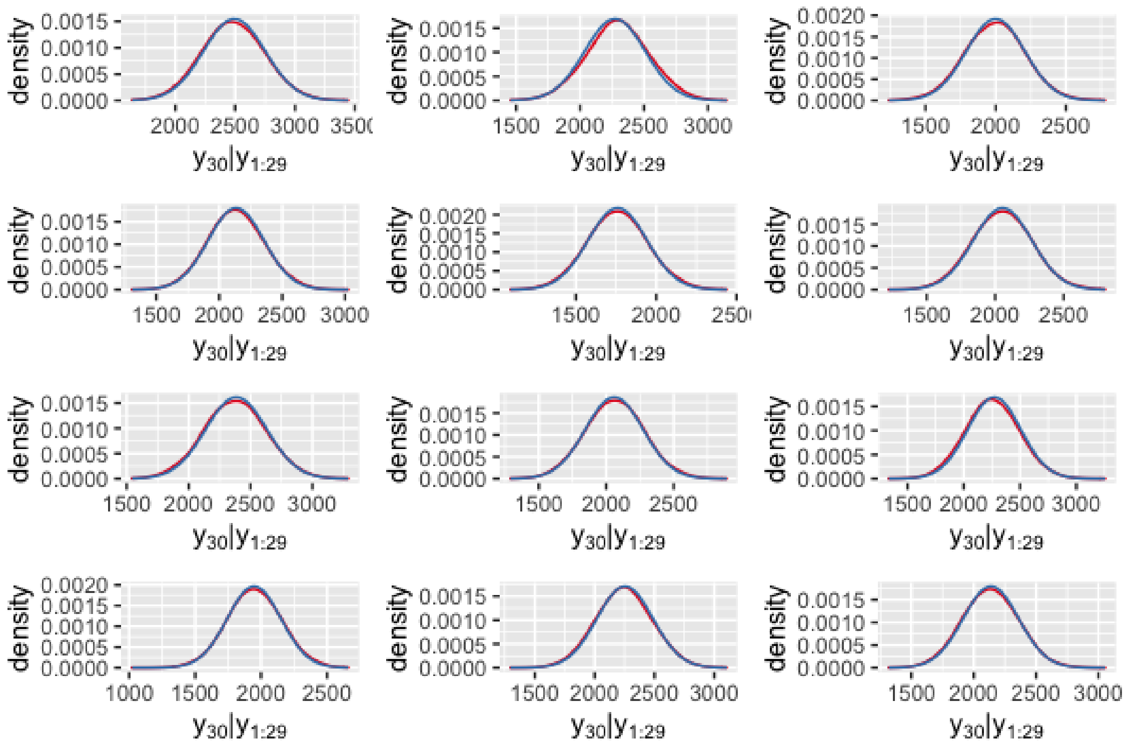


**Figure 4.2:** Density plots of $y_{30}|y_{1:29}$ (generated from 100,000 samples) for fixed parameter values ($\phi_{p,\max} = 0.7, \phi_a = 0.9, \alpha = 0.9, \chi = 2500, \rho = 6, \tau = 100$) and 12 different simulations of $y_{1:29}$ in the 7-state model (🟥) and its NDLM approximation (🟦).

### 4.2.4 Quality of the NDLM Approximation

The goal of creating an NDLM to approximate the true model is to estimate the posterior distribution of the parameters. Our main interest therefore lies in how close the posterior distribution of the parameters, given the NDLM, is to the true posterior distribution of the parameters, given the true model. This is difficult, as we do not know what the true posterior distribution is. For fixed parameter values, however, it is much easier to estimate or calculate the distribution of the states or the observations. We can use these distributions to gain some initial insights into how close the NDLM is to the true model. Here, we compare a few distributions of the true seal model in just one region (so the 7-state model) and its NDLM approximation for fixed parameter values. The first distribution we compare is $y_t|y_{1:t-1}$. As Equation 2.9 shows, this distribution is particularly important to calculate the likelihood. For the NDLM, we use the prediction $\hat{y}_{t|t-1}$ and its variance $F_t$ provided by the Kalman filter in Algorithm 8. For the true model, we use the bootstrap filter determined in Chapter 2 with 10,000 particles to generate samples of states from $x_t|y_{1:t-1}$ and simulate an observation for each of these states. The density plots of the distribution for 12 different simulations of $y_{30}|y_{1:29}$ in both models is shown in Figure 4.2. While the densities do not match exactly, they are close. Comparing the differences in mean and standard deviation between the two distributions for 1000 simulations, we obtained a 95% confidence interval of $(-16.31, 17.28)$ for the mean and $(-5.45, 2.09)$ for the standard deviation. There is therefore no evidence for a bias in the mean or the variance of the NDLM approximation.

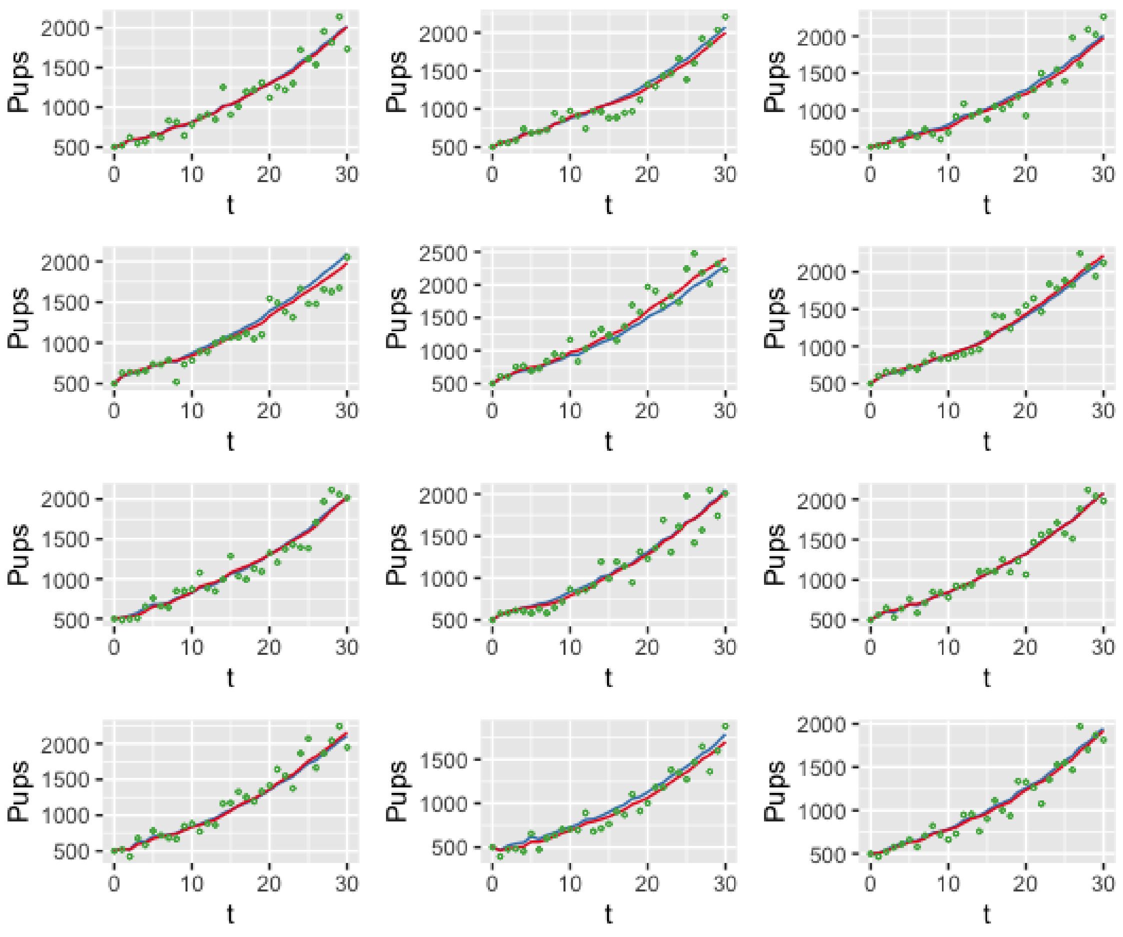


**Figure 4.3:** Filtered estimates of number of pups $x_t|y_{1:t}$ for $t = 0, \ldots, 30$ with fixed parameter values ($\phi_{p,\max} = 0.7, \phi_a = 0.9, \alpha = 0.9, \chi = 2500, \rho = 6, \tau = 100$) and 12 different simulations of $y_{1:30}$ (●) with a bootstrap filter (🟥) and with a Kalman filter using the NDLM approximation (🟦).

Another relevant comparison is the filtered estimates of the number of pups $x_t$ given the observations $y_{1:t}$. Again, the Kalman filter is used to calculate filtered estimates of $x_t|y_{1:t}$ for the NDLM, and a bootstrap filter with 10,000 particles is used for the true model. The resulting estimates can be seen in Figure 4.3 for fixed parameters and 12 simulated sets of observations $y_{1:30}$. The expected value of the filtered distribution $x_t|y_{1:t}$ is estimated with the Kalman filter and with a bootstrap filter. Again, the estimates are quite close to each other with no obvious bias of the NDLM estimates. In a further simulation with 1000 sets of observations $y_{1:30}$, the differences between estimates of the filtered state computed by the Kalman filter and a bootstrap filter were calculated. For each timepoint, the central 95% of the differences were calculated. Each of these intervals contain 0, and so the two estimates do not differ significantly. Figure 4.4 shows boxplots of these 1000 differences for each time point. We note that the range of differences increases as $t$ increases. This may be due to the increasing observation error with increasing $x_t$ which results in a higher variance of the filtered estimate under either model.

Lastly, we compare the observation density of the independent estimate of all adult seals with its normal approximation for a fixed underlying state. As can be seen in Figure

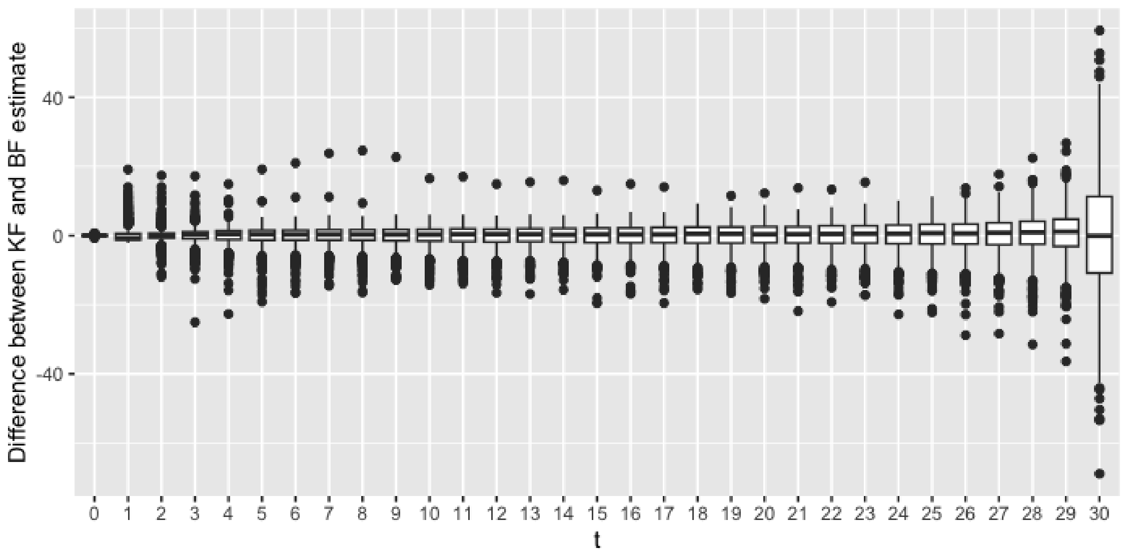


**Figure 4.4:** Boxplot of the differences between the filtered estimates of number of pups $x_t|y_{1:t}$ for $t = 0, \dots, 30$, produced by the Kalman filter (KF) applied to the NDLM approximation and a particle filter (BF), with fixed parameter values ($\phi_{p,\max} = 0.7, \phi_a = 0.9, \alpha = 0.9, \chi = 2500, \rho = 6, \tau = 100$) for 1000 different simulations of $y_{1:30}$.

4.5, the two distributions are similar but not quite the same. It is clear that the normal distribution cannot capture the asymmetry of the Gamma distribution but the resulting difference in the distributions seems to be relatively small. To quantify this difference we use the distribution-free overlapping index suggested by Pastore and Calcagnì (2019), which measures the amount of shared area under the two curves of the densities. For two probability density functions $p_1$ and $p_2$, this index is

$$\eta(p_1, p_2) = \int \min\left(p_1(x), p_2(x)\right) dx.$$

An index of 0 would mean that there is no overlap in the pdfs and an index of 1 would mean that the pdfs are identical almost everywhere except possibly a subset of measure zero. For the normal and gamma densities shown in Figure 4.5, the overlap was evaluated numerically to be 0.929.

### 4.2.5 Algorithms for Parameter Estimation

We describe here the three different algorithms to estimate the posterior distribution of the parameters: a Kalman filter within Metropolis-Hastings (KFMH) algorithm, the Particle Marginal Metropolis-Hastings (PMMH) algorithm (Algorithm 6), and an MCMC algorithm that uses data augmentation rather than exploiting the sequential nature of the model. Similarly to the PMMH, the KFMH relies on the Metropolis-Hastings (MH) algorithm (see Section 3.2.2.1) to perform parameter inference. However, the two algorithms differ in the way in which they compute or estimate the likelihood $p(y_{1:T}|\theta)$ which is required for the MH algorithm. Lastly, we briefly outline how an MCMC algorithm with data augmentation (MCMC-DA) performs parameter inference for SSMs. While the MCMC-DA is not expected to perform well for this type of problem, it is still often the "go-to" solution. We want to show that alternative general algorithms perform better for SSMs and therefore we include this algorithm to demonstrate our improvements.

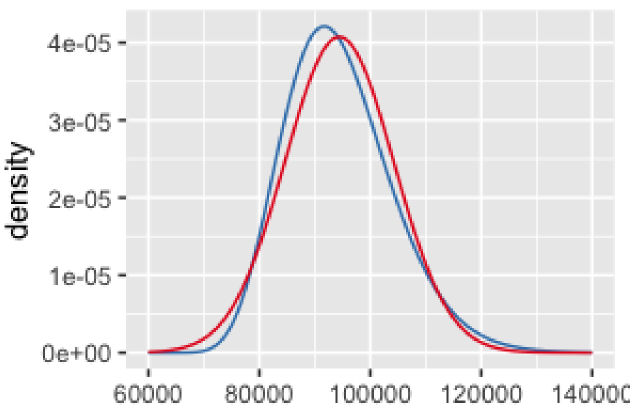


**Figure 4.5:** Observation density of the independent estimate for fixed $x_{t=24}$ in the 7-state model (🟥) and its NDLM approximation (🟦).

#### 4.2.5.1 KFMH

To generate samples from the posterior distribution of the NDLM approximation, we use the KFMH algorithm. In that algorithm, the MH algorithm is used as described in Section 3.2.2.1. Any required likelihood of a parameter value is calculated by running the Kalman filter with that parameter value. This is possible because we are only applying this algorithm to the NDLM and so have the Kalman filter readily available for likelihood calculation. As the Kalman filter is deterministic and consists mostly of a sequence of matrix multiplications, one iteration of the resulting chain is relatively fast. For a fixed total computer time, therefore, compared with other methods that take longer to evaluate the likelihood, more MCMC samples can be generated.

The algorithm was implemented in `nimble`, where we relied on the following observation: to calculate the likelihood for the MCMC algorithm, the states $x_{0:T}$ do not need to be modelled explicitly. Instead, only the likelihoods $p(y_t|y_{t-1}, \theta)$ are necessary. With the Kalman filter, these distributions can be calculated and are

$$y_t|y_{t-1} \sim \mathcal{N}(\hat{y}_{t|t-1}|F_t), \tag{4.5}$$

with $\hat{y}_{t|t-1}$ and $F_t$ calculated as specified in Algorithm 8. Instead of first writing the entire model in the `nimble` language and then writing a Kalman filter implementation to interact with `nimble`'s MCMC engine, we integrated the states out already in the model definition. The only variables (in addition to the parameters $\theta$) in the model as written for `nimble` are then the observations $y_{1:T}$ whose distribution can be given exactly using the Kalman filter with Equation 4.5. With this trick, we can invoke `nimble`'s MCMC engine on this model. The implementation of the Kalman filter for the 2-state model in `nimble` can be found in a Github repository (Empacher, 2023).

#### 4.2.5.2 PMMH

The second algorithm used here to generate samples from the posterior distribution of the true model is the PMMH. As discussed extensively in Chapter 3, this algorithm takes a long time to converge when applied to the true model. However, for the simpler 2-state model, it does provide an unbiased estimate of the true posterior distribution in feasible computation time. It is therefore a useful tool to assess the performance of the KFMH for the simpler model and we can use it as a basis for comparison.

The specific settings of this algorithms are as described in Section 3.3.1.3.

#### 4.2.5.3 Data Augmentation

To highlight the advantage of using an algorithm that utilises the sequential structure of the model, we contrast the performance of the two sequential algorithms above with an MCMC algorithm that uses data augmentation (MCMC-DA) to generate estimates from the posterior distribution (see Section 3.2.2.1).

Here, we used `nimble` to create and execute this algorithm and relied on its default configuration to generate samples from $p(x_{0:T}, \theta | y_{1:T})$ (de Valpine et al., 2022). This means that all parameters but $\tau$ were sampled independently from a Metropolis-Hastings adaptive random-walk sampler with a univariate normal proposal distribution. Since there is a conjugate relationship between the prior distribution of $\tau$ and its dependants $y_{1:T}$, a Gibbs sampler was assigned to this parameter and the discrete states $x_{1:T}$ were generated with a slice sampler Neal (2003), both done automatically done by `nimble`.

The tuning of the MCMC algorithm for all three algorithms, KFMH, PMMH and MCMC-DA, for example adapting the variance of the random walk proposal distribution, was left up to `nimble`. This was done to ensure a fair comparison between the three algorithms.

### 4.2.6 Assessment of Performance

In comparing the performance of the three algorithms, there are two aspects to consider. The first is Monte Carlo accuracy of each of the algorithms, i.e., how long the algorithms take to converge to their respective target distribution and how efficient the algorithms are once converged. The second is the crucial aspect of whether the KFMH applied to the NDLM approximation provides posterior estimates that are close to the true posterior distribution.

We use the potential scale reduction factor $\hat{R}$ (see Section 3.2.2.1) to assess whether convergence has been achieved. We report the upper limit of a 95% confidence interval for $\hat{R}$ for the parameter with the largest such value. In addition, we examined trace plots to determine whether convergence had been achieved.

The Monte Carlo accuracy of the three algorithms once converged was assessed using the effective sample size (ESS, see Equation 3.4). The computation time of the algorithms was incorporated by measuring their computational efficiency, defined as effective sample size per unit time. As the ESS was computed individually for each parameter, we used the lowest of the values to compare convergence speed. In Section 4.4, we instead use the multivariate ESS (Vats et al., 2019) for better comparability with Chapter 3.

To compare the posterior distributions of the approximated NDLM and the true model, we relied on density plots and summary statistics of the posterior. While the approximation provides a biased estimate of the true posterior distribution, a great reduction in Monte Carlo error compared to the unbiased algorithms can justify the use of the approximation if the bias is not too great.

## 4.3 Simulation Study

### 4.3.1 Simulation Scenarios

To assess the performance of the KFMH, we examined the closeness of the estimate of the posterior of the NDLM to the estimate of the true posterior under different circumstances. These highlight various challenges that might also appear in real applications, for example having a true parameter value that is close to the edge of the support of the prior distribution, uninformative data or a mis-specified model.

To explore these aspects, we performed parameter inference in five different scenarios. The first was a baseline scenario where no additional challenges were included. The other four scenarios varied just one aspect each to see how each of those aspects changed the behaviour of the approximation. The scenarios were studied on the 2-state model (see Section 1.2), as only a relatively simple model allowed us to estimate the posterior distribution in feasible computation time to explore different scenarios.

**Scenario 0: Baseline** In this scenario, we simulated $T = 50$ observations. The initial observation was $y_0 = 200$ and the parameters used for the simulation were $\phi_{p,\max} = 0.48, \phi_a = 0.9, \alpha = 0.8, \chi = 2500, \rho = 6, \tau = 100$. The resulting time series showed both the exponential growth at the start of the population development, and the effect of the density dependent pup survival as the population reaches carrying capacity. The observations were unbiased and informative with a coefficient of variation of 10%. The parameters were not close to the edge of the support of the prior distribution.

**Scenario 1: Uninformative Data** This scenario used only the first $T = 10$ observations. This means that only the exponential growth section of the population trajectory was observed, before density dependence started to have an effect. There was therefore very little information about the carrying capacity in the data.

**Scenario 2: Edge of Parameter Support** Here, the adult survival to simulate the data was set to $\phi_a = 0.965$. As before (see Table 1.1), the prior was a shifted and scaled beta distribution with support from 0.8 to 0.97 and the true value was therefore only marginally lower than the upper limit of the prior.

**Scenario 3: Mis-specified Model** In this scenario, the data were simulated from a different model but the algorithms were used to fit the same model as in the baseline scenario. The model to simulate the date was similar to the baseline model but changed in four respects. First, a random effect on the fecundity $\alpha$ was introduced by adding a normally distributed error with a standard deviation of 0.5 to the logit of $\alpha$. Second, the observation precision was assumed to be known as $\tau = 400$ even though the data were simulated with $\tau = 100$. Third, 20% of the observations were outliers, with a much higher coefficient of variation of 50% instead of 10%. Last, there was a slight linear decline in adult survival from 0.9 to 0.81 throughout the time series.

**Scenario 4: Estimating Only One Parameter** All parameters but $\phi_a$ were assumed known, so only the posterior distribution of this parameter was estimated.

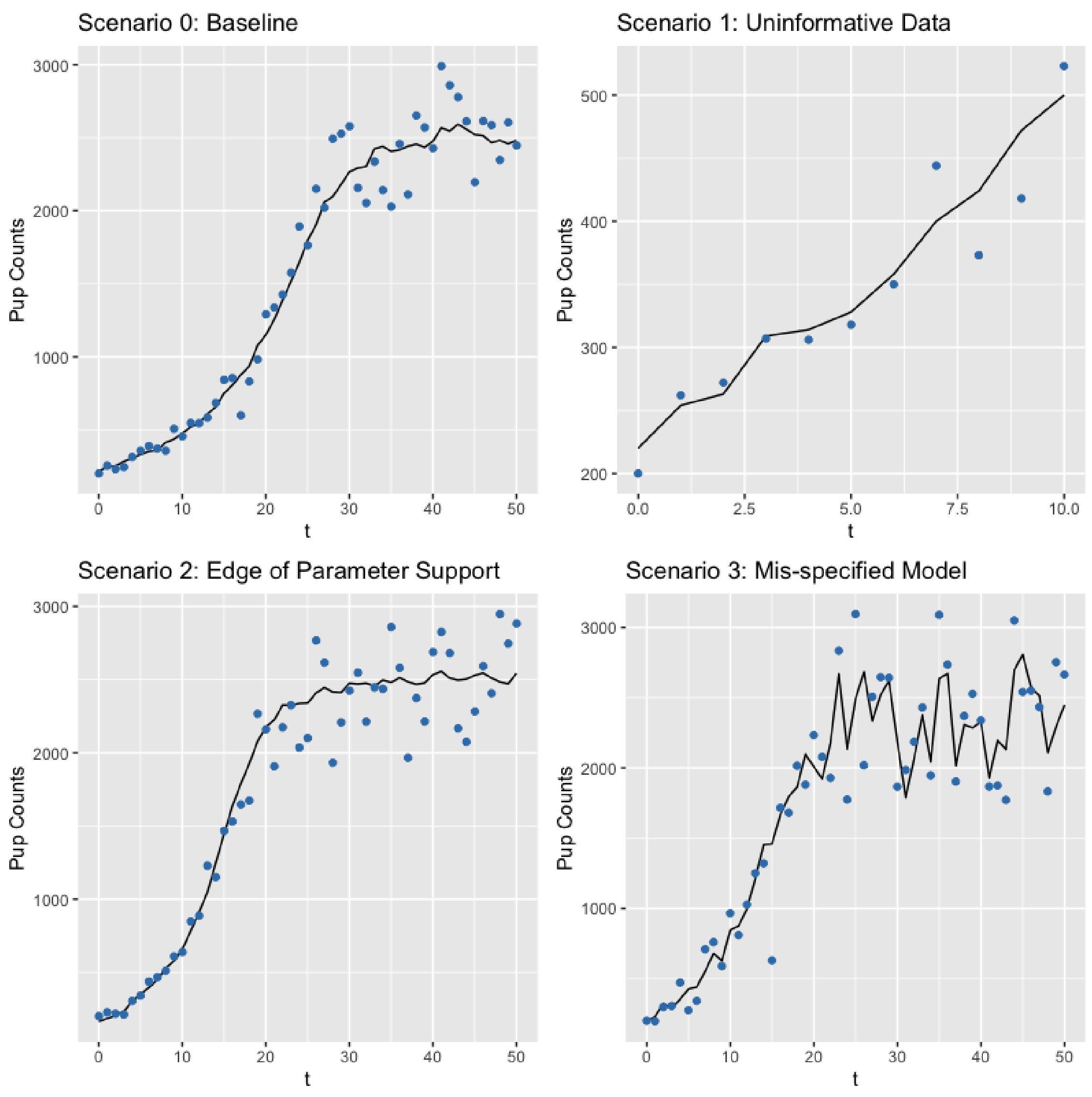


**Figure 4.6:** Simulated trajectories of pup numbers (■) and annual observations (🟦) in Scenarios 0 through 3. The values in Scenario 4 (estimating only one parameter) were simulated in the same way as in Scenario 0 (baseline) and are not shown here.

### 4.3.2 Set-Up

For each of the five scenarios, we simulated five replicate sets of observations $y_{1:T}$. Only the plots for the first of the five replicates of the simulation are shown in the results (Figure 4.6) but the range of values of all five replicates are given in the tables detailing runtimes and performance. For each of the five simulated datasets, the posterior distribution of the parameters was estimated with the three algorithms described in Section 4.2.5: a standard MCMC algorithm, the PMMH algorithm, and the KFMH algorithm applied to the NDLM approximation of the true model. As there was no reason to expect multimodality or other complicating factors, only two chains were run for each algorithm.

To estimate the posterior distributions in each scenario, two chains each with 1,000,000 iterations plus an initial burn-in of 10,000 iterations were run for the MCMC and the KFMH algorithms. As each iteration in the PMMH is much more computationally expensive, the number of iterations was reduced by a factor of 50. Each chain was therefore run for only 20,000 iterations plus a burn-in of 200 iterations. The bootstrap filter within each iteration of the PMMH was run with 1000 particles. These values were chosen to keep the runtimes of the algorithms within the same order of magnitude. In some cases, it was necessary to change the burn-in period of the PMMH algorithm after examining some initial diagnostic plots.

### 4.3.3 Results

#### 4.3.3.1 Scenario 0: Baseline

Table 4.1 shows that mixing in the MCMC-DA algorithm was relatively poor as indicated by the ESS values. The low ESS produced by that algorithm is also visible in the density plots in Figure 4.7 which exhibit an unevenness that points to a high Monte Carlo error. While the same issue is noticeable for the PMMH algorithm, it is much less pronounced as can be seen both by the smoother density plots in Figure 4.7 and by the much higher ESS. The KFMH algorithm suffers none of these problems. Mixing is fast and the ESS per runtime is more than 100 times greater than for the other two algorithms. It is interesting to note that the slowest mixing parameter was different across the three algorithms: for the MCMC-DA algorithm, $\alpha$ had the lowest ESS, for the PMMH algorithm it was $\rho$ and $\tau$ (in different replicates) and for the KFMH algorithm it was $\phi_{p,\max}$.

To compare the posterior distribution of the approximated NDLM to the true posterior distribution, we rely here mostly on the estimated posterior by the PMMH, as it showed a higher ESS and therefore introduced less Monte Carlo error into the estimate. Figure 4.7 shows that the posterior distributions of the PMMH and KFMH algorithm are quite similar. While there are some differences that appear to be systematic, for example a very slight shift to the right for $\tau$, these differences are smaller than the Monte Carlo error in, e.g., the estimated posterior distribution for $\phi_a$. The correlation structure of the posterior distribution is shown in Figure 4.8. It appears to be very similar across all three algorithms. The other replicates showed a similar behaviour to the first one concerning the closeness of the approximation.

#### 4.3.3.2 Scenario 1: Uninformative Data

All three algorithms performed much better in terms of convergence speed and ESS than in Scenario 0 (see Table 4.2). This is likely because there is very little information in the data and therefore the likelihood component of the algorithms is less important compared

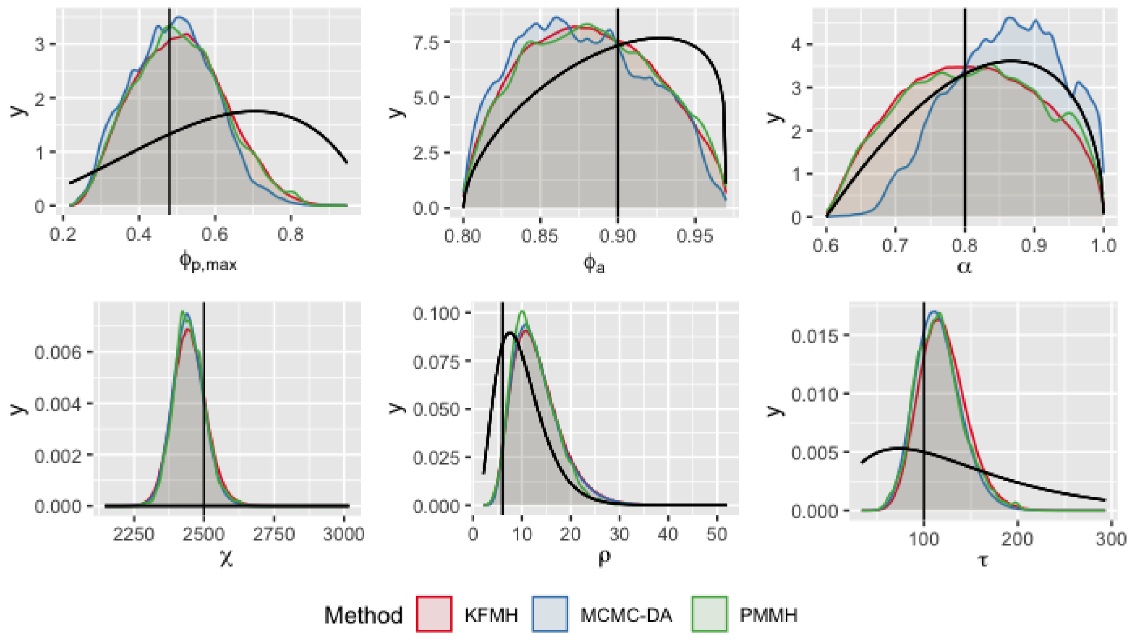


**Figure 4.7:** Scenario 0: posterior distributions of all six parameters for simulated observations in Scenario 0 (baseline) with the 2-state model, estimated by the Particle Marginal Metropolis-Hastings algorithm (PMMH), a MCMC with data augmentation (MCMC) algorithm and the KFMH algorithm. The parameter used for the simulation is indicated by the black vertical line, and the prior distribution is drawn as a black curve.

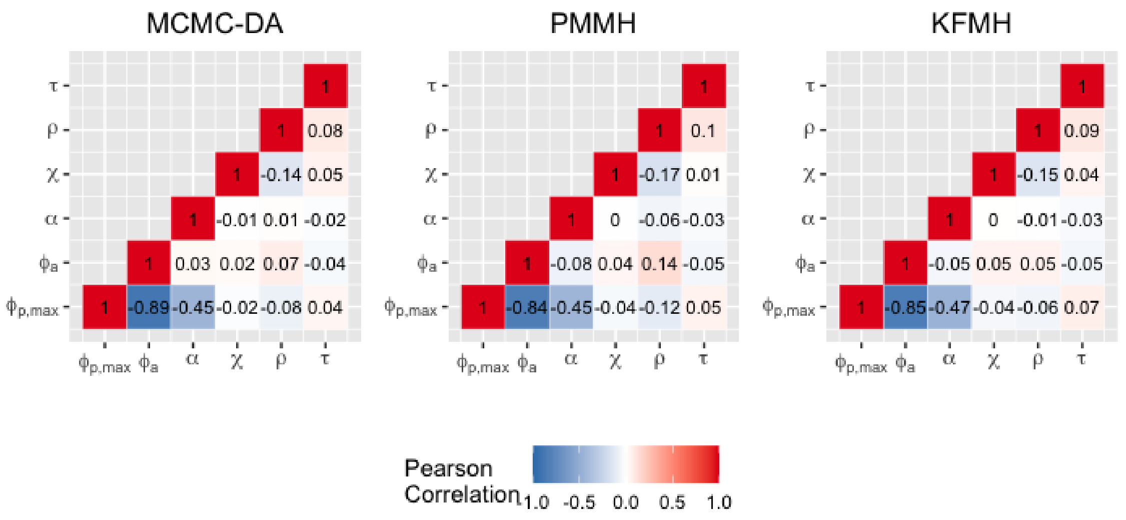


**Figure 4.8:** Scenario 0: correlations between the parameters in the posterior distributions in Scenario 0 (baseline) with the 2-state model, as estimated by (from left to right) the MCMC-DA algorithm, the PMMH algorithm and the KFMH algorithm.

| Algorithm | MCMC-DA | PMMH | KFMH |
|---|---|---|---|
| Max. upper limit of 95% CI of $\hat{R}$ | 1.03 (1.03-2.01) | 1.23 (1.03-1.23) | 1.02 (1-1.03) |
| MCMC iterations (after adjusted burn-in) | 2,000,000 | 24,000 (24,000-30,000) | 2,000,000 |
| Min. ESS | 66.29 (65.28-75.00) | 477.59 (457.22-681.06) | 2,188.64 (2,188.64-3,206.27) |
| Runtime (in sec) | 1,322.93 (1,208.23-1,322.93) | 1,863.21 (1,764.07-1,863.21) | 607.51 (556.35-652.14) |
| Min. ESS/sec | 0.050 $(0.027 - 0.059)$ | 0.263 $(0.259 - 0.370)$ | 3.60 $(3.40 - 4.93)$ |
| Min. ESS/iteration | $3.31 \times 10^{-5}$ $(1.76 \times 10^{-5}$-$3.75 \times 10^{-5})$ | $2.04 \times 10^{-2}$ $(1.66 \times 10^{-2}$-$2.62 \times 10^{-2})$ | $1.09 \times 10^{-3}$ $(1.09 \times 10^{-3}$-$1.60 \times 10^{-3})$ |

**Table 4.1:** Scenario 0: measures to assess the convergence speed of the three algorithms in Scenario 0 (baseline) with the 2-state model. For each entry, first the value in replicate 1 is given and then the range of values across all 5 replicates.

to the prior. For some parameters (such as $\rho$) the algorithms were therefore almost only sampling from the prior. Still the KFMH algorithm was more than 50 times faster than the other two algorithms. The posterior densities and correlation structure were very similar to each other across all three algorithms. This is unsurprising as the likelihood only forms a small component of the posterior density compared to the prior. We expected there to be particularly little information about the carrying capacity, given the way that the data were simulated (see Section 4.3 and Figure 4.6). While the posterior distribution of the carrying capacity shape parameter $\rho$ fulfilled this expectation, the posterior distribution of $\chi$ did not. This might be because the starting value of the simulated population trajectory was chosen too high and that the approaching carrying capacity was already noticeable in the declining pup survival probability. Figure 4.6 shows that the pup numbers seem to grow linearly and not exponentially as they would without density dependence. We note that there was still much more variance in the posterior distribution compared to the posterior distribution of $\chi$ in Scenario 0 (baseline). Again, the other replicates showed similar behaviours to this one.

#### 4.3.3.3 Scenario 2: Edge of Parameter Support

In this scenario, the Markov chains produced by the MCMC-DA had high values for some of the five replicates (see Table 4.3), indicating the difficulty of this algorithm with convergence. The unevenness of the density plot in Figure 4.11 confirms this. Both the PMMH algorithm and the KFMH seemed to converge and seemed largely unaffected by the challenging parameter location. The density plots show that there was relatively little information about the parameters in the data but a pull towards the right edge of the parameter support can be seen for the parameter $\alpha$. The correlation structure looks very similar across all three algorithms (see Figure 4.12).

#### 4.3.3.4 Scenario 3: Mis-specified Model

In this scenario, the observations used to fit the model were simulated with a different model. In practice this means that no parameter vector fit particularly well. It is still important to be able to fit the model to the data. Only then can diagnostic tools be applied

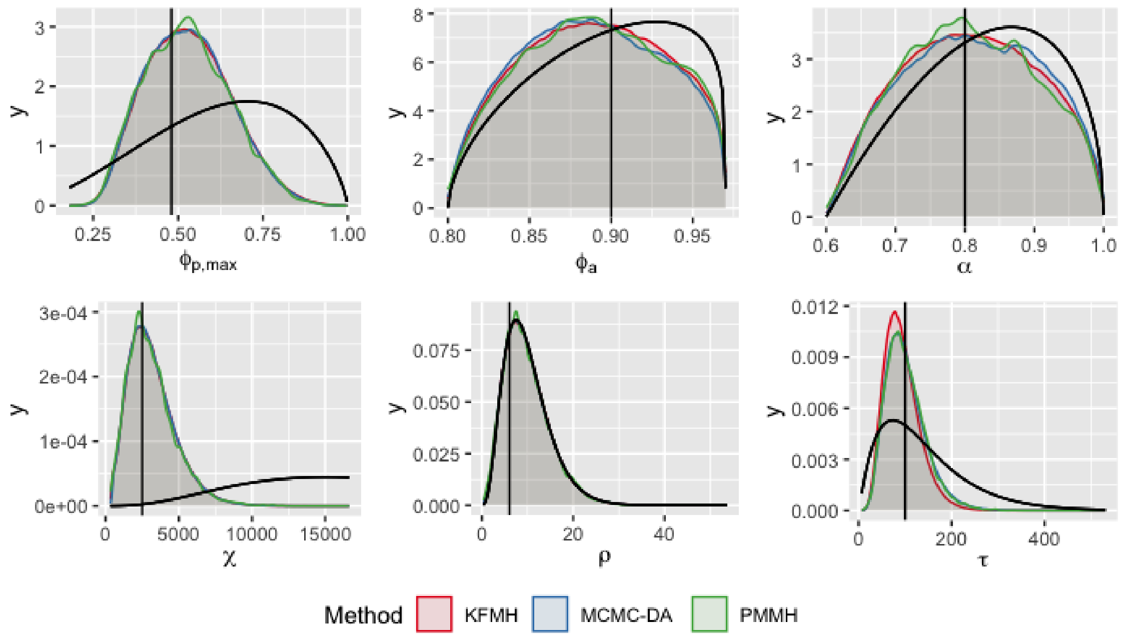


**Figure 4.9:** Scenario 1:Posterior distributions of all six parameters for simulated observations in Scenario 1 (uninformative data) with the 2-state model, estimated by the Particle Marginal Metropolis-Hastings algorithm (PMMH), an MCMC with data augmentation algorithm and the KFMH algorithm. The parameter used for the simulation is indicated by the black vertical line, and the prior distribution is drawn as a black curve.

