Statistical issues in ecological simulation models

Michael Spence
October 2015

Thesis submitted to the
University of Sheffield for
the degree of Doctor of
Philosophy
School of Mathematics and
Statistics

Acknowledgements

I would like to thank NERC and NCSE for funding my PhD. Thank you very much to my supervisor Professor Paul Blackwell for his continued guidance, knowledge and life advice. Thanks to Dr. Julia Blanchard for helping me with the North Sea model.

Thanks to other staff and students for their advice and help during my studies.

Thanks to all of my friends and family for continued support throughout my studies.

We are the Boro!!!

Abstract

Complex simulation models are being increasingly used in ecological modelling as a way of trying to understand a system by examining the processes that make up that system. Complex simulation models generally model behaviour of a system through a series of rules or algorithms, rather than describing it in a formal mathematical way and this can be a good way of capturing an ecologist's expertise and intuition. When interpreting outputs from such a model, it is important to allow for uncertainty due to parameter values which may not be known precisely and structural or implementation aspects. This thesis develops and applies a number of new statistical methods for handling uncertainty in such models.

For stochastic simulation models with intractable likelihoods, parameter estimation can be done using Approximate Bayesian Computation with Markov Chain Monte Carlo (ABC-MCMC). This method does not mix well in the tails of the distribution. In this thesis we develop a version of ABC-MCMC that treats the random inputs as unknown as well as the unknown model parameters and we show empirically that this improves the efficiency of the ABC-MCMC algorithm on a queuing model and an individual-based model (IBM) of the group-living bird, the woodhoopoe.

For models that are expensive to run, inference may be challenging even if the likelihood can be evaluated. We consider a deterministic multi-species size-based marine ecosystem model, with unknown initial states and parameters, and carry out Bayesian inference using a combination of MCMC and optimisation algorithms.

Stochastic simulation models, especially IBMs, often have model uncertainty that is down to some seemingly arbitrary choices, for example spatial or temporal scales, the timing of different events or the spatial configuration. Ideally the outputs of the model should be insensitive to these choices. We develop methods for variance-based sensitivity analysis for stochastic models, allowing us to assess the sensitivity of the model outputs to stochasticity of the inputs and to partition out the variance between submodels. This enables us to test the arbitrary choices made by the modeller and thus test the robustness of the model. We demonstrate these methods on two IBMs: the woodhoopoe model and a bird breeding synchrony model.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In ecology, as in many other areas, the need for answering the question "what makes something happen?" as opposed to "what actually happens?" is becoming increasingly recognised. This question often leads to building simulation models, or complex models as they are sometimes known, where the different aspects of the system are modelled separately and give rise to the collective behaviour of the system. Simulation models generally model behaviour through a series of rules or algorithms, rather than describing it in a formal mathematical way. They describe the system processes in a number of submodels: for example in a marine ecosystem model, consumption, production, migration, predation, recruitment, habitat dependancy and mortality may all be modelled individually (Pinnegar, 2014). By building a simulation model in this way, it enables the user to create a "virtual laboratory" which would allow them to perform potentially expensive or impossible real life experiments cheaply and with very little risk to the environment. For example, in a marine ecosystem model, one could experiment with different fishing strategies and see their consequences without the risk of irreversible change in the real ecosystem.

One class of simulation models that are being increasingly used in ecology are individual-based models (IBMs). Grimm (1999) describes IBMs as a "bottom up" approach to modelling. By modelling the individuals, the properties of the complex system can be traced to the behaviour or the individuals in it (Kaiser, 1979). There is no strict definition to distinguish IBMs from more classical models (Grimm and Uchmański, 1996) but in IBMs the individuals of the system are explicitly modelled and all of the modelling is done at the individual level as opposed to the population level as in more conventional models. Interactions of these individuals result in global consequences (Reynolds, 1997) that are often the aim of the IBM to understand; for example plants and animals in ecosystems, vehicles in traffic or people in crowds.

An example of an IBM is the marmot model by Grimm et al (2003).

In this model a marmot goes through its life, first being young living in a group and then trying to become the dominant animal of that group before eventually dying; subdominant marmots only know what is going on in their own territory, so they will have to leave the territory to go to another one; the number of marmots at anytime in the model is a discrete count and marmots are born with different weaning weights and can have different social ranks throughout their lives.

One common way of building IBM is to build an agent-based model (ABM). ABMs are a relatively new approach to modelling complex systems (Chen et al, 2011). They consist of interacting agents that behave automatically, often described by simple rules. The rules are dependent on the agents themselves, the environment or other agents and may have stochastic elements to them. For example, in a territory model, an agent might leave their territory if there is a dominant animal in that territory, or else with a certain probability. Agents are active and make independent decisions, often trying to achieve their own goals. In ecology the goal is usually to seek fitness (an attempt to pass on their genes to future generations) (Grimm and Railsback, 2005). Through these interactions, patterns and behaviours emerge that were not explicitly programmed in the models (Macal and North, 2010).

One of the major advantages of ABMs is that it allows each agent to be different. This is a major attraction to ecology as individuals differ from each other and interact in different ways with the environment (Grimm and Railsback, 2005). Not only this, but the environment may also be heterogeneous. This is also an attraction to IBMs, as again the environment is not the same everywhere in nature (for example, there are parts of land that are extremely fertile and other parts that are barren). ABMs allow these differences to be explicitly programmed into the model.

ABMs allow the experts to use their expertise and intuition in building the model. The aim of ABMs is not only to recreate the system outputs, but trying to recreate aspects of the real system. This is because the majority of ABMS are built to try to understand the system as opposed to predict it (Heath et al, 2009). Agent-based models are very popular as they seem a natural way of modelling a system and because of this ABMs are being used increasingly in ecology (Grimm et al, 2003; Franz et al, 2010; Hovel and Regan, 2008) so much so that standardised ways of building and describing these models have been developed (Grimm et al, 2006, 2010, 2014, 1996).

Simulation models are more “problem orientated” than more conventional models at a cost of tractability (Silverman, 1984). They are often developed with algorithms that are not well tuned from the beginning and require parameters that are either not precisely known in the literature, or simply not concretely measurable (Piou et al, 2009). As the probabilistic behaviour of the model is implicit in the rules of the model, the likelihood is generally intractable which means that traditional statistical methods of parameter estimation are not possible.

Sometimes it is possible to perform likelihood-free inference (Wilkinson, 2010a) by running the model for a parameter set in order to estimate the likelihood for these inputs. This is often used when the model is deterministic and some kind of tractable model error is assigned to the output of the model to relate the model to the observed data. One such class of models that this could be applied to are size-based models of the marine ecosystem.

However when the model is stochastic, the output of the model for a particular set of parameters is still uncertain, even after the model has been run, and thus another approach is required. One possible approach in this case is Approximate Bayesian Computation (Beaumont, 2010; Tavaré et al, 1997), which is described in Section 3.3. This method has been developed and different hybrid algorithms have been created in order to better estimate the posterior distribution. One of these, ABC-MCMC (Marjoram et al, 2003) which is based on the well-established Metropolis-Hastings algorithm (Section 3.2), has been suggested but has poor mixing qualities especially in the tails of the distribution. In this thesis we will develop a method of coupling the random inputs in order to improve the mixing of this algorithm.

In order to create a simulation model often assumptions need to be made to simplify the model enough to make it useful. These could include the scale of the model (Holland et al, 2009; Chen and Mynett, 2003) or the order in which submodels are run with in the model. For the model to be useful, the output should not be sensitive to the assumptions or details of the model (Grimm and Railsback, 2005; Railsback and Grimm, 2012). Although some efforts have been made to test these assumptions (Chen and Mynett, 2003; Kloprogge et al, 2011; Maclean, 2010) there is no formal quantitive mathematical framework to investigate the sensitivity of the structure of the model. However there is a framework to investigate the sensitivity of the parameters.

Sensitivity analysis is often performed on simulation models with quite an extensive literature on global sensitivity analysis (Sobol', 1993; Saltelli et al, 2000; Oakley and O'Hagan, 2004) for deterministic simulation models however there has been little work for stochastic models (Iooss and Lemaître, 2015). The lack of work on stochastic models means that global sensitivity analysis in not performed on quite a number of IBMs (Railsback and Grimm, 2012). Instead local sensitivity is often performed as an alternative but this does not give the sensitivity of the model as a whole but just part of it. In this thesis we develop a method of performing global sensitivity analysis on stochastic simulation models that we apply to test whether the model is sensitive to its assumptions and details.

In Chapter 2 we describe two IBMs that we are going to use in the rest of the thesis. In Chapter 3 we review potential methods of estimating the parameters both when the likelihood is tractable and when it is not. We will also introduce some algorithms that are used in later chapters. In Chapter 4 we develop a method of coupling the random inputs of a stochastic model and show empirically that it enables us to improve the performance of ABC-MCMC. In Chapter 5 we introduce size spectrum models and perform parameter estimation on a multi-species size-based model of the North Sea (Blanchard et al, 2014). The work in this chapter has been motivated by a specific problem brought to us by Julia Blanchard in the Animal and Plant Sciences department of the University of Sheffield. In Chapter 6 we briefly review global sensitivity analysis before developing a method of performing global sensitivity analysis on stochastic simulation models and using this to perform robustness analysis on two IBMs in Chapter 7. A brief overview to the findings of the thesis and the future work is described in Chapter 8.

Chapter 2

Models

In this chapter we describe the main models that are used later in the thesis. We describe the bird synchrony chapter in Section 2.1 and the woodhoopoe model in Section 2.2. The multi-species size-based model is described in Chapter 5 as it only appears in there.

The models in this chapter are described using the standard way used to describe IBMs, namely the ODD protocol (Overview, Design, Details) (Grimm et al, 2006). Using this protocol, the author first gives an overview of their model under the subheadings: Purpose; Entities, State variables and scales and Process overview and scheduling. Then they describe the Design concepts and finally they describe the details of the model under the subheadings: Initialisation; Input data and Submodels.

2.1 Bird synchrony model

Jovani and Grimm (2008) built an individual based model that modelled the laying times of birds. They say that in order for the birds to lay their eggs it is important that there is calm and that each bird assesses her neighbours' stress levels, and when they are calm enough, she lays her eggs.

First we will describe the original model, based on the descriptions of Jovani and Grimm (2008) and Railsback and Grimm (2012), which we have re-parameterised and then describe some possible extensions which we will test in Chapter 7.

2.1.1 Original model

Purpose

The purpose of the model is to see how local interactions affect the breeding synchrony of colonial birds.

State variables and scales

The entities are female birds that each occupy a stationary nest and are characterised by their own stress level (OSL) and the coordinates of their nest site. The nest sites are on a homogeneous square 15×15 grid that is arranged on a torus. One time step in the model represents 1 day with each model run until all of the birds have laid their eggs. Simulations are run with all of the nests fully occupied.

Process overview and scheduling

At each time step, the stress level of each bird is updated according to its own OSL and that of its eight neighbours. If an individual's stress level falls below a threshold (re-parameterised to 0 for this work) she will lay her eggs and her stress level will be to set to 0 for the rest of the simulation. The updating of the stress levels is done simultaneously for all of the birds.

Design Concepts

Breeding synchrony at global level is caused by synchrony at local levels. Birds adapt their stress level to those around them. If an individual's neighbours are stressed then the individual's stress level will increase and her laying day will therefore be delayed. It is assumed that an individual can sense the stress level of their eight neighbours but no further. Stochasticity is assumed in the initial distribution of a stress level. The laying date of each bird is observed.

Initialisation

Initially each of the birds have stress levels generated stochastically from a uniform distribution between 10 and $m + 10$. m is thus the range of initial stress levels.

Input

The model does not have any external inputs.

Submodels

The model has only one submodel that is how the new stress levels are calculated. The new stress level, OSL_{t+1} for bird i is

$$OSL^i_{t+1} = NR \times \text{mean}NSL^i_t + (1 - NR) \times OLS^i_t - 1$$

where $\text{mean}NSL^i_t$ is the mean OSL_t of the eight neighbours of of bird i. NR is the neighbourhood relevance which is the amount that an individual's stress level is affected by its neighbours. If $NR = 1$, then the stress level of the individuals becomes the mean of the stress levels of the neighbours less 1. If $NR = 0$ then there is no interaction between the birds and the stress level reduces by 1 per time step.

2.1.2 Variate submodels

Neighbours

Instead of meanNSL_t^i being the mean OSL_t of an individual's 8 neighbours (Moore neighbourhood) each bird could examine the mean OSL_t of her 4 neighbours (Von Neumann neighbourhood).

Stochastic Arrival

Rather than the model being initialised with all of the grid full of birds, the grid begins empty and each bird's arrival time is sampled from a geometric distribution with parameter λ. Once a bird arrives her initial stress level is sampled uniformly between 10 and $10 + m$.

Stochastic Reduction

The stress level for bird i at time $t + 1$ is

$$OSL_{t+1}^i = NR \times \text{mean}NSL_t^i + (1 - NR) \times OLS_t^i - 1 + \sigma\epsilon_t^i$$

where $\epsilon_t^i \sim N(\cdot|0, 1)$.

2.2 Woodhoopoe model

Woodhoopoes are birds that can be found in the sub-Saharan (du Plessis, 1992). They live in groups just like wolves with one dominant pair which are the only ones that breed. Quite often two of these groups will meet and engage in a conflict. The conflict, which is performed by all the adults but with the sub-dominants taking a larger part, involve both groups taking it in turn to sing choruses at each other. As the conflict continues some of the sub-dominants start waving bits of bark around like flags which is rewarded by petting from the dominants after the conflict.

Neuert et al (1995) used an individual-based approach in order to model the population and group dynamics of the woodhoopoes. Railsback and Grimm (2012) simplified this model for use as examples in their textbook, and it is this simpler version which we will describe here. We are going to fit this model to data using the method developed in Chapter 4. We also perform robustness analysis on this model in Chapter 7. The model is described below using the ODD protocol.

Purpose

“The purpose of the model is to see illustrate the dynamics of a population of group-living woodhoopoe, and the dynamics of its social groups, depend on the trait individuals use to decide when to leave their group. The model provides a laboratory for developing theory for the woodhoopoes’ scouting foray traits.” (Railsback and Grimm, 2012).

Entities, State variables, and Scales

The model’s entities are territories and the birds. A territory is a collection of birds and the space that the group occupy. Territories can be empty (no birds in the territory). There are 25 territories that are positioned in a one dimensional row that is wrapped so that the two ends of the row are considered next to each thus creating a circle. The territories have state variables that determine their position in the row and whether or not they have a dominant male and a dominant female. Birds have state variables that determine whether or not they are a dominant, which territory they are in, their sex and their age (in months).

The time step for the model is one month and it is run for 25 years with the first 5 years being burn-in.

Process, Overview and Scheduling

During a time step, the following processes will occur:

1. Dates and ages are updated.
2. Territories fill vacant dominant positions.
3. Birds undertake forays and experience predation.
4. Dominant females reproduce.
5. Birds experience natural mortality.
6. Output is then produced.