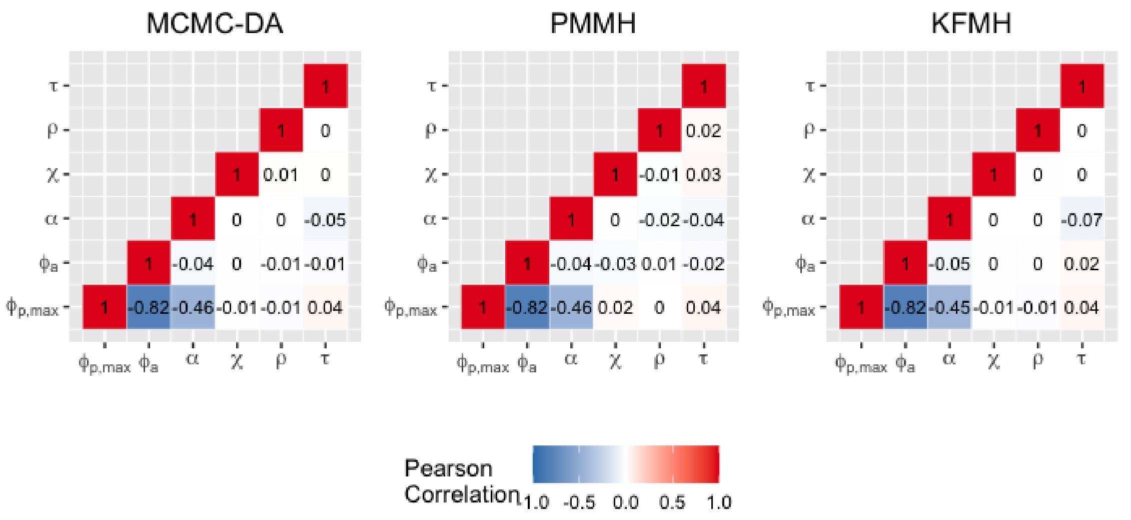


**Figure 4.10:** Scenario 1: correlations between the parameters in the posterior distributions in Scenario 1 (uninformative data) with the 2-state model, as estimated by (from left to right) an MCMC with data augmentation algorithm, the PMMH algorithm and the KFMH algorithm.

| Algorithm | MCMC-DA | PMMH | KFMH |
|---|---|---|---|
| Max. upper limit of 95% CI of $\hat{R}$ | 1.01 (1-1.04) | 1.01 (1.01-1.04) | 1 (1-1) |
| MCMC iterations (after adjusted burn-in) | 2,000,000 | 34000 (34000-36000) | 2,000,000 |
| Min. ESS | 721.45 (721.45-923.20) | 987.73 (987.73-1158.62) | 19357.70 (18749.04-30454.16) |
| Runtime (in sec) | 265.60 (262.75-271.96) | 372.18 (370.84-378.00) | 123.30 (121.63-126.25) |
| Min. ESS/sec | 2.72 (2.72-3.39) | 2.65 (2.65-3.12) | 156.77 (126.25-208.07) |
| Min. ESS/iteration | $3.61 \times 10^{-4}$ ($3.61 \times 10^{-4}$-$4.61 \times 10^{-4}$) | $2.90 \times 10^{-2}$ ($2.90 \times 10^{-2}$-$3.36 \times 10^{-2}$) | $9.68 \times 10^{-3}$ ($9.37 \times 10^{-3}$-$1.52 \times 10^{-2}$) |

**Table 4.2:** Scenario 1: measures to assess the convergence speed of the three algorithms in Scenario 1 (uninformative data) with the 2-state model. For each entry, first the value in replicate 1 is given and then the range of values across all 5 replicates. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the effective sample size by ESS.

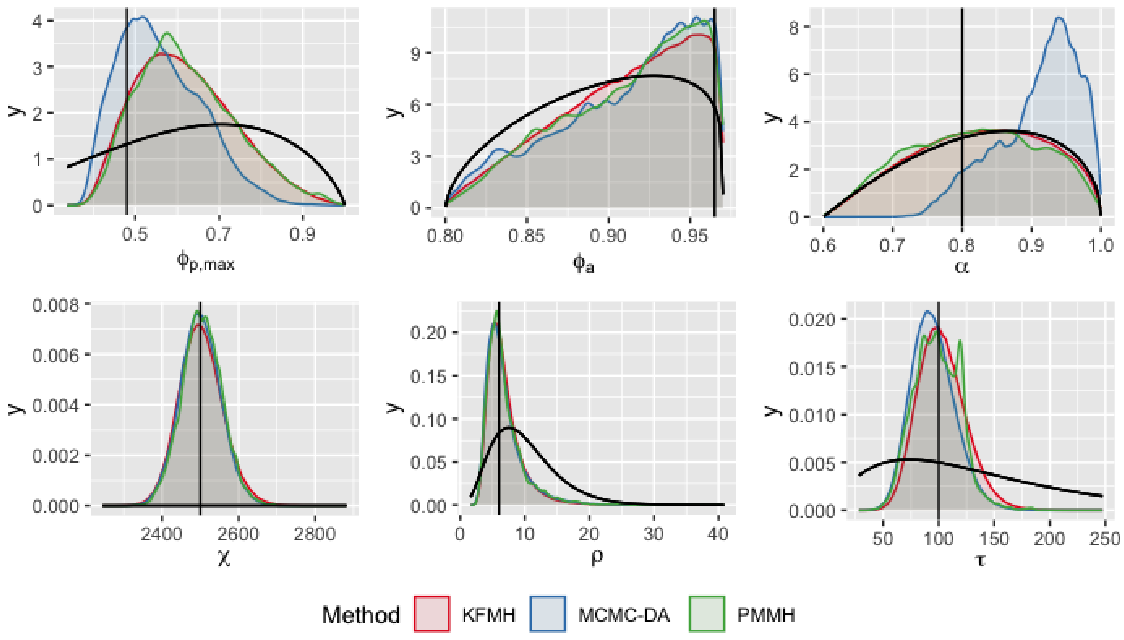


**Figure 4.11:** Scenario 2: posterior distributions of all six parameters for simulated observations in Scenario 2 (edge of parameter support) with the 2-state model, estimated by the Particle Marginal Metropolis-Hastings algorithm (PMMH), an MCMC with data augmentation algorithm and the KFMH algorithm. The parameter used for the simulation is indicated by the black vertical line, and the prior distribution is drawn as a black curve.

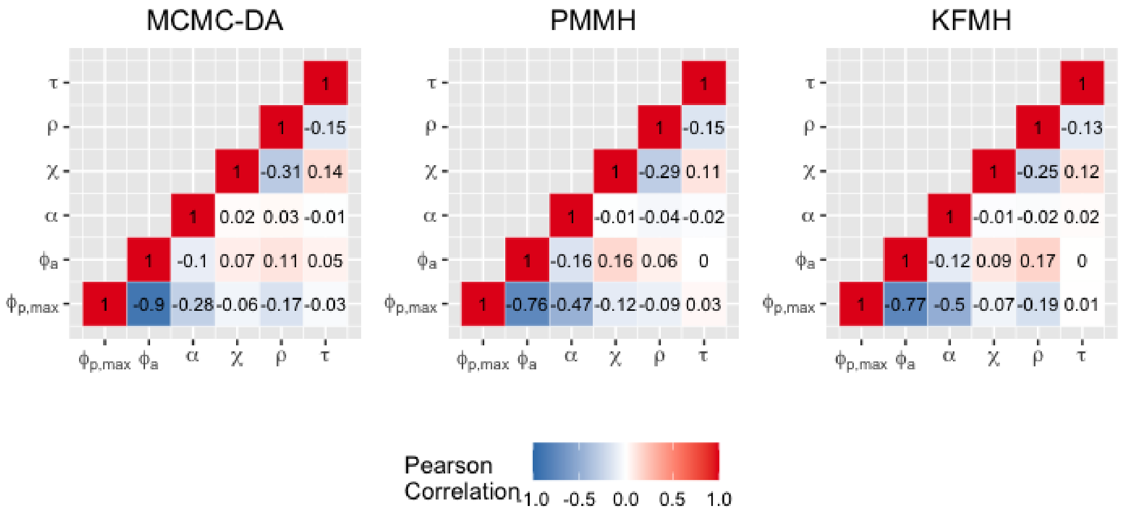


**Figure 4.12:** Scenario 2: correlations between the parameters in the posterior distributions in Scenario 2 (edge of parameter support) with the 2-state model, as estimated by (from left to right) an MCMC with data augmentation algorithm, the PMMH algorithm and the KFMH algorithm.

| Algorithm | MCMC-DA | PMMH | KFMH |
|---|---|---|---|
| Max. upper limit of 95% CI of $\hat{R}$ | 1.09(1.09–1.27) | 1.02 (1.02–1.15) | 1 (1–1) |
| Sample Size | 2,000,000 | 40,000 | 2,000,000 |
| Min. ESS | 53.93 (53.93–60.60) | 556.91 (556.91–735.99) | 8909.04 (8909.04–12124.60) |
| Runtime (in sec) | 1328.91 (1328.91–1392.97) | 1773.02 (1773.02–1791.42) | 726.93 (718.36–726.93) |
| Min. ESS/sec | 0.041 (0.041 – 0.044) | 0.314 0.314 – 0.411 | 12.26 (12.26–16.88) |
| Min. ESS/iteration | $2.70 \times 10^{-5}$ $(2.70 \times 10^{-5} - 3.03 \times 10^{-5})$ | $1.39 \times 10^{-2}$ $(1.39 \times 10^{-2} - 1.84 \times 10^{-2})$ | $4.45 \times 10^{-3}$ $(4.45 \times 10^{-3} - 6.06 \times 10^{-3})$ |

**Table 4.3:** Scenario 2: measures to assess the convergence speed of the three algorithms in Scenario 2 (edge of parameter support) with the 2-state model. For each entry, first the value in replicate 1 is given and then the range of values across all 5 replicates. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the effective sample size by ESS.

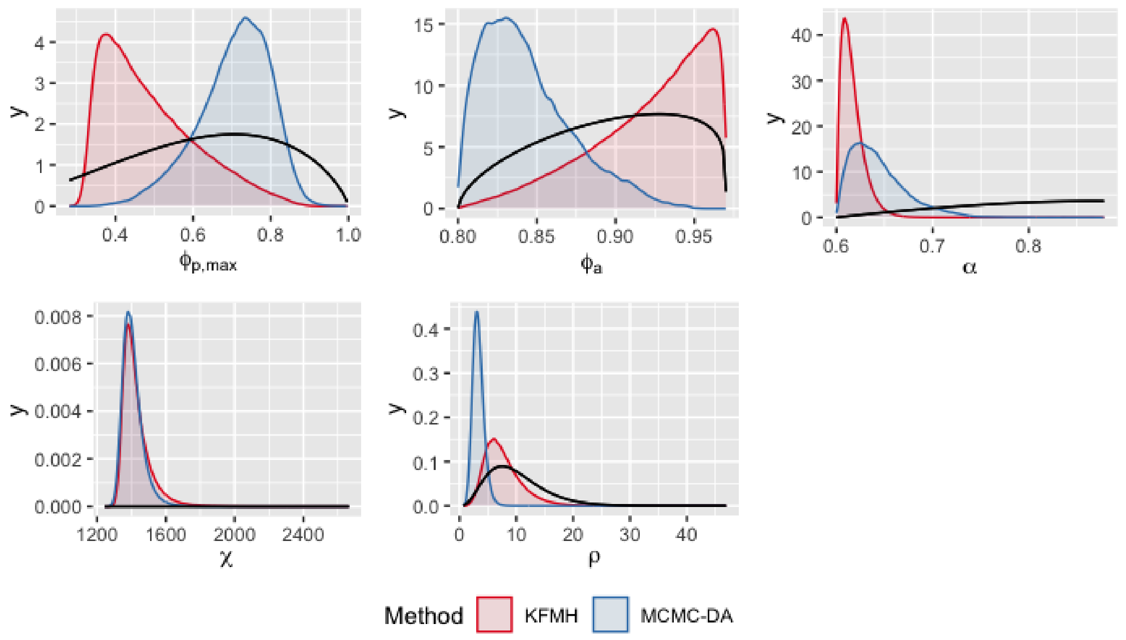


**Figure 4.13:** Scenario 3: posterior distributions of all five parameters for simulated observations in Scenario 3 (mis-specified model) with the 2-state model, estimated by an MCMC with data augmentation algorithm and the KFMH algorithm. The density plot from the PMMH algorithm is not included, because this algorithm never converged. The prior distribution is drawn as a black curve. There is no vertical line to indicate the true parameter value, as the data were simulated from a different model.

to assess the model fit. Here, the PMMH algorithm did not converge at all, as indicated by the very low ESS of 7.23 in Table 4.4, and also by the number of unique values in the chains which was only 38 across all 40,000 iterations. While the algorithm performed better in some of the other replicates, this improvement was not enough to produce any meaningful output for the posterior estimate. The MCMC-DA algorithm performed better with an ESS of 376.94, as did the KFMH algorithm with an ESS of 1900.33. This algorithm outperformed the MCMC-DA algorithm by a factor of 10 in terms of ESS per unit time. It is noticeable that the range of ESS for the KFMH algorithm was very large. This might stem from differences between the specific sets of simulated observations. Some of these sets might be easier to explain with the mis-specified model and therefore simplify the fitting process. In Figure 4.13, the estimated posterior distributions of the MCMC-DA and the KFMH algorithm are compared. It is clear that the estimates are very different. The correlation structure as shown in Figure 4.14 was similar between the two algorithms, although the correlation between $\phi_{p,\max}$ and $\phi_a$ was even stronger for the KFMH algorithm and that algorithm also estimated a higher correlation between some of the parameters (e.g., $\rho$ and $\phi_a$) where the MCMC-DA algorithm estimated almost none. The other four replicates showed similar results.

#### 4.3.3.5 Scenario 4: Estimating Only One Parameter

The inference task here was much easier compared to the other scenarios. This was reflected clearly in the increased ESS for all three algorithms as seen in Table 4.5. The $\hat{R}$-value also indicated that convergence was not problematic in this scenario. The Kalman filter was much faster than the other two algorithms in terms of ESS per unit time, by a

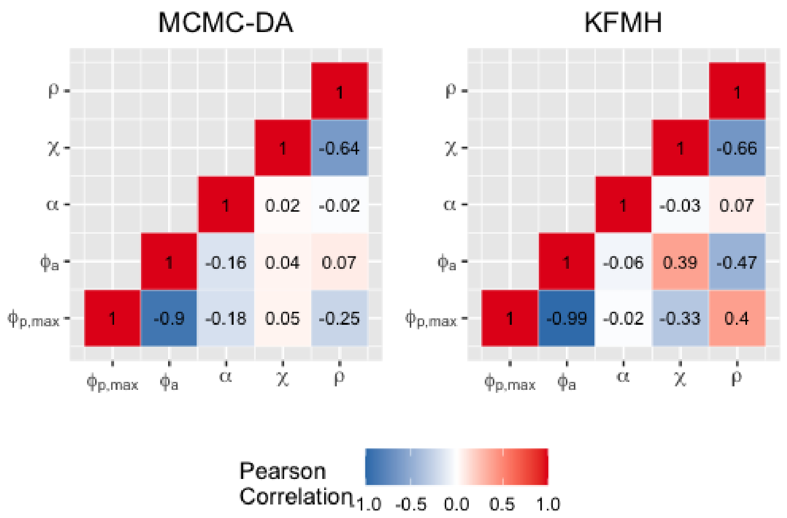


**Figure 4.14:** Scenario 3: correlations between the parameters in the posterior distributions in Scenario 3 (mis-specified model) with the 2-state model, as estimated by (from left to right) an MCMC with data augmentation and the KFMH algorithm. The areas of the circles are proportional to the absolute value of corresponding correlation coefficients.

| Algorithm | MCMC-DA | PMMH | KFMH |
|---|---|---|---|
| Max. upper limit of 95% CI of $\hat{R}$ | 1.08 (1.01−1.29) | 946 (946−1686417) | 1 (1−1) |
| MCMC iterations (after adjusted burn-in) | 2,000,000 (not always converged) | 40,000 (never converged) | 2,000,000 |
| Min. ESS | 376.94 (267.76−376.94) | 7.23 (7.23−45.66) | 1900.33 (1900.33−34185.11) |
| Runtime (in sec) | 1324.74 (1324.50−1334.76) | 2061.52 (1994.17−2097.91) | 511.04 (500.86−541.05) |
| Min. ESS/sec | 0.285 (0.201 – 0.285) | 0.004 (0.004 – 0.023) | 3.72 (3.72−68.25) |
| Min. ESS/iteration | $1.88 \times 10^{-4}$ ($1.34 \times 10^{-4}$-$1.88 \times 10^{-4}$) | $7.18 \times 10^{-5}$ ($7.18 \times 10^{-5}$-$1.14 \times 10^{-3}$) | $9.50 \times 10^{-4}$ ($9.50 \times 10^{-4} - 1.71 \times 10^{-2}$) |

**Table 4.4:** Scenario 3: measures to assess the convergence speed of the three algorithms in Scenario 3 (mis-specified model) with the 2-state model. For each entry, first the value in replicate 1 is given and then the range of values across all 5 replicates. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the effective sample size by ESS.

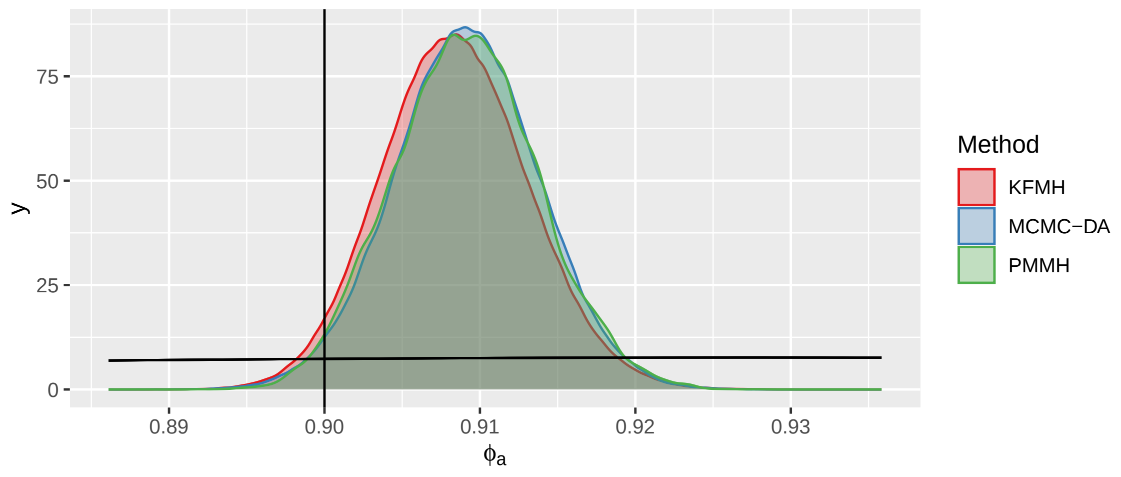


**Figure 4.15:** Scenario 4: posterior distributions of the parameter $\phi_a$ for simulated observations in Scenario 4 (estimating only one parameter) with the 2-state model, estimated by an MCMC with data augmentation algorithm and the KFMH algorithm. The prior distribution is drawn as a black curve and the true value used to simulate the observation is indicated with a black vertical line.

factor of 855. The density plot in Figure 4.15 shows that the estimated posterior density was very similar for the three algorithms. In this case, the data were very informative about the parameter $\phi_a$ relative to the prior, as can be seen by the low variance in the posterior estimate and the almost horizontal black line that indicates the prior. This means that here, contrary to, e.g., Scenario 1 (uninformative data), the likelihood formed an important part of the posterior distribution. This points to the fact that the approximation of the likelihood with the Kalman filter is very close to the true value.

| Algorithm | MCMC-DA | PMMH | KFMH |
|---|---|---|---|
| Upper limit of 95% CI of $\hat{R}$ | 1 (1−1) | 1.01 (1−1.01) | 1 (1−1) |
| MCMC iterations (after adjusted burn-in) | 2,000,000 | 4000 | 2,000,000 |
| ESS | 5474.21 (4382.30 −7873.83) | 8344.70 (8344.70−8557.13) | 452997.40 (452997.40−455635.58) |
| Runtime (in sec) | 1064.91 (1061.88−1073.16) | 1910.01 (1910.01−1945.28) | 103.48 (102.06−115.16) |
| ESS/sec | 5.14 (4.13−7.40) | 4.37 (4.37−4.43) | 4395.50 (3949.15−4443.93) |
| ESS/iteration | $2.74 \times 10^{-3}$ $(2.19 \times 10^{-3} - 3.94 \times 10^{-3})$ | $2.09 \times 10^{-1}$ $(2.09 \times 10^{-1}$-$2.14 \times 10^{-1})$ | $2.27 \times 10^{-1}$ $(2.26 \times 10^{-1} - 2.28 \times 10^{-1})$ |

**Table 4.5:** Scenario 4: measures to assess the convergence speed of the three algorithms in Scenario 4 (estimating only one parameter) with the 2-state model. For each entry, first the value in replicate 1 is given and then the range of values across all 5 replicates if there was some variation. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the effective sample size by ESS.

## 4.4 Application to Real Data

### 4.4.1 Methods

We applied the KFMH algorithm to the real data and the complete seal model. To be able to determine the impact of the independent estimate on the closeness of the approximation and the convergence speed, we estimated the posterior distribution both with and without the independent estimate of all adult seals, $y_{IE}$. The MCMC-DA that was used for comparison in the simulation study in Section 4.3 did not converge at all when applied to the real data in the full seal model and is therefore omitted here. The estimates produces by the KFMH algorithm were compared with those produced by the PMMH algorithm as described in Algorithm 6. When the independent estimate was included, we used the settins and estimates from Chapter 3.4. When the independent sample was excluded, the factorised version of the particle filter as described in Chapter 2.5 was used. Here, the settings that were determined to be optimal in Chapter 6 were used to produce the parameter estimates. In addition, we compared the results with those obtained with the modified Liu-West algorithm described in Section 3.2.1.1.

### 4.4.2 Results

#### 4.4.2.1 Complete Seal Model Without Independent Estimate

| Algorithm | PMMH | KFMH |
|---|---|---|
| Max. upper limit of 95% CI of $\hat{R}$ | 1.06 | 1 |
| MCMC iterations | 4,000,000 | 2,000,000 |
| ESS | 4554.9 | 608835.5 |
| Runtime (in hours) | 425.9 | 16.6 |
| ESS/sec | 0.002971038 | 10.17655 |
| ESS/iteration | $1.14 \times 10^{-3}$ | $3.04 \times 10^{1}$ |

**Table 4.6:** Without $y_{IE}$: measures to assess the convergence speed of the KFMH algorithm and the PMMH algorithm applied to the complete seal model without independent estimate. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the multivariate effective sample size by ESS.

In Table 4.6, we can see that both algorithms converged. The overall computing time for the PMMH algorithm was much higher than for the KFMH. The KFMH algorithm had a high ESS, and in particular also had a higher ESS per iteration than the PMMH algorithm, even though a single iteration with the PMMH is much more computationally expensive. Similarly, the ESS per time was much higher for the KFMH than for the PMMH, namely more than 3400 times higher. The runtime for the KFMH algorithm was quite long at 16.6 hours. However, as the ESS was also very high, the algorithm could easily be run with fewer iterations, depending on how large of an ESS is required in any given situation. Assuming a linear relationship between time and ESS, the algorithm would produce an ESS of 1000 in about half an hour. The Liu-West algorithm is omitted from Table 4.6 because it does not rely on MCMC and so the equivalent diagnostics are not available; there is also no readily available measure of effective sample size (see Section 3.2.1.1).

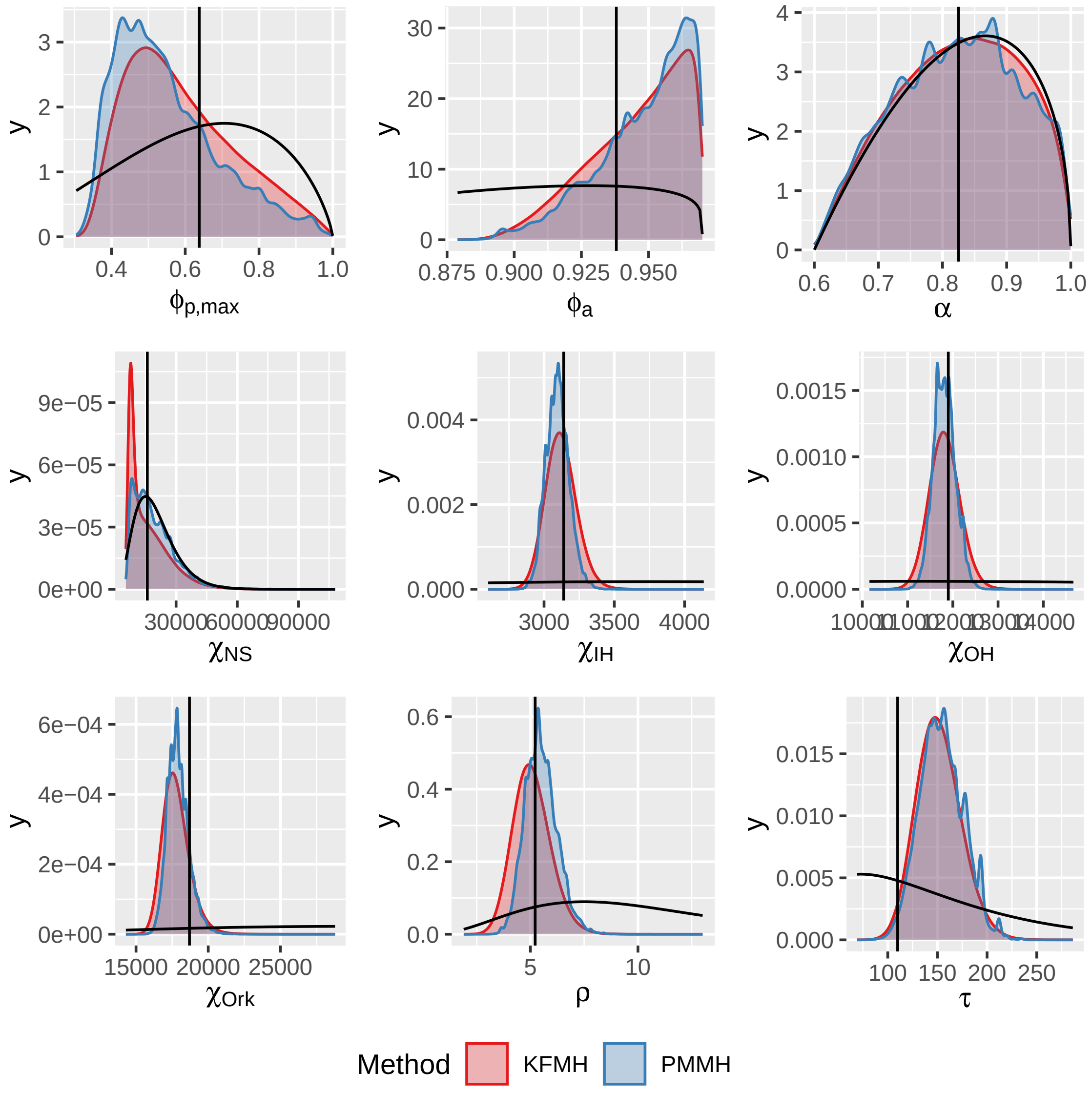


**Figure 4.16:** Without $y_{IE}$: posterior distributions of all nine parameters for simulated observations in the complete seal model without the independent estimate, estimated by the KFMH algorithm and the PMMH algorithm. The prior distribution is drawn as a black curve. The black vertical lines indicate the posterior mean that was obtained by Thomas et al. (2019) with the Liu-West algorithm.

| Parameter | PMMH | KFMH | Liu-West |
|---|---|---|---|
| Maximum pup survival $\phi_{p,\max}$ | 0.550 (0.138) | 0.591 (0.145) | 0.638 (0.155) |
| Adult survival $\phi_a$ | 0.948 (0.0170) | 0.946 (0.0177) | 0.938 (0.0196) |
| Fecundity $\alpha$ | 0.822 (0.0947) | 0.824 (0.0935) | 0.825 (0.0919) |
| DD shape $\rho$ | 5.49 (0.741) | 5.07 (0.871) | 5.22 (1.5) |
| NS carrying cap. $\chi_{NS}$ | 19100 (9470) | 16100 (9740) | 15900 (7870) |
| IH carrying cap. $\chi_{IH}$ | 3100 (81.2) | 3120 (110) | 3140 (180) |
| OH carrying cap. $\chi_{OH}$ | 11800 (243) | 11800 (341) | 11900 (552) |
| Ork carrying cap. $\chi_{Ork}$ | 17900 (749) | 17800 (937) | 18700 (2590) |
| Observation precision $\tau$ | 155 (22.0) | 156 (22.4) | 110 (32.1) |

**Table 4.7:** Without $y_{IE}$: mean and standard deviation of the posterior distributions for the complete seal model without the independent estimate, as estimated by the PMMH algorithm and the KFMH algorithm.

The density plots in Figure 4.16 and the posterior means and standard deviations in Table 4.7 show that the marginal posterior density estimates of most of the parameters were quite close to each other. The relatively low ESS produced by the PMMH algorithm is visible in the unevenness of the density estimate. Even so, the estimates appear to be very similar for most of the parameters, in particular for $\phi_{p,\max}$, $\phi_a$ and $\alpha$. With the carrying capacity parameters, there are slight differences visible. In the case of $\chi_{IH}$, $\chi_{OH}$ and $\chi_{Ork}$ the Kalman filter overestimated the variance of the posterior distributions, whereas it underestimated the mean for $\chi_{NS}$. The posterior distribution for the North Sea carrying capacity parameter $\chi_{NS}$ has a distinctly right-skewed shape even when estimated with the PMMH algorithm. The spike towards the lower end of the distribution was even more pronounced in the estimate provided by the KFMH algorithm. There is very little information about carrying capacity in the observations for the North Sea as the population appears to be growing close to exponentially and seems not yet affected by any density dependence. It is therefore intriguing where the information, especially about the lower limit of the distribution, comes from. It might be that the observed pup numbers in the North Sea region, reaching 8119 in the last year, exclude carrying capacity values below a certain threshold. Another parameter with noteable difference in posterior estimates was $\rho$, where the KFMH estimate showed a negative bias.

Table 4.7 also gives the mean and standard deviation of the posterior estimate given in Thomas et al. (2019). There, the Liu-West algorithm (Liu and West, 2001) was used for estimating the posterior distribution (see Section 3.2.1.1). Similar to the Kalman filter algorithm, this algorithm does not target the true posterior but contains a bias. The mean parameter values computed by this algorithm are also shown in Figure 4.16 as a black vertical line. Comparing the mean values and standard deviations of the three algorithms, we note that the Kalman filter was usually closer to the true posterior distribution as estimated by the PMMH algorithm. The correlation structure in the posterior distributions, as seen in Figure 4.17, was very similar between the two algorithms.

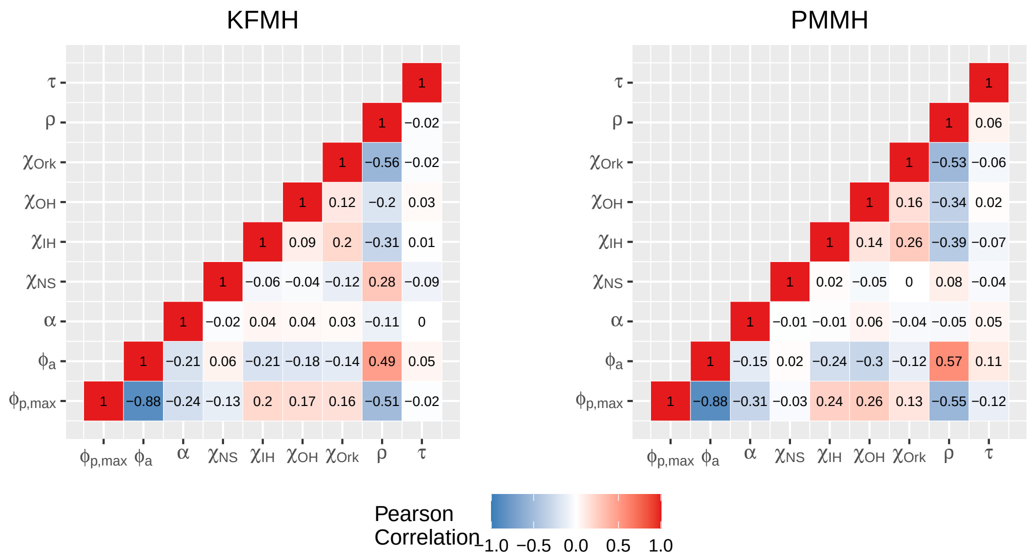


**Figure 4.17:** Without $y_{IE}$: correlations between the parameters in the posterior distributions in the complete seal model without the independent estimate, as estimated by the KFMH algorithm and the PMMH algorithm.

| Algorithm | PMMH | KFMH |
|---|---|---|
| Max. upper limit of 95% CI of $\hat{R}$ | see Table 3.11 | 1 |
| MCMC iterations | 20,000,000 | 20,000,000 (thinned by factor 10) |
| ESS | 4801.642 | 1238435 |
| Runtime (in hours) | 2775 | 64 |
| ESS/sec | $4.81 \times 10^{-4}$ | 5.37 |
| ESS/iteration | $2.40 \times 10^{-4}$ | $6.19 \times 10^{-2}$ |

**Table 4.8:** With $y_{IE}$: measures to assess the convergence speed of the KFMH algorithm and the PMMH algorithm applied to the complete seal model with independent estimate. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the mutlivariate effective sample size by ESS.

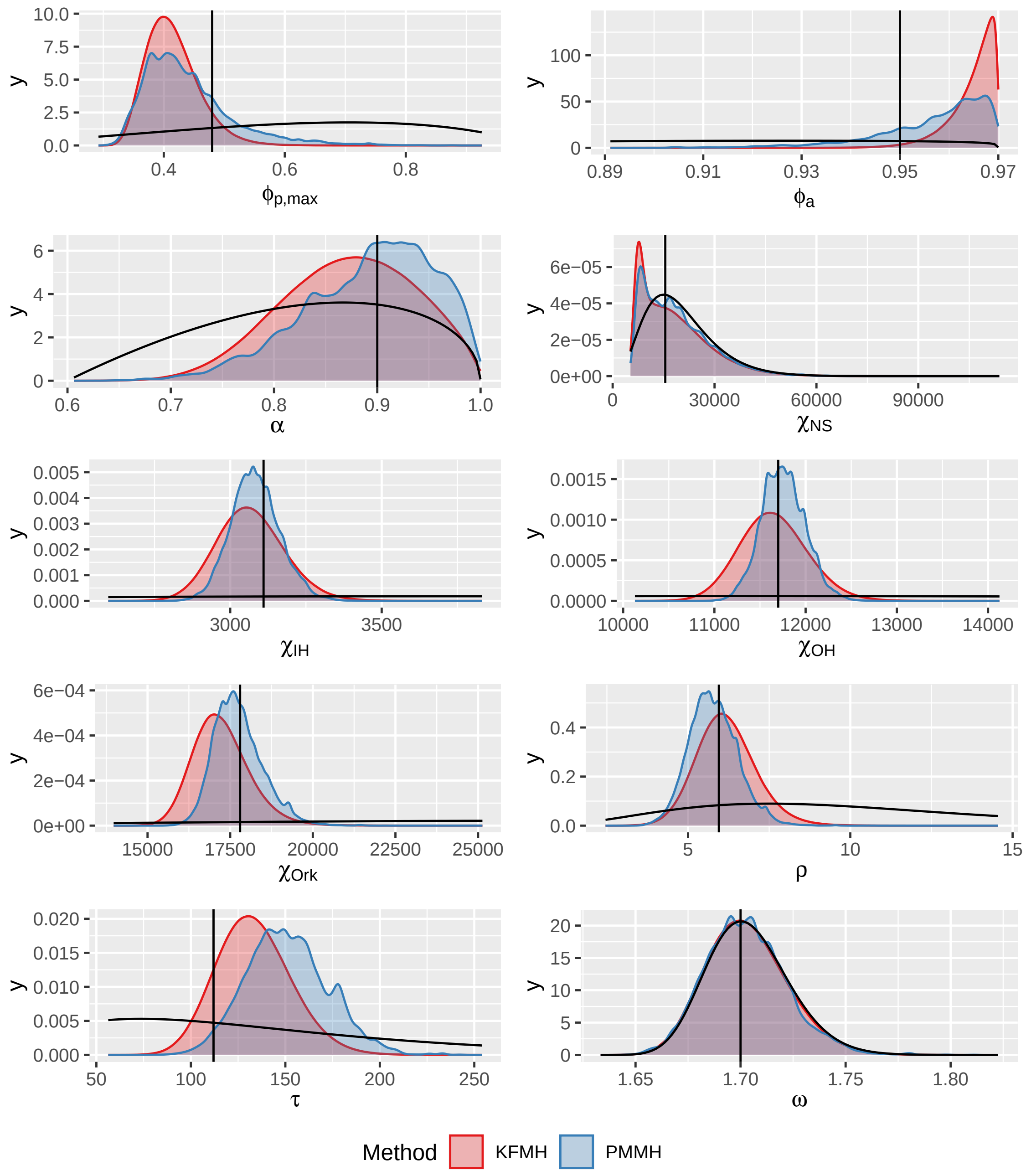


**Figure 4.18:** With $y_{IE}$: posterior distributions of all ten parameters in the complete seal model using the real data with the independent estimate, estimated by the KFMH algorithm and the PMMH algorithm. The prior distribution is drawn as a black curve. The black vertical lines indicate the posterior mean that was obtained by Thomas et al. (2019) with the Liu-West algorithm.

| Parameter | PMMH | KFMH | Liu-West |
|---|---|---|---|
| Maximum pup survival $\phi_{p,\max}$ | 0.439(0.073) | 0.415 (0.0436) | 0.478 (0.0895) |
| Adult survival $\phi_a$ | 0.957(0.011) | 0.965 (0.00467) | 0.952 (0.0129) |
| Fecundity $\alpha$ | 0.893(0.063) | 0.871 (0.0638) | 0.896 (0.0627) |
| DD shape $\rho$ | 5.78(0.746) | 6.17 (0.920) | 5.95 (1.73) |
| NS carrying cap. $\chi_{NS}$ | 18,800(9870) | 17,900 (10,000) | 15,500 (8200) |
| IH carrying cap. $\chi_{IH}$ | 3,080(80.5) | 3070 (112) | 3110 (173) |
| OH carrying cap. $\chi_{OH}$ | 11,800(238) | 11700 (372) | 11700 (534) |
| Ork carrying cap. $\chi_{Ork}$ | 17,700(734) | 17200 (856) | 17800 (1670) |
| Observation precision $\tau$ | 151(21.4) | 133 (19.7) | 111 (34.6) |
| Sex ratio $\omega$ | 1.70(0.019) | 1.70 (0.0194) | 1.7 (0.0192) |

**Table 4.9:** With $y_{IE}$: mean and standard deviation of the posterior distributions for the complete seal model with the independent estimate, as estimated by the PMMH algorithm, the KFMH algorithm, and the Liu-West algorithm (Thomas et al., 2019).

#### 4.4.2.2 Complete Seal Model With Independent Estimate $y_{IE}$

As can be seen in Table 4.8, both algorithms converged. The KFMH was run for 20,000,000 iterations, taking around 64 hours, as a precaution. However, such a long runtime does not seem to be necessary. Assuming a linear relationship between runtime and ESS, an ESS of 1000 would already be achieved after 15 minutes. Surprisingly, while the algorithm was faster per iteration in the model without the independent estimate, including the independent estimates increased the ESS per unit time. This points to the fact that the independent estimate simplified the fitting task. As in the previous section, the KFMH was much faster than the PMMH when measured as ESS per time.

The posterior distributions as seen in Figure 4.18, Table 4.9 and Figure 4.19 show that the differences between the two estimated posterior distributions were greater here than when the independent estimate was not included. The parameters $\phi_{p,\max}$, $\phi_a$ and $\alpha$ in particular exhibit a much greater difference in posterior distribution than previously, where these were the distributions that seemed the most similar. For $\phi_{p,\max}$ and $\phi_a$, the variance of the distributions was underestimated by the KFMH algorithm. A clear difference can also be seen in the estimated posterior distribution of $\tau$, where the KFMH shows a negative bias. This could be due to the non-normal distribution of the observation when conditioned on the previous observations, as shown in Figure 4.1. The correlation structure was also different between the two algorithms. The PMMH algorithm estimated a stronger negative correlation between $\phi_{p,\max}$ and $\phi_a$, whereas the KFMH algorithm estimated a stronger negative correlation between $\phi_{p,\max}$ and $\alpha$. Comparing the Kalman filter summary statistics with those estimated by the Liu-West algorithm, the advantage of the Kalman filter was less clear than in the previous section. Across all parameters, the KFMH algorithm seemed to be slightly closer on average to the posterior distribution estimated by the PMMH algorithm but this was not a clear-cut picture. A possible reason for this might be that the gamma distribution of the independent estimate is not well matched at higher moments by the normal distribution.