The woodhoopoe execute these processes in a random order and the state variables are updated after each action.

Design Concepts

Basic principles

“The model explores the “stay-or-leave” question: when should a subordinate individual leave a group that provides safety and group success but restricts opportunities for individual success?” (Railsback and Grimm, 2012).

Emergence

We are interested in reproducing the characteristic group size pattern as described by Railsback and Grimm (2012) and the fact that more subordinate woodhoopoe leave their groups earlier in the year than later. Also the subordinates that leave their groups in order to become a dominant with another group are generally younger than those that stay in their groups.

Adaptation

"The only adaptive decision that the woodhoopoe make is whether or not to undertake a scouting foray." (Railsback and Grimm, 2012).

Objectives

Each woodhoopoe's goal is to become a dominant.

Sensing

Woodhoopoe know nothing about other territories until they foray and try to become a dominant in another group. At this stage the woodhoopoe knows whether or not the group in question has a vacant dominant position that the woodhoopoe could fill.

Stochasticity

There is a number of different stochastic parts for the model to run. Firstly the initial age of the woodhoopoe that are alive when the model is initialised. The order in which subordinates leave their group to foray with a certain probability (θ_2) is determined randomly. If the subordinate leaves their group they die with a certain probability (θ_3) and assuming they survive scout to the left or right of their own territory with equal probability. When a woodhoopoe is born the sex of the woodhoopoe is determined randomly. Each woodhoopoe also dies with a probability (θ_1).

Collectives

"The social groups are the collectives: their state affects the individual birds, and the behaviour of individuals determines the state of the collectives. Because the model's "territory" entities represent the social groups as well as their space, the model treats behaviours of the social groups (promoting dominants) as territory traits." (Railsback and Grimm, 2012).

Observation

The total population of woodhoopoe, the number of adult woodhoopoe in each group, the number of vacant dominant positions in the whole model, the mean age of subordinates that foray, the mean age of subordinates that don't foray and the total number of forays each month will all be observed.

Initialisation

Simulations start in January (month 1). Every territory starts with two males and two females all aged uniformly between 1 and 24 months. The oldest of each sex becomes the dominant in that territory.

Input

"The model does not use any external input." (Railsback and Grimm, 2012).

Submodels

Dates and ages are updated

The current year and month are advanced by one month and the ages of the birds are advanced by one month.

Territories fill vacant alpha positions

If a territory lacks a dominant position and has a subordinate of the right sex then the oldest subordinate of that sex becomes the new dominant.

Birds undertake forays

A subordinate adult (aged a year or over) will scout and look to become a dominant in another territory if they are not the oldest subordinate of their sex in their territory and with a probability θ_2 (leave submodel). If a bird decides to scout it will be subject to a predication mortality. The bird will die with a probability θ_3 (predation submodel). If a bird survives this additional mortality they either move left or right with an equal chance and inspect the territory next to their current territory (left/right submodel). If there is an vacant dominant position that the bird can occupy in that territory it will occupy it (take-over submodel). If not it will move to the territory next to that one and continue until either it has become a dominant in a territory or has searched in 5 territories. If the bird is unsuccessful it will return to its home territory and continue to be a subordinate.

Dominant females reproduce

In the twelfth month every year, dominant females that have a dominant male in their territory will produce two offspring. The offspring have their age set to zero and their sex is chosen randomly with equal probability of being a male and female.

Birds experience mortality

All birds will die with a probability θ_1 (natural mortality submodel).

2.2.1 Absorbing state

There is an absorbing state in this model where all of the woodhoopoe die. This happens for high values of θ_1. In Chapter 4 we use Approximate Bayesian Computation (described in Section 3.3) to fit this model to simulated data. If during a model run, the absorbing state is reached then the proposed parameter set would be rejected.

In Chapter 7 we perform variance based sensitivity on outputs of the model. The outputs that we examine are still defined if the model reaches the absorbing state. Having said this, we did not find that we reached the absorbing state in any runs of the model either in Chapter 4 or Chapter 7.

Chapter 3

Methods of parameter estimation

3.1 Introduction

Once you have a model parameterisation and structure it is often necessary to fit the model to data. This means finding values of the parameters, θ, that could generate the data $\boldsymbol{x}$ from the model. This is often done when the model is used to forecast events and to validate the model. In statistics there are two different approaches of performing parameter estimation.

In classical frequentist statistics, the number of times an event occurs in n independent and identical trials tends towards the probability of the event taking place as $n \to \infty$. This means that in frequentist statistics the trial under which the event occurs must be repeatable. Furthermore a $i\%$ confidence interval for a parameter θ means that if we were to repeat the trials, as the number of trials $n \to \infty$, $i\%$ of the intervals would contain the true value of the parameter. This does not mean that the probability of the parameter being in a particular interval, for example the one calculated from the data, is $i/100$, as in the classical approach, the parameter is fixed even if unknown, so given the data, the event $\{\theta \in \text{the confidence interval}\}$ is either true or false, deterministically.

In subjective probability it is possible to state probabilities of any event including non-repeatable ones. In subjective probability, the probability of an event is an individual's belief in the likelihood of the event occurring on a scale from 0 to 1. Bayesian inference is conducted using subjective probability and therefore can be used to make probability statements about parameters.

In order to make these statements we use Bayes theorem,

$$p(\theta|\boldsymbol{x}) = \frac{p(\theta)p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x})}.$$

This equation can be separated into the prior $p(\theta)$, the likelihood $p(\boldsymbol{x}|\theta)$, sometimes written $l(\theta|\boldsymbol{x})$, and the evidence $p(\boldsymbol{x})$. The evidence, sometimes known as the normalising constant or just the marginal distribution for $\boldsymbol{x}$, can be written as

$$p(\boldsymbol{x}) = \int_{\Theta} p(\theta)p(\boldsymbol{x}|\theta)d\theta.$$

The prior, $p(\theta)$, is the decision-maker's prior beliefs about the parameters before we look at the data. This can be determined from literature or past experiments, elicited from an expert or set to be non-informative. For a description of elicitation see O'Hagan et al (2006). The likelihood, $l(\theta|\boldsymbol{x})$, is the same as in the frequentist case which is the probability (or density) of observing the data $\boldsymbol{x}$ conditional on the parameter value. If the prior is non-informative, the posterior, $p(\theta|\boldsymbol{x})$, is equivalent to the normalised likelihood.

Sometimes it is possible to evaluate the normalising constant analytically meaning that the posterior can be solved analytically which is the case when we have conjugate priors (Lee, 2004). This was done by Johnson and Briggs (2011) when they performed parameter estimates on their IBM, examining chytridiomycosis in frogs. However it is often the case that we only know the posterior up to a normalising constant and therefore we have to estimate it numerically. There are a number of Monte Carlo methods (Metropolis and Ulam, 1949) including rejection-samplers (Rubin, 1984), population Monte Carlo (Cappé et al, 2004) and Markov Chain Monte Carlo (Metropolis et al, 1953; Hastings, 1970) that are used to estimate or sample from the posterior distribution. A brief overview of the latter is described in Section 3.2. It is mostly just review but we describe a novel hybrid algorithm in Section 3.2.3 that we use in Chapter 5.

For models that are built using mechanistic rules it is often the case that the likelihood is intractable as it either cannot be written down or is expensive to calculate. If this is the case then other methods of parameter estimation are required (Sottoriva and Tavaré, 2010). One natural method is Approximate Bayesian Computation (ABC) and a brief overview is described in Section 3.3. There are many other methods that could be used for this problem and we have given a brief overview of some of them in Section 3.4 with some comments on our experience of using them. This section is not essential reading for the rest of the thesis.