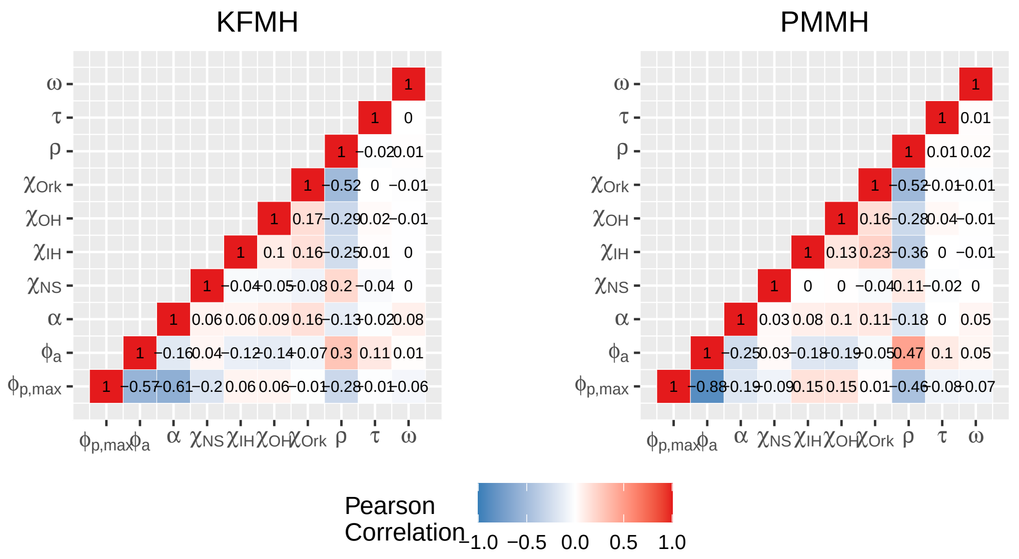


**Figure 4.19:** With $y_{IE}$: correlations between the parameters in the posterior distributions in the complete seal model with the independent estimate, as estimated by the KFMH algorithm and the PMMH algorithm.

## 4.5 Improving the Linear Approximation

We explored two ideas for improving the approximation of the posterior distribution obtained with the Kalman filter. Firstly, we attempted to improve the likelihood estimate by replacing the Kalman filter with the *unscented Kalman filter* (UKF, Wan and van der Merwe, 2001) which allows non-linear and non-Gaussian models. Secondly, we aimed to transform the approximated posterior to the true posterior by using the more general theory of SMC samplers (Del Moral et al., 2006). Neither of these approaches was successful and therefore only a brief summary of the ideas and results are given here.

### 4.5.1 Improving the Approximated Likelihood Estimate: The Unscented Kalman Filter

While the Kalman filter only allows linear and Gaussian process and measurement equations, the unscented Kalman filter is an alternative for more complex models. It calculates approximations of the expected values and covariance matrices used in the Kalman filter using so-called *sigma points*. We note that the *extended Kalman filter* is another popular extension of the Kalman filter that improves the linear approximation by using the Taylor expansion (see Harvey, 1990). In the MSc thesis of this author (Empacher, 2017), the extended Kalman filter was compared with the UKF using similar models but performed considerably worse than the UKF. It was therefore not explored in this thesis.

We first briefly outline the algorithm of the UKF as given in Wan and van der Merwe (2001). Then, we describe how the seal model needs to be transformed to fit the requirements of the algorithm and discuss some of the difficulties with this process. Lastly, some of the results of our initial exploration of this methods are given.

#### 4.5.1.1 Algorithm

The UKF relies on specifying the transition and observation processes of an SSM slightly differently than in Equations 1.1 and 1.2. Instead of specifying them as probability density functions they need to be written as deterministic functions of both the state and a random noise variable that is independent of the state. We therefore have

$$\begin{aligned} y_t &= g_t(x_t, \eta_t) \\ x_t &= f_t(x_{t-1}, \epsilon_t), \end{aligned} \tag{4.6}$$

where $\eta_t$ and $\epsilon_t$ are random variables independent of the states $x_{1:T}$ with variance-covariance matrices $P_{\epsilon_t}$ and $P_{\eta_t}$. The functions $g$ and $f$ are deterministic.

We augment the state vector with the random noise variables, leading to the augmented state vector

$$x_t^a := \begin{pmatrix} x_t \\ \epsilon_t \\ \eta_t \end{pmatrix}.$$

The distribution of the state vector $x_t^a|y_{1:t}$ is then approximated through a discrete probability distribution defined by $2L+1$ sigma points with associated probability weights, where $L$ is the length of the augmented state vector $x_t^a$. The sigma points and weights are chosen such that the distribution has the same first and second moment as the true distribution. This is done by taking the augmented state estimate and the augmented covariance matrix

$$\hat{x}_t^a := \begin{pmatrix} \hat{x}_t \\ \mathrm{E}(\epsilon_t) \\ \mathrm{E}(\eta_t) \end{pmatrix}$$

$$P_t^a := \begin{pmatrix} P_t & 0 & 0 \\ 0 & P_{\epsilon_t} & 0 \\ 0 & 0 & P_{\eta_t} \end{pmatrix},$$

where $\hat{x}_t$ is the estimate of $x_t^a|y_{1:t}$ and $P_t$ the covariance matrix of the estimate. We calculate the Cholesky decomposition (Anderson, 1958) of the augmented covariance matrix, which we write as $\sqrt{P_t^a}'\sqrt{P_t^a} = P_t^a$. Here, $\sqrt{P_t^a}$ denotes the upper triangular matrix of the decomposition. The sigma points are calculated by adding a multiple of the rows of $\sqrt{P_t^a}$ to the point estimate $\hat{x}_t^a$, so

$$\begin{aligned} \mathcal{X}_{0,t} &= \hat{x}_t^a, \\ \mathcal{X}_{i,t} &= \hat{x}_t^a + \sqrt{(L+\kappa)P_t^a}_i, & i &= 1, ..., L, \\ \mathcal{X}_{i,t} &= \hat{x}_t^a - \sqrt{(L+\kappa)P_t^a}_i, & i &= L+1, ..., 2L, \end{aligned}$$

where the subscript $i$ denotes the $i$-th row of the matrix, and $\kappa$ is a tuning parameter with $\kappa > -L$. This creates a collection of sigma points $\mathcal{X}_{0:2L,t}$ regularly dispersed around the augmented filtered point estimate. The sigma points are associated with probability weights, which are defined as

$$\begin{aligned} W_0 &= \frac{\kappa}{\kappa + L} \\ W_i &= \frac{1}{2(\kappa+L)} & i &= 1, ..., 2L. \end{aligned}$$

For this new discrete distribution defined by the sigma points with their probability weights, the expected value of the new state vector $x_{t+1|t}$ can be calculated exactly, simply by applying the transition process function to each of the sigma points and calculating the weighted mean of the resulting values. This weighted mean is then taken as the prediction for the new state vector $\hat{x}_{t+1|t}$. Similarly, the covariance matrix of the predicted state and the predicted observation $\hat{y}_{t+1|t}$ can be calculated. We omit here the details of selecting the sigma points and instead refer to Wan and van der Merwe (2001) but note that depending on the choice of a tuning parameter $\kappa$ it is possible to approximate the higher moments of the distribution. With the predicted state vector $\hat{x}_{t+1|t}$ and covariance matrix $P_{t+1|t}$ and the discrete approximation of the the distribution of $x_{t+1}|y_{1:t}$, the updating step can be calculated similarly as for the basic Kalman filter in Algorithm 8.

#### 4.5.1.2 Applying the UKF to the Seal Model

When we compare the model specifications for the UKF algorithm in Equation 4.6 and the model specifications in Equations 1.1 and 1.2, two discrepancies become apparent. First, the UKF requires the transition and observation processes to be specified as deterministic functions that take random variables as arguments. However, the transition and observation processes in an SSM are specified as probability density functions. Second, the random variables used in the UKF should have infinite support as the sigma points are generated by adding a multiple of the error's standard deviation, dependent on the tuning parameter $\kappa$, to its expected value. This only works reliably if the random variable has support $\mathbb{R}^n$. We present here a partial solution to this problem that allows the specification of pdfs with certain conditions to fit the requirements of the UKF.

Let $x_t \sim p(\cdot|x_{t-1})$ where $p(\cdot|x_{t-1})$ is any pdf. We are now looking for a random variable $\epsilon_t$ that is independent of $x_{t-1}$ and a deterministic function $f(x_{t-1}, \epsilon_t)$ such that $x_t = f(x_{t-1}, \epsilon_t)$. We would also prefer $\epsilon_t$ to have support $\mathbb{R}^n$ and ideally to be normally distributed, as most is known about the performance of the UKF in this case (see Julier and Uhlmann, 2004).

In the one-dimensional case, we suggest the following approach. Set $\epsilon_t \sim \mathrm{N}(0, 1)$. We denote with $F_d$ the cumulative distribution function of distribution $d$ and with $Q_d$ its quantile function which is $F_d^{-1}$ in the continuous case and $Q_d(q) = \inf\{x \in \mathbb{R} : q \leq F(x)\}$ else. We set

$$x_t = f(x_{t-1}, \epsilon_t) = Q_{p(\cdot|x_{t-1})}\left(F_{\mathrm{N}(0,1)}(\epsilon_t)\right).$$

Since $F_{\mathrm{N}(0,1)}(\epsilon_t) \sim \mathrm{Unif}(0, 1)$, this randomly selects a quantile of the distribution $p(\cdot|x_{t-1})$. The same approach can be used in the multidimensional case, if the probability distribution can be written as a concatenation of several one-dimensional distributions. We then simply generate a normally distributed random variable for each required sub-distribution. This is sufficient for the models used here (see Section 1.2).

A difficulty that we have not been able to solve yet is that the requirement for infinite and continuous support applies not only to the errors $\epsilon_t$ and $\eta_t$ but also to the states $x_t$. The choice of $\kappa$ can sometimes help to circumvent this problem by dispersing the sigma points close enough around the mean such that no negative values are selected. This was successful for the 2-state and 7-state model but not for the complete seal model.

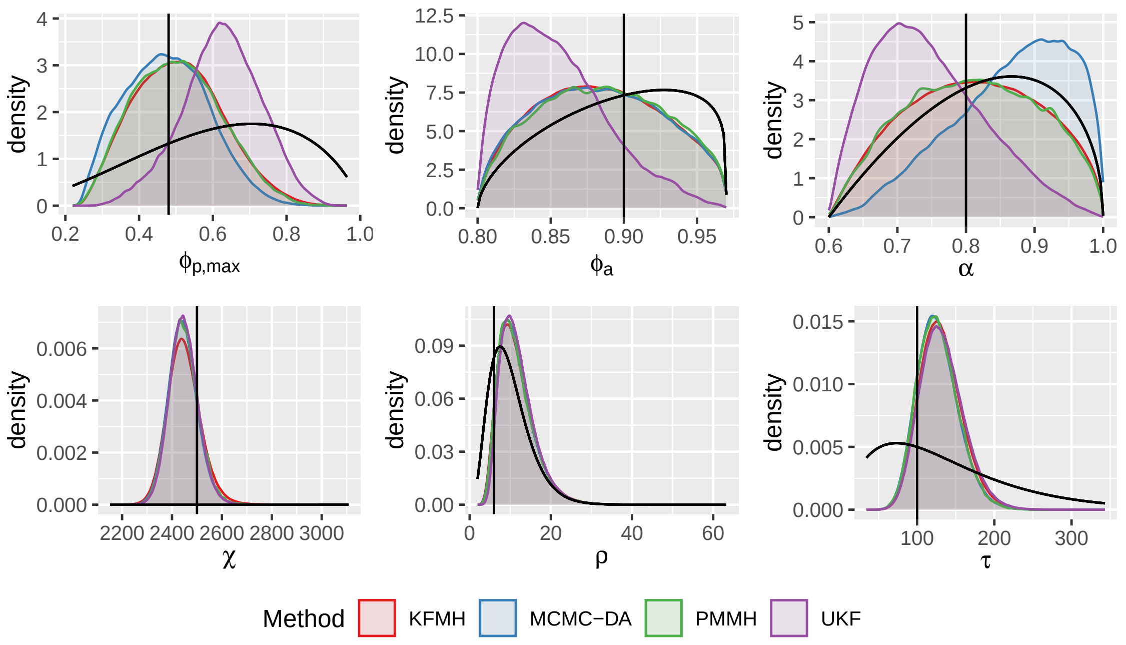


**Figure 4.20:** Marginal posterior distributions of all six parameters for simulated observations in Scenario 0 (baseline), estimated by the unscented Kalman filter (UKF) and compared with the estimated posterior distributions by the PMMH, the MCMC-DA and the KFMH algorithm. The parameter used for the simulation is indicated by the black vertical line, and the prior distribution is drawn as a black curve.

| Algorithm | MCMC-DA | PMMH | KFMH | UKF |
|---|---|---|---|---|
| Max. upper limit of 95% CI of $\hat{R}$ | 1.07 | 1 | 1 | 1.01 |
| Sample Size | 2,000,000 | 400,000 | 2,000,000 | 2,000,000 |
| Min. ESS | 443.66 | 7971.67 | 19196.71 | 938.08 |
| Min. ESS/iteration | $2.22 \times 10^{-4}$ | $1.99 \times 10^{-2}$ | $9.60 \times 10^{-3}$ | $4.69 \times 10^{-4}$ |

**Table 4.10:** Measures to assess the convergence speed of the UKF in Scenario 0 (baseline), compared to the MCMC-DA, PMMH and Kalman filter. The estimated multivariate potential scale reduction factor is denoted by $\hat{R}$ and the effective sample size by ESS.

#### 4.5.1.3 Results

In initial investigations, we attempted to estimate the posterior distribution of the 2-state model in Scenario 0 (see Section 4.3.1) with the UKF and compare the result to the other three methods (Kalman filter, MCMC-DA and PMMH). In Table 4.10, we see that the mixing of the UKF is almost as slow as with an MCMC-DA algorithm when the ESS per iteration is considered. Figure 4.20 shows that while the UKF approximation seems to work well for the parameters $\chi$, $\rho$ and $\tau$, it is not at all close to the marginal posterior distributions from the other three methods for $\phi_{p,\max}$, $\phi_a$ and $\alpha$. It might be possible to achieve a better approximation by choosing a different value for the tuning parameter $\kappa$. However, this is not a feasible approach, as achieving a better tuning parameter amounts to guessing if no true posterior is available to compare the results. Additionally, we are heavily restricted in the choice of $\kappa$ by having to select it such that no negative values of the states are generated in any of the sigma points, as discussed in the previous section. In addition to these concerns, we found that the runtime of the UKF per iteration is around 100 times as long as the runtime of the Kalman filter. Since this great increase in computation time is not in any way balanced by an increase in performance, we did not further investigate this approach.

### 4.5.2 Improving the approximated posterior with an SMC sampler

The second attempt to improve the approximation with the Kalman filter was to use the posterior distribution estimated by the KFMH and use an importance sampling scheme to correct any bias. The idea is that the approximated posterior is much closer to the true posterior than the prior is. It might therefore serve as a useful proposal distribution in an importance sampling algorithm.

We first used a standard importance sampling procedure, where the KFMH approximation of the posterior was used as the proposal distribution and a particle filter as determined optimal in Chapter 2 to calculate the importance weight. With this procedure, we found that in many cases the two distributions were already so close to each other that the Monte Carlo error introduced by importance sampling negated any positive effect in the reduction of the bias.

In a few cases with the 7-state model (not shown here), the difference between the KFMH approximation and the true posterior distribution was slightly larger. In these cases the importance sampling approach failed because for some areas of the true posterior with positive density the KFMH posterior contained very few or no samples. This led to a few sampled points with very high importance weights in the tails of the distribution and sometimes no points at all where the true posterior should have positive density.

We therefore proceeded to the more sophisticated technique of the *SMC sampler*, developed in Del Moral et al. (2006). This is a framework for algorithms that can track a sequence of related probability distributions and uses many of the ideas of standard SMC methods. It avoids some of the problems with standard importance sampling, such as a lack of support of the proposal distribution in important areas of the target distribution. The idea is to design a sequence of distributions $\gamma_k, k = 0, ..., K$ to smoothly transition from a proposal distribution—in our case the KFMH posterior—to a target distribution—in our case the true posterior—in a few steps. The general algorithm is given in Algorithm 9.

A popular strategy to design the intermediate distribution is likelihood tempering, so $\gamma_k(\theta) \propto p(y|\theta)^{\tau_k} p(\theta)$ and $\tau_k$ some sequence of exponents from 0 to 1, e.g., $\tau_k = k/K$. We used a modified version of this with

$$\gamma_k(\theta) \propto p(y|\theta)^{\tau_k} p(\theta) p_{NDLM}(y|\theta)^{1-\tau_k},$$

where the weight of the Kalman filter likelihood $p_{NDLM}$ was gradually reduced.

This procedure worked well for one unknown and biased parameter. However it still resulted in more Monte Carlo error and a much longer runtime than if the parameter had been estimated directly with a PMMH algorithm. For the higher-dimensional cases where the KFMH is much faster than a PMMH, the Monte Carlo error of the SMC sampler was too high to be able to detect any reduction of the bias. While there are many tuning choices when using an SMC sampler—like the sequence of intermediate distributions, the MCMC

**Algorithm 9** SMC Sampler

```
for $i \leftarrow 1, ..., N$ do                                    ▷ Initialisation
    Sample $\theta_0^{(i)}$ from $\gamma_0$
    Set weights $w_0^{(i)} = 1/N$
end for
for $k \leftarrow 1, ..., K$ do
    for $i \leftarrow 1, ..., N$ do
        Set weights $w_k^{(i)} = w_{k-1}^{(i)} \frac{\gamma_k(\theta_{k-1}^{(i)})}{\gamma_{k-1}(\theta_{k-1}^{(i)})}$
    end for
    if ESS too low then
        Resample and set $w_k^{(i)} = 1/N$ for all $i = 1, ..., N$
    end if
    Sample $\theta_k^{(i)}$ from MCMC kernel with stationary distribution $\gamma_k$
end for
return weighted sample $\{(\theta_k^i, w_k^i)\}_{k=1,...,N}$ from $\gamma_K$
```

kernel and the number of particles—we were not able to find settings that produced a visible improvement of the bias in feasible computation time.

## 4.6 Discussion and Conclusion

The KFMH algorithm applied to the NDLM approximation of the true model performed in general quite well. It easily outperformed both an MCMC with data augmentation and the PMMH algorithm in terms of speed. In many but not all cases, the approximated posterior distributions were very close to the true posterior distribution. A particular strength of the algorithm is that it is very robust to challenging fitting problems as demonstrated with the results for Scenario 2 (edge of parameter support) and 3 (mis-specified model). As was demonstrated in Chapter 3 in Section 3.4, it is possible to develop methods that generate samples from the true posterior distribution. However, these methods require often unfeasibly long runtimes or bespoke algorithms with specific hardware which cannot easily be generalised. The benefit of the KFMH algorithm is that it is relatively easy to implement and fast to run. In cases where extensive computing time or fitting expertise is not available, it provides an inviting alternative. There might also be cases where an exact solution is less important than a fast one. For example, the KFMH algorithm could be employed in a model selection process where many alternatives need to compared. In many practical applications, the survey set-up might introduce more uncertainty into any estimates than the small bias in the Kalman filter. Apart from practical considerations, the approximated posterior distribution generated by the KFMH algorithm might be useful as an intermediate step to get to the exact distribution. The algorithm very quickly provides a rough estimate of the location of the posterior distribution. This might help focus the effort of computationally more expensive methods such as SMC methods on the important areas of the posterior distribution. However, in the two approaches considered here with the UKF and the SMC sampler, we were not able to further improve the approximation.

# Chapter 5

# Effect of process and observation errors on inference in a nonlinear non-normal SSM

## 5.1 Introduction

While the other chapters of this thesis focus on the methods for producing samples from the posterior distribution of the parameters and the filtering or smoothing distributions of the states, here we investigate some of the aspects of the posterior distribution of both the states and the parameters. In particular, we noticed a curious phenomenon when analysing the posterior distribution of the 2-state model. We would usually expect the posterior distribution to move closer towards the true parameter value, starting from the prior distribution. In Gelman et al. (2013), this is described as the posterior reaching a compromise between the given data and the prior distribution. However, as seen in Figure 5.1 and Table 5.1, the mean of the marginal posterior distribution (blue vertical line) moves further away from the prior mean than the true value for three of the parameters, i.e., $\phi_a$, $\tau$ and $\chi$, rather than staying between the prior mean and the true value. We also notice that there seems to be very little learning about some of the parameters when looking at the marginal posterior distributions, especially $\phi_a$ and $\alpha$, where the variance is larger than that of the induced prior distribution.

| | $\phi_{p,\max}$ | $\phi_a$ | $\alpha$ | $\chi$ | $\rho$ | $\tau$ |
|---|---|---|---|---|---|---|
| Prior | 0.643 (0.184) | 0.900 (0.0423) | 0.831 (0.0928) | 3200 (1600) | 10 (5) | 140 (96.6) |
| Posterior | 0.507 (0.121) | 0.892 (0.0425) | 0.804 (0.0943) | 2426 (64.6) | 9.20 (4.23) | 89.1 (18.5) |
| True value | 0.480 | 0.900 | 0.800 | 2500 | 6.00 | 100 |

**Table 5.1:** Means (with standard deviations in parentheses) of the posterior and induced prior distributions of the 2-state model, as well as the true parameter values used to simulate the data.

We note that the sample used here was produced by applying the PMMH (particle marginal Metropolis-Hastings) algorithm with the settings as in Table 3.5 to the same simulated data used there and running 24 chains with 2 million iterations each, leading to an ESS of 90100.07. This phenomenon is therefore unlikely to be due to Monte Carlo error but rather a real feature of the posterior distribution.

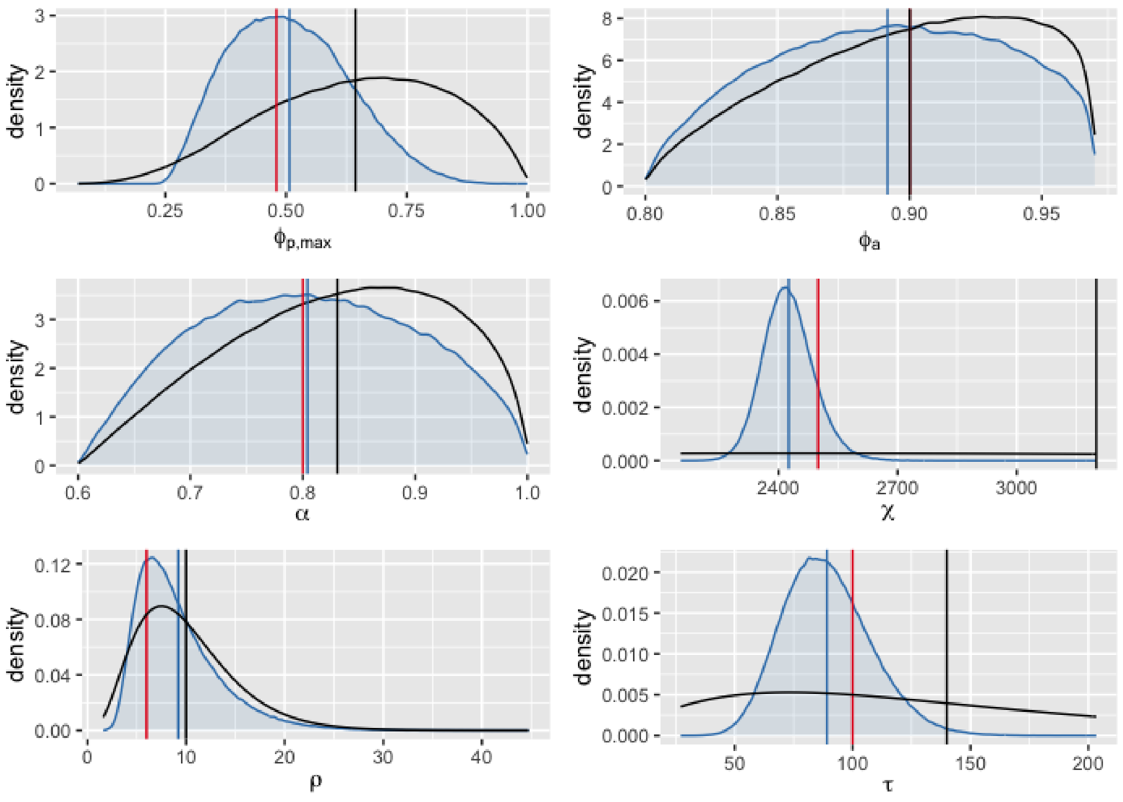


**Figure 5.1:** Posterior (blue) and induced prior (black) densities for the 2-state model. The true parameter values used to simulate the data are indicated by the vertical red line (🟥), the prior means by a vertical line in black (■), and the posterior means by a vertical line in blue (🟦). For $\phi_a$, the red line indicating the true parameter value is hidden behind the black vertical line.

When communicating our results, we often report the mean parameter values as summary statistics. It is therefore important to find out if this measure is appropriate compared to the true parameter value or biased in any way and if the observed behaviour of the mean moving beyond the true value is systematic. We also want to explore if more and better data might help to better be able to recover the true parameter value and reduce the variance in the posterior distribution. Studying this is a difficult task with the complete model because fitting is so computationally expensive. We therefore use only the 2-state model, albeit with some extensions, for these explorations in this chapter.

First, we discuss convergence properties of the posterior distribution. Even though these only hold in the limit as the number of i.i.d. data points $n \to \infty$, they can still help our understanding when an application with a finite sample size is considered. We also discuss the results of Auger-Méthé et al. (2016) where similar behaviours were observed when analysing the posterior distribution of a linear Gaussian SSM.

Next, we define the model and data we are using to study this behaviour of the posterior distribution. The starting point is the 2-state model as introduced in Section 1.2.6. However, we increase the length of the observed time series and include further observations about both age classes. We also vary the size of the observation error to explore its effect on the bias and variance of the posterior distribution. Lastly, a random effect is introduced into the transition process to increase its variance and compare the effect to the changes produced by varying the observation error.

The effect of these four changes to the model are explored in two simulation studies. In the first, the observation error is varied while the information in the data remains otherwise unchanged. We find that even with a very low observation error and observations of both age classes, some estimation problems remain. In the second study, we simulate many time series and estimate the posterior distribution for each of these and for varying sizes of random effect variance. This reveals the systematic trends in the estimation of the posterior mean and other summary statistics.

## 5.2 Methods

### 5.2.1 Previous Results

We begin by citing some of the relevant convergence theorems in Bayesian statistics and discussing their relevance to the observed issue when estimating parameters in the seal model. We discuss posterior variance, properties of different point estimators and the notion of posterior consistency in general. Then we lay the foundation for the remaining chapter by discussing the effect of having more or "better" data in the case of the seal model, and summarising the work in Auger-Méthé et al. (2016) for NDLMs, which we emulate for the non-linear non-Gaussian case.

First, we look at the variance of the posterior distribution. We note that the law of total variance (e.g., Proposition 5.2 in Ross, 2011) states that, under some mild conditions,

$$\mathrm{Var}(\theta) = \mathrm{E}(\mathrm{Var}(\theta|y)) + \mathrm{Var}(\mathrm{E}(\theta|y)).$$

In particular, this means that from a Bayesian perspective, the variance of the posterior distribution $p(\theta|y)$ is on average smaller than that of the prior distribution. This matches our intuition that having more data available should lead to less uncertainty about the distribution of a parameter. However, this is only true in the expectation, and for some data $y$, as seen for $\phi_a$ and $\alpha$ in the motivating example in Table 5.1, the variance can increase.

Second, we look at point estimates $\hat{\theta}(y)$ for the parameter $\theta$. While unbiasedness, i.e., $\mathrm{E}\left(\hat{\theta}(Y)|\theta\right) = \theta$ for any $\theta \in \Theta$ with the expectation taken with respect to the data $Y \sim p(y|\theta)$, might seem like an appealing property for an estimator, insisting on this can lead to bad estimators, as discussed in, e.g., Bickel and Blackwell (1967) and Reich and Ghosh (2019), and the trade off between variance and bias needs to be considered. Using the mean of the posterior distribution $\hat{\theta}(y) = \mathrm{E}(\theta|y)$ as a point estimate for $\theta$ is common and is the Bayes rule estimator under squared error loss (Reich and Ghosh, 2019), i.e., it minimizes the posterior mean square error $\mathrm{E}_\theta((\hat{\theta}(y) - \theta)^2)$. Other common estimators are the median, which minimises the absolute loss $\mathrm{E}_\theta(|\hat{\theta}(y) - \theta|)$, and the mode.

Another aspect we might be interested in is the behaviour of the posterior distribution as more and more data are used to estimate it. For independent and identically distributed data $y = (y_1, ..., y_n)$, the following theorem can be established (see Appendix B in Gelman et al., 2013 for a proof).

**Theorem 2.** *If $\theta$ is defined on a compact set and $A$ is a neighborhood of $\theta_0$ with nonzero prior probability, then $\mathbb{P}(\theta \in A) \to 1$ as $n \to \infty$, where $\theta_0$ is the unique value of $\theta$ that* *minimizes the Kullback-Leibler divergence $KL(\theta)$ of the distribution $p(y_i|\theta)$ relative to the true distribution $f(y_i)$*

$$KL(\theta) = \mathrm{E}\left(\log\left(\frac{f(y_i)}{p(y_i|\theta)}\right)\right).$$

In particular, this means that if the true distribution of $y_i$ is of the form in the model, i.e., $f(y_i) = p(y_i|\theta_0)$ for some value $\theta_0$ (and the minimizing value of $\mathrm{KL}(\theta)$ is unique) then the posterior distribution converges to a point mass at the true parameter. This is the case here, where data are simulated from the model with a known true parameter value.

In the seal model, it is unrealistic to obtain $n$ i.i.d. copies of the data. This would mean observing $n$ copies of seal populations under the exact same conditions and recording a time series of observations $y_{1:T}$ for each of them. A more realistic way to obtain more data is to increase the number of observations by increasing the length of the time series $T$ and by making observations on more components of the state vector $x_t$, e.g., by recording an estimated count for each age class. This way, the data are of course not independent and identically distributed but could still lead to less uncertainty in the posterior distribution.

Moving from general Bayesian statistics to the special case of parameter estimation in SSMs, we refer to the work in Auger-Méthé et al. (2016). The model and estimation framework there is different from our work, but they observe similar behaviour in their estimators. The model under consideration in that paper is linear and Gaussian which simplifies the estimation process considerably (see Chapter 4). In contrast to the work in this thesis, they work within a frequentist framework and therefore use the maximum likelihood estimator (MLE) to check for the size of the bias. While it is unsurprising that the MLE is biased, it has the property of consistency and so converges in probability to the true parameter value $\theta_0$ as the number of i.i.d. data points tends to infinity (see Section 6 in Chernoff, 1972). As argued above, it is unrealistic to obtain observations of multiple i.i.d. time series. Instead, they studied how the ratio of observation error to variance in the transition process affects the size of the bias of the MLE. The study found that with decreasing process stochasticity and constant observation error the bias in the parameter estimates increased. This also affected the quality of the state estimates. This was measured by calculating the root mean square error (RMSE) of the state estimates

$$\mathrm{RMSE}(\hat{x}_{1:T}) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{x}_t - x_t)^2},$$

where $x_t$ is the true state and $\hat{x}_t$ is the mean of the smoothed state estimates at time $t$, and comparing this value when the states were estimated using the MLE with when the true parameter values were used. Using the MLE led on average to a much higher error than using the true parameter value. While the error decreased when the process stochasticity was decreased, the discrepancy between using the true parameter value and the MLE grew for these cases.

Through a simulation study, we aim to determine whether the same issues arise in a Bayesian context using the more complex non-linear and non-Gaussian seal model.

### 5.2.2 Changes to the Model

We want to explore the behaviour of the posterior distribution and any point estimates derived from it when more and better data are available. We also want to determine whether the estimation problems described in Auger-Méthé et al. (2016) are present in our model when the ratio of observation error to process stochasticity is decreased. Since it is unreasonable to assume obtaining many i.i.d. copies of the observed time series, we look for an alternative that is closer to reality.

#### 5.2.2.1 Changing $T$ and Observation Vector Length

As in the motivating example, we use the 2-state seal model to simulate data and to estimate the posterior distribution. We first reduce the complexity of the estimation problem by reducing the number of unknown parameters from six to four. In practice, it is common to assume the observation precision $\tau$ to be known, e.g., by externally estimating it. It was also shown in Knape (2008) that estimating this parameter when the strength of density dependence and the process error variance are unknown can lead to identifiability issues. Next, we assume that the density dependence shape parameter $\rho$ is known to simplify estimation of the remaining parameters. This enables us to better focus on the effect of changing the error ratios of the model. Additionally, the length of the time series is increased to $T = 60$ which might help with the estimation of the parameters. The last step to increase the quality of the data is to include observations of the number of adults in every year. This requires some decisions about how to model these observations. In the complete seal model, the observations of the number of adult seals are modelled with a shifted Gamma distribution (see Section 1.2.3.4):

$$\kappa_{0,24} \sim x_{adults,24} - \mathrm{Ga}(\kappa_{1,24}, \kappa_{2,24}),$$

where $\kappa_{0,24}$ is treated as the observation, and the shape parameter $\kappa_{1,24}$ and scale parameter $\kappa_{2,24}$ are treated as known parameters.

When extending this to the entire time series and simulating adult observations $\kappa_{0,1:T}$, we need to decide what the shape and scale parameters should be for each time point. The uncertainty in the pup observations is modelled by using a constant CV $c$ for the observation error. We adopt the same choice for the adult observations, which means that the variance of the observations is

$$\kappa_{1,t}\kappa_{2,t}^2 = (cx_{2,t})^2.$$

This still leaves some freedom for choosing $\kappa_{1,t}$ and $\kappa_{2,t}$. Here, we decide to use the same shape parameter $\kappa_1$ as in the complete seal model for all time points, and adapting the scale parameter to achieve the correct variance. This leads to a relatively symmetric distribution (see Figure 4.5 for a comparison with the normal distribution). We note that, as discussed in Section 1.2.3.4, the change of the mean $\kappa_{1,t}\kappa_{2,t}$ of the Gamma distribution is irrelevant, as this mean is known and simply results in a (known) shift of the observation $\kappa_{0,t}$.

Using adult observations as described in addition to the pup observation, we have a comprehensive data set comprising information about all components of the state in every year for a much longer time series.

#### 5.2.2.2 Changing Observation Variance

Similar to Auger-Méthé et al. (2016), we want to explore the effect of changing the observation variance to process stochasticity ratio and test whether their theory that a higher ratio leads to a larger bias is true for our model. In the seal model the transition process consists of several binomial distributions. These have a variance of $p(1-p)n$ where $p$ is the success probability and $n$ is the number of trials. This variance can therefore not be altered without changing the success probability or the number of seals. We discuss possibilities to vary the process stochasticity in the next section, but first focus on the observation error instead. If it is indeed the ratio between the two variances that affects the size of the bias, varying only the observation error should allow us to observe the same effects.

Changing the precision parameter $\tau$ for the pup observations is straightforward and does not affect any other part of the model while having the same effect of changing the ratio of observation to process error. We therefore use observations with a CV of $c = 0.1, c = 0.01$ and $c = 0.005$, corresponding to $\tau = 100$, $10,000$ and $40,000$. We note that using an even higher precision was not possible because the estimation algorithm no longer converged.

For the adult observations, the shape parameter $\kappa_1$ is kept constant across all time points and choices of CV. Any change of the CV is therefore reflected in a change of the scale parameter $\kappa_{2,t}$ which is calculated by

$$\kappa_{2,t} = \frac{cx_{2,t}}{\sqrt{\kappa_1}}.$$

Varying only the observation error rather than the process stochasticity allows us to use the same states $x_{1:T}$ for all three cases which makes the results between the different observation errors more comparable. We want to further increase comparability by eliminating the difference between the three simulated observation time series as much as possible from the simulated data. To achieve this, we simulated observations from the model only for the time series with $\tau = 100$. For the other two time series, "quantile matching" was used. This means that each observed value was shrunk down towards the true state value such that the cumulative distribution function of the observation density $F_c$ was the same for all three observation time series across all time points $t$:

$$F_{c=0.1}(y_t(c=0.1)|x_t) = F_{c=0.01}(y_t(c=0.01)|x_t) = F_{c=0.005}(y_t(c=0.005)|x_t),$$

where $y_t(c = 0.1)$ refers to the observation at time $t$ with a CV of 10%. The resulting states and (shifted) observations can be seen in Figure 5.2.

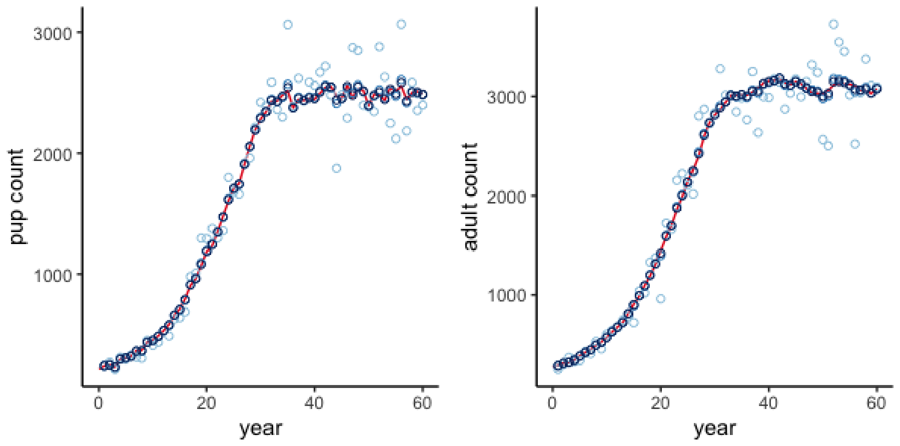


**Figure 5.2:** Simulated pup and adult numbers (🟥 line) with circles for the observations with a CV of 10% (🟦), 1% (🟦) and 0.5% (🟦)

#### 5.2.2.3 Changing Process Stochasticity

While changing the process error in the seal model cannot be done by simply altering a variance parameter, the model can be adapted slightly to allow for more flexibility in the process error. One possibility for this is to include a random effect on one of the demographic parameters. Here, we choose the fecundity parameter $\alpha$ to vary across time. This is a sensible choice from a biological perspective becayse grey seals are capital breeders and rely on acquired fat stores accumulated through the year before a breeding attempt. Poor environmental conditions in a year can lead to decreased resource availability and a resulting lower population fecundity (Smout et al., 2020). It was even expressed in Thomas and Harwood (2003) that being able to add such a random effect would be desirable because it might more accurately reflect the true seal population dynamics and fit the data better.

There are a few options to implement this random effect. A standard choice is letting $\alpha$ vary according to a normal distribution, that is, $\alpha \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$. The advantages of the normal distributions are its intuitive interpretation and the clear separate parametrisation of variance and mean. It is also symmetric around the mean which makes it easy to construct the new values of $\alpha_t$ around the existing value of $\alpha$ by adding normally distributed errors to it. However, the normal distribution does not restrict $\alpha_t$ to be between 0 and 1 and is therefore not a good choice, as $\alpha_t$ is the success probability in a binomial distribution. While a logit-normal distribution, i.e., $\text{logit}(\alpha) \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$, solves this domain issue, it loses the intuitive interpretation of the parameters. It does not even allow the analytic calculation of mean and variance, making it difficult to calibrate the distribution to have the original fecundity parameter $\alpha$ as its mean. A more natural choice for the distribution of a probability is the Beta distribution $\text{Beta}(a, b)$ which is naturally restricted to values between 0 and 1. In the implementation, it even allows us to integrate over $\alpha_t$ and draw the new number of pups directly from a Beta-binomial distribution, so $x_{1,t} \sim \text{BetaBin}(x_{2,t-1}, a, b)$. We note that in the seal model the prior of $\alpha$ restricts it to lie between 0.6 and 1. While we could reflect this in the random effect by choosing a shifted and scaled Beta distribution, this would add two further parameters to the model and break the connection to the Beta-binomial distribution. Instead, we can place priors on the Beta distribution shape parameters $a$ and $b$ such that $\alpha_t$ lies between these two values with a high probability. For example, when simulating values for $\alpha$ with priors

$$\begin{aligned} a &\sim \text{Ga}(40, 0.1) \\ b &\sim \text{Ga}(40, 0.4) \end{aligned}$$

we found that 99.9946% of values lie between 0.6 and 1. For the simulation of the time series, we set $a$ and $b$ such that the mean value of $\alpha$ is 0.8 which corresponds to the fixed value used in the 2-state model. To increases the stochasticity of the process, we varied only the variance of the distribution and calculated $a$ and $b$ accordingly. In Figure 5.3, we show the distribution of $\alpha$ when $a = 400$ and $b = 100$, which results in a mean value of 0.8 of $\alpha$ with a standard deviation of 0.0179. It also results in a relatively symmetric distribution of $\alpha$ with a skewness of -0.134. The figure also compares the distribution of $\alpha$ when it is sampled from the (induced) prior and when it is sampled from $\text{Beta}(a, b)$ with fixed and random $a$ and $b$.