3.2 Markov Chain Monte Carlo

If we are trying to sample from a the distribution $\pi(\theta)$ but are unable to write it down analytically, we can create a Markov chain whose stationary distribution is $\pi(\theta)$. This means that once the Markov chain has reached its stationary distribution, we can use it to take samples of $\pi(\theta)$. This is known as the Markov Chain Monte Carlo method (MCMC).

3.2.1 Metropolis Hastings

One method of creating a Markov chain with the required stationary distribution is the Metropolis-Hasting algorithm (Algorithm 1) (Metropolis et al, 1953; Hastings, 1970). Given the current point θ_t a new point is proposed, θ', according to a proposal distribution $q(\cdot|\theta_t)$ and is then accepted with probability

$$\min\left(1, \frac{\pi(\theta')q(\theta_t|\theta')}{\pi(\theta_t)q(\theta'|\theta_t)}\right).$$

When trying to estimate more than one parameter it is possible to change a number of parameters at a time. Sometimes the normalising constant can be calculated analytically by marginalising the target distribution. If this is the case the point can be sampled from this distribution and accepted with probability 1. This is known as a Gibbs sampler (Geman and Geman, 1984). Sampling from this algorithm involves simulating the chain until it has reached its stationary distribution and the desired sample size has been taken. However the Markov chain may take a while to find the stationary distribution. The period of time it takes to get into the stationary distribution is known as the "burn in" period and is discarded as it is not sampled from the target distribution. A common application of MCMC is where the target distribution $\pi(\theta)$ is the posterior distribution for θ. This simply requires

Algorithm 1 Metropolis-Hastings algorithm

```
Generate a candidate point $\theta' \sim q(\cdot|\theta_t)$
$\alpha \leftarrow \min\left(1, \frac{\pi(\theta')q(\theta_t|\theta')}{\pi(\theta_t)q(\theta'|\theta_t)}\right)$
Sample $u \sim \mathrm{U}(0,1)$
if $u < \alpha$ then
    $\theta_{t+1} \leftarrow \theta'$
else
    $\theta_{t+1} \leftarrow \theta_t$
end if
```

$$\pi(\theta) \propto p(\theta)l(\theta|\boldsymbol{x})$$

and the normalising constant is not needed.

It is important to choose a good proposal distribuition $q(\cdot|\theta)$. It has been shown that for a multivariate normal posterior and proposal distribution that the optimal acceptance rate is 0.234, and this result seems (by simulation) robust (Roberts and Rosenthal, 2001; Neal and Roberts, 2006).

There are a number of adaptive algorithms that help choose the proposal distribution (Andrieu and Thoms, 2008) as well as other methods that change how the proposal is chosen including Multiple-try MCMC, (J. Liu and Wong, 2000) Differential Evolution Adaptive Metropolis (DREAM) (ter Braak and Vrugt, 2008; Vihola, 2012; Vrugt et al, 2009; ter Braak, 2006), Hamiltonian Monte Carlo (HMC) (Duane et al, 1987), Langevin adjusted Monte Carlo (Roberts and Tweedie, 1996) and Riemann manifold Metropolis adjusted Langevin algorithm (Girolami and Calderhead, 2011).

When the likelihood, $l(\theta|\boldsymbol{x})$, is slow to calculate, the Metropolis-Hastings algorithm could be sped up by a method suggested by Cui et al (2011). At each time step the MCMC chain proposes n points to move to. At each of these points the likelihood is calculated and then the first of the point is considered. If it is rejected the second is considered, then the third and so on until either one of the points has been accepted or all of the points have been rejected. It is essentially saying that conditional on the current point, a proposed point being rejected and the next proposed point are independent of one another and therefore can be run in parallel. This algorithm speeds up the the Metropolis-Hastings algorithm up as a number of likelihoods are estimated in parallel. This is summed up in Algorithm 2. If $n = 1$, the algorithm is the same as the Metropolis-Hastings algorithm (Algorithm 1).

Algorithm 2 Parallel Metropolis-Hastings algorithm

```
Generate $n$ candidate points $\theta'_i \sim q(\cdot|\theta_t)$
$j \leftarrow 0$
while $j < n$ or there is an accepted point do
    $j \leftarrow j + 1$
    $\alpha \leftarrow \min\left(1, \frac{\pi(\theta')q(\theta_t|\theta'_j)}{\pi(\theta_t)q(\theta'_j|\theta_t)}\right)$
    Sample $u \sim \mathrm{U}(0, 1)$
    if $u < \alpha$ then
        $\theta_{t+1} \leftarrow \theta'$
    else
        $\theta_{t+1} \leftarrow \theta_t$
        $t \leftarrow t + 1$
    end if
end while
```

Suppose we were unable to estimate the likelihood $l(\theta|\boldsymbol{x})$ exactly but we were able to sample

$$Z|\theta, \boldsymbol{x} \sim f(\theta, \boldsymbol{x})$$

where $f(\theta, \boldsymbol{x})$ is a noisy estimate of the likelihood, with the properties that $Pr(z < 0) = 0$ and $E(Z) = l(\theta|\boldsymbol{x})$, then an MCMC algorithm (for example see Algorithm 3 (Beaumont, 2003)) can be set up with stationary distribution

$$\pi_{PM}(\theta, z) \propto p(\theta) z f(z|\theta, \boldsymbol{x})$$

which when marginalised becomes

$$\int p(\theta) z f(z|\theta, \boldsymbol{x}) dz = p(\theta) p(\boldsymbol{x}|\theta).$$

This means that the sample of θs found using Algorithm 3 will follow the posterior distribution. This is known as Pseudo-Marginal MCMC (Andrieu and Roberts, 2009).

Algorithm 3 Pseudo-Marginal MCMC

```
Generate a candidate point $\theta' \sim q(\cdot|\theta_t)$
Generate a candidate likelihood estimate $Z' \sim f(\cdot|\theta', \boldsymbol{x})$
$\alpha \leftarrow \min\left(1, \frac{\pi(\theta')Z'q(\theta_t|\theta')}{\pi(\theta_t)Zq(\theta'|\theta_t)}\right)$
Sample $u \sim \mathrm{U}(0,1)$
if $u < \alpha$ then
    $\theta_{t+1} \leftarrow \theta'$
    $Z \leftarrow Z'$
else
    $\theta_{t+1} \leftarrow \theta_t$
end if
```

3.2.2 Parallel tempering

If $\pi(\theta)$ is multi-modal, the Metropolis-Hastings algorithm would not move between the modes in a reasonable amount of time. One way of getting round this is by parallel tempering (Swendsen and Wang, 1986). Parallel tempering uses the idea that if the target distribution, $\pi(\theta)$, is smoothed, that allows a Metropolis-Hastings algorithm to move between modes easily. We define

$$\begin{aligned} \pi_i(\theta) &= \frac{\pi(\theta)^{\tau_i}}{\int \pi(\theta)^{\tau_i} d\theta} \\ &\propto \pi(\theta)^{\tau_i} \end{aligned}$$

where $0 \leq \tau_i \leq 1$. The scaling parameter, τ_i, is often referred to as the temperature of the distribution. The distribution, $\pi_i(\theta)$, will become more peaked the larger τ_i is. If $\tau_i = 0$ the distribution will be completely flat. For smaller values of τ_i a Metropolis-Hastings algorithm will be able to make bigger moves whilst keeping good mixing properties. In parallel tempering we run several chains at different temperatures, moving around by Metropolis-Hastings algorithm, that are able to exchange points which allows us to move around and between the modes of $\pi(\theta)$ much more easily.