For the simulation study, we selected four different standard deviation values for the random effect on $\alpha$ and calculated $a$ and $b$ such that the mean of $\text{Beta}(a, b)$ was 0.8 and the standard deviation as selected (see Table 5.2). In addition, we compared the results

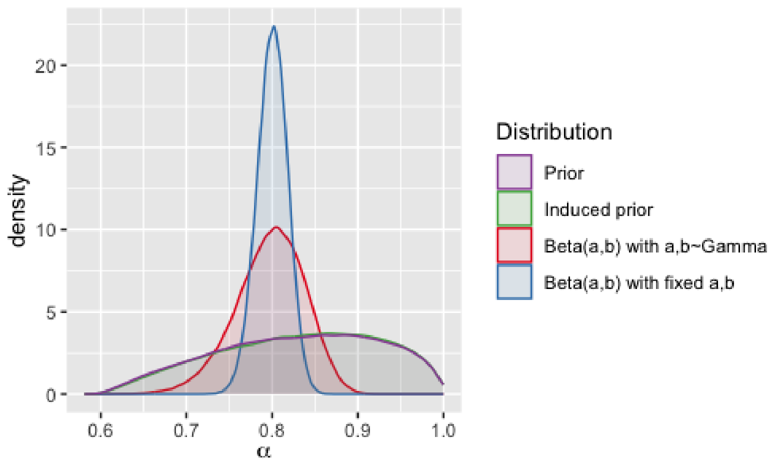


**Figure 5.3:** Distributions of $\alpha$ in different model formulations. 🟪 shows the prior density in the seal model $\alpha \sim 0.6 + 0.4\text{Beta}(2, 1.5)$, 🟩 shows the prior induced by the Beverton-Holt density dependence. 🟥 shows the distribution of $\alpha$ as a random effect, with $\alpha \sim \text{Beta}(a, b)$ with priors $a \sim \text{Ga}(40, 0.1)$ and $b \sim \text{Ga}(40, 0.4)$. 🟦 shows the distribution of $\alpha$ as a random effect $\alpha \sim \text{Beta}(a, b)$ when the shape parameters are fixed to $a = 400$ and $b = 100$.

| SD($\alpha$) | $a$ | $b$ |
|---:|---:|---:|
| 0 | - | - |
| 0.01 | 1279.2 | 319.8 |
| 0.03 | 141.4 | 35.4 |
| 0.10 | 12.0 | 3.0 |
| 0.20 | 2.4 | 0.6 |

**Table 5.2:** Parameter values of $a$ and $b$ for a Gamma distribution for $\alpha$, with mean 0.8 and varying standard deviation.

from the four different standard deviations with results from simulations where there was no random effect on $\alpha$. Figure 5.4 shows the distribution of $\alpha \sim \text{Beta}(a, b)$ when $a$ and $b$ are set as in Table 5.2. The observation CV was fixed and set to 10% across all simulations here.

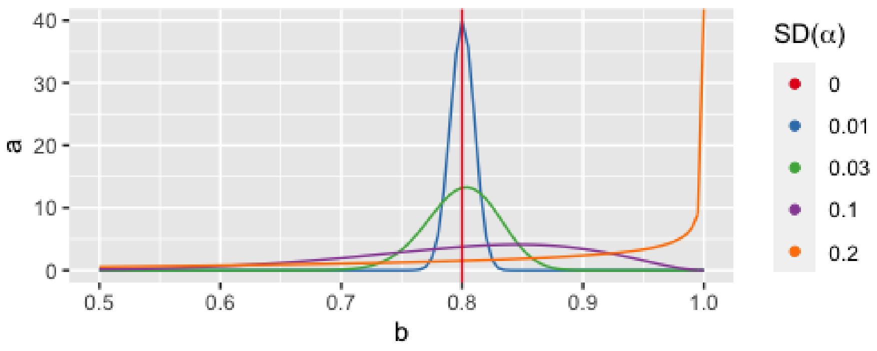


**Figure 5.4:** Density of the distribution of the random effect on $\alpha$ when $\alpha \sim \text{Beta}(a, b)$ with $a$ and $b$ as in Table 5.7.

For the parameter estimation, we assumed $a$ and $b$ (and $\alpha$ for the simulations with no random effect) to be known. Only the three parameters $\phi_{p,\max}$, $\phi_a$ and $\chi$ were estimated. As $a$ and $b$ vary across the simulations, fixing these to the true value allows a better comparison between the posterior estimates of the other three parameters.

Lastly, we need to consider how the pup survival probability is affected by replacing a fixed $\alpha$ with a random effect. As described in Equations 1.4 and 1.5, the pup survival probability $\phi_{p,t}$ in year $t$ is a function of the current number of pups and all model parameters but $\tau$. We therefore need to decide how to model this probability when $\alpha_t$ varies in each year. Updating $\phi_{p,t}$ each year based not only on the current number of pups but also on the current value for $\alpha_t$ does not work because the formula for the Beverton-Holt density dependence does not allow some combinations of $\alpha_t$, $\phi_{p,\max}$ and $\phi_a$. When sampling parameter values from the prior in the seal model, we can pre-select valid combinations, resulting in the induced priors discussed in Section 1.2.3.6. This is no longer possible when placing a random effect on $\alpha$. We therefore fix the value that is used to calculate the annual pup survival probability to the mean of $\alpha \sim \text{Beta}(a, b)$ given the shape parameters $a$ and $b$. The resulting state trajectories for two different distributions for $\alpha$ (in blue and green) can be seen in Figure 5.5, compared with a trajectory without a random effect (in red).

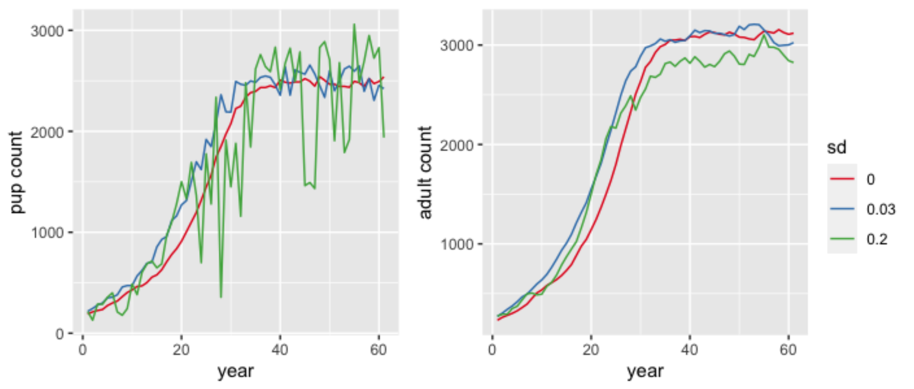


**Figure 5.5:** Simulated trajectories of pup and adult numbers with and without (🟥) a random effect on the fecundity $\alpha$. For the two simulations with a random effect, the parameters of the Beta distribution for $\alpha$ were chosen such that the mean of $\alpha$ was 0.8, and the standard deviation was 0.03 (🟦) or 0.2 (🟩).

Rather than varying both the observation error and the process error at the same time, we studied the effect of these two changes separately. Fitting the model for very low observation errors requires a lot of computation power, as it does for very high variances of the random effect. We therefore set up two different simulation studies for the two sources of error but note that combining these and studying interaction effects is a possible route for further research.

### 5.2.3 Diagnostic Tools

#### 5.2.3.1 Changing Observation Variance

We first confirmed that the Markov chains had converged, using the tools described in Section 3.2.2.1. To analyse the resulting posterior distributions, we considered three aspects. First, the marginal distributions of the parameters were examined. Here, we relied on density plots and compared means and standard deviations with the true parameter value. We also calculated the percentage of overlapped area between some of the distributions (Pastore and Calcagnì, 2019). The position of the true parameter value within the three posterior distributions (representing the three levels of observation error examined) was quantified by evaluating the empirical cumulative distribution function at the true parameter value. This is equivalent to calculating the proportion of sampled values that are smaller than the true value:

$$\hat{F}(\theta_i) = \frac{1}{M} \sum_{m=1}^{M} \mathbb{1}_{\theta_i^{(m)} < \theta_i}.$$

Here, $\theta_i$ is the $i$-th component of the true parameter value, and $\theta_i^{(m)}$ is the $i$-th component of the $m$-th sampled value from the posterior distribution.

Second, the joint distributions of all parameter component pairs was examined. We used scatter plots and the Pearson correlation coefficient to study the multivariate distribution of the parameters. To assess how far away the true parameter value $\theta$ was from the estimated posterior distribution, we calculated the Mahalanobis distance (Mahalanobis, 1936) between these. This is defined as

$$d_M(\theta, Q) = \sqrt{(\theta - \mu)\Sigma^{-1}(\theta - \mu)},$$

where $Q$ is the distribution in question with mean $\mu$ and covariance matrix $\Sigma$.

Finally, the posterior state estimates were considered. Often, parameter estimation is only a necessary intermediate step for making state inference. It is therefore important to know whether difficulties in recovering the true parameter value with a point estimate affects the quality of the state estimates. To sample from the smoothed posterior state distribution $p(x_{0:T}|y_{1:T})$, we first sampled $\theta_i$ for $i = 1, ..., 100$ from the posterior distribution. We then ran a bootstrap filter for each sampled parameter value $\theta_i$ to generate smoothed state estimates from $p(x_{0:T}|y_{1:T}, \theta_i)$ with 1000 particles and randomly selected one of the 1000 state trajectories for each of the parameter values. The high number of particles was necessary for two reasons. First, as discussed in Section 2.2.2, a particle filter is an importance sampling method and therefore only asymptotically unbiased. Using a high number of particles reduces this bias. Second, here the naïve method of sampling from the smoothed distribution was used, where we saved the entire state trajectory for each particle and then sampled from these trajectories. This can lead to particle degeneracy for the early time points but can be mitigated by using enough particles. More sophisticated methods are available for sampling smoothed state estimates, as briefly discussed in Section 2.2.8, but we found that here the basic approach produced satisfactory results.

Similar to Auger-Méthé et al. (2016) with the RMSE, we assessed the quality of the state estimates by calculating a measure to quantify the error of the estimated state values compared to the true simulated values. To transfer the idea of the RMSE to the 2-state model, we considered two things. First, the state $x_t$ consists of a pup count and an adult count of potentially different orders of magnitude. Second, using a constant CV leads to the observation error increasing as the states increase. This means that simply summing the unscaled squared errors disproportionally attaches importance to the larger errors towards the end of the time series. We therefore scaled the errors by dividing each difference $\hat{x}_{age,t} - x_{age,t}$ by the true state $x_{age,t}$. This also means that errors for pup and adults could be compared on the same scale. The error measure is then a scaled root mean square error

$$\mathrm{SRMSE}(\hat{x}_{1:T}) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\left(\frac{\hat{x}_{1,t} - x_{1,t}}{x_{1,t}}\right)^2 + \left(\frac{\hat{x}_{2,t} - x_{2,t}}{x_{2,t}}\right)^2\right)}.$$

#### 5.2.3.2 Changing Process Stochasticity

We examined the marginal posterior distributions, the joint posterior distributions and the posterior state estimates with the same tools as described in the previous paragraph. However, when examining the effect of varying the process stochasticity on the posterior distribution, we could no longer use the same underlying time series of states, as this changes when the variance of the random effect changes. It was therefore no longer possible to see any trends by only examining one posterior distribution for each of the selected variances, as any differences could be due to the different states and observations. Therefore, we simulated states and observations 50 times for each of the selected variances and present summaries of these 50 runs. This allows us to see if any estimation problem are systematic. Additionally, the results from one randomly selected simulation for each variance value are shown to illustrate some features of the posterior distributions.

## 5.3 Results

### 5.3.1 Changing Observation Variance

#### 5.3.1.1 Marginal Distributions of Parameters

We first confirmed that the samples produced by the PMMH algorithm have low enough Monte Carlo error to be reliable for further analysis. Table 5.3 summarizes the diagnostics of the MCMC. The chains seemed to have converged according to the $\hat{R}$-value and examination of the trace plots (not shown). The ESS was large enough that further analysis is feasible.

| CV | $M$ | $\hat{R}$ | ESS |
|---|---|---|---|
| 10% | 2,000,000 | 1.00 | 120174.61 |
| 1% | 2,000,000 | 1.00 | 71451.42 |
| 0.5% | 2,000,000 | 1.00 | 35426.74 |

**Table 5.3:** Summaries from running the PMMH for the 2-state model with four unknown parameters with three different observation CVs.

Examining the marginal densities of the four unknown parameters with their means and standard deviations, we see in Table 5.4 and Figure 5.6 that there were clear differences between the three observation variances. The marginal densities for $\alpha$ and $\chi$ were very similar when comparing the posterior distributions for CV=1% with CV=0.5% (overlapped area 98.23% respectively 99.01%) but still showed a clear difference for the other two parameters (88.42% overlap for $\phi_{p,\max}$ and 87.82% overlap for $\phi_a$).

Even with the lowest CV of 0.5%, there was still a lot of uncertainty for some of the parameters. For all four parameters the variance of the marginal posterior distributions was lower than in the motivating example with only $T = 26$ observations of pup counts and 6 unknown parameters. For three of the four parameters, the standard deviation was decreased by about a factor of 2—from 0.121 to 0.065 for $\phi_{p,\max}$, from 0.0425 to 0.0263

| CV | $\phi_{p,\max}$ | $\phi_a$ | $\alpha$ | $\chi$ |
|---|---|---|---|---|
| Prior | 0.643 (0.184) | 0.900 (0.0423) | 0.831 (0.0928) | 3200 (1600) |
| 10% | 0.544 (0.095) | 0.876 (0.0385) | 0.808 (0.0142) | 2499 (41.5) |
| 1% | 0.562 (0.075) | 0.867 (0.0302) | 0.796 (0.0102) | 2463 (33.7) |
| 0.5% | 0.580 (0.065) | 0.859 (0.0263) | 0.796 (0.0101) | 2462 (33.5) |
| True value | 0.480 | 0.900 | 0.800 | 2500 |

**Table 5.4:** Means (with standard deviations in parentheses) of the prior distribution of the 2-state model, as well as the the posterior distributions estimated using observations with three different CVs and the true parameter values used to simulate the data.

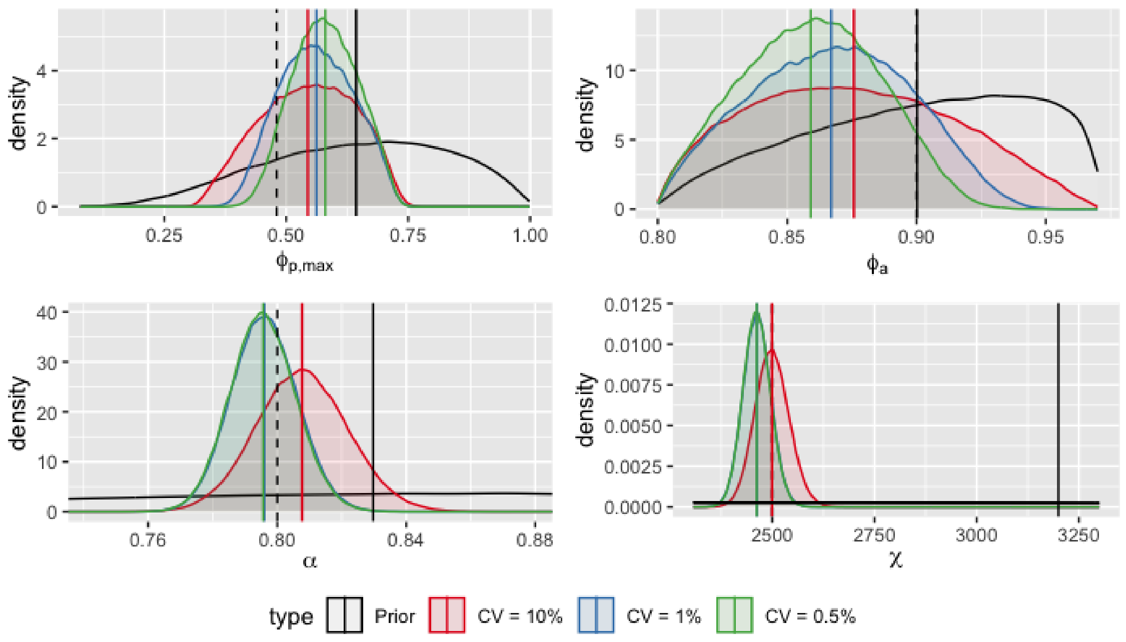


**Figure 5.6:** Marginal posterior distributions and means (vertical lines) for the 2-state model, estimated using observations with three different CVs. The marginal prior distributions and means are indicated in black. The true value used to simulate the data is shown as the vertical black dashed line (hidden behind the red vertical line for $\chi$).

for $\phi_a$, and from 64.6 to 33.5 for $\chi$. However, for the fecundity $\alpha$ the change was much more dramatic. In the motivating example, the posterior distribution was relatively close to the prior with an overlapped area of 88.17% whereas in the simulation here, it was only 17.40%. This is unsurprising given that there were precise observations about both the pup and adult number in every year and the ratio of newborn pups to adults could easily be calculated with only a small observation error.

When looking for trends as the observation variance decreases, we note that the standard deviations of all four marginal distributions decreased. We also note that the mean of the distribution moved in one consistent direction for each parameter as the observation variance decreased. For all four parameters, this direction was away from the true value. For $\alpha$ and $\chi$ the mean even moved beyond the true parameter value for the two smaller observation variances, when the mean for a CV of 10% was in between the prior mean and the true value or very close to the true value. We also note that for those parameters, there was very little difference in marginal posterior distribution between the two smaller observation variances. This could indicate that even with better data, some uncertainty about the parameters remains due to the nature of the data and the model. However, we also note in Table 5.5 that the true value lay within the range of plausible values for each of the marginal distributions.

| | $\phi_{p,\max}$ | $\phi_a$ | $\alpha$ | $\chi$ |
|---|---|---|---|---|
| Prior | 0.21 | 0.47 | 0.38 | 0.38 |
| CV = 10% | 0.27 | 0.71 | 0.29 | 0.51 |
| CV = 1% | 0.15 | 0.85 | 0.66 | 0.87 |
| CV = 0.5% | 0.06 | 0.94 | 0.67 | 0.87 |

**Table 5.5:** Empirical cumulative distribution function of the marginal prior and three posterior distributions, evaluated at the true parameter value.

In summary, even with a long time series with very precise observations of both adult and pup counts, there remained a lot of uncertainty for some of the parameters, namely $\phi_{p,\max}$ and $\phi_a$. It also seems that for the two parameters where the posterior distribution indicated a much greater learning, almost no improvement could be achieved by reducing the CV from 1% to 0.5%. Importantly, decreasing the CV did not lead to a better point estimate of the true parameter value—indeed the posterior means moved away from the true value as CV decreased.

#### 5.3.1.2 Joint Distributions of Parameters

As there was seemingly little learning about some of the parameters when examining only the marginal distributions, even with very precise observations about both age classes, we then analysed the joint distributions of the parameter pairs.

In Figure 5.7 we note that for all combinations the parameter space was greatly reduced from the prior distribution, even when only a CV of 10% was used. Rows 2 to 6 of the figure show scatter plots where the parameters $\alpha$ or $\chi$ are part of the joint distributions. These are the two parameters where the variance of the marginal posterior distributions was already greatly reduced, and the reduction in posterior space seems largely due to their contribution and not $\phi_a$ or $\phi_{p,\max}$. In the first row, we see the joint distribution of $\phi_{p,max}$ and $\phi_a$, where even for a CV of 0.5% a large uncertainty remained in the marginal posterior distributions. In the scatter plot of the samples from the joint distribution, we note that the two-dimensional parameter space was decreased significantly. Here, the points of the sample are highly correlated and lie on a straight line.

Again, we observed that a reduction in CV did not lead to a better point estimate of the true parameter value. However, it is also apparent that even when the joint parameter space was considered, the true value still lay within the area of plausible values of the posterior distribution. We quantified this by calculating the Mahalanobis distance between the true parameter value and the three posterior distributions. This was 1.25 for a CV of 10%, 2.10 for a CV of 1%, and 2.45 for a CV of 0.5%.

As indicated by the scatter plots in Figure 5.7 and confirmed by the correlation coefficients in Figure 5.8, there was a strong correlation between $\phi_{p,\max}$ and $\phi_a$, as well as between $\alpha$ and $\chi$. For the first of these pairs, this strong correlation was apparent for all three of the observation variances. However, for the second pair $\alpha$ and $\chi$, the strength of the correlation increased as the observation CV decreased, with a particularly sharp increase from 0.55 to 0.92 when the CV decreased from 10% to 1%.

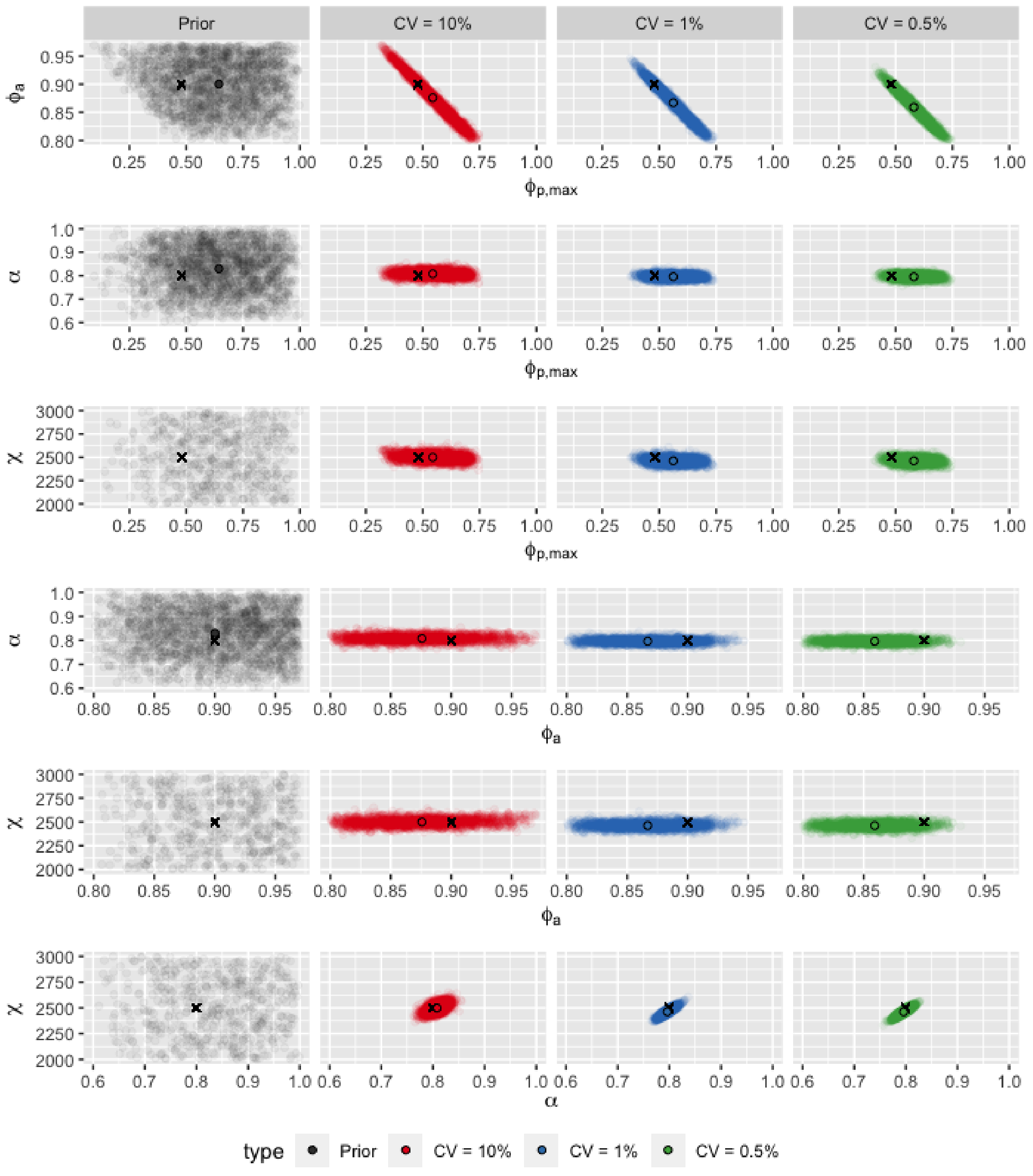


**Figure 5.7:** Samples from the prior and posterior distributions for the 2-state model, estimated using observations with three different CVs. The mean of each joint distribution is indicated with a black circle, other than for the prior of $\chi$, which is outside the plot limits. The true value used to simulate the data is shown as a black cross. For the parameter $\chi$, the axis limits were restricted to focus on the joint posterior distributions, resulting in an omission of some of the sampled values from the prior distribution.

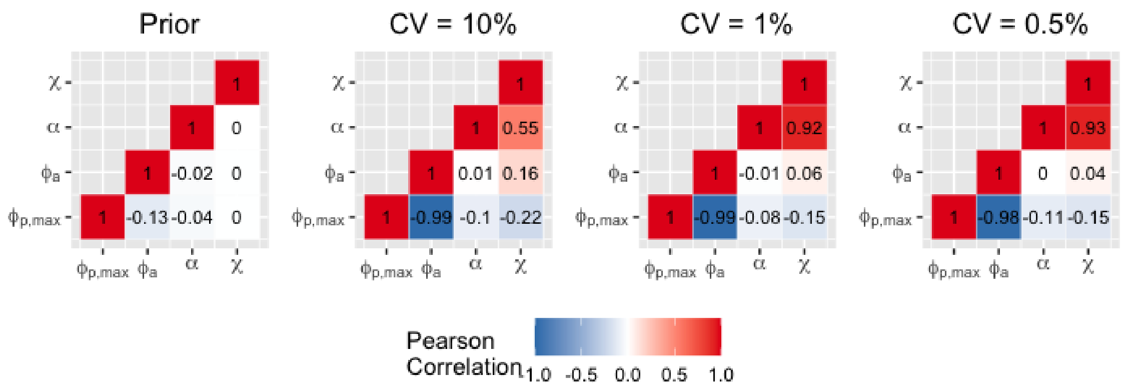


**Figure 5.8:** Pearson correlation coefficient between the four parameters for the prior distribution and the posterior distributions estimated using observations with three different CVs.

The strong correlation between $\phi_{p,\max}$ and $\phi_a$ in combination with their wide marginal posterior distributions indicates parameter identifiability issues. This means that even given precise observations of both age classes, it might be impossible to learn the true value of these parameters without further information about abundance after a sub-process such as pup survival.

#### 5.3.1.3 State Estimates

Figure 5.9 shows adult and pup trajectories sampled from the smoothed posterior distributions for the three different CVs. Even though the true parameter values could not be recovered even with observations with a CV of 0.5%, it seems that it was possible to recover the true state values. We also note that the small change in parameter estimates from a CV of 1% to a CV of 0.5% still led to a visible reduction in variance of the smoothed state estimates, while also reducing the scaled root mean square error seen in Table 5.6.

| | CV=10% | CV=1% | CV=0.5% |
|---|---|---|---|
| Posterior distribution | 0.0419 | 0.0183 | 0.0152 |
| True parameter value | 0.0679 | 0.0188 | 0.0154 |

**Table 5.6:** Scaled root mean squared error (SRMSE) of the state estimates compared to the true state values. The SRMSE was calculated for all three posterior distributions, as well as when the true parameter value is assumed to be known and the three different sets of observations are used.

We also assessed the quality of the smoothed posterior state estimates by comparing them with the smoothed state estimates when the true parameter value is known and the observations with the different CVs are used. The resulting estimated state trajectories can be seen in the Appendix in Figure E.1. Table 5.6 compares the SRMSE for these two scenarios. For all three CVs the SRMSE was larger when the true parameter value was used. Comparing the estimated state trajectories in Figures 5.9 and E.1, we note that this seems due to the state estimates being much closer to the observations—and therefore sometimes further away from the true state value—when the true parameter is known, than when it is not. This is confirmed in Figure 5.10 where the standard deviations of the state estimates across time are shown. We note that the difference in standard deviation was much larger for the larger CV and decreased as the CV decreases.

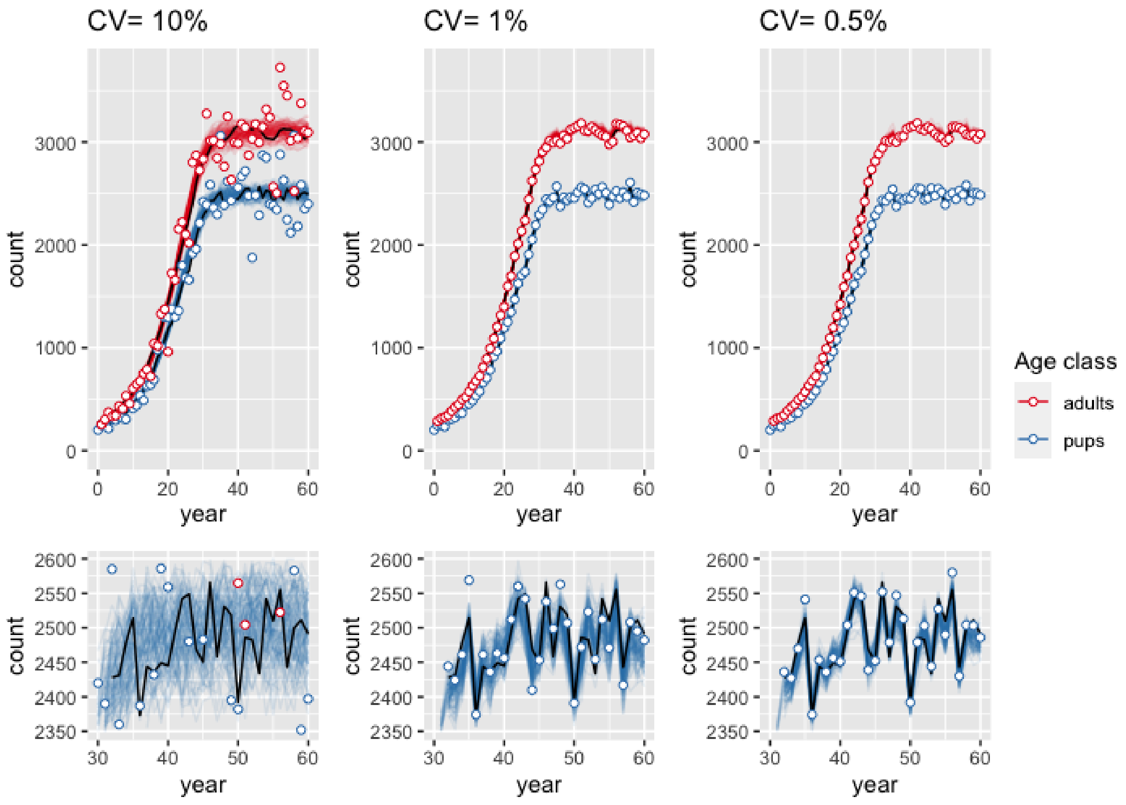


**Figure 5.9:** Smoothed posterior state estimates of both pups and adults for the three different observation variances. The white circles show the observed pup and adult counts. In the second row, both axis limits are reduced to better illustrate the reduction in variance of the pup state estimates as the CV decreases.

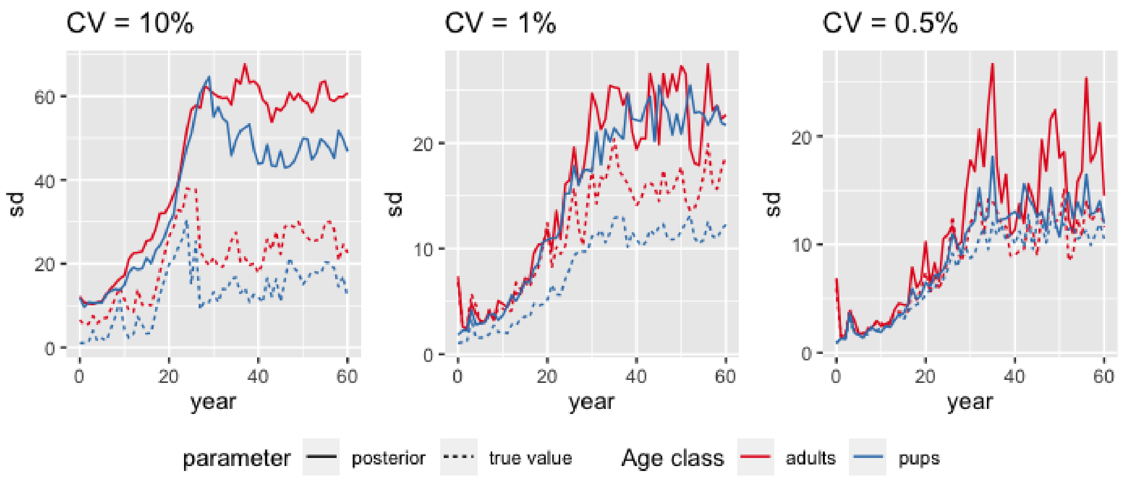


**Figure 5.10:** Standard deviations of the smoothed state estimates across time. The standard deviations are shown for both the adult and pup estimates and for all three observation CVs. The solid line shows the standard deviations when the parameters are sampled from the posterior distributions, and the dashed line shows it when the parameter is fixed to the true value. Note the change in y-axis scale between the three plots.

In conclusion, there did not appear to be any bias in the state estimates. Even though some of the marginal posterior parameter distributions were very wide, and the joint distributions revealed issues with parameter identifiability, the state estimates were not affected by these issues. On the contrary, they did not appear to exhibit any bias and showed a decrease in variance as the CV decreased.

### 5.3.2 Changing Process Stochasticity

| SD($\alpha$) | $a$ | $b$ | $N$ | $M$ | $h$ | ESS | $\hat{R}$ | Runtime (sec) | ESS/sec |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| 0 | - | - | 30 | 100000 | 0.8 | 3279.72 | 1.00 | 262.18 | 12.509 |
| 0.01 | 1279.2 | 319.8 | 30 | 100000 | 0.8 | 5607.05 | 1.00 | 274.19 | 20.449 |
| 0.03 | 141.4 | 35.4 | 100 | 100000 | 0.6 | 6808.53 | 1.00 | 848.96 | 8.020 |
| 0.10 | 12.0 | 3.0 | 300 | 100000 | 0.4 | 8035.41 | 1.00 | 2652.20 | 3.030 |
| 0.20 | 2.4 | 0.6 | 300 | 100000 | 0.1 | 1223.13 | 1.02 | 3104.30 | 0.394 |

**Table 5.7:** Details from fitting the 2-state model with a random effect on $\alpha$ with varying variance. ESS, $\hat{R}$, and runtime are given for one randomly selected run each.

For each of the selected standard deviations, 50 posterior distributions were estimated with a PMMH with the settings detailed in Table 5.7.

#### 5.3.2.1 Marginal Distributions of Parameters

First, we examine the marginal distributions of the three unknown parameters for one randomly selected run each. Figure 5.11 shows the marginal densities for the 5 different standard deviations and for the prior distribution. For calculation of the induced prior, $\alpha$ was fixed to 0.8, as this is the value that was used to calculate the annual pup survival probability for the Beverton-Holt density dependence. We note that this induced prior for $\phi_{p,\max}$ and $\phi_a$ was much closer to the theoretical prior distribution when $\alpha$ was fixed to 0.8 as opposed to when $\alpha$ was distributed according to its prior distribution (compare with Figure 1.6).

We also note that the marginal posterior distributions varied greatly between the five different simulations. This confirms that comparing results between the five different random effect variances is only feasible when many simulations are considered, as any difference between two single posterior distributions can be due to the different underlying states and observations.

Figure 5.12 shows boxplots of the means of the marginal posterior distributions from 50 different simulations, for each of the 5 random effect variances. We note that the distributions of the posterior means were similar for the 5 different variances. For $\phi_{p,\max}$, the posterior means tended to lie between the true value and the prior mean—between 60% and 84% of the means lay within this range.

For $\chi$, the mean values were dispersed around the true parameter value. The medians of the mean values were close to the true value, ranging from 2496 to 2506. As seen before (e.g., in Figure 3.7a), there was a lot of information about $\chi$ in the data compared to the relatively wide prior and it was therefore unsurprising that the prior mean had such little effect on the posterior means.

For $\phi_a$, a shift of the posterior mean away from the true value and prior mean could be observed for all five standard deviations values. Between 62% (for SD = 0.1) and 84%

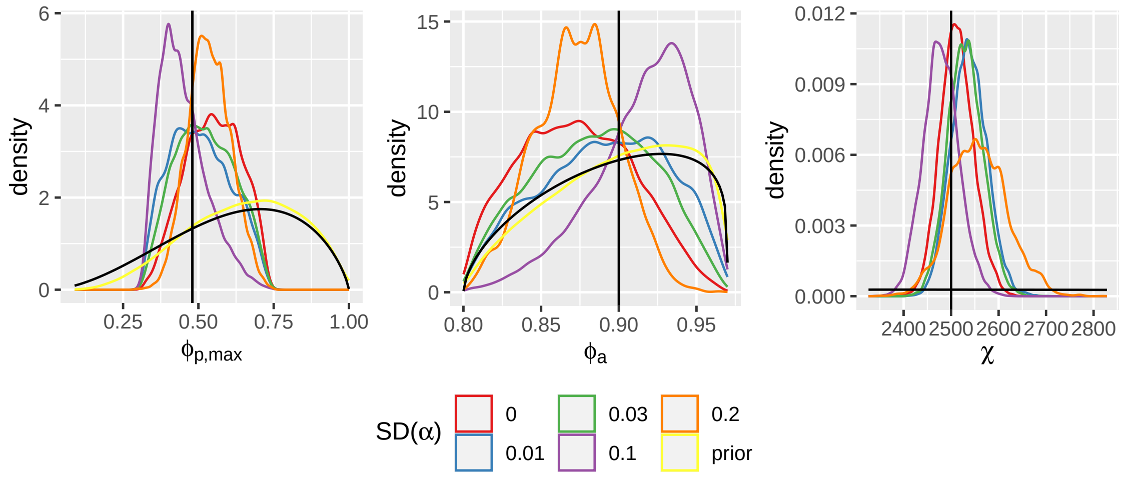


**Figure 5.11:** Marginal posterior distributions for the 2-state model, using simulated state values with 5 different standard deviations for the random effect on $\alpha$. The marginal prior distribution is indicated in black. The induced marginal prior distributions for $\phi_{p,\max}$ and $\phi_a$ are indicated in yellow. The true value used to simulate the data is shown as the vertical solid black line.

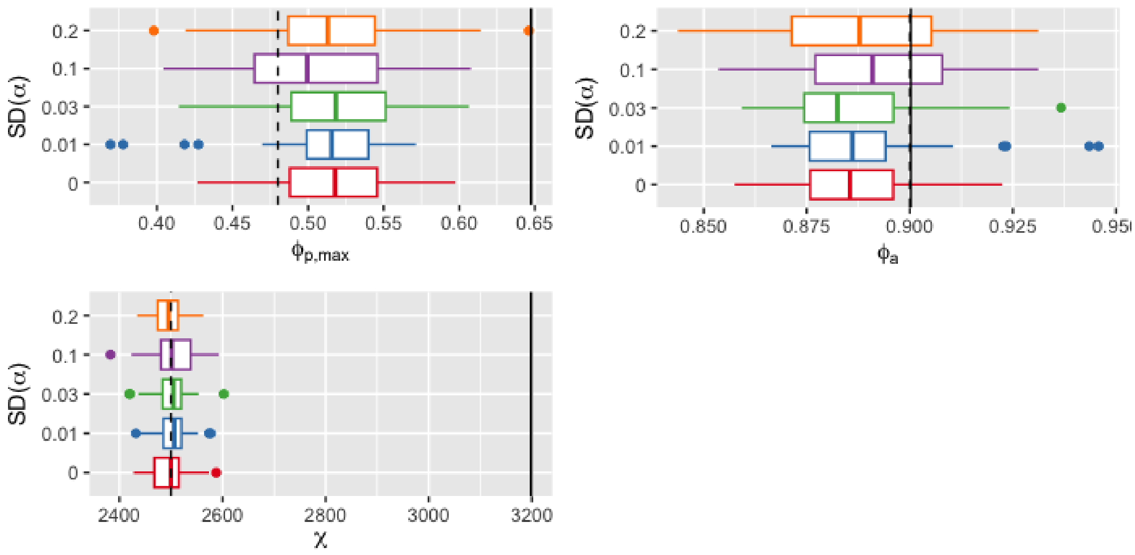


**Figure 5.12:** Boxplots of posterior means from 50 simulations for each of the 5 values for the random effect standard deviation. The black dashed line indicates the true parameter value used to simulate the data, the black solid line indicates the mean of the prior distribution.