After a Metropolis-Hastings update at each temperature two chains i and j with current points $\theta^{(i)}$ and $\theta^{(j)}$ respectively are proposed and are exchanged with probability

$$\alpha(i,j) = \min\left(1, \frac{\pi(\theta^{(i)})^{\tau_j}\pi(\theta^{(j)})^{\tau_i}h(j,i)}{\pi(\theta^{(i)})^{\tau_i}\pi(\theta^{(j)})^{\tau_j}h(i,j)}\right),$$

where $h(i,j)$ is the probability of selecting chains i and j to be exchanged. This is summed up in Algorithm 4.

Algorithm 4 Parallel tempering algorithm

```
Set a starting value $[\theta_0^k]_{k=1}^M$ and temperatures $[\tau_k]_{k=1}^M$
for $t = 1 : N$ do
    for $k = 1 : M$ do
        $\theta_t^k \leftarrow$ One update of the Metropolis-Hastings algorithm with distribution $\pi_k(\theta)$ and proposal $q(\cdot|\theta_{t-1}^k, \sigma_k)$
    end for
    $i, j \sim h(\cdot,\cdot)$ {Sample $i$ and $j$ from the distribution $h(i,j)$.}
    $g \leftarrow \min\left(1, \frac{\pi(\theta^{(i)})^{\tau_j}\pi(\theta^{(j)})^{\tau_i}h(j,i)}{\pi(\theta^{(i)})^{\tau_i}\pi(\theta^{(j)})^{\tau_j}h(i,j)}\right)$
    Sample $u \sim U(\cdot|0,1)$
    if $u < g$ then
        $\phi \leftarrow \theta_t^i$
        $\theta_t^i \leftarrow \theta_t^j$
        $\theta_t^j \leftarrow \phi$
    end if
end for
```

Parallel tempering defines an ergodic Markov chain which therefore has a stationary distribution and if we marginalise the stationary distribution at temperature i we have a sample from $\pi_i(\theta)$. Thus if we run a parallel tempering algorithm with M temperatures, with $\tau_i = 1$ for some i, then we can use the values of just the ith chain as a sample from $\pi(\theta)$, the original target distribution.

3.2.3 Doubly parallel tempering

We are going to combine the parallel tempering algorithm and the parallel MCMC algorithm of Cui et al (2011) in order to try and maximise the efficiency of the parallel tempering algorithm when the likelihood is expensive to calculate. The parallel tempering algorithm remains mathematically sound if the chains move at different speeds. If we assume that at a given time step t each chain is in its stationary distribution $\pi(\theta)^{\tau_i}$, then, if we only allow Metropolis-Hastings updates, at time $t+s$ the chain will also be in its stationary distribution. However if we naively use the method suggested by Cui et al and run each chain for a maximum of χ times steps if all of the proposals are rejected or until the first of the χ proposals is accepted and then proposed an exchange move the method would no longer satisfy the Markov property and stationary distribution of chain i will no longer be $\pi(\theta)^{\tau_i}$. This is because there would be more chance of an exchange after a Metropolis-Hastings move has been accepted than after a rejection. So to be able to use this method we need to move each chain s_i steps before an exchange occurs in chain i where s_i does not depend on anything in the history of the chain.

In order to perform an exchange move, all of the chains involved in the swap need to be in their respective stationary distributions. Having said this, when the exchange occurs and between which chains, doesn't have to be simulated just before the proposed exchange occurs but can be done in advance as the proposed exchanges are completely independent of chain locations and other exchanges. In fact, as with standard parallel tempering, it is possible to completely simulate the proposed exchanges for the whole algorithm before the algorithm is even run. This means that if a chain is going to be proposed in an exchange in s time steps, then we can use Cui et al's method to move this chain forward s steps.

Suppose we have M chains each with the time to exchange being s_i for $i = 1 \dots M$ and we only have the computational power to calculate χ likelihoods in parallel, then we need to find a way of dividing the power up so that we run the algorithm efficiently. We suggest that the amount of power assigned be inversely proportional to the acceptance rate of the recent history of the chain. In this step α is varied so that $\sum_{i=1}^{M} P_i = \chi$. The reason for dividing the computational effort in this way is so that for a given chain, once a point has been accepted, any subsequent simulation is wasted. If we assume that the local acceptances for that chain hold the information of whether or not a point will be accepted we want to simulate more from the points that may be rejected. This can be updated as the algorithm continues as it doesn't affect the Markovian assumption of the algorithm.

Algorithm 5 One iteration of the doubly parallel tempering with χ available cores.

```
Given current values $\left[\theta^k\right]_{k=1}^M$, temperatures $[\tau_k]_{k=1}^M$, exchange rates $[s_k]_{k=1}^M$ and the acceptance rates, $a_k$, over the last $n$ Metropolis Hastings updates.
if $\sum_{k=1}^M s_k \leq \chi$ then
    $P_k \leftarrow s_k \forall k$
else
    for $k = 1 \dots M$ do
        $\chi_1 \leftarrow \sum_{k=1}^M \mathbb{I}(s_k \geq 1)$
        $P_k \leftarrow$
            $$\mathbb{I}(s_k \geq 1) + \left\lfloor \left( \chi_1 \times \frac{\frac{1}{a_k}}{\sum_{i=1}^M \frac{1}{a_k}} + \alpha \right) \right\rfloor$$
        {$\alpha$ is adjusted so that $\sum_{k=1}^M P_k = \chi$.}
    end for
end if
for $k = 1 \dots M$ do
    Generate $P_k$ candidate points $\theta'_{1:P_k} \sim q_k(\cdot|\theta^k)$
    $j \leftarrow 0$
    $t \leftarrow 1$
    while $j < P_k$ do
        $j \leftarrow j + 1$
        $\alpha' \leftarrow \min\left(1, \frac{\pi(\theta'_j)^{\tau_k} q(\theta^k|\theta'_j)}{\pi(\theta^k)^{\tau_k} q(\theta'_j|\theta^k)}\right)$
        Sample $u \sim \mathrm{U}(0, 1)$
        if $u < \alpha'$ then
            $\theta^k \leftarrow \theta'_j$
            $j \leftarrow P_k$
        else
            $t \leftarrow t + 1$
        end if
    end while
    Update $a_k$ so that it is the proportion of the last $n$ proposed Metropolis Hastings moves accepted moves for chain $k$.
    $s_k \leftarrow s_k - t$
end for
```

```
while $\sum_{k=1}^{M} \mathbb{I}_{s_k=0} > 1$ do
    $i, j \sim h(\cdot,\cdot)$ {Sample $i$ and $j$ from the distribution $h(i,j)$ where $s_i = s_j = 0$.}
    $g \leftarrow \min\left(1, \frac{\pi(\theta^{(i)})^{\tau_j}\pi(\theta^{(j)})^{\tau_i}h(j,i)}{\pi(\theta^{(i)})^{\tau_i}\pi(\theta^{(j)})^{\tau_j}h(i,j)}\right)$
    Sample $u \sim U(\cdot|0,1)$
    if $u < g$ then
        $\phi \leftarrow \theta^i$
        $\theta^i \leftarrow \theta^j$
        $\theta^j \leftarrow \phi$
    end if
end while
```

Combining these ideas with Algorithm 4 gives Algorithm 5. In addition to this, the times between exchanges will also be different. Chains that move faster will require more clock time spent on evaluating the likelihoods so we suggest making these chains slower. i.e. fewer moves between exchanges. Although it may be appealing to update these exchange rates whilst the algorithm is running, it cannot be done in practise as this would violate the Markovian assumption of the algorithm.