(for no random effect) of posterior means were lower than the true value. This shift was more pronounced when the variance of the random effect was lower. However, the main issue was that the estimation of this parameter remained difficult across all random effect variances.

While the posterior means were occasionally further away from the prior mean than the true value, this did not appear to be a systematic behaviour for the most part. Rather, the mean was centred around the true value for $\chi$, where the data were highly informative, and lay on average between the true value and the prior mean for $\phi_{p,\max}$, where the prior was more important compared to the only weakly informative data. For $\phi_a$, the posterior mean was not centred around the true value and the prior mean. This could be due to identifiability issues with this parameter. There was no clear trend in the distributions of the posterior means when the standard deviation of the random effect was changed. Even with a large random effect on $\alpha$, the estimated marginal posterior distributions of the three unknown parameters seemed unaffected.

#### 5.3.2.2 Joint Distributions of Parameters

Next, the joint distributions were examined. In Figure 5.13, the results from one randomly selected run each are shown. We see that there was a very strong negative correlation between $\phi_{p,\max}$ and $\phi_a$ across all standard deviations. Again, the different underlying states made it difficult to identify any trends when the random effect size was changed.

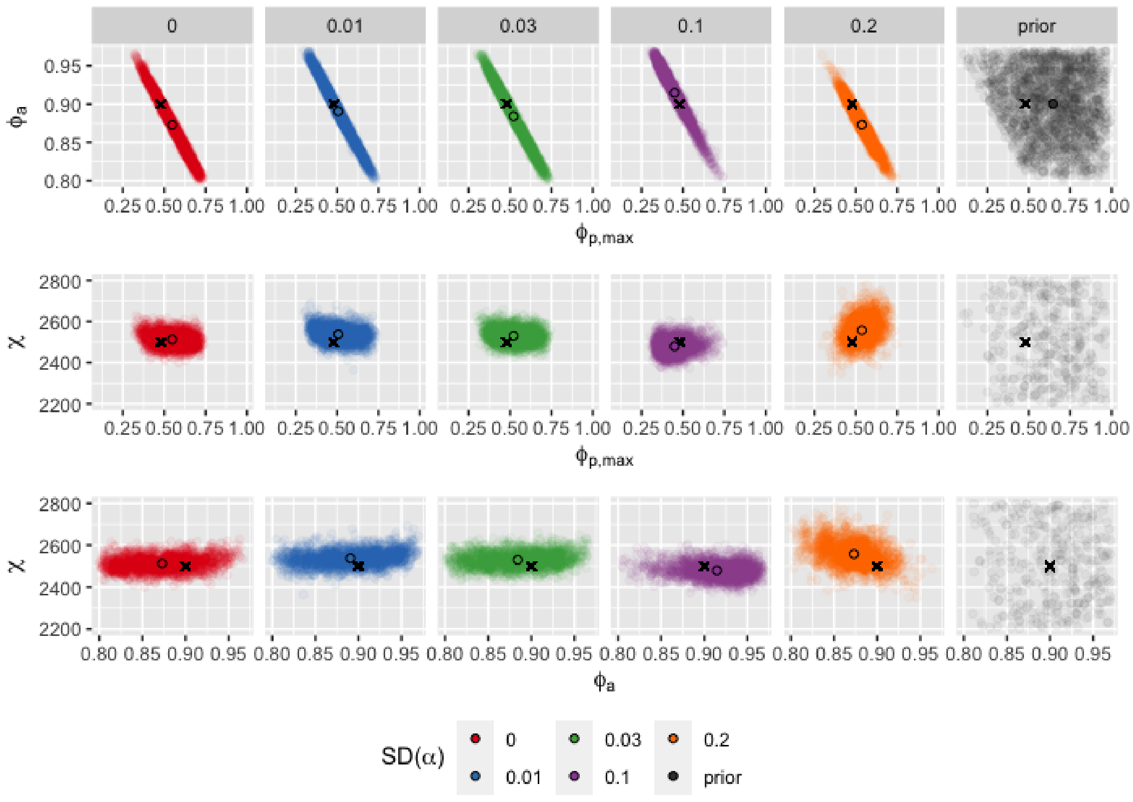


**Figure 5.13:** Samples from the prior and posterior distributions for the 2-state model, using simulated state values with 5 different standard deviations for the random effect on $\alpha$. The mean of each joint distribution is indicated with a black circle. The true value used to simulate the data is shown as a black cross. For the parameter $\chi$, the axis limits were restricted to focus on the joint posterior distributions, resulting in an omission of some of the sampled values from the prior distribution.

In Figure 5.14, boxplots of the correlation coefficients between each of the three parameter pairs from 50 different simulations for each random effect standard deviation are shown. Contrary to the marginal distributions, a clear trend can be seen when the standard deviation of the random effect on $\alpha$ was increased. The standard deviation of the 50 observed correlation coefficients was similar for the three lower values for the random effect standard deviation but increased with the two larger values. There also appeared to be a

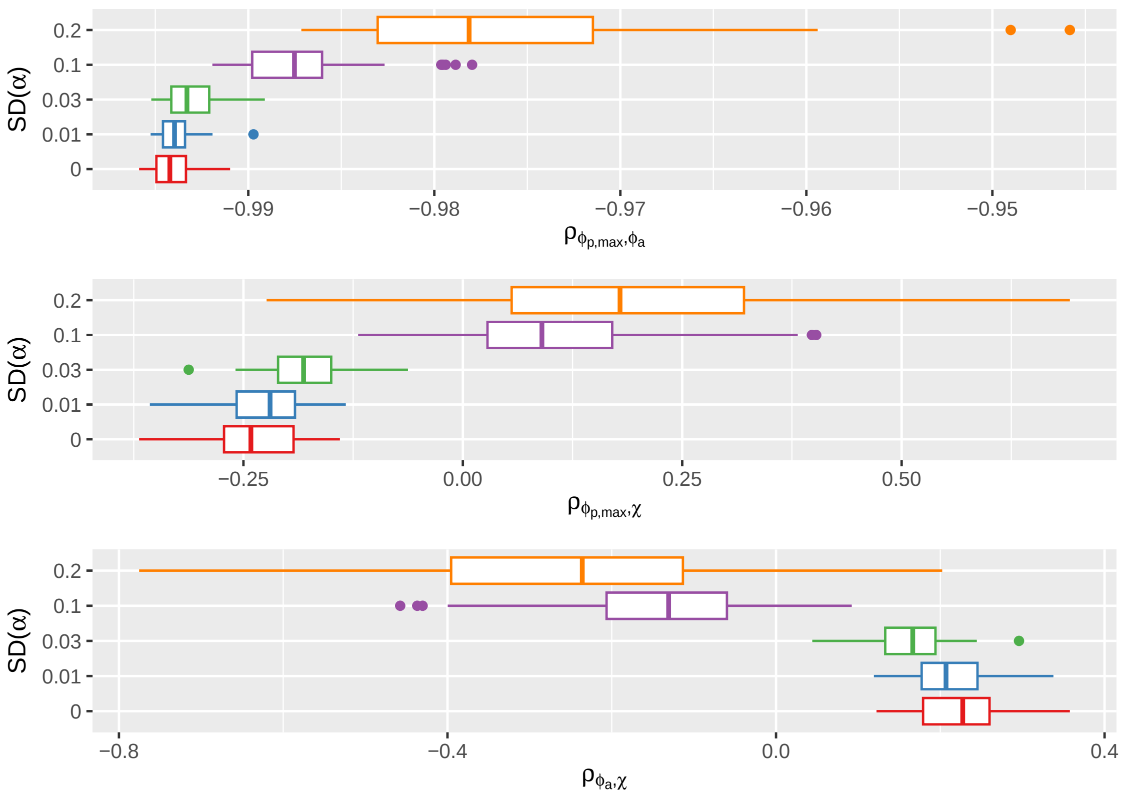


**Figure 5.14:** Boxplots of the posterior correlation coefficients $\rho$ using 50 simulations each, for each of the 5 different standard deviations for the random effect on $\alpha$ and for each of the three parameter pairs.

shift in the mean of these correlation coefficients. For the pair $(\phi_{p,\max}, \phi_a)$, it decreased from 0.994 for the model with no random effect to 0.976 for the model with a random effect standard deviation of 0.2. For the pairs $(\phi_{p,\max}, \chi)$ and $(\phi_a, \chi)$, these changes were from -0.239 to 0.183 and from 0.226 to -0.246 respectively. The mean correlation coefficient even moved across 0 for the pairs $(\phi_{p,\max}, \chi)$ and $(\phi_a, \chi)$ as the random effect standard deviation changed.

### 5.3.2.3 State Estimates

To sample from the smoothing distribution of the posterior states, we sampled 100 parameter vectors from each posterior distribution, ran a particle filter with 1000 particles for each of them and selected one smoothed state trajectory for each of the parameter vectors. Figure 5.15 shows the distribution of the smoothed posterior state estimates for one randomly selected simulation for each of the five values of random effect standard deviation. It seems that even when the standard deviation of the random effect was large, that the estimates could track the true state trajectory relatively well due to the many observations of both the adult and pup numbers.

This was confirmed when the results from all 50 simulations were analysed. Figure 5.16 shows boxplots of the scaled RMSEs from the smoothed posterior state estimates, and compares these with the scaled RMSEs when the true parameter value was known. While the median SRMSE was lower for the true parameter value across all 5 random effect standard deviations, the difference was not large. In fact, the median SRMSE of the

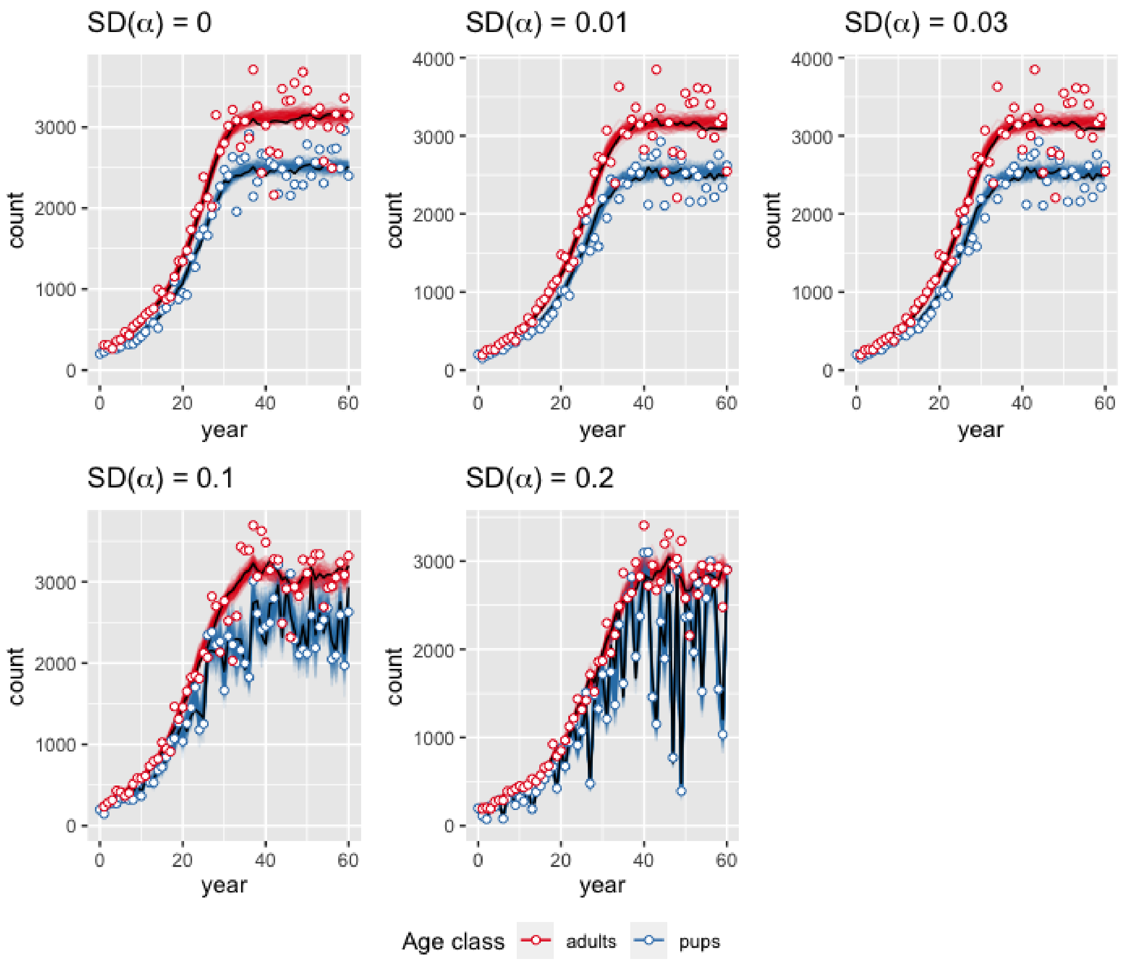


**Figure 5.15:** Smoothed posterior state estimates of both pups and adults, using simulated state values with 5 different standard deviations for the random effect on $\alpha$. The white circles show the observed pup and adult counts and the black lines indicate the true counts.

posterior smoothed state estimates lay within the interquartile range of the SRMSEs when the true parameter value was known, and in 2 out of 5 cases, the largest observed SRMSE was produced when the true parameter value was used.

As in the previous section where the observation variance was varied, the estimation difficulties of the marginal parameter distributions did not translate to a difficulty in estimating the posterior states. Even though some non-idenfiability issues appeared to affect the estimation of $\phi_{p,\max}$ and $\phi_a$, the posterior state estimates were almost as close to the truth as when the true parameter value was used.

## 5.4 Conclusion

We found that even with very good data consisting of a long time series observations of both age classes with a small error, estimation of the posterior distribution remained difficult for some of the parameters in this model. In particular the two survival probabilities $\phi_{p,\max}$ and $\phi_a$ suffered from identifiability issues. When the joint distribution of these two parameters was examined, we noted the very high correlation of between -0.95

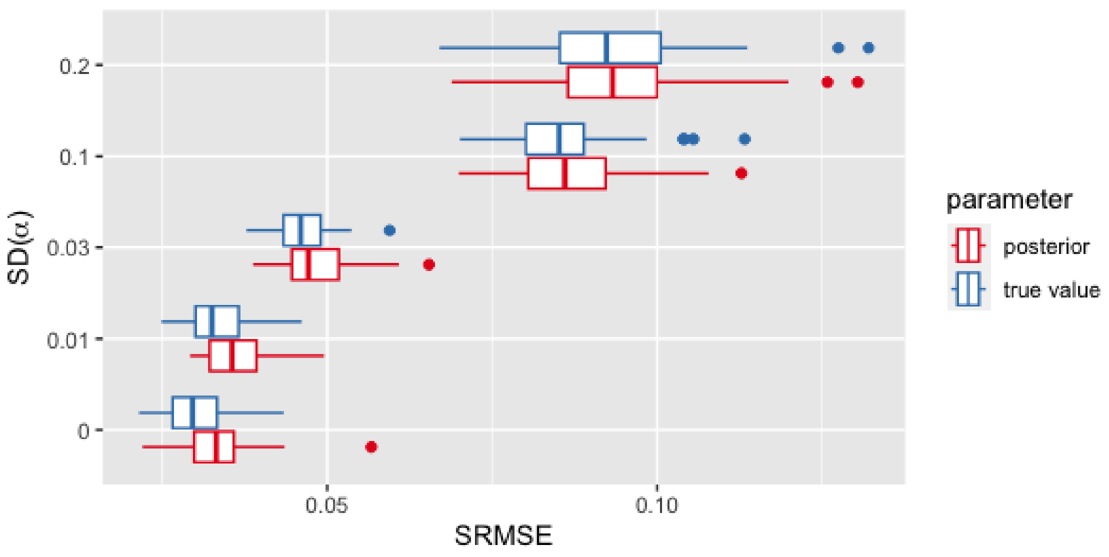


**Figure 5.16:** Boxplots (🟥) of the scaled RMSEs of the smoothed posterior state estimates from 50 simulations with 5 different random effect standard deviations. The boxplots in blue (🟦 show the scaled RMSEs when the same observations are used but parameters are fixed to the true values.)

and -0.99 depending on the exact model. In the future, a re-parameterisation of the seal model could be considered to alleviate these identifiability issues and to reflect the strong relationship between these two parameters. For example, one might consider having a parameter for adult survival and calculating maximum pup survival as a function of adult survival. Other remedies might be to obtain data on an intermediate life stage, e.g., abundance of age 1 animals in a 7-state model to obtain a better estimate of $\phi_{p,\max}$, though this is likely not feasible (see Buckland et al., 2004, p. 164, on the importance of collecting data on relevant life stages), or adapting the survey effort to a more suitable observation model (Knape et al., 2011).

Returning to the motivating question of a possible bias when the posterior mean is estimated, we found that this estimate can indeed sometimes be further away from the true parameter value when the observation variance is reduced. It is important to consider this when communicating results. We could not confirm a systematic bias when the process stochasticity is increased. While the median of estimated posterior means of the parameter $\phi_a$ is significantly smaller than the true value and prior mean for all random effect sizes, there was no clear relationship of this behaviour with the random effect size. This might also be due to the identifiability issues of this parameter.

Considering the issues with the estimation of the posterior distribution of the parameters, it is remarkable that the posterior state estimation produces very precise estimates with errors only marginally larger than when the true fixed parameter is used. From a practical perspective this is very important. Often, estimating the parameters is only a necessary intermediate step for the final goal of estimating the states. Here, we can be be confident that the state estimates are unbiased, and that they improve as the observations get closer to the truth. However, it is important to clearly communicate the identifiability issues and how this affects the estimates. While it seems that the model and collected data are adequate when only an estimate of the states is required, we cannot produce precise posterior estimates of the parameters.

In future research, we would like to explore the combination of a random effect in the process equations with a changing observation variance and study a possible interaction between the two features. With a lot of computational power, it could also be feasible to repeat the observation variance simulation study to discover more general trends rather than only analysing the results from one time series. This chapter has also highlighted the identifiability issues of some of the parameters in this model even when great data are available. A next step is therefore to analyse these identifiability issues and to identify and implement methods for addressing these challenges.

# Chapter 6

# Factorising the Likelihood

## 6.1 Introduction

In this chapter, we return to the idea of factorising the likelihood first presented in Section 2.5. There, we investigated the effect of utilising the independence of the four regions when the independent estimate is omitted. This showed promising results. In particular, for achieving the same variance of the log-likelihood estimates produced by a particle filter, the factorised formulation required around 10 times fewer particles than the joint formulation.

However, transferring this idea to the complete seal model with independent estimate requires some modifications. Through this independent estimate, which is an estimate of the total number of adult seals, the regions are no longer independent from each other. In this chapter, we describe in Section 6.2.1 two formulations for how the independent estimate can be included in the likelihood term. Using these formulations, we lay out three methods that allow elements of the factorisation idea to be used even for the complete seal model with independent estimate (Section 6.2.2). In Section 6.2.3, we describe the methods used for measuring a potential gain in efficiency when the first of the three options is implemented. Section 6.3 describes the results that were obtained when the likelihood and the posterior distribution were estimated with the factorised formulation compared to a joint formulation.

## 6.2 Methods

### 6.2.1 Factorisation of the Independent Estimate

In Chapter 2, we derived in Equation 2.15 that the likelihood for the complete seal model without independent estimate can be estimated separately for each region and then multiplied for an estimate of the total likelihood, and one factor of the likelihood at time $t$ is

$$p(y_{t,r=1:4}|y_{1:t-1,r=1:4}) = \prod_{r=1}^{4} p(y_{t,r}|y_{1:t-1,r}) \approx \prod_{r=1}^{4} \frac{1}{N} \sum_{i=1}^{N} w_{t,r}^{(i)}.$$

The unbiasedness of the estimate is preserved due to the independence of the regions (see Section 2.5). It is possible to preserve this idea of factorising the likelihood for parts of the likelihood estimation algorithm even when including the independent estimate. Before describing the options for this, we show how the likelihood can be rewritten when the independent estimate is part of the data.

Including the independent estimate $y_{IE,24}$ in the state space model in addition to the annual pup production estimates allows two different formulations of the marginal density related to the inclusion of $y_{IE,24}$, which we label *embedded formulation* and *appended formulation*, that affect implementation of the algorithm. We denote the year to which the independent estimate is assigned with $t^*$ and use $y_{IE,24}$, $y_{IE}$ and $y_{IE,t^*}$ somewhat interchangeably depending on the context. As a reminder, the observation density of the independent estimate is

$$y_{IE,24}|x_{24} \sim x_{\text{adults},24} - \text{Ga}(\kappa_1, \kappa_2)$$

where $x_{\text{adults},24}$ denotes the total number of all seals (male and female) aged 1 and above in all regions and can be directly calculated from the state vector $x_{24}$ using the sex ratio parameter $\omega$, i.e.,

$$x_{\text{adults},24} = \omega \sum_{r=1}^{4} \sum_{a=1}^{6} x_{a,r,t=24},$$

see also Equation 1.6.

The embedded formulation augments the observation in year $t^*$, which is the formulation that has been used so far throughout this thesis:

$$p(y_{1:T}, y_{IE}) = \left( \prod_{t=1}^{t^*-1} p(y_t|y_{1:t-1}) \right) p(y_{t^*}, y_{IE}|y_{1:t^*-1}) \left( \prod_{t=t^*+1}^{T} p(y_t|y_{1:t-1}, y_{IE}) \right). \tag{6.1}$$

In a particle filter this means the importance weight of a particle $i$ is $p(y_{t^*}, y_{IE}|x^i_{t^*}) = p(y_{t^*}|x^i_{t^*})p(y_{IE}|x^i_{t^*})$. Importantly, and independently of the choice of algorithm, this means that from time $t^*$ onwards the state estimates for the four regions are no longer independent. From an implementation point of view this means that the updating and predicting equations no longer factorise into four separate parts but need to be considered jointly.

The appended formulation considers the independent estimate only at the end, after all standard observations $y_{1:T}$ have been looked at:

$$p(y_{1:T}, y_{IE}) = \left( \prod_{t=1}^{T} p(y_t|y_{1:t-1}) \right) p(y_{IE}|y_{1:T}) \tag{6.2}$$

This means that the likelihood of the independent estimate is calculated given all other observations, including the ones for later times $t > t^*$. In the implementation of the various algorithms this means that the smoothed states need to be used in the evaluation of this factor. In a particle filter, this means averaging over all smoothed particles $x^i_{t^*}$ at time $t^*$, so $1/N \sum p(y_{IE}|x^i_{t^*})$.

### 6.2.2 Incorporating the Independent Estimate in a Factorised Formulation

We present here three options for incorporating the independent estimate as a shared data point while still benefiting from some of the advantages of the factorised formulation. The first option incorporates the independent estimate into the likelihood estimate produced by a bootstrap filter using the embedded formulation, the second option incorporates it using the appended formulation, as described in the previous section. This likelihood estimate can then be used as part of a PMMH algorithm. The third option is to initially disregard the independent estimate when the posterior distribution is estimated and only to include it in a second step with an adjustment of the initially estimated posterior distribution, again using the appended formulation.

**Option 1** For the first option of factorising while including the shared data point, we use the embedded formulation of the likelihood as written in Equation 6.1. The first of the factors in that equation can be estimated as follows,

$$\begin{aligned}\prod_{t=1}^{t^*-1} p(y_t|y_{1:t-1}) &= \prod_{r=1}^{4}\prod_{t=1}^{t^*-1} p(y_{t,r}|y_{1:t-1,r}) \\ &\approx \prod_{r=1}^{4}\prod_{t=1}^{t^*-1} \frac{1}{N}\sum_{i=1}^{N} w_{t,r}^{(i)}.\end{aligned}$$

For the second factor, we introduce the state $x_{t^*}$ and write

$$\begin{aligned}p(y_{t^*}, y_{IE}|y_{1:t^*-1}) &= \int p(y_{t^*}, y_{IE}, x_{t^*}|y_{1:t^*-1})dx_{t^*} \\ &= \int p(y_{t^*}|x_{t^*})p(y_{IE}|x_{t^*})p(x_{t^*}|y_{1:t^*-1})dx_{t^*}\end{aligned}$$

From a bootstrap filter we obtain a (potentially weighted) sample from the distribution in the last factor of the integrand $p(x_{t^*}|y_{1:t^*-1})$. This applies regardless of whether the sample of the states $x_{t^*}|y_{1:t^*-1}$ was obtained using the joint formulation or from the factorised formulation. There, we use

$$\begin{aligned}p(x_{t^*}|y_{1:t^*-1}) &= p(x_{t^*,1}, x_{t^*,2}, x_{t^*,3}, x_{t^*,4}|y_{1:t^*-1}) \\ &= \prod_{r=1}^{4} p(x_{t^*,r}|y_{1:t^*-1,r}),\end{aligned}$$

and obtain a sample from each of the four distributions on the right-hand side from four separate bootstrap filters. These can be combined into a sample from the left-hand side distribution by combining randomly chosen particles from each of the four bootstrap filters into a joint particle, where the importance weight is the product of the four weights assigned to each of the particles, so $w_{t^*-1}^{(i)} = \prod_{r=1}^{4} w_{t^*-1,r}^{(i)}$ . Using these newly created particles and weights, a bootstrap filter for the joint formulation can be set up, starting with the observation $y_{t^*}, y_{IE}$, and updating the weights with $w_{t^*}^{(i)} = w_{t^*-1}^{(i)} g(y_{IE}|x_{t^*}^{(i)}) g(y_{t^*}|x_{t^*}^{(i)})$. This bootstrap filter can then be run until time $T$ to obtain estimates for the factors in the last term in Equation 6.1. We note that even though the observations $y_{t^*+1:T}$ no longer contain a shared data point, the particles cannot be separated into the factorised formulation. They are samples from the filtering distribution $p(x_t|y_{1:t}, y_{IE})$ and through their dependence on $y_{IE}$ no longer independent.

The advantage of this approach is that no additional calculations are necessary to obtain the required samples and weights. The existing bootstrap filter framework can be used with only a slight modification to produce a likelihood estimate. In the case of the seal model, where the shared data point is introduced at time $t = 24$ out of a total number of observations $T = 26$, the factorised structure can be used for almost the entire time series. However, this also illustrates the drawback of this option as a general method for models that consist of sections that are independent other than one shared data point. If this shared data point occurs towards the start of the time series, the factorised structure can only be used for a few iterations before that time point. If several shared data points need to be incorporated, the expected benefit of this option depends on the time point at which the first if the shared time points appear. After that, the number of shared time points does not affect the ability to factorise the model. The different sections cease to be independent after the introduction of the first shared data point and so the joint formulation has to be used regardless of whether more shared data points are introduced. This is relevant for the seal model because two further independent estimates of total adult seal counts have been undertaken since 2010, the year of the last observations used in this thesis.

**Option 2** The second option is based on the appended formulation as in Equation 6.2, thus the independent estimate $y_{IE}$ is only included after all standard observations $y_{1:T}$ have been incorporated. We can obtain an estimate of the first term using the factorised form since the four regions are independent when $y_{IE}$ is omitted. For the second term in Equation 6.2 we integrate over $x_{t^*}$ and obtain

$$\begin{aligned} p(y_{IE}|y_{1:T}) &= \int p(y_{IE}, x_{t^*}|y_{1:T}) dx_{t^*} \\ &= \int p(y_{IE}|x_{t^*}) p(x_{t^*}|y_{1:T}) dx_{t^*} \\ &\approx \sum_{i=1}^{N} p(y_{IE}|x_{t^*}^{(i)}), \quad x_{t^*}^{(i)} \sim p(x_{t^*}|y_{1:T}). \end{aligned}$$

The last approximation is a Monte Carlo estimate of the integral, where $x_{t^*}^{(i)}$ is sampled from the smoothing distribution $p(x_{t^*}|y_{1:T})$. This sample is produced by a bootstrap filter if the ancestor particles of each resampled particle are saved, as in Algorithm 1 (Gordon et al., 1993). As discussed in Section 2.2.8, this sample can suffer from particle degeneracy resulting in high Monte Carlo error if $t^*$ is much smaller than $T$. For the complete seal model, this is not an issue, since the time point of the independent estimate is close to the end of the time series. The estimate of the smoothed distribution at that time point produced by the bootstrap filter is thus not expected to suffer from particle degeneracy and does not require any further calculations besides the bootstrap filter that was run to obtain the estimate for $p(y_{1:T})$. We note that this is how the independent estimate is incorporated in Thomas et al. (2019), although there the initial weighted sample of the posterior distribution without independent estimate is obtained with a modified Liu-West algorithm (Algorithm 4). Generalising this option for shared data points at earlier time points is still possible. The large Monte Carlo variance of the estimate of $p(y_{IE}|y_{1:T})$ could be combated by using a more sophisticated smoothing algorithm (see Section 2.2.8). While this requires an additional step to obtain an estimate of $p(y_{IE}|y_{1:T})$, the factorised structure can be used for the entire time series regardless of how early in the time series the shared data point appears. In addition, multiple shared data points after the first appearance of such a point can easily be incorporated since obtaining a sample from $p(x_{t^*}|y_{1:T})$ usually yields samples from $p(x_t|y_{1:T})$ for all $t > t^*$.

**Option 3** The third option is to estimate the posterior distribution in two steps. First, we apply an algorithm such as the PMMH to estimate the posterior distribution without the independent estimate where the factorised formulation can be used throughout. In a second step, we use importance sampling to incorporate the independent estimate. The posterior without independent estimate is used as the proposal distribution, and the importance weight can be calculated with

$$\begin{aligned}\frac{p(\theta|y_{IE}, y_{1:T})}{p(\theta|y_{1:T})} &= \frac{p(\theta)p(y_{IE}, y_{1:T}|\theta)}{p(\theta)p(y_{1:T}|\theta)} \\ &= \frac{p(y_{IE}|y_{1:T}, \theta)p(y_{1:T}|\theta)}{p(y_{1:T}|\theta)} \\ &= p(y_{IE}|y_{1:T}, \theta)\end{aligned}$$

An estimate of the last term $p(y_{IE}|y_{1:T}, \theta)$ can be obtained through a smoothing algorithm (see Section 2.2.8). Since the values of $\theta$ are drawn from a sample where a bootstrap filter was run for each value, an implementation can be considered where this estimate is generated and saved as the bootstrap filter is run. If standard importance sampling is not feasible due to large difference between the target and the proposal distribution, a more sophisticated importance sampling technique such as the SMC sampler could be considered (see Section 4.5.2).

In the remainder of this chapter we demonstrate application of the approach described in Option 1 and leave demonstrations of the other two options as future research. This option is straightforward to implement, requires no additional smoothing algorithm and leads to an unbiased likelihood estimate.

### 6.2.3 Assessing Gain in Efficiency

To assess the benefits of exploiting the factorised structure when using SMC methods, we examined both the likelihood estimates and the output from the PMMH algorithm using the factorised formulation.

For the likelihood estimate, we used the same methods as in Chapter 2 to compare the factorised formulation with the joint formulation. We estimated the likelihood at four different values in the parameter space: the prior mean $\theta_1$, the posterior mean $\theta_4$ determined in Thomas et al. (2019), and two values in between these. The first was the mean, so $\theta_2 = (\theta_1 + \theta_4)/2$ and the second was shifted further towards the posterior mean with $\theta_3 = (\theta_1 + 3\theta_4)/4$. The parameter values can be seen in Table 6.1. For both the joint and the factorised formulation we used the settings for the particle filter that were determined optimal in Chapter 2 and varied the number of particles from 3 to 100,000. For each number of particles, the likelihood was estimated 100 times. Then, the CV of the likelihood and the variance of the log-likelihood estimates were compared between the joint and the factorised formulation across all numbers of particles. We also recorded the runtimes of the two formulations to evaluate the effect of modifying the likelihood estimation.

For the posterior distribution, we ran the PMMH with the optimal settings determined in Chapter 3 but varied the number of particles and the number of iterations while keeping the computational effort relatively constant. Using the effective sample size per time as the criterion, we then compared the output of the PMMH using the joint formulation for the likelihood estimate with the same when the factorised formulation is used. For the joint formulation, we used the results from the run detailed in Section 3.3.2.3. We used the same settings for the algorithm here but tested only a subset of the range of the number of particles, since we were able to utilise the results from Chapter 3 to limit the search for the optimal number of particles.

| | $\theta_1$ Prior mean | $\theta_2$ | $\theta_3$ | $\theta_4$ Posterior mean |
|---|---|---|---|---|
| $\phi_{p,\max}$ | 0.62 | 0.55 | 0.52 | 0.48 |
| $\phi_a$ | 0.90 | 0.93 | 0.94 | 0.95 |
| $\alpha$ | 0.83 | 0.86 | 0.88 | 0.90 |
| $\chi_{NS}$ | 20000 | 17750 | 16625 | 15500 |
| $\chi_{IH}$ | 5000 | 4055 | 3582 | 3110 |
| $\chi_{OH}$ | 15000 | 13350 | 12525 | 11700 |
| $\chi_{Ork}$ | 40000 | 28900 | 23350 | 17800 |
| $\rho$ | 10.00 | 7.97 | 6.96 | 5.95 |
| $\tau$ | 140 | 126 | 119 | 112 |
| $\omega$ | 1.70 | 1.70 | 1.70 | 1.70 |

**Table 6.1:** Parameter vectors used for the likelihood estimation

## 6.3 Results

### 6.3.1 Likelihood Estimation

| | Factorised | | Joint | |
|---|---|---|---|---|
| $N$ | CV | Var(log-L) | CV | Var(log-L) |
| 3 | 8.40 | 2718.19 | 5.80 | 3289.02 |
| 10 | 7.88 | 228.56 | 8.99 | 1369.14 |
| 30 | 5.58 | 21.76 | 7.86 | 744.02 |
| 100 | 2.55 | 4.81 | 8.32 | 320.53 |
| 300 | 1.75 | 1.70 | 6.07 | 101.71 |
| 1000 | 0.85 | 0.69 | 5.01 | 23.58 |
| 3000 | 0.55 | 0.29 | 2.88 | 6.62 |
| 10000 | 0.38 | 0.13 | 2.50 | 2.51 |
| 30000 | 0.29 | 0.07 | 0.85 | 0.80 |
| 100000 | 0.23 | 0.04 | 0.54 | 0.22 |
| log-L | -807.11 | | -806.33 | |

**Table 6.2:** Measures of variation of 100 likelihood estimates computed by a particle filter with varying numbers of particles $N$. The likelihood was estimated with a particle filter at the posterior mean ($\theta_4$ in Table 6.2), using the factorised and the joint formulation. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 30,000$ particles.

The runtimes between the two formulations were similar. For small numbers of particles, the joint formulation was slower. This advantage of the factorised formulation was visible for all number of particles between 3 and 1000, where the joint formulation was slower by a factor of between 1.15 and 2.65, although no trend was visible in relation to the number of particles. For $N$ between 3000 and 100,000, the joint formulation was faster and took between 0.56 to 0.85 of the time of the factorised formulation. This could be due to issues with computer memory.

For all four tested parameter vectors, the variability of the likelihood estimates decreased as the number of particle increased for both formulations, as can be seen in Table 6.2 and

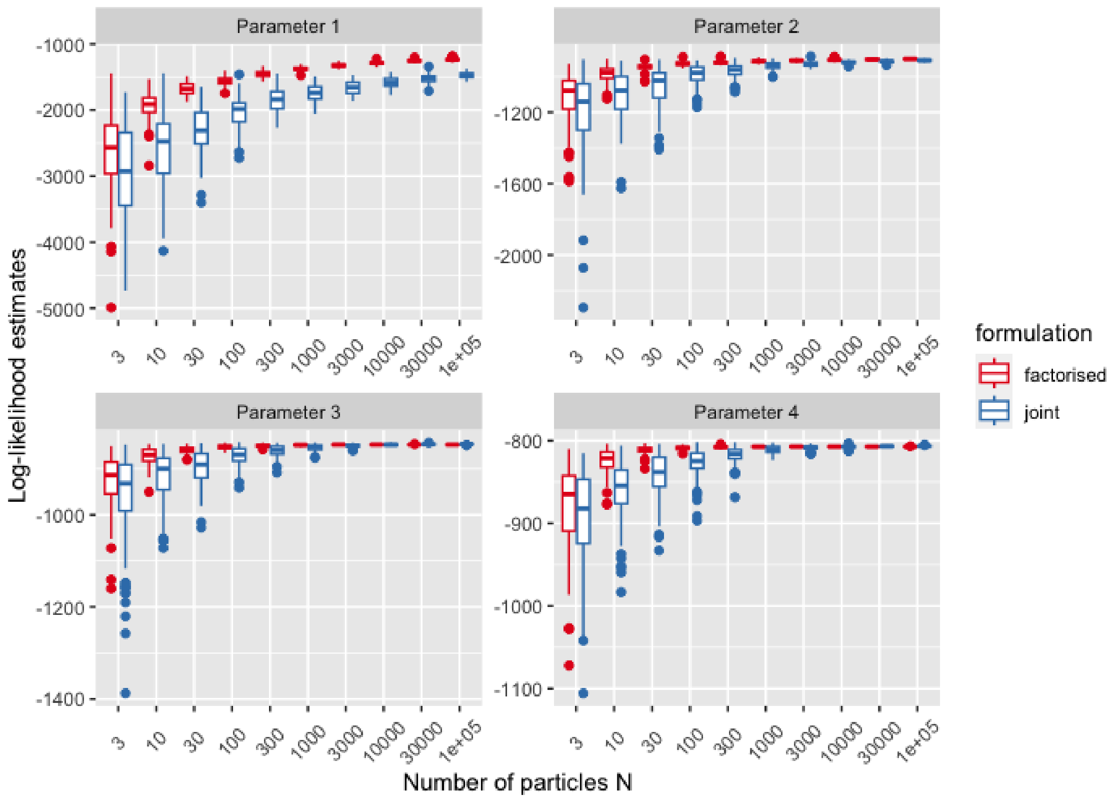
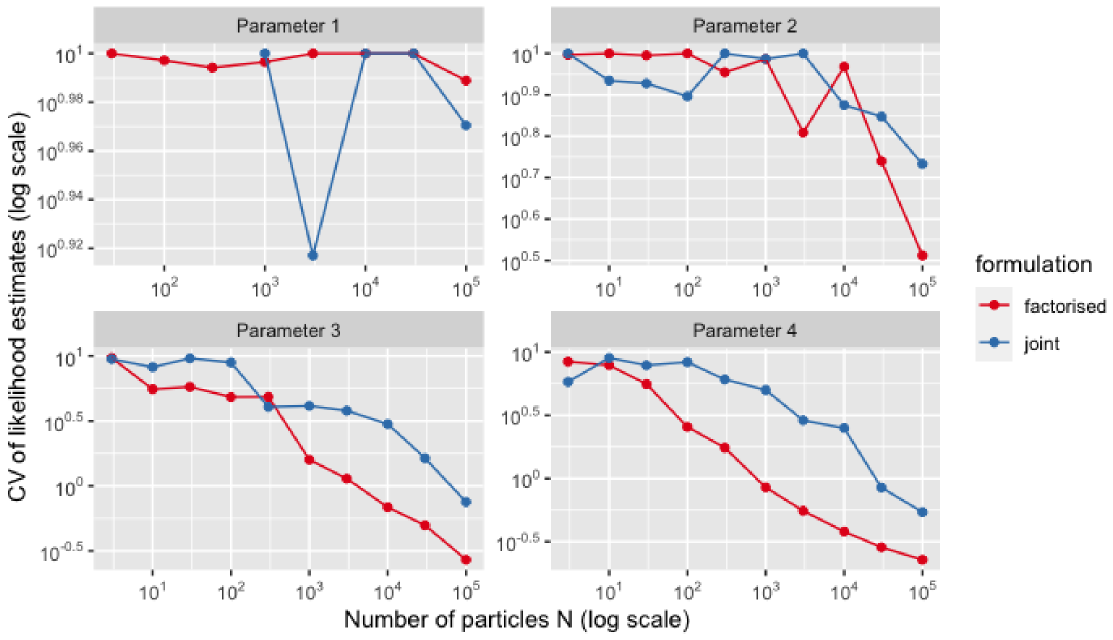
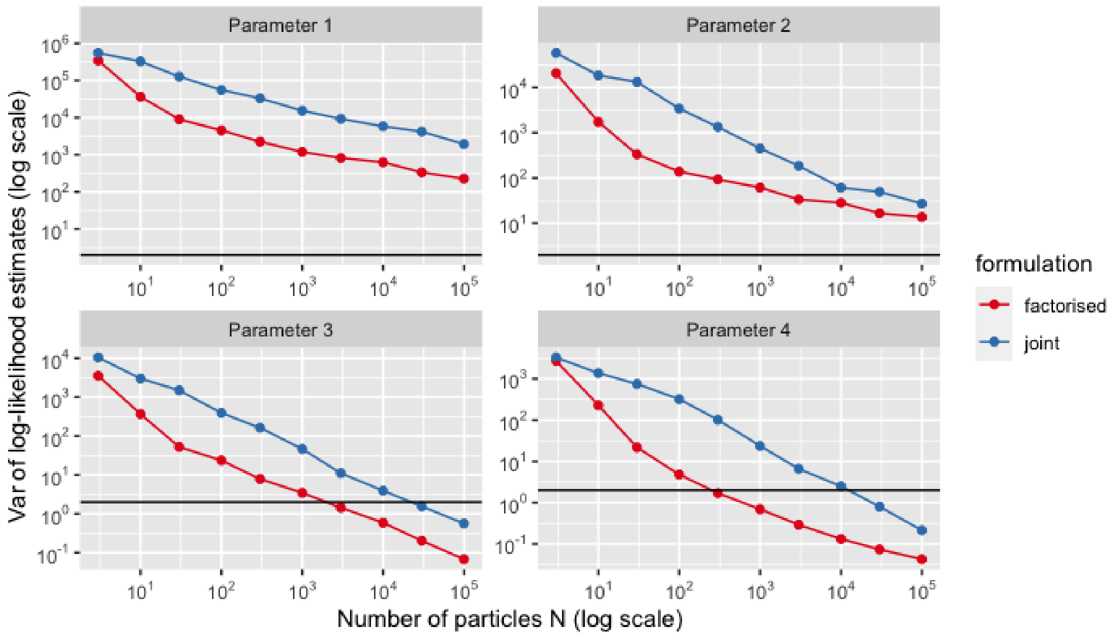


**Figure 6.1:** Boxplots of 100 likelihood estimates of four different parameter values (see Table 6.1) in the complete seal model with independent estimate for varying numbers of particles $N$ and for the two different model formulations.