3.3 Approximate Bayesian Computation

Sometimes we are unable to sample from the posterior distribution because the likelihood is intractable but we are able to run the model at any parameter set with relatively small computational expense. Rubin (1984) says that one way of sampling from the posterior distribution $p(\theta|\boldsymbol{x})$ is to sample values $\theta_1, \ldots, \theta_s$ from the prior, $p(\theta)$, and then define pseudo-observations

$$\boldsymbol{y}_j \sim M(\cdot|\theta_j)$$

where $M(\cdot|\theta_j)$ is the model run at parameter values θ_j. For the values of $[\boldsymbol{y}_j]_{j=1}^{s}$ that equal $\boldsymbol{x}$, the θ values that correspond to these values will be a sample from the posterior distribution. This method later became known as Approximate Bayesian Computation although in principle, this version involves no approximation (Tavaré et al, 1997; Beaumont, 2010) and is described in Algorithm 6.

This is valid in theory but unfortunately Algorithm 6 rarely works in practice. Often the model output and the data are both continuous (or measured to many decimal places) meaning that $Pr(\boldsymbol{y} = \boldsymbol{x}) = 0$. This means that the algorithm would be very inefficient, if useful at all, so an approximation is taken. If $\rho(\boldsymbol{y}, \boldsymbol{x}) < \epsilon$ then θ is accepted where $\rho(\cdot)$ is a

Algorithm 6 Simple rejection ABC method

```
while $i < N$ do
    $\theta' \sim p(\cdot)$
    $\boldsymbol{y} \sim M(\cdot|\theta')$
    if $\boldsymbol{y} = \boldsymbol{x}$ then
        $\theta_i \leftarrow \theta'$
        $i \leftarrow i + 1$
    end if
end while
```

measure of the distance between $\boldsymbol{x}$ and $\boldsymbol{y}$. This means that if the actual data and the simulated data are within some tolerance threshold ϵ then the point will be accepted.

$\boldsymbol{x}$ may also be high dimensional which again means that recreating the data is difficult and sometimes impossible. So the dimension of $\boldsymbol{x}$ can be reduced by using summary statistics $S(\boldsymbol{x})$. This method was first used in population genetics (Pritchard et al, 1999) and is described in Algorithm 7.

Algorithm 7 Rejection ABC method with summary statistics

```
while $i < N$ do
    $\theta' \sim p(\cdot)$
    $\boldsymbol{y} \sim M(\cdot|\theta')$
    if $\rho(S(\boldsymbol{y}), S(\boldsymbol{x})) < \epsilon$ then
        $\theta_i \leftarrow \theta'$
        $i \leftarrow i + 1$
    end if
end while
```

Using some ϵ means that the inference is an approximation rather than exact. Instead of sampling from the target distribution

$$\pi(\theta|\boldsymbol{x}) \propto p(\theta)p(\boldsymbol{x}|\theta),$$

we are sampling from

$$\begin{aligned} \pi_{ABC}(\theta|\boldsymbol{x}) &\propto p(\theta)p(R|\theta) \\ &\propto p(\theta|R) \end{aligned} \tag{3.1}$$

where $R = \{\boldsymbol{z} : \rho(S(\boldsymbol{z}), S(\boldsymbol{x})) < \epsilon\}$. If $\epsilon = 0$ the target distribution becomes

$$\begin{aligned} \pi_{ABC}(\theta|\boldsymbol{x}) &\propto p(\theta)p(S(\boldsymbol{x})|\theta) \\ &\propto p(\theta|S(\boldsymbol{x})) \end{aligned}$$

and if the summary statistics are sufficient then

$$\begin{aligned}\pi_{ABC}(\theta|\boldsymbol{x}) &\propto p(\theta)p(\boldsymbol{x}|\theta) \\ &\propto p(\theta|\boldsymbol{x})\end{aligned}$$

and therefore we would be sampling from the target distribution. The basic idea behind ABC is that using a representative summary statistic coupled with a small enough tolerance should produce a good approximation to the posterior distribution (Marin et al, 2012).

3.3.1 Tolerance level

The tolerance should be set as small as possible in order to give reasonable results but not so small that nothing is accepted. Fearnhead and Prangle (2012) and Blum (2010) showed that optimal values of ϵ that depend on the true likelihood. As these are not computable other methods of setting the tolerance need to be used. There are a number of ways to remove the approximation and make the simulation exact, some of which are described below.

Adjustment

Beaumont et al (2002) proposed a method of improving the approximation by using a regression method that shifts the accepted points so that their pseudo-observations $S(\boldsymbol{y})$ in R are mapped to $S(\boldsymbol{x})$

Each proposed θ is given a weight (rejected values are given a weight of 0) depending how far away $S(\boldsymbol{y})$ is from $S(\boldsymbol{x})$. The authors suggest using an Epanechnikov kernel

$$K(t) = \begin{cases} c\epsilon^{-1}(1 - t/\epsilon^2)) & t \leq \epsilon, \\ 0 & t > \epsilon, \end{cases}$$

where c is a normalizing constant and ϵ is chosen so that approximately 1% of the proposed θ values are accepted. The weight is $K(||S(\boldsymbol{x}))-S(\boldsymbol{y}))||)$ where $||S(\boldsymbol{x})) - S(\boldsymbol{y}))||$ is the distance between $S(\boldsymbol{x})$ and $S(\boldsymbol{y})$. The regression parameters $\hat{\boldsymbol{\beta}}$ are calculated from

$$\hat{\boldsymbol{\beta}} = (X'WX)^{-1}X'W\boldsymbol{\theta}$$

where

$$X = \begin{pmatrix} 1 & S_1(\boldsymbol{x}) - S_1(\boldsymbol{y}_1) & \cdots & S_n(\boldsymbol{x}) - S_n(\boldsymbol{y}_1) \\ 1 & S_1(\boldsymbol{x}) - S_1(\boldsymbol{y}_2) & \cdots & S_n(\boldsymbol{x}) - S_n(\boldsymbol{y}_2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & S_1(\boldsymbol{x}) - S_1(\boldsymbol{y}_m) & \cdots & S_n(\boldsymbol{x}) - S_n(\boldsymbol{y}_m) \end{pmatrix} \quad \text{and} \quad \boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \end{pmatrix}$$

for n summary statistics and m proposed θ values and where W is a diagonal matrix

$$W = \begin{pmatrix} K(||S(\boldsymbol{x}) - S(\boldsymbol{y}_1)||) & 0 & \cdots & 0 \\ 0 & K(||S(\boldsymbol{x}) - S(\boldsymbol{y}_2)||) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K(||S(\boldsymbol{x}) - S(\boldsymbol{y}_m)||) \end{pmatrix},$$

$\boldsymbol{\beta}$ is a $m+1$ column vector and the regression model

$$\boldsymbol{\theta} = X\hat{\boldsymbol{\beta}}.$$

The first element of $\hat{\boldsymbol{\beta}}$ is a constant, however every ith subsequent element relates to the ith summary statistic. Each of the N accepted θ_js are then adjusted for each summary statistic

$$\theta_j^* = \theta_j + \hat{\beta}_{i+1}(S_i(\boldsymbol{x}) - S_i(\boldsymbol{y})) \quad \text{for } i = 1...n, \quad j = 1...m$$

and therefore θ^* are sampled from a better approximation than rejection-ABC.

Beaumont et al (2002) assume that, around the region in the parameter space where the model is accepted, the relationship between $\boldsymbol{\theta}$ and $S_i(\boldsymbol{x}) - S_i(\boldsymbol{y})$, for $i = 1 \ldots n$, can be represented using a linear model and therefore are able, with a small enough tolerance, to move the points as described above. Blum and François (2010) extended this by using non-linear models to adjust the accepted points.