**(a)** CV of the likelihood estimates.



**(b)** Variance of the log-likelihood estimates.

**Figure 6.2:** Measures of variation of 100 likelihood estimates of four different parameter values (see Table 6.1, Parameters 1 to 4 correspond to $\theta_1$ to $\theta_4$) in the complete seal model with independent estimate for varying numbers of particles $N$ and for the two different model formulations. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates.

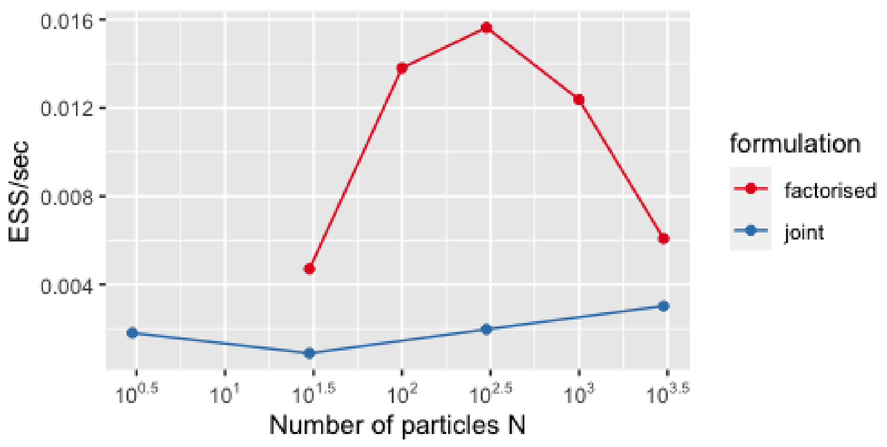


**Figure 6.3:** Relationship between the number of particles $N$ and ESS per runtime when using the PMMH to generate samples from the posterior distribution of the complete seal model with independent estimate, comparing the joint formulation (blue line) with the factorised formulation (red line). The proposal distribution is a multivariate normal distribution on the transformed parameter space with a covariance $h = 0.8$ times the posterior covariance.

Figures 6.1 and 6.2. Table 6.2 shows the CV of the likelihood and the variance of the log-likelihood only for the estimates at the posterior mean ($\theta_4$ in Table 6.1). The tables for the other three parameter vectors can be found in the appendix in Tables F.1, F.2 and F.3. The variance of the log-likelihood was lower for the factorised formulation than for the joint formulation for almost all numbers of particles and parameters. For example for the posterior mean (Parameter 4), the threshold of a variance of less than 2 was passed with 300 particles for the factorised formulation but only with 30,000 particles for the joint formulation. In general, the factorised formulation needed about 30 times fewer particles to obtain a similar variability of the likelihood estimates.

We note that Figures 6.1 and 6.2 reveal large differences between the four parameters. Their mean likelihood value and the variability of the estimates differed greatly. For the prior mean (Parameter 1 in Figure 6.2), the CV seemed to be affected by a particularly high likelihood estimate for $N = 3000$ particles. Nevertheless, the factorised formulation showed an advantage even for the parameters with high-variance estimates.

### 6.3.2 Posterior Distribution

| $N$ | $M$ | Runtime (sec) | ESS | ESS/sec | $\hat{R}$ |
|---:|---:|---:|---:|---:|---:|
| 30 | 4000000 | 122631.04 | 577.59 | 0.0047 | 1.09 |
| 100 | 2000000 | 105500.60 | 1456.30 | 0.0138 | 1.03 |
| 300 | 1000000 | 114815.45 | 1795.14 | 0.0156 | 1.02 |
| 1000 | 300000 | 100374.46 | 1241.47 | 0.0124 | 1.01 |
| 3000 | 100000 | 107275.54 | 652.34 | 0.0061 | 1.03 |

**Table 6.3:** Results from running the PMMH for the complete seal model with independent estimate, using the factorised formulation for the likelihood estimation. The algorithms settings were the ones that led to the two best results in Section 3.3.2.3. The number of particles $N$ for the likelihood estimation was varied between $N = 3$ and $N = 3,000$ with the number of iterations $M$ of the Markov chain adapted to ensure comparable computation times.

The PMMH algorithm appeared to have converged for all tested numbers of particles, as can be seen from the $\hat{R}$ value in Table 6.3 and which was confirmed by inspecting traceplots of the two chains. Figure 6.3 and Table 6.3 show that the effective sample size per second was highest for $N = 300$ particles. We also note that the effective sample size per second was higher for the factorised formulation than for the joint formulation across all tested numbers of particles. The best value for the joint formulation was achieved with $N = 3000$. Comparing the two best values for both formulations, we see that the factorised formulation generated a 5.17 times higher ESS than the joint formulation in the same time.

## 6.4 Discussion and Conclusion

In this chapter, we built on the findings in Section 2.5, where we found that estimating the likelihood separately for the four independent regions and then combining the four values resulted in a much lower variance of the likelihood estimates. We first described three options for including the independent estimate, which leads to dependence between the four regions, in the likelihood estimation while retaining some of the benefits of the factorised formulation. One of these options was implemented where the likelihood was estimated in separate particle filters until the time point of the independent estimate and then combined for the remaining iterations.

We found that there is a large benefit to using the factorised structure of the likelihood even in the modified version when the shared data point of the independent estimate $y_{IE}$ is incorporated using the embedded formulation. For the likelihood estimation at the posterior mean, we found that about 30 times fewer particles were necessary to achieve a similar variability of the likelihood estimates compared with the joint formulation. For parameter estimation, we measured the gain in efficiency by calculating the effective sample size per second and found that the factorised formulation produced a more than 5 times higher ESS per second compared to the joint formulation, when the best results for each of the formulations were compared.

In light of these results, the factorised formulation should be implemented instead of the joint formulation going forward, both for parameter and for likelihood estimation.

In further research, we aim to implement the other two options and compare the performance of the three. We also want to generalise this idea by exploring the advantage of using the factorisation when the data contains more than just one data point that combines the otherwise independent regions. We suspect that the benefit of using a factorised formulation might depend on time points at which the shared data points occur, and investigating this relationship is a further direction of research. In case of the UK grey seals, two more recent estimates of total adult seal population have been undertaken since 2010. Incorporating these into the factorised formulation is a first step.

# Chapter 7

# Concluding Remarks and Further Research

The aim of this thesis was to explore methods for inference for Bayesian non-linear and non-Gaussian state space models, using the case study of the UK grey seal population dynamics model. We conclude with a summary of the key findings and an outlook to future research.

## 7.1 Summary

Chapters 2 and 3 were dedicated to SMC methods for likelihood estimation and parameter inference. We found that the distribution of likelihood estimates derived with a bootstrap filter can be heavily right-skewed and can exhibit a large variance, in particular at parameter values with a low true likelihood value. While small improvements could be achieved by adjusting the resampling frequency of the particles in a bootstrap filter, a clear decrease was only possible by increasing the computational power, i.e., the number of particles, devoted to the estimation procedure.

For parameter inference with the PMMH algorithm, we found that the optimal number of particles increased with the complexity of the model. For the 2- and the 7-state model, the highest ESS per runtime was achieved by estimating the likelihood with 30 and 10 particles, respectively. However, for the complete seal model using the real data, 3000 particles per likelihood estimate were necessary optimise the ESS per runtime. Even with this large number of particles, occasional high likelihood estimates led to poor mixing due to long runs of unchanging parameter values in the Markov chain, requiring a runtime of approximately 11 days in 10 parallel processes to achieve an ESS of 4799. This high runtime is in many cases infeasible and decreases the usefulness of the algorithm for many applications.

While re-estimation of the likelihood each time an acceptance probability is calculated seems like a possible solution to the problem of high likelihood estimates, we showed that re-estimating at every iteration of the Markov chain leads to a clear bias. Re-estimating less frequently might reduce this bias and is sometimes done in the literature, but we showed on a toy example that even rare re-estimations can heavily bias the resulting estimate of the posterior distribution estimate.

Other options for parameter inference were briefly explored but showed no promise for the UK grey seal model. Among these were the SMC$^2$ algorithm and the Liu-West algorithm with improved initial samples.

A deterministic alternative to SMC methods is the Kalman filter which calculates the exact likelihood for linear and Gaussian SSMs. We investigated in Chapter 4 the appropriateness of this algorithm when applied to an NDLM approximation of the UK grey seal model. To obtain a sample of the approximated posterior distribution, we used the likelihood obtained with the Kalman filter within a Metropolis-Hastings algorithm (KFMH). We found that in many cases, the approximated posterior distribution was close to the true posterior distribution while greatly reducing the computational cost. For the complete seal model using the real data, the multivariate ESS per runtime of the KFMH algorithm was 5.37 ESS/sec, while it was 0.0030 ESS/sec when the PMMH was used. However, there remained a clear difference between the KFMH approximation and the true posterior distribution. We also note that implementing the Kalman filter required lengthy calculations to derive the NDLM approximation of the true model. Whether the approximation is useful therefore depends on the application. We suggest using the KFMH when the posterior distribution needs to be estimated several times, e.g., for model selection.

In Chapter 5, we used the PMMH to examine the effect of observation error variance and process stochasticity on the posterior distribution. We found that challenges with parameter inference are likely due to the identifiability issues of the two parameters $\phi_{p,\max}$ and $\phi_a$. These issues could not be resolved by increasing the observation accuracy, increasing the length of the observation time series or by adding observations of another age class. We also investigated whether increasing the variation in the underlying state process with a random effect affected parameter inference. Marginal posterior parameter distributions showed no clear change but the estimated correlation coefficients between the parameters showed a larger variance when calculated for many simulations. State estimation was unaffected by the identifiability issues and had a lower error when observation accuracy was increased. Even when a large random effect was placed on one of the parameters, the estimated mean smoothed states had a similar error as when the true parameter values were used for the estimation.

Finally, we explored in Chapter 6 the idea of factorising the likelihood into four distinct components to reduce the computational effort. By estimating the four components separately for the first 23 of the 26 observed years and only combining them for the independent estimate in year 24 and the last two pup production estimates, we were able to reduce the computational effort for estimating the parameter posterior distribution by a factor of 5.

## 7.2 Potential Future Research

Based on the results in this thesis, we identify areas that may be of interest for future research.

Within the area of SMC metods, there are some avenues for further improvements that have not fully been explored yet. A promising algorithm is the correlated pseudo-marginal algorithm of Deligiannidis and Doucet (2018) where the likelihood estimates within a PMMH algorithm for two successive parameter values are made dependent on one another. This has been shown to decrease the likelihood estimate variance by guiding the state particles towards higher likelihood areas given the observations and the parameter value.

While we are cautious about re-estimating the likelihood in the PMMH algorithm due to the introduced bias, further research could highlight cases in which this bias is outweighed by a large reduction in Monte Carlo variance. Since long runs without acceptance of a new parameter value were prevalent when the PMMH was applied to the complete seal model and led to a low effective sample size, investigating this approach further is warranted.

For the SMC$^2$ algorithm, more settings could be trialled to explore the usefulness of this algorithm for the seal model. One factor to be investigated is the balance between the state particles assigned to each parameter for likelihood estimation, and the number of parameter particles. Other factors are increasing the number of state particles for each parameter particle as the number of iterations increases, and tuning the proposal distribution for the parameter particles.

While initial attempts to improve the Kalman filter approximation of the posterior distribution were not successful, this could be further investigated by using other techniques such as the extended Kalman filter. In addition, it would be helpful to simplify or automate the process of linearising and normalising the model. This would facilitate the use of this approximation for fast inference when exact posterior distributions are not required.

A technique that was outside of the scope of this thesis is using the Laplace approximation to estimate the likelihood. This would allow the use of existing software such as the `R` package TMB (Kristensen et al., 2016).

Building on the results in Chapter 5, it is of interest to further investigate the identifiability issues with some of the model parameters. These could be analytically assessed, e.g., using the methods developed by Cole and McCrea (2016). Potential solutions to these identifiability issues might be a re-parametrisation of the model, fixing the observation error variance to an externally estimated value using the pup production model of Russell et al. (2019), and—in the long term—collecting more data on intermediate life stages of the population, such as on pups after their survival process.

Finally, it seems promising to expand the idea of exploiting the factorised model structure even when a shared data point is available to improve likelihood estimation with SMC methods. In Chapter 6, only one of three theoretical options was investigated. Investigating all three options might lead to an even greater decrease in computation time. The factorisation technique could also be explored under different conditions, such as varying the time point of the shared data point or including more such data points. This might lead to a general strategy in the larger context of combining two sources of data that impose different dependency structures on the underlying states.

## 7.3 Concluding Remarks

In the immediate future, we hope to use the results in Chapter 3 combined with the factorising approach of Chapter 6 for future inference of the UK grey seal model. For further development of the model, we point to the Kalman filter as a robust and fast inference method. We hope that illustrating the various methods in this thesis is not only helpful for the further work with the UK grey seal population model but also for challenging inference problems for state space models in ecology and other areas.

# Bibliography

Anderson, T. W. (1958). *An Introduction to Multivariate Statistical Analysis*, volume 2. Wiley New York.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725.

Auger-Méthé, M., Field, C., Albertsen, C. M., Derocher, A. E., Lewis, M. A., Jonsen, I. D., and Flemming, J. M. (2016). State-space models' dirty little secrets: even simple linear Gaussian models can have estimation problems. *Scientific Reports*, 6(1).

Auger-Méthé, M., Newman, K., Cole, D., Empacher, F., Gryba, R., King, A. A., Leos-Barajas, V., Mills Flemming, J., Nielsen, A., Petris, G., and Thomas, L. (2021). A guide to state–space modeling of ecological time series. *Ecological Monographs*, 91(4):e01470.

Beraha, M., Falco, D., and Guglielmi, A. (2021). JAGS, NIMBLE, Stan: a detailed comparison among Bayesian MCMC software.

Besbeas, P. and Morgan, B. J. T. (2020). A general framework for modeling population abundance data. *Biometrics*, 76(1):281–292.

Beverton, R. J. H. and Holt, S. J. (1957). *On the dynamics of exploited fish populations.* Great Britain. Ministry of Agriculture, Fisheries and Food. Fishery investigations; ser.II, v.19. H.M. Stationery Office, London.

Bickel, P. J. and Blackwell, D. (1967). A note on Bayes estimates. *The Annals of Mathematical Statistics*, 38(6):1907–1911.

Borowska, A. and King, R. (2023). Semi-complete data augmentation for efficient state space model fitting. *Journal of Computational and Graphical Statistics*, 32(1):19–35.

Brooks, S. P. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455.

Buchholz, A., Chopin, N., and Jacob, P. E. (2021). Adaptive tuning of Hamiltonian Monte Carlo within sequential Monte Carlo. *Bayesian Analysis*, 16(3):745 – 771.

Buckland, S., Newman, K., Thomas, L., and Koesters, N. (2004). State-space models for the dynamics of wild animal populations. *Ecological Modelling*, 171(1):157–175.

Chernoff, H. (1972). *Sequential Analysis and Optimal Design.* Society for Industrial and Applied Mathematics.

Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411.

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). SMC$^2$ : an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 75(3):397–426.

Chopin, N. and Papaspiliopoulos, O. (2020). *An Introduction to Sequential Monte Carlo.* Springer.

Chopin, N., Ridgway, J., Gerber, M., and Papaspiliopoulos, O. (2015). Towards automatic calibration of the number of state particles within the SMC$^2$ algorithm. *arXiv: 1506.00570.*

Cole, D. J. and McCrea, R. S. (2016). Parameter redundancy in discrete state-space and integrated models. *Biometrical Journal*, 58(5):1071–1090.

de Valpine, P., Paciorek, C., Turek, D., Michaud, N., Anderson-Bergman, C., Obermeyer, F., Wehrhahn Cortes, C., Rodrìguez, A., Temple Lang, D., and Paganin, S. (2022). *NIMBLE User Manual.* R package manual version 0.13.0.

de Valpine, P., Paciorek, C., Turek, D., Michaud, N., Anderson-Bergman, C., Obermeyer, F., Wehrhahn Cortes, C., Rodrìguez, A., Temple Lang, D., and Paganin, S. (2023). *NIMBLE: MCMC, Particle Filtering, and Programmable Hierarchical Modeling.* R package version 1.0.1.

de Valpine, P., Turek, D., Paciorek, C., Anderson-Bergman, C., Temple Lang, D., and Bodik, R. (2017). Programming with models: writing statistical algorithms for general model structures with NIMBLE. *Journal of Computational and Graphical Statistics*, 26:403–417.

Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems With Applications*, volume 100. Springer.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436.

Deligiannidis, G. and Doucet, A. (2018). The correlated pseudomarginal method. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 80(5):pp. 839–870.

Douc, R., Cappé, O., and Moulines, E. (2005). Comparison of resampling schemes for particle filtering. *CoRR*, abs/cs/0507025.

Doucet, A., De Freitas, N., and Gordon, N. (2001). An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.

Doucet, A. and Johansen, A. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12.

Doucet, A., Pitt, M. K., Deligiannidis, G., and Kohn, R. (2015). Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 102(2):295–313.

Dubbin, G. and Blunsom, P. (2012). Unsupervised Bayesian part of speech inference with particle Gibbs. In Flach, P. A., De Bie, T., and Cristianini, N., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 760–773, Berlin, Heidelberg. Springer Berlin Heidelberg.

Duck, C. (2007). *Seals: Naturally Scottish.* Scottish Natural Heritage.

Durbin, J. and Koopman, S. J. (2012). *Time Series Analysis by State Space Methods.* Oxford University Press.

Empacher, F. (2017). The Kalman filter and extensions. inferring animal movement from low-fidelity animal-borne tags.

Empacher, F. (2023). Nimble Implementation of 2-State Model Kalman Filter.

Fagard-Jenkin, C. (2024). PhD thesis in preparation, University of St Andrews. unpublished.

Fasiolo, M., Pya, N., and Wood, S. N. (2016). A Comparison of Inferential Methods for Highly Nonlinear State Space Models in Ecology and Epidemiology. *Statistical Science*, 31(1):96–118.

Fernández-Villaverde, J. and Rubio-Ramírez, J. F. (2007). Estimating macroeconomic models: A likelihood approach. *The Review of Economic Studies*, 74(4):1059–1087.

Finke, A., King, R., Beskos, A., and Dellaportas, P. (2019). Efficient sequential Monte Carlo algorithms for integrated population models. *Journal of Agricultural, Biological and Environmental Statistics*, 24(2):204–224.

Flegal, J. M., Hughes, J., Vats, D., Dai, N., Gupta, K., and Maji, U. (2021). *MCMCse: Monte Carlo Standard Errors for MCMC.* Riverside, CA, and Kanpur, India. R package version 1.5-0.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. (2013). *Bayesian Data Analysis, Third Edition.* Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis.

Gelman, A., Gilks, W. R., and Roberts, G. O. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target—Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(1):127–146.

Golightly, A., Bradley, E., Lowe, T., and Gillespie, C. S. (2019). Correlated pseudo-marginal schemes for time-discretised stochastic kinetic models. *Computational Statistics and Data Analysis*, 136:92–107.

Golightly, A. and Wilkinson, D. (2011). Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface focus*, 1:807–20.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140:107–113(6).

Grewal, M. S. and Andrews, A. P. (2010). Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Systems Magazine*, 30(3):69–78.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242.

Hall, A. J. and Russell, D. J. (2018). Gray seal: Halichoerus grypus. In Würsig, B., Thewissen, J., and Kovacs, K. M., editors, *Encyclopedia of Marine Mammals (Third Edition)*, pages 420–422. Academic Press, third edition edition.

Harvey, A. (1990). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press.

Harwood, J. and Prime, J. H. (1978). Some factors affecting the size of British grey seal populations. *Journal of Applied Ecology*, 15(2):401–411.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Houston, A. I., Stephens, P. A., Boyd, I. L., Harding, K. C., and McNamara, J. M. (2006). Capital or income breeding? A theoretical model of female reproductive strategies. *Behavioral Ecology*, 18(1):241–250.

Iannelli, M. and Milner, F. (2017). *The Basic Approach to Age-Structured Population Dynamics*. Lecture Notes on Mathematical Modelling in the Life Sciences. Springer.

Ionides, E. L., Nguyen, D., Atchadé, Y., Stoev, S., and King, A. A. (2015). Inference for dynamic and latent variable models via iterated, perturbed Bayes maps. *Proceedings of the National Academy of Sciences*, 112(3):719–724.

Jonsen, I. D., Flemming, J. M., and Myers, R. A. (2005). Robust state–space modeling of animal movement data. *Ecology*, 86(11):2874–2880.

Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., and Chopin, N. (2015). On particle methods for parameter estimation in state-space models. *Statist. Sci.*, 30(3):328–351.

Kattwinkel, M. and Reichert, P. (2017). Bayesian parameter inference for individual-based models using a particle markov chain monte carlo method. *Environmental Modelling & Software*, 87:110–119.

King, R. (2011). *Handbook of Markov Chain Monte Carlo*. CRC Press, United States.

King, R. (2012). A review of Bayesian state-space modelling of capture–recapture–recovery data. *Interface Focus*, 2(2):190–204.

Kitagawa, G. (1998). A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215.

Kitagawa, G. and Sato, S. (2001). *Monte Carlo Smoothing and Self-Organising State-Space Model*, pages 177–195. Springer New York, New York, NY.

Knape, J. (2008). Estimability of density dependence in models of time series data. *Ecology*, 89(11):2994–3000.

Knape, J. and de Valpine, P. (2012). Fitting complex population models by combining particle filters with Markov chain Monte Carlo. *Ecology*, 93(2):256–263.

Knape, J., Jonzén, N., and Sköld, M. (2011). On observation distributions for state space models of population survey data. *Journal of Animal Ecology*, 80(6):1269–1277.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288.

Kristensen, K., Nielsen, A., Berg, C. W., Skaug, H., and Bell, B. M. (2016). TMB: Automatic differentiation and Laplace approximation. *Journal of Statistical Software*, 70(5):1–21.

Lambert, R. A. (2002). The grey seal in Britain: A twentieth century history of a nature conservation success. *Environment and History*, 8(4):449–474.

Lefebvre, T., Bruyninckx, H., and Schutter, J. D. (2004). Kalman filters for non-linear systems: A comparison of performance. *International Journal of Control*, 77(7):639–653.

Lele, S. R., Dennis, B., and Lutscher, F. (2007). Data cloning: easy maximum likelihood estimation for complex ecological models using Bayesian Markov chain Monte Carlo methods. *Ecology Letters*, 10(7):551–563.

Lindsten, F. and Schön, T. B. (2013). Backward simulation methods for Monte Carlo statistical inference. *Found. Trends Mach. Learn.*, 6(1):1–143.

Liu, J., Wang, W., and Ma, F. (2011). A regularized auxiliary particle filtering approach for system state estimation and battery life prediction. *Smart Materials and Structures*, 20(7):075021.

Liu, J. and West, M. (2001). Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo methods in practice*, pages 197–223. Springer.

Liu, J. S. (2001). *Monte Carlo strategies in scientific computing.* Springer Science & Business Media.

Lonergan, M., Duck, C., Thompson, D., Moss, S., and Mcconnell, B. (2011). British grey seal (halichoerus grypus) abundance in 2008: An assessment based on aerial counts and satellite telemetry. *ICES Journal of Marine Science*, 68:2201–2209.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55.

Malthus, T. R. (1798). *An Essay on the Principle of Population.* Number malthus1798 in History of Economic Thought Books. McMaster University Archive for the History of Economic Thought.

Mendelssohn, R. (1988). Some problems in estimating population sizes from catch-at-age data. *Fishery Bulletin*, 86(4).

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Michaud, N., de Valpine, P., Turek, D., Paciorek, C. J., and Nguyen, D. (2021). Sequential Monte Carlo methods in the nimble and nimbleSMC R packages. *Journal of Statistical Software*, 100(3):1–39.

Monnahan, C. C. and Kristensen, K. (2018). No-U-turn sampling for fast Bayesian inference in ADMB and TMB: Introducing the adnuts and tmbstan R packages. *PLOS ONE*, 13(5):1–10.

Moore, J. B. and Anderson, B. (1979). *Optimal Filtering.* Prentice-Hall.

Moral, P. D., Doucet, A., and Jasra, A. (2012). On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli*, 18(1):252 – 278.

Murray, L. M., Lee, A., and Jacob, P. E. (2016). Parallel resampling in the particle filter. *Journal of Computational and Graphical Statistics*, 25(3):789–805.

Neal, R. (2012). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo.*

Neal, R. M. (2003). Slice sampling. *The Annals of Statistics*, 31(3):705 – 767.

Newman, K., King, R., Elvira, V., de Valpine, P., McCrea, R. S., and Morgan, B. J. T. (2023). State-space models for ecological time-series data: Practical model-fitting. *Methods in Ecology and Evolution*, 14(1):26–42.

Newman, K. B., Buckland, S. T., Morgan, B. J., King, R., Borchers, D. L., Cole, D. J., Besbeas, P., Gimenez, O., and Thomas, L. (2014a). *Modelling population dynamics: model formulation, fitting and assessment using state-space methods.* Springer.

Newman, K. B., Buckland, S. T., Morgan, B. J. T., King, R., Borchers, D. L., Cole, D. J., Besbeas, P., Gimenez, O., and Thomas, L. (2014b). *Fitting State-Space Models*, pages 51–82. Springer New York, New York, NY.

Newman, K. B., Buckland, S. T., Morgan, B. J. T., King, R., Borchers, D. L., Cole, D. J., Besbeas, P., Gimenez, O., and Thomas, L. (2014c). *Introduction*, pages 1–5. Springer New York, New York, NY.

Newman, K. B., Buckland, S. T., Morgan, B. J. T., King, R., Borchers, D. L., Cole, D. J., Besbeas, P., Gimenez, O., and Thomas, L. (2014d). *Matrix Models as Building Blocks for Population Dynamics*, pages 7–37. Springer New York, New York, NY.

NIMBLE Development Team (2021). nimbleSMC: Sequential Monte Carlo methods for NIMBLE. R package version 0.10.1.

Osada, Y., Kuriyama, T., Asada, M., Yokomizo, H., and Miyashita, T. (2019). Estimating range expansion of wildlife in heterogeneous landscapes: A spatially explicit state-space matrix model coupled with an improved numerical integration technique. *Ecology and Evolution*, 9(1):318–327.

Papoulis, A. and Pillai, S. (2002). *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill series in electrical and computer engineering. McGraw-Hill.

Pastore, M. and Calcagnì, A. (2019). Measuring distribution similarities between samples: A distribution-free overlapping index. *Frontiers in Psychology*, 10.

Persson, S., Welkenhuysen, N., Shashkova, S., Wiqvist, S., Reith, P., Schmidt, G. W., Picchini, U., and Cvijovic, M. (2022). Scalable and flexible inference framework for stochastic dynamic single-cell models. *PLOS Computational Biology*, 18(5):1–24.

Pitt, M. K., dos Santos Silva, R., Giordani, P., and Kohn, R. (2012). On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2):134–151. Bayesian Models, Methods and Applications.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599.

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using gibbs sampling. *3rd International Workshop on Distributed Statistical Computing (DSC 2003); Vienna, Austria*, 124.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). Coda: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11.

Poyiadjis, G., Doucet, A., and Singh, S. (2005). Particle methods for optimal filter derivative: application to parameter estimation. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v/925–v/928 Vol. 5.

R Core Team (2023). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Reich, B. and Ghosh, S. (2019). *Bayesian Statistical Methods.* ASA-CRC series on statistical reasoning in science and society. CRC Press, Taylor & Francis Group.

Ross, S. (2011). *A First Course in Probability.* Pearson Education.

Russell, D. J. F., Duck, C. D., Morris, C., and Thompson, D. (2016). *Independent estimates of grey seal population size: 2008 and 2014: Special Committee on Seals. Briefing Paper 16/03.* Sea Mammal Research Unit.

Russell, D. J. F., McConnell, B., Thompson, D., Duck, C., Morris, C., Harwood, J., and Matthiopoulos, J. (2013). Uncovering the links between foraging and breeding regions in a highly mobile mammal. *Journal of Applied Ecology*, 50(2):499–509.

Russell, D. J. F., Morris, C. D., Duck, C. D., Thompson, D., and Hiby, L. (2019). Monitoring long-term changes in UK grey seal pup production. *Aquatic Conservation: Marine and Freshwater Ecosystems*, 29(S1):24–39.

Schön, T. B., Gustafsson, F., and Nordlund, P.-J. (2005). Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289.

Schön, T. B. and Lindsten, F. (2017). Learning of dynamical systems. Unpublished course notes for the 2017 PhD course on Sequential Monte Carlo methods by Uppsala University.

Schön, T. B., Svensson, A., Murray, L., and Lindsten, F. (2018). Probabilistic learning of nonlinear dynamical systems using sequential Monte Carlo. *Mechanical Systems and Signal Processing*, 104:866–883.

Sherlock, C., Thiery, A. H., Roberts, G. O., and Rosenthal, J. S. (2015). On the efficiency of pseudo-marginal random walk Metropolis algorithms. *The Annals of Statistics*, 43(1):238–275.

Slutsky, E. (1925). Über stochastische Asymptoten und Grenzwerte. Metron 5, Nr. 3, 3–89 (1925).

Smout, S., King, R., and Pomeroy, P. (2020). Environment-sensitive mass changes influence breeding frequency in a capital breeding marine top predator. *Journal of Animal Ecology*, 89(2):384–396.

Soulsby, R. L. and Thomas, J. A. (2012). Insect population curves: modelling and application to butterfly transect data. *Methods in Ecology and Evolution*, 3(5):832–841.

Special Committee on Seals (2021). Scientific advice on matters related to the management of seal populations: 2021. Technical report, SNatural Environment Research Council.

Spiegelhalter, D., Thomas, A., Best, N., and Lunn, D. (2003). *Winbugs 1.4. User Manual.* MRC Biostatistics Unit.

Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540.

Thomas, L. (2021). Estimating the size of the UK grey seal population between 1984 and 2020. Technical Report SCOS-BP 21/05, Scottish Oceans Institute and Centre for Research into Ecological and Environmental Modelling.

Thomas, L. and Harwood, J. (2003). Estimating grey seal population size using a Bayesian state-space model. Technical report, Scottish Oceans Institute and Centre for Research into Ecological and Environmental Modelling.

Thomas, L. and Harwood, J. (2005). Estimating the size of the UK grey seal population between 1984 and 2004: model selection, survey effort and sensitivity to priors. Technical report, Scottish Oceans Institute and Centre for Research into Ecological and Environmental Modelling.

Thomas, L. and Harwood, J. (2008). Estimating the size of the UK grey seal population between 1984 and 2007. Technical report, Scottish Oceans Institute and Centre for Research into Ecological and Environmental Modelling.

Thomas, L., Russell, D. J., Duck, C. D., Morris, C. D., Lonergan, M., Empacher, F., Thompson, D., and Harwood, J. (2019). Modelling the population size and dynamics of the British grey seal. *Aquatic Conservation: Marine and Freshwater Ecosystems*, 29(S1):6–23.

Turchin, P. (2001). Does population ecology have general laws? *Oikos*, 94(1):17–26.

Unit, S. M. R. (2019). Preface to sea mammal research unit 40th anniversary issue. *Aquatic Conservation: Marine and Freshwater Ecosystems*, 29(S1):5–5.

van Neer, A., Jensen, L. F., and Siebert, U. (2015). Grey seal (halichoerus grypus) predation on harbour seals (phoca vitulina) on the island of Helgoland, Germany. *Journal of Sea Research*, 97:1–4.

Vats, D., Flegal, J. M., and Jones, G. L. (2019). Multivariate output analysis for Markov chain Monte Carlo. *Biometrika*, 106(2):321–337.

Wan, E. A. and van der Merwe, R. (2001). *The Unscented Kalman Filter*, chapter 7, pages 221–280. Wiley Online Library.

West, M. and Harrison, J. (1997). *The Dynamic Linear Model*, pages 97–142. Springer New York, New York, NY.

White, J. W., Nickols, K. J., Malone, D., Carr, M. H., Starr, R. M., Cordoleani, F., Baskett, M. L., Hastings, A., and Botsford, L. W. (2016). Fitting state-space integral projection models to size-structured time series data to estimate unknown parameters. *Ecological Applications*, 26(8):2677–2694.

Whiteley, N. (2013). Stability properties of some particle filters. *The Annals of Applied Probability*, 23(6):2500–2537.

Wills, A. and Schön, T. (2023). Sequential Monte Carlo: A unified review. *Annual Review of Control, Robotics, and Autonomous Systems*, 6.

Yu, C., El-Sheimy, N., Lan, H., and Liu, Z. (2017). Map-based indoor pedestrian navigation using an auxiliary particle filter. *Micromachines*, 8(7).

Zhou, Y., Johansen, A. M., and Aston, J. A. (2016). Toward automatic model comparison: An adaptive sequential Monte Carlo approach. *Journal of Computational and Graphical Statistics*, 25(3):701–726.

# Appendix A

# Chapter 1

## A.1 Data

| Year | North Sea | Inner Hebrides | Outer Hebrides | Orkney |
|---|---|---|---|---|
| 1984 | 1325 | 1332 | 7594 | 4741 |
| 1985 | 1711 | 1190 | 8165 | 5199 |
| 1986 | 1834 | 1711 | 8455 | 5796 |
| 1987 | 1867 | 2002 | 8777 | 6389 |
| 1988 | 1474 | 1960 | 8689 | 5948 |
| 1989 | 1922 | 1956 | 9275 | 6773 |
| 1990 | 2278 | 2032 | 9801 | 6982 |
| 1991 | 2375 | 2411 | 10617 | 8653 |
| 1992 | 2436 | 2816 | 12215 | 9854 |
| 1993 | 2710 | 2923 | 11915 | 11034 |
| 1994 | 2652 | 2719 | 12054 | 11851 |
| 1995 | 2757 | 3050 | 12713 | 12670 |
| 1996 | 2938 | 3117 | 13176 | 14531 |
| 1997 | 3698 | 3076 | 11946 | 14395 |
| 1998 | 3989 | 3087 | 12434 | 16625 |
| 1999 | 3380 | 2787 | 11759 | 15720 |
| 2000 | 4303 | 3223 | 13472 | 16546 |
| 2001 | 4134 | 3032 | 12427 | 18196 |
| 2002 | 4520 | 3096 | 11248 | 17952 |
| 2003 | 4870 | 3386 | 12741 | 18652 |
| 2004 | 5015 | 3385 | 12319 | 19123 |
| 2005 | 5232 | 3427 | 12397 | 18126 |
| 2006 | 5484 | 3470 | 11719 | 19332 |
| 2007 | 5771 | 3118 | 11342 | 19184 |
| 2008 | 6501 | 3317 | 12279 | 17813 |
| 2009 | 7360 | - | 11887 | 18548 |
| 2010 | 8119 | 3108 | 11831 | 18582 |

**Table A.1:** Regional pup production estimates from 1984 to 2010. In 2009, there is a missing value for the Inner Hebrides due to too few aerial surveys in that region. The data was provided by the Sea Mammal Research Unit, University of St Andrews, and is published in Table S1 in Thomas et al. (2019).

| Year | Shift parameter $\kappa_0$ | Shape parameter $\kappa_1$ | Scale parameter $\kappa_2$ |
|---|---|---|---|
| 2008 | 59167.84 | 12.96 | 2719.38 |

**Table A.2:** Parameters of the shifted Gamma distribution used in Thomas et al. (2019) to model the uncertainty in the independent estimate derived by Russell et al. (2016).

## A.2 Proofs

### A.2.1 Carrying Capacity

*Proof.* We proof the relationship between the pup carrying capacity $\chi_r$ and the parameter $\beta_r$ in the Beverton-Holt function

$$\phi_{p,r,t} = \frac{\phi_{p\,\max}}{1 + (\beta_r x_{0,r,t-1})^{\rho}}, \tag{A.1}$$

which is

$$\chi_r = \frac{1}{\beta_r}\left(\frac{0.5\alpha\phi_{p\,\max}\phi_a^5}{1-\phi_a} - 1\right)^{1/\rho} \tag{A.2}$$

as stated in Equation 1.5. When the population is in equilibrium and random variability is ignored (treating probabilities as exact rates), there should be a stable age structure where the number of newborn pups each year is exactly the pup carrying capacity $\chi_r$:

$$\chi_r = x_{0,r,t} = \alpha x_{6,r,t} \tag{A.3}$$

If this is the number of newborn pups every year, then the rest of the population structure is as follows, where we use the properties of a geometric series on the last line:

$$\begin{aligned}
x_{0,r,t} &= \chi_r, \\
x_{1,r,t} &= 0.5\chi_r\phi_{p,r,t}, \\
x_{2,r,t} &= 0.5\chi_r\phi_{p,r,t}\phi_a, \\
x_{3,r,t} &= 0.5\chi_r\phi_{p,r,t}\phi_a^2, \\
&\vdots \\
x_{6,r,t} &= 0.5\chi_r\phi_{p,r,t}(\phi_a^5 + \phi_a^6 + \dots) = 0.5\chi_r\phi_{p,r,t}\phi_a^5\left(\sum_{k=0}^{\infty}\phi_a^k\right) = 0.5\chi_r\phi_{p,r,t}\frac{\phi_a^5}{1-\phi_a}
\end{aligned} \tag{A.4}$$

When substituting Equation A.4 for the value of $x_{6,r,t}$ in Equation A.3 and use the Beverton-Holt function A.1 to derive $\phi_{p,r,t}$, we obtain

$$\frac{\chi_r}{\alpha} = 0.5\chi_r\frac{\phi_{p\,\max}}{1+\beta_r x_{0,r,t-1}}\frac{\phi_a^5}{1-\phi_a}.$$

Solving for $\chi_r$ leads to Equation A.2. □

### A.2.2 Overdispersion of $x_0$

In Thomas et al. (2019), the prior for the number of pups in each region in year 0, $x_{a=0,r,t=0}$ is derived from the observation in year 0, $y_0$. The observation density is effectively reversed but then the data are further dispersed to ensure that all likely values are included in the prior. While unlikely values are easily excluded in the algorithms used here later on, it is harder to add new values (and impossible in some of the algorithms).