Kernel ABC

A more general way of looking at the accept/reject part of the algorithm is to accept or weight points according to a kernel $K(\cdot)$. If $K(\cdot)$ is a probability density then the target distribution is

$$\pi_{Kernel-ABC}(\theta|S(\boldsymbol{x})) \propto p(\theta) \int_D K\left(\frac{S(\boldsymbol{x}) - S(\boldsymbol{z})}{\epsilon}\right) p(\boldsymbol{z}|\theta) d\boldsymbol{z}. \tag{3.2}$$

This is the posterior distribution of $p(\theta|Q)$ where Q is the perturbation of $S(\boldsymbol{x})$ with distribution $K(\cdot)$ scaled by ϵ or more formally

$$Q = S(\boldsymbol{x}) + \epsilon \boldsymbol{z},$$

where $\boldsymbol{z} \sim K(\cdot)$. A rejection algorithm using Kernel ABC is presented in Algorithm 8.

If we can say that

$$\boldsymbol{x} = M(\cdot) + \zeta,$$

where M can be learned exactly by simulation, and ζ is an error term with known distribution, $K_\zeta(\cdot)$, that does not depend on θ and is independent of

Algorithm 8 The Kernel-ABC algorithm. c is set to $\sup\left(K\left(\frac{S(\boldsymbol{x})-S(\boldsymbol{y})}{\epsilon}\right)\right)$.

```
while $i < N$ do
    $\theta' \sim p(\cdot)$
    $\boldsymbol{y} \sim M(\cdot|\theta')$
    $u \sim U(0,1)$
    if $u < \frac{K\left(\frac{S(\boldsymbol{x})-S(\boldsymbol{y})}{\epsilon}\right)}{c}$ then
        $\theta_i \leftarrow \theta'$
        $i \leftarrow i+1$
    end if
end while
```

M and if we sample the ABC posterior using Algorithm 8, with line 5 being replaced by

$$\textbf{if } u < \frac{K_\zeta\left(\boldsymbol{x}-\boldsymbol{y}\right)}{c} \textbf{ then},$$

then we would be sampling from the exact posterior distribution (Wilkinson, 2013).

A similar idea is that of Noisy ABC (Fearnhead and Prangle, 2012). If we define

$$S_{noisy} = S(\boldsymbol{x}) + \epsilon \boldsymbol{z},$$

where $\boldsymbol{z} \sim K(\cdot)$, and used Algorithm 8, with line 5 being replaced by

$$\textbf{if } u < \frac{K\left(\frac{S_{noisy}-S(\boldsymbol{y})}{\epsilon}\right)}{c} \textbf{ then},$$

to sample the ABC posterior, then we would be sampling from the true posterior distribution, $p(\theta|S(\boldsymbol{x}))$ (Fearnhead and Prangle, 2012). Furthermore, as the amount of data increases, the target of the noisy ABC will converge to a point mass on the true parameter value. This is not the case with standard ABC. For more information on this topic see Fearnhead and Prangle (2012, Section 2.2). An advantage of Noisy ABC is that we can specify any $K(\cdot)$ and we are still sampling from the true posterior distribution.

3.3.2 Summary statistics

Carrying out inference based on summary statistics instead of data sets implies discarding potential useful information (Csillėry et al, 2010). $S(\cdot)$ may be a vector value; however if you increase the number of summary statistics, you can increase the amount of information available to the ABC algorithm (Sousa et al, 2009). This can reduce the efficiency of the inference as when the number of dimensions increases, the probability of accepting a simulation decreases exponentially and the Monte Carlo error of the estimation increases as the number of summary statistics increases (Fearnhead and Prangle, 2012; Beaumont et al, 2002).

In response to this Joyce and Marjoram (2008) developed a method that scores summary statistics according to whether their inclusion in the analysis improves the quality of inference. Suppose we have a list of summary statistics $S_1, S_2, \cdots, S_{k-1}$ and a candidate statistic S_k. The candidate summary statistic is scored by

$$\frac{P(\theta|S_1, S_2, \cdots, S_k)}{P(\theta|S_1, S_2, \cdots, S_{k-1})}, \tag{3.3}$$

with a summary statistic that greatly improves the quality of the inference giving a lower score. If the summary statistics $S_1, S_2, \cdots, S_{k-1}$ were sufficient, no other summary statistic could be calculated from the data that would provide any additional information about the data, then the likelihood of the summary statistics given the parameter would be

$$P(S_k|S_1, S_2, \cdots, S_{k-1}, \theta) = P(S_k|S_1, S_2, \cdots, S_{k-1})$$

which would mean that expression 3.3 would be approximately one (Joyce and Marjoram, 2008).

Fearnhead and Prangle (2012) showed that the optimal choice of summary statistics, that minimises the quadratic loss of an estimate of the parameter, is $E(\theta|\boldsymbol{x})$. That is summary statistics that are equal to the posterior mean. However we are unable to calculate the posterior means but we can use simulation to estimate these. Fearnhead and Prangle suggest

1. simulating sets of parameter values and data from the prior.

2. using the simulated sets of parameter values and data to estimate the summary statistics.

3. running ABC with this choice of summary statistics.

There could be an additional step, before step 1, that determines in which regions of the parameter space the posterior distribution has negligible mass using arbitrary summary statistics similar to History Matching (Vernon et al, 2010). This means that these regions do not need to be simulated from in steps 1 and 3.

In the first step T parameter values $[\theta_i]_{i=1}^T$ are sampled from the prior (or from the region where there is not negligible posterior mass) and the model is run at these values to generate pseudo data $[\boldsymbol{y}_i]_{i=1}^T$.

There are a number of methods of using the simulated values to estimate the parameter values, but Fearnhead and Prangle suggest building a regression model. They suggest using a function $f(\cdot)$ so that $f(\boldsymbol{y}_i)$ is a vector of, possibly non-linear, transformations of the data so that $[f(\boldsymbol{y}_i)]_{i=1}^T$ are the explanatory variables and the parameter values $[\theta_i]_{i=1}^T$ are the responses. This means that we fit a model

$$\begin{aligned} \theta &= E(\theta|\boldsymbol{y}) + \xi \\ &= \beta_0 + f(\boldsymbol{y})\boldsymbol{\beta} + \xi \end{aligned} \qquad (3.4)$$

where ξ is some zero-mean noise. The fitted function in equation 3.4 is then an estimate of $E(\theta|\boldsymbol{y})$. If more than one parameter is required to be estimated this is done independently for each of them. This step then becomes a familiar statistical problem with standard model checks being used to decide between different models. Equation 3.4 is then used as the summary statistic for step 3, with the summaries of the data being

$$\beta_0 + f(\boldsymbol{x})\boldsymbol{\beta}.$$

Fearnhead and Prangle suggest that twice as much processing time is spent on step 3 as on step 1 with step 2 having negligible processing time.

3.3.3 Other methods

In higher dimensions, or when the prior is not very informative, rejection-ABC can be very inefficient. This lead to the development of a number of different methods that are described in this subsection.

ABC-MCMC

Marjoram et al (2003) developed a method of using ABC to approximate the likelihood with an MCMC approach (Algorithm 9). In this algorithm a new candidate point, θ', is sampled from the current point, θ using $q(\cdot|\theta)$ and then the model is run with parameter θ' to generate pseudo data, $\boldsymbol{y}$. If this sampled output is close enough to the observed data, $\boldsymbol{x}$, or more formally if $\rho(S(\boldsymbol{y}), S(\boldsymbol{x}) < \epsilon)$, then the new candidate point is accepted with probability

$$\alpha(\theta', \theta) = \min\left(1, \frac{p(\theta')q(\theta|\theta')}{p(\theta)q(\theta'|\theta)}\right).$$

Although the ABC-MCMC algorithm does sample from $\pi_{ABC}(\theta|\boldsymbol{x})$, it doesn't mix very well (Sisson et al, 2007). If the chain reaches a part with a low posterior distribution then, with a poor proposal mechanism, the chain will not move and will stick in regions of the state space for long periods. To counter this the algorithm should be run for a long time.