The initial pup numbers are sampled as follows, where $a \geq 1$ is the dispersion factor. This factor has been chosen as $a = 1.3$ which was found by trial and error as described in Thomas et al. (2019).

$$\begin{aligned} x^*_{a=0,t=0} &\sim \mathcal{N}(y_0, y_0^2/\tau) \\ x_{a=0,t=0} &\sim \text{Unif}(\frac{x^*_{a=0,t=0}}{a}, ax^*_{a=0,t=0}) \end{aligned}$$

This dispersion factor influences the mean and variance of $x_{a=0,r,t=0}$. The following calculations show this influence. For better readability, the subscript $a = 0, r, t = 0$ is dropped in the following calculations.

$$\begin{aligned} \text{E}(X) = \text{E}(\text{E}(X|X^*)) &= \int |E(X|x^*)p(x^*)dx^* = \iint xp(x|x^*)p(x^*)dxdx^* \\ &= \int p(x^*) \int_{\frac{x^*}{a}}^{ax^*} \frac{x}{\frac{x^*}{a} - ax^*} dxdx^* \\ &= \int \frac{1}{2} p(x^*) \frac{1}{\frac{x^*}{a} - ax^*} \left( (ax^*)^2 - \left( \frac{x^*}{a} \right)^2 \right) dx^* \\ &= \frac{1}{2} \left( a + \frac{1}{a} \right) \int p(x^*)x^* dx^* = \frac{1}{2} \left( a + \frac{1}{a} \right) \mu \end{aligned}$$

$$\begin{aligned} \text{E}(X^2) &= \int p(x^*) \int_{\frac{x^*}{a}}^{ax^*} \frac{x^2}{\frac{x^*}{a} - ax^*} dxdx^* \\ &= \int \frac{1}{3} p(x^*) \frac{1}{\frac{x^*}{a} - ax^*} \left( (ax^*)^3 - \left( \frac{x^*}{a} \right)^3 \right) dx^* \\ &= \frac{1}{3} \left( a^2 + 1 + \frac{1}{a^2} \right) \int p(x^*)(x^*)^2 dx^* \\ &= \frac{1}{3} \left( a^2 + 1 + \frac{1}{a^2} \right) (\sigma^2 + \mu^2) \end{aligned}$$

$$\begin{aligned} \text{Var}(X) &= \text{E}(X^2) - (\text{E}(X))^2 \\ &= \frac{1}{3} \left( a^2 + 1 + \frac{1}{a^2} \right) (\sigma^2 + \mu^2) - \left( \frac{1}{2} \left( a + \frac{1}{a} \right) \mu \right)^2 \\ &= \frac{1}{3} \left( a^2 + 1 + \frac{1}{a^2} \right) \sigma^2 + \frac{1}{12} \left( a - \frac{1}{a} \right)^2 \mu^2 \end{aligned}$$

Setting $a$ to a value other than 1 increases the mean and variance of the prior distribution of $x_{0,pups}$, as

$$\begin{aligned} \frac{1}{2} \left( a + \frac{1}{a} \right) &= \frac{1}{2a} \left( a^2 + 1 \right) = \frac{1}{2a} \left( a^2 - 2a + 1 + 2a \right) \\ &= \frac{1}{2a} \left( (a-1)^2 + 2a \right) \geq 1, \end{aligned}$$

and

$$\left(a - \frac{1}{a}\right)^2 \geq 0$$

# Appendix B

# Chapter 2

## B.1 Results for 2-State Model—Second Dataset

| | True parameter | | False parameter 1 | | False parameter 2 | |
|---|---|---|---|---|---|---|
| $N$ | CV(L) | Var(logL) | CV(L) | Var(log-L) | CV(L) | Var(log-L) |
| 1 | 3.93E+00 | 1.50E+04 | 9.12E+00 | 4.59E+04 | 1.00E+01 | 1.73E+06 |
| 3 | 2.87E+00 | 1.72E+02 | 6.40E+00 | 6.86E+02 | 1.00E+01 | 4.48E+04 |
| 10 | 1.41E+00 | 4.12E+00 | 5.96E+00 | 2.87E+01 | 1.00E+01 | 3.14E+03 |
| 30 | 8.55E-01 | 7.83E-01 | 1.92E+00 | 3.51E+00 | 8.70E+00 | 5.65E+02 |
| 100 | 3.87E-01 | 1.40E-01 | 8.48E-01 | 7.68E-01 | 1.00E+01 | 1.74E+02 |
| 300 | 2.21E-01 | 5.12E-02 | 4.73E-01 | 2.03E-01 | 7.54E+00 | 7.60E+01 |
| 1000 | 1.23E-01 | 1.64E-02 | 3.22E-01 | 1.05E-01 | 5.05E+00 | 3.54E+01 |
| 3000 | 7.66E-02 | 5.83E-03 | 1.69E-01 | 2.86E-02 | 4.89E+00 | 2.33E+01 |
| 10000 | 3.84E-02 | 1.46E-03 | 1.04E-01 | 1.05E-02 | 6.61E+00 | 1.43E+01 |
| 30000 | 2.39E-02 | 5.76E-04 | 6.17E-02 | 3.68E-03 | 6.27E+00 | 6.21E+00 |
| log-L | -331.3 | | -388.3 | | -384.8 | |

**Table B.1:** Measures of variation of 100 likelihood estimates computed by a standard bootstrap filter with varying numbers of particles $N$ for the 2-state model. The likelihood was estimated at the true parameter value and at two false parameter values (defined in Table 2.1). The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 30,000$ particles.

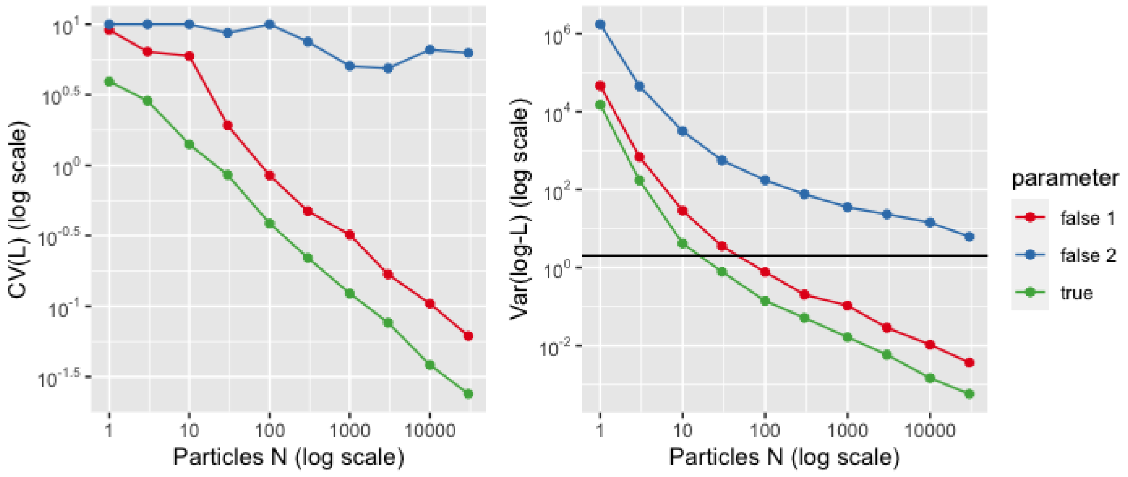
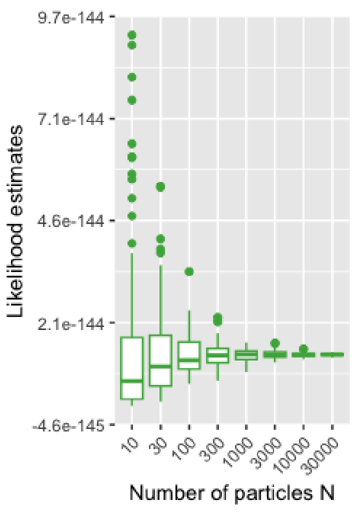
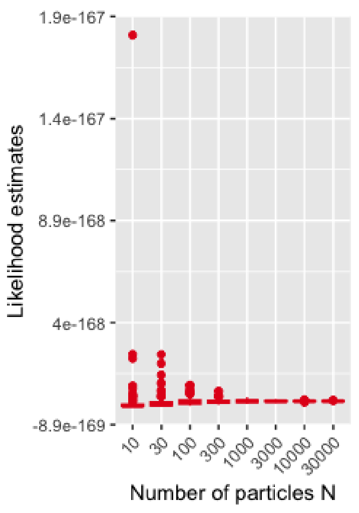


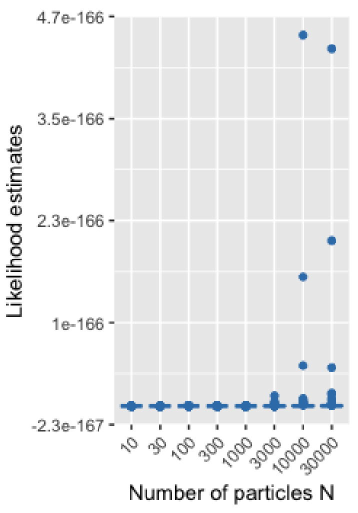
**Figure B.1:** Measures of variation of 100 likelihood estimates computed by a standard bootstrap filter with varying numbers of particles $N$ at three different parameter values. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates



**(a)** True parameter value



**(b)** False parameter value 1



**(c)** False parameter value 2

**Figure B.2:** Boxplots of 100 likelihood estimates for varying numbers of particles $N$ at three different parameter values.

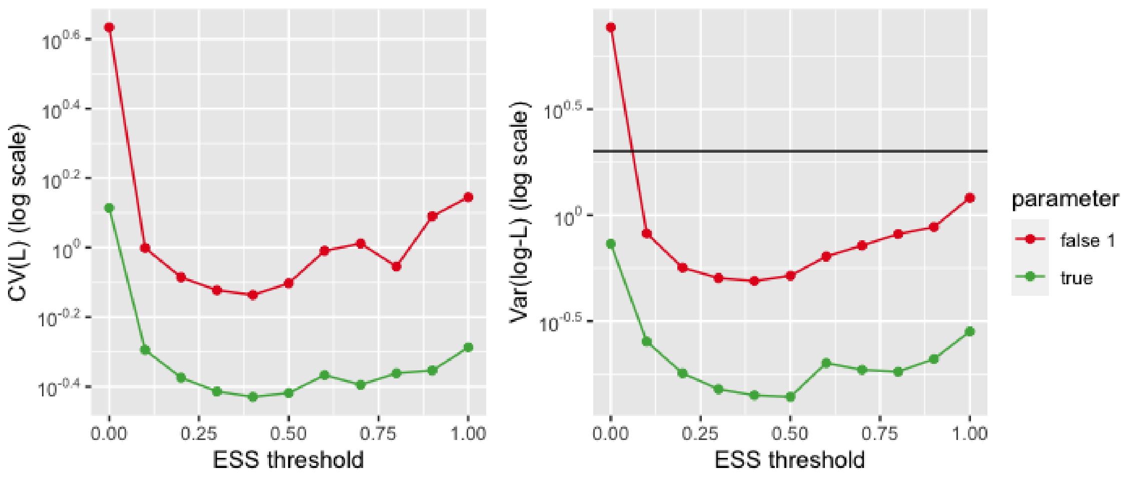
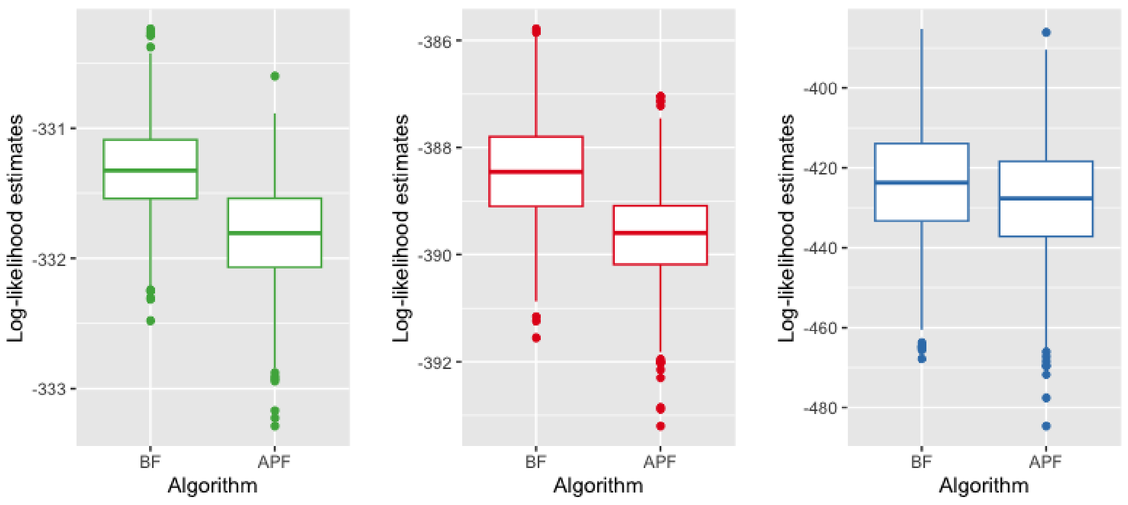


**Figure B.3:** Measures of variation of 1000 likelihood estimates computed by a bootstrap filter with varying ESS threshold for resampling. False parameter 2 is not included in the plot because the values are too large and show no discernible trends. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates.



**(a)** True parameter value **(b)** False parameter value 1 **(c)** False parameter value 2

**Figure B.4:** Boxplots of 1000 log-likelihood estimates comparing a bootstrap filter (BF) and an auxiliary particle filter (APF) using the expected value in the adjustment multiplier, for three different parameter values.

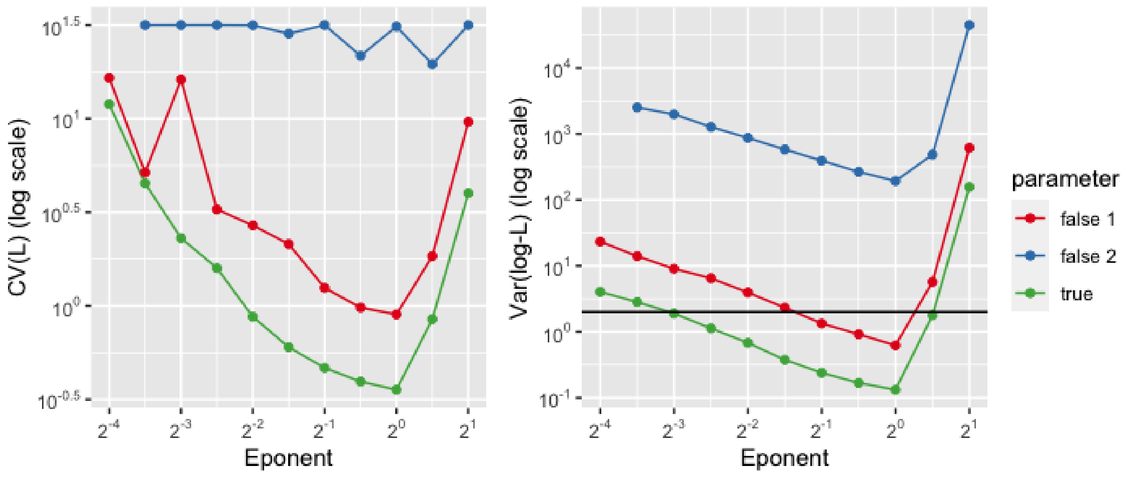
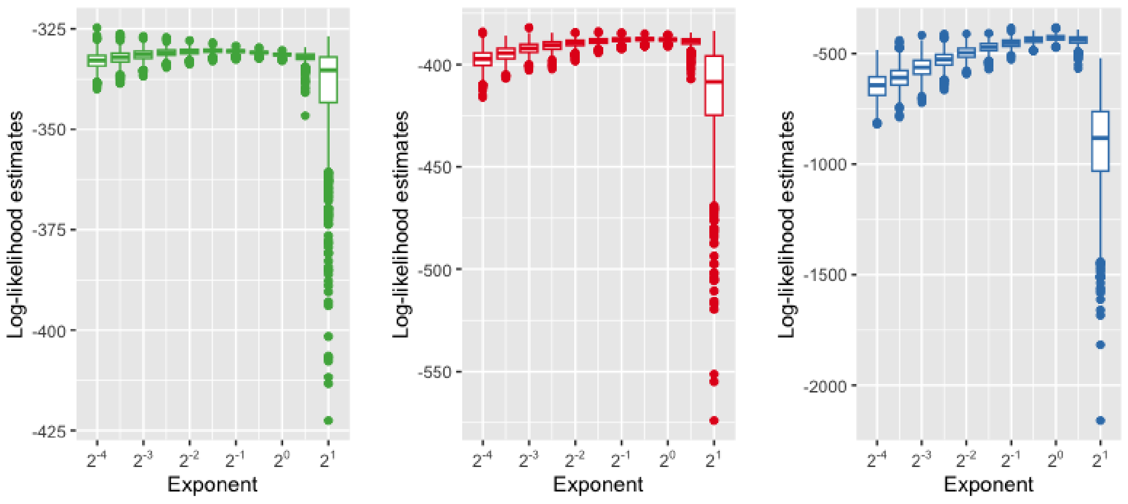


**Figure B.5:** Measures of variation of 1000 likelihood estimates computed by an auxiliary particle filter using tempered weights with varying exponents in the resampling step. The first figure shows the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates.



**(a)** True parameter value **(b)** False parameter value 1 **(c)** False parameter value 2

**Figure B.6:** Boxplots of 1000 log-likelihood estimates computed by an auxiliary particle filter using tempered weights with varying exponents in the resampling step, for three different parameter values.

| | True parameter | | False parameter 1 | | False parameter 2 | |
|---|---|---|---|---|---|---|
| ESS | CV(L) | Var(log-L) | CV(L) | Var(log-L) | CV(L) | Var(log-L) |
| 0 | 1.30E+00 | 7.32E-01 | 4.30E+00 | 7.69E+00 | 3.16E+01 | 1.31E+04 |
| 0.1 | 5.08E-01 | 2.53E-01 | 9.97E-01 | 8.20E-01 | 3.15E+01 | 4.90E+02 |
| 0.2 | 4.22E-01 | 1.79E-01 | 8.21E-01 | 5.65E-01 | 3.16E+01 | 3.35E+02 |
| 0.3 | 3.86E-01 | 1.51E-01 | 7.54E-01 | 5.05E-01 | 3.04E+01 | 2.56E+02 |
| 0.4 | 3.72E-01 | 1.41E-01 | 7.31E-01 | 4.89E-01 | 3.09E+01 | 2.64E+02 |
| 0.5 | 3.82E-01 | 1.39E-01 | 7.90E-01 | 5.18E-01 | 1.69E+01 | 2.49E+02 |
| 0.6 | 4.30E-01 | 2.00E-01 | 9.79E-01 | 6.38E-01 | 3.16E+01 | 2.05E+02 |
| 0.7 | 4.03E-01 | 1.86E-01 | 1.03E+00 | 7.19E-01 | 2.47E+01 | 1.89E+02 |
| 0.8 | 4.35E-01 | 1.82E-01 | 8.82E-01 | 8.14E-01 | 2.40E+01 | 1.92E+02 |
| 0.9 | 4.43E-01 | 2.10E-01 | 1.23E+00 | 8.79E-01 | 2.03E+01 | 1.89E+02 |
| 1 | 5.17E-01 | 2.83E-01 | 1.40E+00 | 1.21E+00 | 2.05E+01 | 2.16E+02 |

**Table B.2:** Measures of variation of 1000 likelihood estimates computed by a bootstrap filter with varying ESS threshold for resampling. The likelihood was estimated at the true parameter value and at two false parameter values for the 2-state model. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates.

| | True parameter | | False parameter 1 | | False parameter 2 | |
|---|---|---|---|---|---|---|
| Alg | CV(L) | Var(logL) | CV(L) | Var(log-L) | CV(L) | Var(log-L) |
| BF | 3.47E-01 | 1.14E-01 | 1.09E+00 | 8.26E-01 | 2.49E+01 | 1.91E+02 |
| APF | 3.89E-01 | 1.56E-01 | 1.06E+00 | 7.93E-01 | 3.09E+01 | 1.98E+02 |

**Table B.3:** Measures of variation of 1000 likelihood estimates of three different parameter values in the 2-state model computed by a bootstrap filter (BF) and by an auxiliary particle filter (APF) using the expected value in the adjustment multiplier. The likelihood was estimated at the true parameter value and at two false parameter values. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates.

| | True parameter | | False parameter 1 | | False parameter 2 | |
|---|---|---|---|---|---|---|
| Alg | CV(L) | Var(logL) | CV(L) | Var(log-L) | CV(L) | Var(log-L) |
| 2 | 3.99E+00 | 1.57E+02 | 9.61E+00 | 6.16E+02 | 3.16E+01 | 4.48E+04 |
| $\sqrt{2}$ | 8.49E-01 | 1.78E+00 | 1.84E+00 | 5.72E+00 | 1.96E+01 | 4.86E+02 |
| 1 | 3.56E-01 | 1.32E-01 | 9.00E-01 | 6.23E-01 | 3.11E+01 | 1.93E+02 |
| $\sqrt{2}/2$ | 3.94E-01 | 1.68E-01 | 9.79E-01 | 9.19E-01 | 2.17E+01 | 2.66E+02 |
| 0.5 | 4.66E-01 | 2.38E-01 | 1.25E+00 | 1.34E+00 | 3.16E+01 | 3.94E+02 |
| $\sqrt{2}/4$ | 6.02E-01 | 3.76E-01 | 2.14E+00 | 2.32E+00 | 2.85E+01 | 5.82E+02 |
| 0.25 | 8.74E-01 | 6.81E-01 | 2.69E+00 | 3.95E+00 | 3.15E+01 | 8.72E+02 |
| $\sqrt{2}/8$ | 1.59E+00 | 1.13E+00 | 3.27E+00 | 6.49E+00 | 3.16E+01 | 1.27E+03 |
| 0.125 | 2.30E+00 | 1.89E+00 | 1.62E+01 | 9.02E+00 | 3.16E+01 | 1.99E+03 |
| $\sqrt{2}/16$ | 4.51E+00 | 2.83E+00 | 5.15E+00 | 1.40E+01 | 3.16E+01 | 2.53E+03 |
| 0.0625 | 1.20E+01 | 4.04E+00 | 1.65E+01 | 2.33E+01 | | |

**Table B.4:** Measures of variation of 1000 likelihood estimates computed by an auxiliary particle filter using tempered weights with varying exponents in the resampling step. The likelihood was estimated for the 2-state model at the true parameter value and at two false parameter values. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates.

## B.2 Results for 7-State Model—Second Dataset

| | True parameter | | False parameter 1 | | False parameter 2 | |
|---|---|---|---|---|---|---|
| $N$ | CV(L) | Var(logL) | CV(L) | Var(log-L) | CV(L) | Var(log-L) |
| 3 | 1.41E+00 | 3.45E+01 | 8.42E+00 | 5.19E+03 | 9.10E+00 | 8.81E+02 |
| 10 | 7.43E-01 | 2.06E+00 | 9.99E+00 | 1.34E+03 | 6.02E+00 | 1.63E+02 |
| 30 | 4.78E-01 | 3.37E-01 | 9.99E+00 | 5.74E+02 | 3.29E+00 | 3.68E+01 |
| 100 | 2.79E-01 | 9.11E-02 | 9.66E+00 | 1.74E+02 | 4.50E+00 | 1.06E+01 |
| 300 | 1.55E-01 | 2.51E-02 | 5.83E+00 | 8.40E+01 | 2.26E+00 | 4.85E+00 |
| 1000 | 8.39E-02 | 7.11E-03 | 5.20E+00 | 3.47E+01 | 9.81E-01 | 1.19E+00 |
| 3000 | 4.75E-02 | 2.28E-03 | 3.71E+00 | 1.67E+01 | 6.83E-01 | 4.39E-01 |
| 10000 | 2.73E-02 | 7.42E-04 | 3.65E+00 | 6.66E+00 | 3.95E-01 | 1.36E-01 |
| 30000 | 1.54E-02 | 2.36E-04 | 1.89E+00 | 3.60E+00 | 2.11E-01 | 4.90E-02 |
| log-L | -184.14 | | -224.23 | | -201.62 | |

**Table B.5:** Measures of variation of 100 likelihood estimates computed by a standard bootstrap filter with varying numbers of particles $N$for the 7-state model. The likelihood was estimated at the true parameter value and at two false parameter values. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 30,000$ particles.

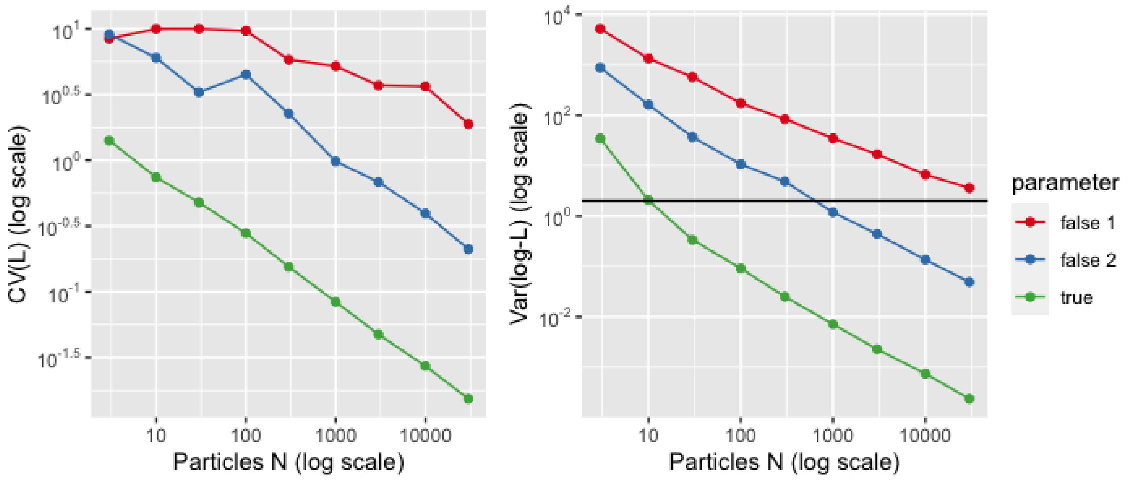
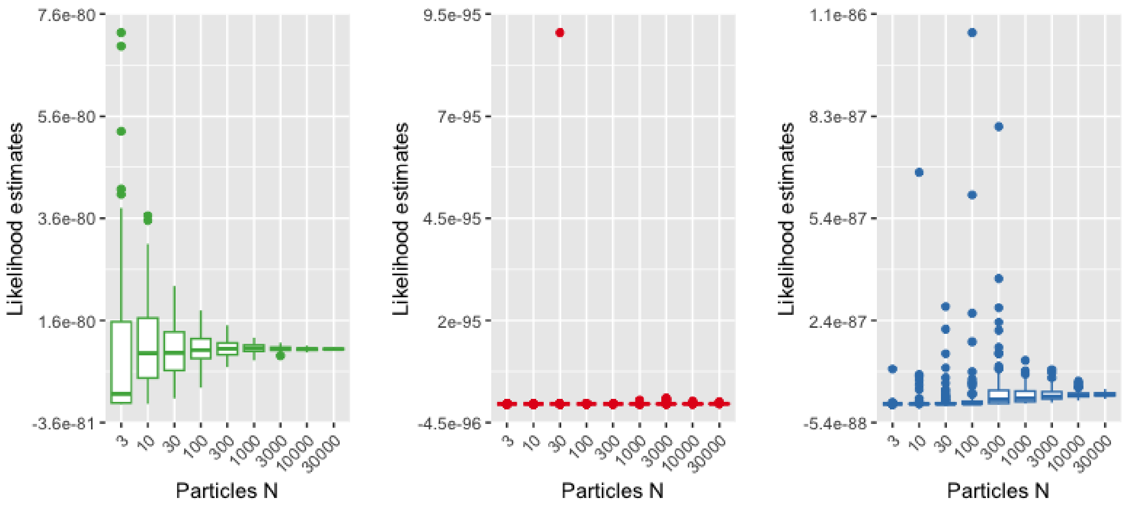


**Figure B.7:** Measures of variation of 100 likelihood estimates of the 7-state model computed by a standard bootstrap filter with varying numbers of particles $N$. The first figure gives the CV of the likelihood estimates, the second figure the variance of the log-likelihood estimates. The black horizontal line shows the limit of 2 for the variance of the log-likelihood estimates.



**(a)** True parameter value **(b)** False parameter value 1 **(c)** False parameter value 2

**Figure B.8:** Boxplots of 100 likelihood estimates of the 7-state model for varying numbers of particles $N$ at three different parameter values.

# Appendix C

# Chapter 3: Additional Results

| $N$ | $M$ | $h$ | Runtime (sec) | $\hat{R}$ | Acceptance Rate | ESS | ESS/sec |
|---:|---:|---:|---:|---:|---:|---:|---:|
| 3 | 1000000 | 0.01 | 9425.93 | 6.27 | 0.11 | | |
| 10 | 300000 | 0.01 | 2901.30 | 1.02 | 0.44 | 479.41 | 0.17 |
| 30 | 100000 | 0.01 | 1212.29 | 1.09 | 0.62 | 276.25 | 0.23 |
| 100 | 30000 | 0.01 | 580.72 | 1.03 | 0.71 | 85.87 | 0.15 |
| 300 | 10000 | 0.01 | 379.82 | 1.48 | 0.69 | | |
| 1000 | 3000 | 0.01 | 320.86 | 5.89 | 0.60 | | |
| 3 | 1000000 | 0.03 | 8456.20 | 1.01 | 0.20 | 2057.37 | 0.24 |
| 10 | 300000 | 0.03 | 2874.81 | 1.00 | 0.44 | 1426.07 | 0.50 |
| 30 | 100000 | 0.03 | 1217.68 | 1.02 | 0.57 | 618.41 | 0.51 |
| 100 | 30000 | 0.03 | 581.85 | 1.06 | 0.66 | 232.97 | 0.40 |
| 300 | 10000 | 0.03 | 380.68 | 1.18 | 0.67 | 69.58 | 0.18 |
| 1000 | 3000 | 0.03 | 323.10 | 1.34 | 0.68 | 38.06 | 0.12 |
| 3 | 1000000 | 0.10 | 8244.27 | 1.00 | 0.17 | 5741.19 | 0.70 |
| 10 | 300000 | 0.10 | 2769.90 | 1.00 | 0.36 | 4004.37 | 1.45 |
| 30 | 100000 | 0.10 | 1162.71 | 1.00 | 0.47 | 1584.92 | 1.36 |
| 100 | 30000 | 0.10 | 556.82 | 1.05 | 0.53 | 577.67 | 1.04 |
| 300 | 10000 | 0.10 | 365.33 | 1.07 | 0.53 | 176.00 | 0.48 |
| 1000 | 3000 | 0.10 | 294.24 | 1.14 | 0.43 | 31.94 | 0.11 |
| 3 | 1000000 | 0.20 | 8286.23 | 1.00 | 0.14 | 9339.82 | 1.13 |
| 10 | 300000 | 0.20 | 2713.19 | 1.01 | 0.29 | 5802.01 | 2.14 |
| 30 | 100000 | 0.20 | 1137.09 | 1.00 | 0.38 | 2507.89 | 2.21 |
| 100 | 30000 | 0.20 | 532.15 | 1.01 | 0.41 | 870.25 | 1.64 |
| 300 | 10000 | 0.20 | 347.34 | 1.02 | 0.40 | 257.93 | 0.74 |
| 1000 | 3000 | 0.20 | 252.36 | 15.33 | 0.22 | | |
| 3 | 1000000 | 0.30 | 7746.05 | 1.00 | 0.12 | 10466.38 | 1.35 |
| 10 | 300000 | 0.30 | 2589.46 | 1.00 | 0.24 | 7235.05 | 2.79 |
| 30 | 100000 | 0.30 | 1092.89 | 1.00 | 0.31 | 3304.82 | 3.02 |
| 100 | 30000 | 0.30 | 522.67 | 1.01 | 0.34 | 1105.74 | 2.12 |
| 300 | 10000 | 0.30 | 335.23 | 1.06 | 0.33 | 323.21 | 0.96 |
| 1000 | 3000 | 0.30 | 282.34 | 1.14 | 0.33 | 86.34 | 0.31 |
| 3 | 1000000 | 0.40 | 7563.62 | 1.00 | 0.10 | 10994.95 | 1.45 |
| 10 | 300000 | 0.40 | 2564.40 | 1.00 | 0.21 | 8185.71 | 3.19 |
| 30 | 100000 | 0.40 | 1068.12 | 1.00 | 0.26 | 3198.63 | 2.99 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 100 | 30000 | 0.40 | 512.12 | 1.03 | 0.29 | 1039.75 | 2.03 |
| 300 | 10000 | 0.40 | 341.18 | 1.04 | 0.30 | 388.99 | 1.14 |
| 1000 | 3000 | 0.40 | 287.42 | 1.12 | 0.31 | 182.74 | 0.64 |
| 3 | 1000000 | 0.50 | 7437.15 | 1.00 | 0.09 | 12816.09 | 1.72 |
| 10 | 300000 | 0.50 | 2499.31 | 1.00 | 0.18 | 8905.01 | 3.56 |
| 30 | 100000 | 0.50 | 1036.60 | 1.01 | 0.23 | 3429.98 | 3.31 |
| 100 | 30000 | 0.50 | 488.31 | 1.01 | 0.25 | 1161.37 | 2.38 |
| 300 | 10000 | 0.50 | 320.62 | 1.05 | 0.26 | 360.88 | 1.13 |
| 1000 | 3000 | 0.50 | 229.33 | 2.32 | 0.15 | | |
| 3 | 1000000 | 0.60 | 7252.47 | 1.00 | 0.08 | 12321.35 | 1.70 |
| 10 | 300000 | 0.60 | 2433.61 | 1.00 | 0.16 | 8330.98 | 3.42 |
| 30 | 100000 | 0.60 | 1015.42 | 1.00 | 0.20 | 3582.34 | 3.53 |
| 100 | 30000 | 0.60 | 474.52 | 1.01 | 0.22 | 1164.87 | 2.45 |
| 300 | 10000 | 0.60 | 305.66 | 1.02 | 0.21 | 364.50 | 1.19 |
| 1000 | 3000 | 0.60 | 262.11 | 1.07 | 0.22 | 96.00 | 0.37 |
| 3 | 1000000 | 0.70 | 7077.63 | 1.00 | 0.07 | 11642.44 | 1.64 |
| 10 | 300000 | 0.70 | 2344.39 | 1.00 | 0.14 | 8253.93 | 3.52 |
| 30 | 100000 | 0.70 | 976.85 | 1.01 | 0.18 | 3615.28 | 3.70 |
| 100 | 30000 | 0.70 | 462.27 | 1.00 | 0.19 | 1177.31 | 2.55 |
| 300 | 10000 | 0.70 | 302.99 | 1.02 | 0.20 | 382.76 | 1.26 |
| 1000 | 3000 | 0.70 | 211.17 | 1.60 | 0.13 | | |
| 3 | 1000000 | 0.80 | 6930.96 | 1.00 | 0.06 | 12390.02 | 1.79 |
| 10 | 300000 | 0.80 | 2293.46 | 1.00 | 0.12 | 8891.20 | 3.88 |
| 30 | 100000 | 0.80 | 945.56 | 1.00 | 0.16 | 3931.97 | 4.16 |
| 100 | 30000 | 0.80 | 446.26 | 1.03 | 0.18 | 1245.63 | 2.79 |
| 300 | 10000 | 0.80 | 292.95 | 1.02 | 0.17 | 497.39 | 1.70 |
| 1000 | 3000 | 0.80 | 224.36 | 1.04 | 0.16 | 98.60 | 0.44 |
| 3 | 1000000 | 0.90 | 6761.22 | 1.00 | 0.05 | 11320.21 | 1.67 |
| 10 | 300000 | 0.90 | 2261.19 | 1.00 | 0.11 | 8563.88 | 3.79 |
| 30 | 100000 | 0.90 | 934.92 | 1.01 | 0.14 | 3581.33 | 3.83 |
| 100 | 30000 | 0.90 | 438.06 | 1.01 | 0.16 | 1145.60 | 2.62 |
| 300 | 10000 | 0.90 | 280.56 | 1.03 | 0.15 | 395.30 | 1.41 |
| 1000 | 3000 | 0.90 | 224.43 | 1.08 | 0.16 | 125.23 | 0.56 |
| 3 | 1000000 | 1.00 | 6434.97 | 1.00 | 0.05 | 11040.03 | 1.72 |
| 10 | 300000 | 1.00 | 2132.95 | 1.00 | 0.10 | 8641.48 | 4.05 |
| 30 | 100000 | 1.00 | 881.54 | 1.00 | 0.13 | 3352.41 | 3.80 |
| 100 | 30000 | 1.00 | 417.99 | 1.01 | 0.14 | 1133.77 | 2.71 |
| 300 | 10000 | 1.00 | 271.34 | 1.04 | 0.14 | 381.40 | 1.41 |
| 1000 | 3000 | 1.00 | 223.78 | 1.07 | 0.15 | 113.72 | 0.51 |
| 3 | 1000000 | 1.10 | 6454.60 | 1.00 | 0.04 | 10560.55 | 1.64 |
| 10 | 300000 | 1.10 | 2159.33 | 1.00 | 0.09 | 8323.76 | 3.85 |
| 30 | 100000 | 1.10 | 885.84 | 1.00 | 0.12 | 3718.65 | 4.20 |
| 100 | 30000 | 1.10 | 409.13 | 1.01 | 0.13 | 1108.84 | 2.71 |
| 300 | 10000 | 1.10 | 258.26 | 1.01 | 0.12 | 312.04 | 1.21 |
| 1000 | 3000 | 1.10 | 214.64 | 1.06 | 0.11 | 100.23 | 0.47 |

**Table C.1:** Results from running the PMMH for the 2-state model with constant $NM$, while varying the number of particles $N$ and the scaling factor $h$. The proposal distribution is a multivariate normal distribution with a covariance proportional to the posterior covariance.

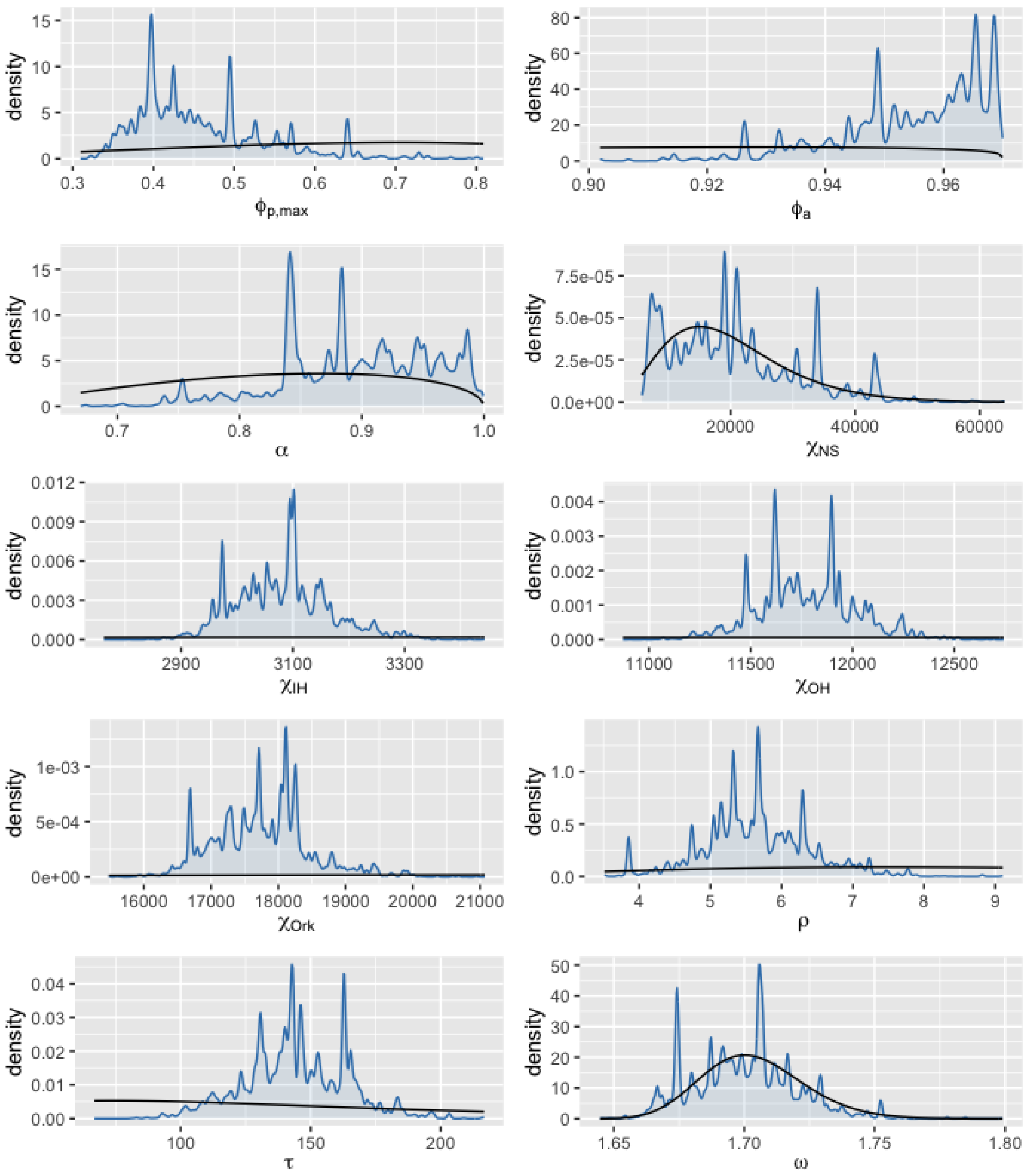


**Figure C.1:** Density plots of the marginal posterior densities for all 10 parameters, generated by running the PMMH for the complete seal model with independent estimate with $N = 3$ particles and 2 chains with $M = 10,000,000$ iterations each. The scaling factor for the covariance of the proposal distribution swas set to $h = 0.8$. The black density line indicates the prior distribution.

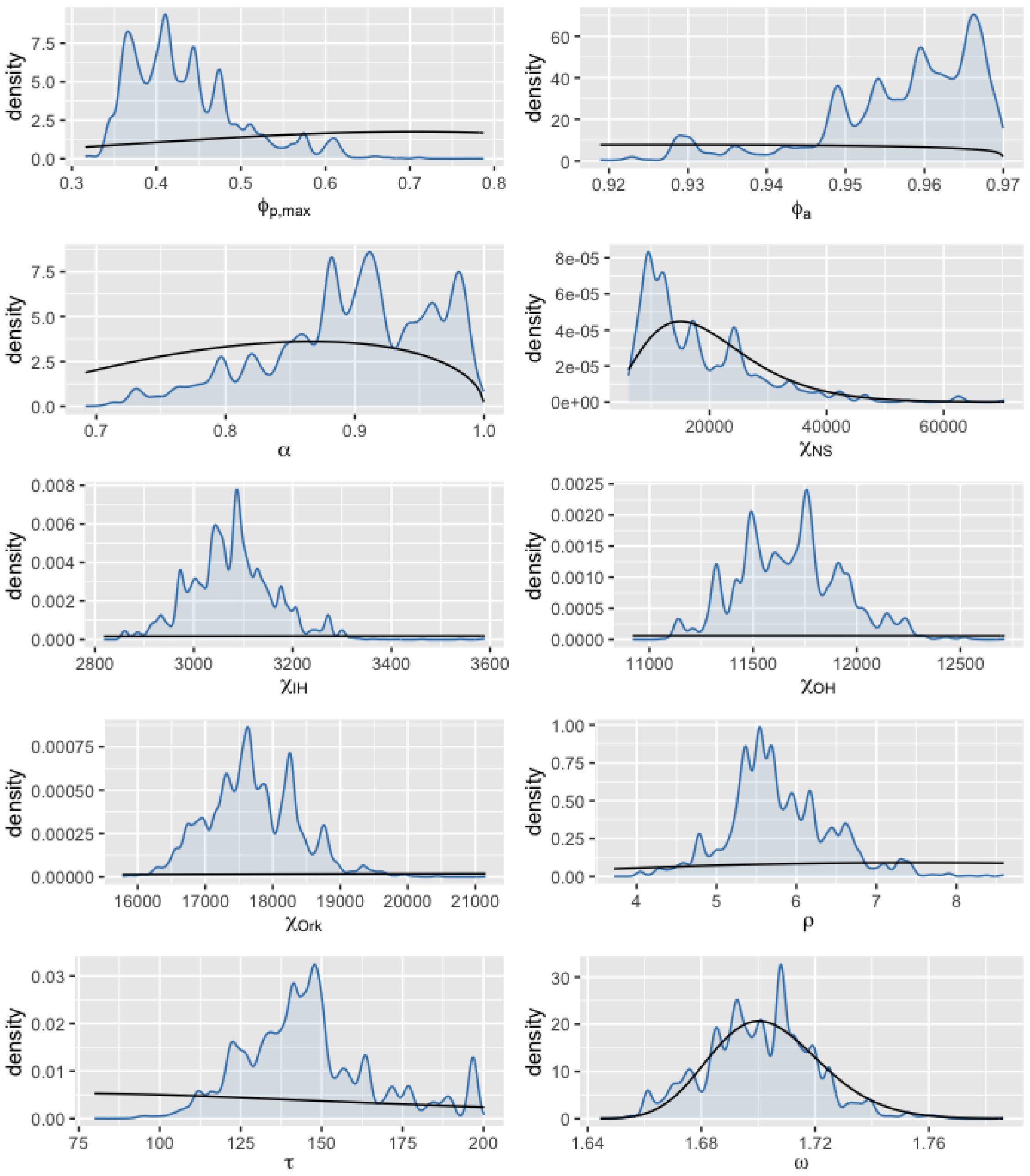
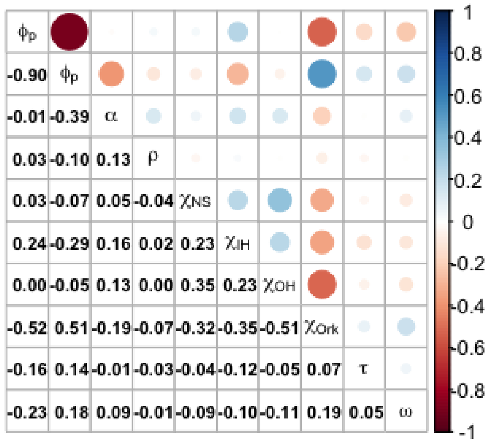
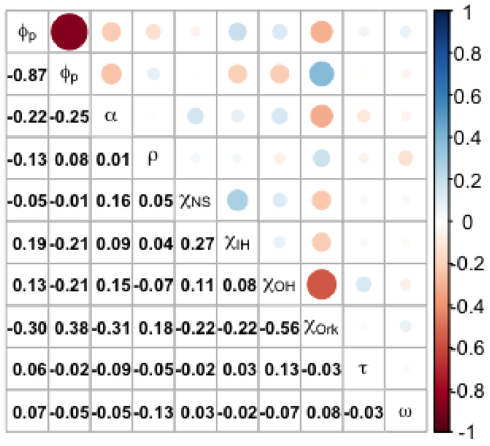


**Figure C.2:** Density plots of the marginal posterior densities for all 10 parameters, generated by running the PMMH for the complete seal model with independent estimate with $N = 3,000$ particles and 2 chains with $M = 100,000$ iterations each. The scaling factor for the covariance of the proposal distribution was set to $h = 0.8$. The black density line indicates the prior distribution.



**(a)** $N = 3$



**(b)** $N = 3,000$

**Figure C.3:** Posterior correlation coefficients of the 10 parameters of the complete seal model, as generated by running the PMMH with two different number of particles $N$.

| $N$ | $M$ | $h$ | Runtime (sec) | $\hat{R}$ | Acceptance Rate | ESS | ESS/sec |
|---:|---:|---:|---:|---:|---:|---:|---:|
| 3 | 1000000 | 0.01 | 8928.07 | 1.24 | 0.23 | | |
| 10 | 300000 | 0.01 | 3037.19 | 1.14 | 0.46 | | |
| 30 | 100000 | 0.01 | 1282.26 | 1.22 | 0.61 | | |
| 100 | 30000 | 0.01 | 601.47 | 2.19 | 0.71 | | |
| 300 | 10000 | 0.01 | 396.95 | 1.62 | 0.73 | | |
| 1000 | 3000 | 0.01 | 339.13 | 8.09 | 0.70 | | |
| 3 | 1000000 | 0.03 | 8850.93 | 1.13 | 0.20 | 1080.34 | 0.12 |
| 10 | 300000 | 0.03 | 2961.13 | 1.07 | 0.41 | 694.38 | 0.23 |
| 30 | 100000 | 0.03 | 1271.99 | 1.24 | 0.52 | | |
| 100 | 30000 | 0.03 | 595.52 | 1.40 | 0.60 | | |
| 300 | 10000 | 0.03 | 391.39 | 1.59 | 0.59 | | |
| 1000 | 3000 | 0.03 | 324.12 | 1.74 | 0.69 | | |
| 3 | 1000000 | 0.10 | 8678.99 | 1.03 | 0.15 | 2472.80 | 0.28 |
| 10 | 300000 | 0.10 | 2931.93 | 1.01 | 0.28 | | |
| 30 | 100000 | 0.10 | 1242.64 | 1.06 | 0.39 | 762.80 | 0.61 |
| 100 | 30000 | 0.10 | 592.67 | 1.11 | 0.42 | | |
| 300 | 10000 | 0.10 | 385.01 | 1.16 | 0.47 | | |
| 1000 | 3000 | 0.10 | 328.85 | 2.23 | 0.45 | | |
| 3 | 1000000 | 0.30 | 8805.31 | 1.01 | 0.09 | 3831.56 | 0.44 |
| 10 | 300000 | 0.30 | 2924.14 | 1.02 | 0.19 | 2797.38 | 0.96 |
| 30 | 100000 | 0.30 | 1230.70 | 1.01 | 0.25 | 1131.36 | 0.92 |
| 100 | 30000 | 0.30 | 591.68 | 1.22 | 0.26 | | |
| 300 | 10000 | 0.30 | 385.89 | 1.04 | 0.27 | 146.52 | 0.38 |
| 1000 | 3000 | 0.30 | 325.32 | 1.48 | 0.30 | | |
| 3 | 1000000 | 1.00 | 9137.21 | 1.02 | 0.04 | 4701.82 | 0.51 |
| 10 | 300000 | 1.00 | 3006.60 | 1.02 | 0.08 | 3559.77 | 1.18 |
| 30 | 100000 | 1.00 | 1260.39 | 1.01 | 0.10 | 1446.06 | 1.15 |
| 100 | 30000 | 1.00 | 607.10 | 1.07 | 0.11 | 463.60 | 0.76 |
| 300 | 10000 | 1.00 | 392.78 | 1.03 | 0.12 | | |
| 1000 | 3000 | 1.00 | 331.78 | 1.24 | 0.12 | | |

**Table C.2:** Results from running the PMMH for the 2-state model with constant $NM$, while varying the number of particles $N$ and the scaling factor $h$. The proposal distribution is a multivariate normal distribution on the transformed parameter space with a covariance proportional to the (transformed) posterior covariance.

| $N$ | $M$ | $h$ | Runtime (sec) | $\hat{R}$ | Acceptance Rate | ESS | ESS/sec |
|---:|---:|---:|---:|---:|---:|---:|---:|
| 3 | 1000000 | 0.20 | 7064.55 | 1.00 | 0.29 | 19087.30 | 2.70 |
| 10 | 300000 | 0.20 | 2380.41 | 1.00 | 0.41 | 8689.39 | 3.65 |
| 30 | 100000 | 0.20 | 1034.95 | 1.00 | 0.46 | 3398.81 | 3.28 |
| 100 | 30000 | 0.20 | 536.82 | 1.01 | 0.49 | 943.83 | 1.76 |
| 300 | 10000 | 0.20 | 381.17 | 1.03 | 0.47 | 198.85 | 0.52 |
| 1000 | 3000 | 0.20 | 345.41 | 1.09 | 0.47 | 81.03 | 0.23 |
| 3 | 1000000 | 0.40 | 6253.14 | 1.00 | 0.22 | 27623.02 | 4.42 |
| 10 | 300000 | 0.40 | 2217.55 | 1.00 | 0.30 | 12175.10 | 5.49 |
| 30 | 100000 | 0.40 | 961.31 | 1.01 | 0.34 | | |
| 100 | 30000 | 0.40 | 506.42 | 1.01 | 0.35 | 1171.81 | 2.31 |
| 300 | 10000 | 0.40 | 366.93 | 1.01 | 0.36 | 410.58 | 1.12 |
| 1000 | 3000 | 0.40 | 319.56 | 1.11 | 0.35 | 126.18 | 0.39 |
| 3 | 1000000 | 0.60 | 6023.85 | 1.00 | 0.17 | 28327.91 | 4.70 |
| 10 | 300000 | 0.60 | 2069.02 | 1.00 | 0.24 | 12998.41 | 6.28 |
| 30 | 100000 | 0.60 | 904.20 | 1.01 | 0.26 | 4427.07 | 4.90 |
| 100 | 30000 | 0.60 | 467.45 | 1.01 | 0.27 | 1373.90 | 2.94 |
| 300 | 10000 | 0.60 | 332.71 | 1.00 | 0.25 | 275.91 | 0.83 |
| 1000 | 3000 | 0.60 | 284.53 | 5.04 | 0.21 | | |
| 3 | 1000000 | 0.80 | 5738.07 | 1.00 | 0.14 | 30304.94 | 5.28 |
| 10 | 300000 | 0.80 | 1982.53 | 1.00 | 0.19 | 13073.27 | 6.59 |
| 30 | 100000 | 0.80 | 859.98 | 1.00 | 0.21 | 5017.22 | 5.83 |
| 100 | 30000 | 0.80 | 448.45 | 1.01 | 0.22 | 1479.16 | 3.30 |
| 300 | 10000 | 0.80 | 311.54 | 1.01 | 0.21 | 373.91 | 1.20 |
| 1000 | 3000 | 0.80 | 274.99 | 1.06 | 0.21 | 138.27 | 0.50 |
| 3 | 1000000 | 1.00 | 5433.13 | 1.00 | 0.11 | 25475.19 | 4.69 |
| 10 | 300000 | 1.00 | 1884.84 | 1.00 | 0.15 | 12087.62 | 6.41 |
| 30 | 100000 | 1.00 | 815.88 | 1.00 | 0.17 | 5104.45 | 6.26 |
| 100 | 30000 | 1.00 | 409.90 | 1.01 | 0.16 | 916.16 | 2.24 |
| 300 | 10000 | 1.00 | 274.96 | 1.03 | 0.15 | 401.44 | 1.46 |
| 1000 | 3000 | 1.00 | 225.92 | 16.51 | 0.11 | | |

**Table C.3:** Results from running the PMMH for the 7-state model with constant $NM$, while varying the number of particles $N$ and the scaling factor $h$. The proposal distribution is a multivariate normal distribution with a covariance proportional to the posterior covariance.

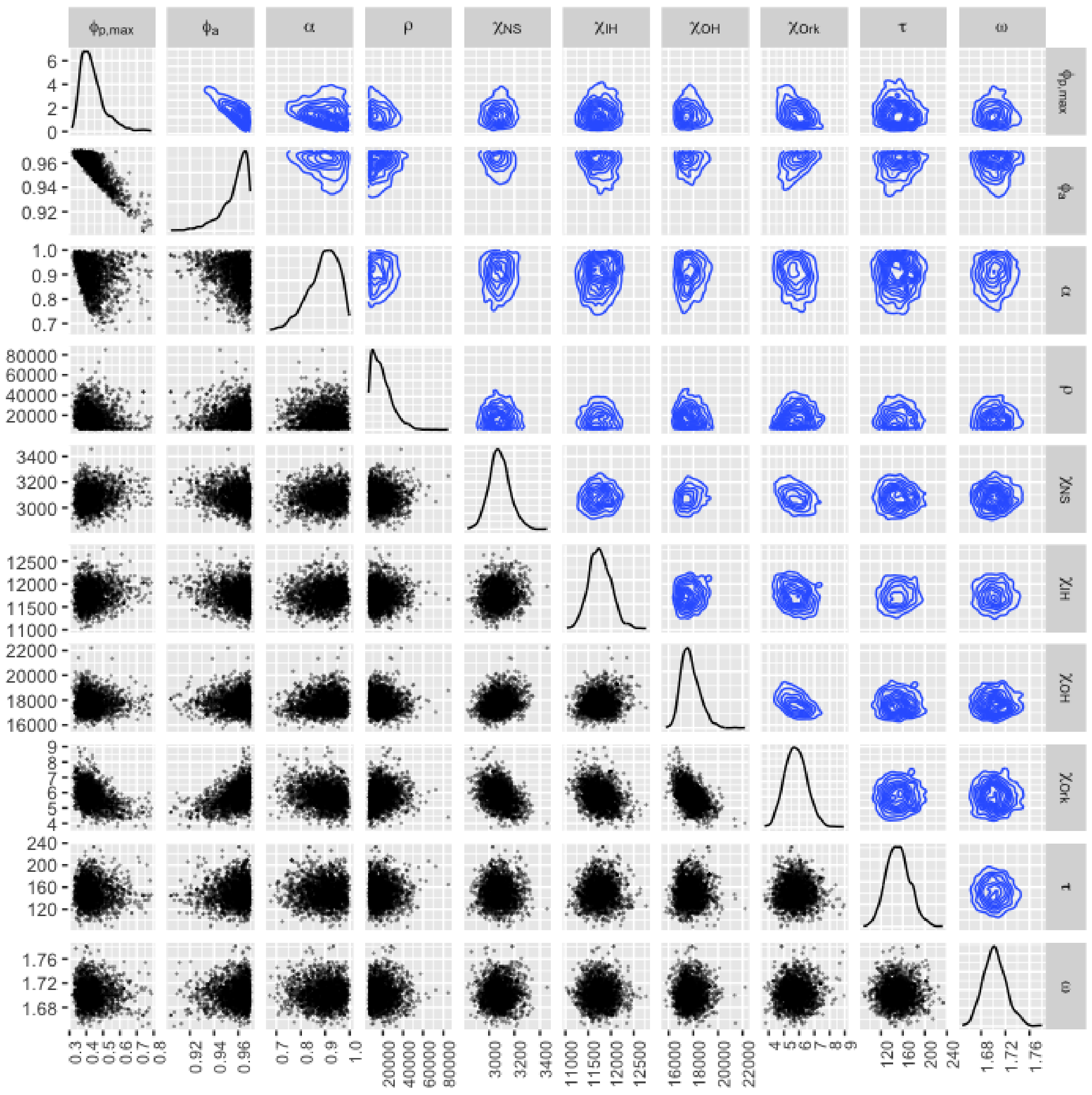


**Figure C.4:** Scatter plots and joint densities of the posterior distribution for the complete seal model using the real data, generated by running the PMMH for the complete seal model with independent estimate with $N = 3,000$ particles and 20 chains with $M = 1,000,000$ iterations each.

# Appendix D

# Chapter 4

## D.1 First and second moments of $X_0$ in the seal model

To calculate the NDLM approximation of the seal model as described in Section 1.2, the first two moments of the distributions are required. The initial distribution of $x_0$ for only one region (and without the dispersion factor $a$) is $x_{0,0} \sim \mathcal{N}(y_0, y_0^2/\tau)$, $x_{0,1} \sim \text{Bin}(x_{0,0}, 0.5\phi_{p,0})$, $x_{0,a=2,...,5} \sim \text{Bin}(x_{0,a-1}, \phi_a)$, $x_{0,6} \sim \text{Negbin}(x_{0,0}, \alpha) + x_{0,0}$. The initial pup survival probability $\phi_{p,0}$ is calculated using the observed number of pups $y_0$ and can therefore be treated as a constant. For these calculations, we omit the time index, since all states here are at time $t = 0$. All calculations of expected values, variances, and covariances are conditional on $y_0$.

$$\begin{aligned}
\mathrm{E}(x_0) &= y \\
\mathrm{E}(x_1) &= 0.5\phi_p \,\mathrm{E}(x_{0,0}) = 0.5\phi_p y \\
\mathrm{E}(x_2) &= \phi_a \,\mathrm{E}(x_1) = 0.5\phi_p\phi_a y \\
\mathrm{E}(x_3) &= \phi_a \,\mathrm{E}(x_2) = 0.5\phi_p\phi_a^2 y \\
\mathrm{E}(x_4) &= \phi_a \,\mathrm{E}(x_3) = 0.5\phi_p\phi_a^3 y \\
\mathrm{E}(x_5) &= \phi_a \,\mathrm{E}(x_4) = 0.5\phi_p\phi_a^4 y \\
\mathrm{E}(x_6) &= \frac{1-\alpha}{\alpha} y + \mathrm{E}(x_0) = \frac{1-\alpha}{\alpha} y + y = \frac{y}{\alpha}
\end{aligned}$$

To calculate the covariance matrix, we compute every individual covariance $\text{Cov}(x_{i,0}, x_{j,0})$ with $i, j = 0, ..., 6$. As an aid variable, we use $\sigma^2 = y_0^2/\tau$. Using the law of total variance (e.g., Proposition 5.2 in Ross, 2011) and $\text{Var}(x_0) = \text{Var}(y) = \sigma^2$, we obtain

$$\begin{aligned}
\text{Var}(x_1) &= \mathrm{E}\left(\text{Var}(x_1|x_0)\right) + \text{Var}\left(\mathrm{E}(x_1|x_0)\right) \\
&= \mathrm{E}\left(0.5\phi_p(1-0.5\phi_p)x_0\right) + \text{Var}\left(0.5\phi_p x_0\right) \\
&= 0.5\phi_p(1-0.5\phi_p)y + (0.5\phi_p)^2\sigma^2 \\
&= 0.5\phi_p y + (0.5\phi_p)^2(\sigma^2 - y).
\end{aligned}$$

Similar calculations give the variances of $x_2$ to $x_5$. For $x_6$, it is

$$\begin{aligned}\operatorname{Var}(x_6) &= \operatorname{E}\left(\operatorname{Var}(x_6|x_0)\right) + \operatorname{Var}\left(\operatorname{E}(x_6|x_0)\right)\\ &= \operatorname{E}\left(\frac{1-\alpha}{\alpha^2}x_0\right) + \operatorname{Var}\left(x_0 + \frac{1-\alpha}{\alpha}x_0\right)\\ &= \frac{1-\alpha}{\alpha^2}y + \sigma^2/\alpha^2.\end{aligned}$$

To calculate the off-diagonal entries of the covariance matrix, we use $\operatorname{Cov}(x_i, x_j) = \operatorname{E}(x_i x_j) - \operatorname{E}(x_i)\operatorname{E}(x_j)$. With

$$\begin{aligned}\operatorname{E}(x_0x_1) &= \iint p(x_1|x_0)p(x_0)x_0x_1dx_0dx_1\\ &= \int x_0p(x_0)\left(\int x_1p(x_1|x_0)dx_1\right)dx_0.\\ &= \int p(x_0)0.5\phi_px_0^2dx_0\\ &= 0.5\phi_p(y^2+\sigma^2),\end{aligned}$$

we then obtain

$$\begin{aligned}\operatorname{Cov}(x_0, x_1) &= 0.5\phi_p(y^2+\sigma^2) - 0.5\phi_py^2\\ &= 0.5\phi_p\sigma^2.\end{aligned}$$

For $\operatorname{Cov}(x_1, x_2)$, it is

$$\begin{aligned}\operatorname{E}(x_1x_2) &= \iint p(x_2|x_1)p(x_1)x_2x_1dx_1dx_2\\ &= \int p(x_1)\phi_ax_1^2dx_1\\ &= \phi_a(\operatorname{Var}x_1 + (\operatorname{E}(x_1))^2)\\ \Rightarrow \operatorname{Cov}(x_1, x_2) &= \phi_a\operatorname{Var}x_1 + \phi_a(\operatorname{E}(x_1))^2 - \phi_a(\operatorname{E}(x_1))^2\\ &= 0.5\phi_a\phi_py + \phi_a(0.5\phi_p)^2(\sigma^2 - y).\end{aligned}$$

Again, the calculations for $\operatorname{Cov}(x_i, x_{i+1})$ for $i \in \{2, 3, 4\}$ are similar and straightforward. For $\operatorname{Cov}(x_5, x_6)$, the dependency on $x_0$ needs to be considered:

$$\begin{aligned}\operatorname{E}(x_5x_6) &= \iiint p(x_0, x_5, x_6)x_5x_6dx_0dx_5dx_6\\ &= \iiint p(x_5|x_0, x_6)p(x_6|x_0)p(x_0)x_5x_6dx_0dx_5dx_6\\ &= \int p(x_0)(\frac{1}{\alpha}x_0)0.5\phi_p\phi_a^4x_0dx_0\\ &= \frac{1}{\alpha}0.5\phi_p\phi_a^4(y^2+\sigma^2).\end{aligned}$$

With the techniques used here, all entries of the covariance matrix can be calculated correspondingly. Since no approximation occurs at this step, this is the exact covariance matrix $\operatorname{Cov}(x_0)$ and therefore positive semi-definite.

## D.2 First and second moments of $X_t|X_{t-1}$ in the seal model

We start by assuming that $x_{t-1}$ is normally distributed with mean $\mathrm{E}(x_{t-1}) = \mu$ and variance $\mathrm{Var}(x_{t-1}) = \Sigma$. We also use the following lemma that describes the relationship of two binomially distributed variables.

**Lemma 4.** *Let* $Y|X \sim \mathrm{Bin}(X, p_1)$ *and* $X \sim \mathrm{Bin}(n, p_2)$*. Then*

1. $Y \sim \mathrm{Bin}(n, p_1p_2)$
2. $\mathrm{Cov}(Y, X) = np_1p_2(1 - p_2)$

*Proof.* 1. We show this by calculating that the probability of $Y = y$ is $\binom{n}{y}(p_1p_2)^y(1 - p_1p_2)^{n-y}$ which is the same as for a binomially distributed random variable with $n$ trials and success probability $p_1p_2$. We use an index shift on line 3, namely $u = x-y$, and the binomial formula on line 5, $(x + y)^n = \sum_{k=0}^{n}\binom{n}{k}x^ky^{n-k}$.

$$\begin{aligned}
\mathbb{P}(Y = y) &= \sum_{x=0}^{n}\mathbb{P}(Y = y, X = x) = \sum_{x=0}^{n}\mathbb{P}(Y = y|X = x)\mathbb{P}(X = x)\\
&= \sum_{x=y}^{n}\frac{x!}{y!(x-y)!}p_1^y(1-p_1)^{x-y}\frac{n!}{x!(n-x)!}p_2^x(1-p_2)^{n-x}\\
&= p_1^yp_2^y\frac{n!}{y!(n-y)!}\sum_{u=0}^{n-y}\frac{(n-y)!}{u!(n-y-u)!}(1-p_1)^up_2^u(1-p_2)^{n-y-u}\\
&= (p_1p_2)^y\binom{n}{y}\sum_{u=0}^{n-y}\binom{n-y}{u}(p_2-p_1p_2)^u(1-p_2)^{n-y-u}\\
&= (p_1p_2)^y\binom{n}{y}(1-p_2+p_2-p_1p_2)^{n-y}
\end{aligned}$$

2. Using a property of the conditional expectation to calculate $\mathrm{E}(XY)$, we get

$$\begin{aligned}
\mathrm{E}(XY) &= \mathrm{E}(\mathrm{E}(XY|X)) = \mathrm{E}(X\,\mathrm{E}(Y|X)) = \mathrm{E}(p_1X^2) = p_1\left(\mathrm{Var}(X) + (\mathrm{E}(X))^2\right)\\
&= p_1\left(np_2(1-p_2) + (np_2)^2\right) = np_1p_2 - np_1p_2^2 + n^2p_1p_2^2\\
\mathrm{Cov}(X, Y) &= \mathrm{E}(XY) - \mathrm{E}(X)\,\mathrm{E}(Y) = np_1p_2 - np_1p_2^2 + n^2p_1p_2^2 - n^2p_1p_2^2\\
&= np_1p_2(1-p_2)
\end{aligned}$$

□

**First moments**

$$\begin{aligned}
x_{t,0}|x_{t-1} &\sim \text{Bin}(x_{t,6+},\alpha)|x_{t-1} \sim \text{Bin}(\text{Bin}(x_{t-1,6+},\phi_a)+\text{Bin}(x_{t-1,5},\phi_a),\alpha)\\
&= \text{Bin}(x_{t-1,6+}+x_{t-1,5},\alpha\phi_a)\\
\text{E}(x_{t,0}|x_{t-1}) &= \alpha\phi_a(x_{t-1,6+}+x_{t-1,5})\\
x_{t,1}|x_{t-1} &\sim \text{Bin}(x_{t-1,0},0.5\phi_{p,t})\\
\text{E}(x_{t,1}|x_{t-1}) &= 0.5\phi_{p,t}x_{t-1,0}\\
x_{t,2}|x_{t-1} &\sim \text{Bin}(x_{t-1,1},\phi_a)\\
\text{E}(x_{t,2}|x_{t-1}) &= \phi_a x_{t-1,1}\\
x_{t,3}|x_{t-1} &\sim \text{Bin}(x_{t-1,2},\phi_a)\\
\text{E}(x_{t,3}|x_{t-1}) &= \phi_a x_{t-1,2}\\
x_{t,4}|x_{t-1} &\sim \text{Bin}(x_{t-1,3},\phi_a)\\
\text{E}(x_{t,4}|x_{t-1}) &= \phi_a x_{t-1,3}\\
x_{t,5}|x_{t-1} &\sim \text{Bin}(x_{t-1,4},\phi_a)\\
\text{E}(x_{t,5}|x_{t-1}) &= \phi_a x_{t-1,4}\\
x_{t,6+}|x_{t-1} &\sim \text{Bin}(x_{t-1,5},\phi_a)+\text{Bin}(x_{t-1,6+},\phi_a)\\
\text{E}(x_{t,6+}|x_{t-1}) &= \phi_a\left(x_{t-1,5}+x_{t-1,6+}\right)
\end{aligned}$$

**Second moments**

$$\begin{aligned}
\text{Var}(x_{t,0}|x_{t-1}) &= \alpha\phi_a(1-\alpha\phi_a)(x_{t-1,6+}+x_{t-1,5})\\
\text{Var}(x_{1,t}|x_{t-1}) &= 0.5\phi_{p,t-1}(1-0.5\phi_{p,t-1})x_{0,t-1}\\
\text{Var}(x_{2,t}|x_{t-1}) &= \phi_a(1-\phi_a)x_{1,t-1}\\
\text{Var}(x_{3,t}|x_{t-1}) &= \phi_a(1-\phi_a)x_{2,t-1}\\
\text{Var}(x_{4,t}|x_{t-1}) &= \phi_a(1-\phi_a)x_{3,t-1}\\
\text{Var}(x_{5,t}|x_{t-1}) &= \phi_a(1-\phi_a)x_{4,t-1}\\
\text{Var}(x_{6+,t}|x_{t-1}) &= \phi_a(1-\phi_a)(x_{5,t-1}+x_{6+,t-1})\\
\text{Cov}(x_{0,t},x_{6+,t}|x_{t-1}) &= \alpha\phi_a(1-\phi_a)(x_{5,t-1}+x_{6+,t-1}).
\end{aligned}$$

All other covariances are 0, as the values are independent given $x_{t-1}$.

## D.3 Distribution of $Y$ when $Y|X \sim N(X, X^2/\tau)$ and $X \sim N(\mu,\sigma^2)$

**Lemma 5.** *Let $X$ be a random variable with $X \sim N(\mu,\sigma^2)$, and $Y$ be conditionally normally distributed with $Y|X \sim N(X, X^2/\tau)$ for $X \neq 0$ and $Y = 0$ for $X = 0$.*

1. *The unconditional expected value of $Y$ is $\text{E}(Y) = \mu$.*
2. *The unconditional variance of $Y$ is $\text{Var}(Y) = \sigma^2\left(1+\frac{1}{\tau}\right)+\frac{\mu^2}{\tau}$.*
3. *The unconditional distribution of $Y$ is not normal.*

*Proof.* 1. Using the properties of the conditional expectation (see for example p. 333 in Ross, 2011), we can calculate the unconditional expected value of $Y$ with

$$\text{E}(Y) = \text{E}(\text{E}(Y|X)) = \text{E}(X) = \mu.$$

2. Using the law of total variance, we calculate

$$\begin{aligned}\mathrm{Var}(Y) &= \mathrm{E}(\mathrm{Var}(Y|X)) + \mathrm{Var}(\mathrm{E}(Y|X)) = \mathrm{E}(X^2/\tau) + \mathrm{Var}(X) = \\ &= \left(\mu^2 + \sigma^2\right)/\tau + \sigma^2 = \mu^2/\tau + \sigma^2(1 + 1/\tau).\end{aligned}$$

3. We show this for the simplified case with $\tau = 1$, $\mu = 0$ and $\sigma^2 = 1$ by calculating the fourth moment of $Y$, so $\mathrm{E}(Y^4)$. With these parameter choices, and using the first and second part of this lemma, $Y$ has mean 0 and variance 2. The random variable $X$ has standard normal distribution, so mean $\mu = 0$ and variance $\sigma^2 = 1$. Its fourth central moment is

$$\mathrm{E}(X^4) = \mu^4 + 6\mu^2\sigma^2 + 3\sigma^4 = 3,$$

using a standard formula for the fourth moment of normally distributed random variables (Papoulis and Pillai, 2002). For $Y$, we use the properties of the conditional expectation for the first equality and the same standard formula as above for the second equality, and obtain

$$\begin{aligned}\mathrm{E}(Y^4) &= \mathrm{E}(\mathrm{E}(Y^4|X)) = \mathrm{E}(X^4 + 6X^2 \cdot X^2 + 3X^4) = 10\,\mathrm{E}(X^4) \\ &= 30.\end{aligned}$$

However, if $Y$ were normally distributed, the formula for the fourth central moment above would yield

$$E(Y^4) = \mathrm{E}(Y)^4 + 6\,\mathrm{E}(Y)^2\,\mathrm{Var}(Y)^2 + 3\,\mathrm{Var}(Y)^4 = 3 \cdot 2^4 = 48.$$

Since the two numbers are not the same, $Y$ is not normally distributed.

□

# Appendix E

# Chapter 5: Additional Results

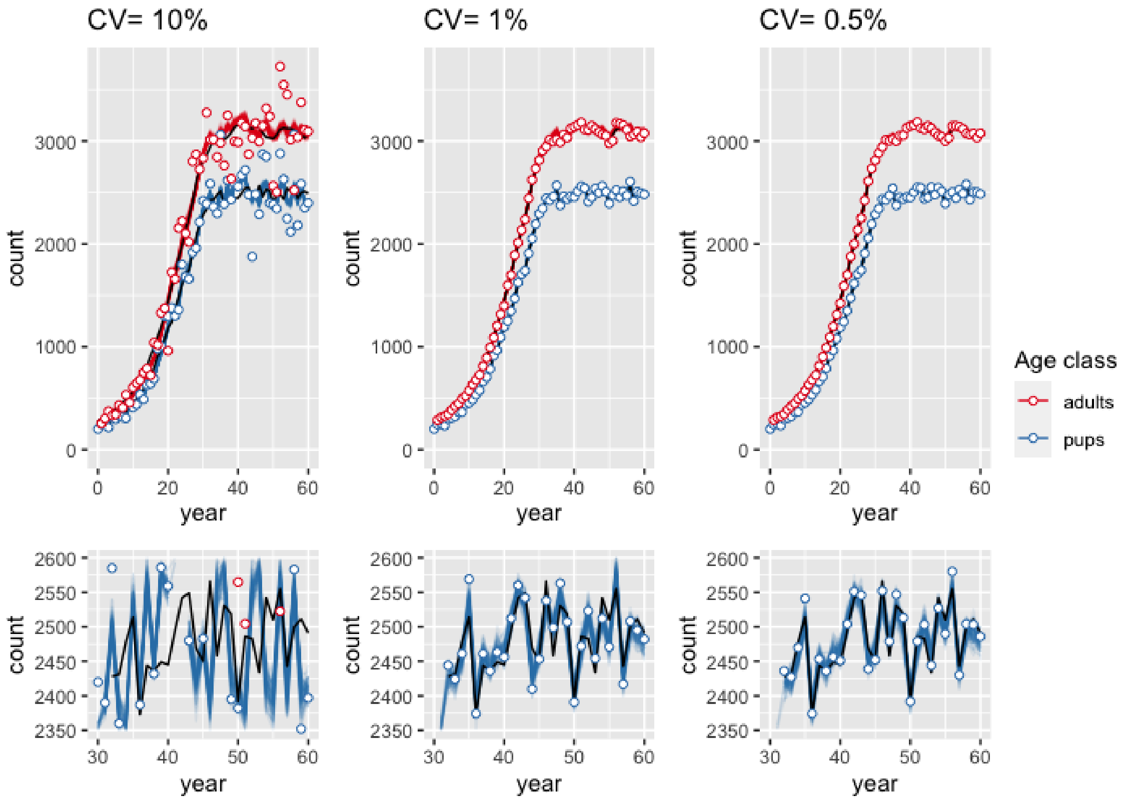


**Figure E.1:** Smoothed posterior state estimates of both pups and adults when the true parameter value is used in combination with adult and pup observations with a varying CV. The white circles show the observed pup and adult counts. In the second row, both axis limits are reduced to better illustrate the reduction in variance of the pup state estimates as the CV decreases.

Appendix F

# Chapter 6: Additional Results

| | Factorised | | Joint | |
|---|---|---|---|---|
| $N$ | CV | Var of log-LH | CV | Var of log-LH |
| 3 | 343368.06 | | 548082.79 | |
| 10 | Inf | 36330.58 | | 325170.27 |
| 30 | 10.00 | 8980.12 | Inf | 127031.70 |
| 100 | 9.93 | 4530.79 | Inf | 55479.05 |
| 300 | 9.87 | 2252.07 | Inf | 33073.51 |
| 1000 | 9.92 | 1188.08 | 10.00 | 15265.25 |
| 3000 | 10.00 | 816.68 | 8.26 | 9237.57 |
| 10000 | 10.00 | 623.22 | 10.00 | 5810.34 |
| 30000 | 10.00 | 333.67 | 10.00 | 4180.68 |
| 100000 | 9.75 | 225.60 | 9.34 | 1936.37 |
| log-L | -1194.2 | | -1379.0 | |

**Table F.1:** Measures of variation of 100 likelihood estimates computed by a particle filter with varying numbers of particles $N$. The likelihood was estimated with a particle filter at the prior mean (Parameter 1 in Table 6.1), using the factorised and the joint formulation. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 100,000$ particles.

| | Factorised | | Joint | |
|---|---|---|---|---|
| $N$ | CV | Var of log-LH | CV | Var of log-LH |
| 3 | 9.93 | 20601.45 | 10.00 | 57692.10 |
| 10 | 10.00 | 1736.32 | 8.60 | 18481.01 |
| 30 | 9.90 | 332.65 | 8.47 | 13264.31 |
| 100 | 10.00 | 138.12 | 7.88 | 3399.35 |
| 300 | 9.01 | 93.25 | 10.00 | 1342.59 |
| 1000 | 9.70 | 61.48 | 9.70 | 451.61 |
| 3000 | 6.44 | 33.63 | 10.00 | 184.85 |
| 10000 | 9.29 | 28.46 | 7.51 | 61.19 |
| 30000 | 5.49 | 16.56 | 7.04 | 49.54 |
| 1e+05 | 3.25 | 13.77 | 5.41 | 27.22 |
| log-L | -895.80 | | -902.01 | |

**Table F.2:** Measures of variation of 100 likelihood estimates computed by a particle filter with varying numbers of particles $N$. The likelihood was estimated with a particle filter at Parameter 2 in Table 6.1, using the factorised and the joint formulation. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 100,000$ particles.

| | Factorised | | Joint | |
|---|---|---|---|---|
| $N$ | CV | Var of log-LH | CV | Var of log-LH |
| 3 | 9.66 | 3528.54 | 9.44 | 10529.58 |
| 10 | 5.54 | 369.05 | 8.20 | 3005.43 |
| 30 | 5.78 | 52.26 | 9.59 | 1492.40 |
| 100 | 4.83 | 23.61 | 8.91 | 393.46 |
| 300 | 4.84 | 7.89 | 4.05 | 162.59 |
| 1000 | 1.59 | 3.45 | 4.13 | 46.35 |
| 3000 | 1.14 | 1.43 | 3.79 | 11.09 |
| 10000 | 0.68 | 0.59 | 2.99 | 3.91 |
| 30000 | 0.50 | 0.20 | 1.64 | 1.55 |
| 100000 | 0.27 | 0.07 | 0.75 | 0.56 |
| log-L | -847.32 | | -845.93 | |

**Table F.3:** Measures of variation of 100 likelihood estimates computed by a particle filter with varying numbers of particles $N$. The likelihood was estimated with a particle filter at Parameter 3 in Table 6.1, using the factorised and the joint formulation. The first column gives the CV of the likelihood estimates, the second column the variance of the log-likelihood estimates. The bottom row gives the log of the mean of 100 likelihood estimates as calculated with $N = 100,000$ particles.