Wegmann et al (2009) suggest treating the tolerance as a random variable with an informative exponential prior and then updating it, along with the uncertain parameters, using ABC-MCMC for N iterations. Then after the chain has finished running the n (for $n < N$) accepted points that were

Algorithm 9 ABC-MCMC

```
Set a starting value $\theta_0$
for $i = 1 : N$ do
    Generate a candidate point $\theta' \sim q(\cdot|\theta_{i-1})$
    $\boldsymbol{y} \sim M(\cdot|\theta')$
    if $\rho(S(\boldsymbol{y}), S(\boldsymbol{x})) < \epsilon$ then
        $\alpha \leftarrow \min\left(1, \frac{p(\theta')q(\theta|\theta')}{p(\theta)q(\theta'|\theta)}\right)$
        Sample $u \sim \mathrm{U}(\cdot|0, 1)$
        if $u < \alpha$ then
            $\theta_i \leftarrow \theta'$
        else
            $\theta_i \leftarrow \theta_{i-1}$
        end if
    else
        $\theta_i \leftarrow \theta_{i-1}$
    end if
end for
```

closest to the data or summaries of the data then become the sample. The other points are then discarded.

Lee et al (2012) described two variations of ABC-MCMC. The first also takes advantage of the pseudo-marginal MCMC features. Instead of the likelihood being either 1 or 0, for a given set of parameters the likelihood is estimated by calculating

$$l(\theta|\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}_{\rho(\boldsymbol{y}_i, \boldsymbol{x})<\epsilon}$$

where $\boldsymbol{y}_i$ is the pseudo data of the ith simulation of m model runs conditional on the parameters θ. This means that the estimation of the likelihood is more accurate and would improve the mixing of the MCMC but will mean that the likelihood evaluation would slow down as the model would need to be run n times. Having said this, the estimate of the likelihood could easily be parallelised.

The second variation of the ABC-MCMC algorithm is known as the one-hit MCMC-ABC. In this algorithm, both the proposed parameter value θ' and the current parameter value θ are used to run simulations. The first parameter value, θ' or θ, to produce an ABC acceptance is accepted as the next point in the MCMC. This would also improve the mixing of the MCMC but would also slow down the likelihood evaluation. Unlike in the previous algorithm the computational effort required to evaluate the likelihood is not capped and one move could take a long time even if parallelised.

ABC-PMC

Sisson et al (2007) proposed a method of sequentially improving the ABC approximation. Sisson et al suggested that rejection-ABC could be run in rounds with the samples for round t being sampled from a weighted average from the points accepted in round $t-1$ (for $t > 2$) with each round getting a smaller tolerance value ϵ (in the first round points are simulated from the prior). So that points are not repeated, once a new θ value is sampled it is perturbed according to some kernel $K_t(\cdot)$. However Sisson et al (2007) method of weighting the points gave a bias.

Beaumont et al (2009) proposed a method that corrected for the bias in Sisson et al (2007). They based their method on the Population Monte Carlo algorithm (Cappé et al, 2004). They built a kernel density estimate (Silverman, 1986) of $\pi_{t-1}(\theta)$ with the points accepted in round $t-1$ with kernel $K_t(\cdot)$ (Beaumont et al (2009) proposed that the smoothing parameter, τ^2, be twice the empirical variance) and sampled the potential points for round t from this kernel density estimate. This means the weight of an accepted point in round t became

$$\frac{\pi(\theta)}{\pi_{t-1}(\theta)}.$$

This method is summed up in Algorithm 10. Del Moral et al (2012) also developed a method of adaptively choosing the tolerance ϵ_t by taking a particle filtering (section 3.4.5) approach to updating the parameters rather than a Population Monte Carlo approach (SMC-ABC).

Coupled ABC

Neal (2012) developed a method known as Coupled-ABC that treats ABC as an inverse problem. If we suppose that we can split up the inference problem into a random vector $\boldsymbol{U}$ such that $\boldsymbol{U}$ is independent of the model parameters θ, then we can say that given a specific parameter set and a realisation of the random vector $\boldsymbol{u}$, the simulated model output is a deterministic function of both of these $h(\boldsymbol{u}, \theta)$. We then solve the inverse problem and say that the set $\Theta = \{\theta : \boldsymbol{x} = h(\boldsymbol{u}, \theta)\}$ which $p(\theta|\boldsymbol{x}, \boldsymbol{u})$, which is exactly the correct inference.

Neal and Huang (2014) state that the level of approximation of an ABC algorithm is often difficult to quantify and extend the idea of coupled ABC to an MCMC framework by treating the random vector $\boldsymbol{u}$ as the random variables used when simulating from the model and proposing new sets of $\boldsymbol{u}$ depending on the current state, essentially treating $\boldsymbol{u}$ as an additional parameter. Neal and Huang, in their forward simulation MCMC algorithm, alternate between updating the random variables, $\boldsymbol{u}$, and the parameters, θ and perform ABC in such a way that $\epsilon = 0$.

Algorithm 10 ABC-PMC algorithm

```
while $i < N$ do
  Sample $\theta'' \sim p(\theta)$
  $\boldsymbol{y} \sim M(\cdot|\theta'')$
  if $\rho(S(\boldsymbol{y}), S(\boldsymbol{x})) < \epsilon_1$ then
    $\theta_1^i \leftarrow \theta''$
    $w_1^i \leftarrow 1$
    $i \leftarrow i + 1$
  end if
end while
$\tau_1^2 \leftarrow$twice the empirical variance of $\theta_i^1$s
for $t = 2 : T$ do
  while $i < N$ do
    Sample $\theta' \sim \theta_{t-1}$ with weights $w_{t-1}^i$
    Sample $\theta'' \sim K_t(\cdot|\theta', \tau_t^2)$
    $\boldsymbol{y} \sim M(\cdot|\theta'')$
    if $\rho(S(\boldsymbol{y}), S(\boldsymbol{x})) < \epsilon_t$ then
      $\theta_t^i \leftarrow \theta''$
      $w_t^i \leftarrow \frac{p(\theta'')}{\sum_{j=1}^N w_{t-1}^j K_t(\theta''|\theta_{t-1}^j)}$
      $i \leftarrow i + 1$
    end if
  end while
  $\tau_t^2 \leftarrow$twice the empirical variance of $\theta_i^t$s
end for
```

The authors cause a simulation to become exact in two ways. For some models they create a bias that means that the model output, $\boldsymbol{y}$, equals the observed data, $\boldsymbol{x}$, exactly. By taking into account of this bias, and the probability of it occurring exactly, it is possible to generate a likelihood $l(\theta, \boldsymbol{u}|\boldsymbol{x})$ which takes pseudo marginal approach (Andrieu and Roberts, 2009) (see Section 3.2.1) to MCMC.

The second way is similar to the way described by Wilkinson (2013) (Section 3.3.1), where the model is ran and then, conditional on the model being run, the likelihood of the data being generated is calculated. This could be measurement error (Wilkinson, 2013) or sampling error when conducting a survey (Neal and Huang, 2014). Another way would be to stop a model early, for example in the woodhoopoe model described in Section 2.2 we could stop the model before the last submodel is run, the mortality, and check the likelihood of getting the observed data from that. This method also takes advantage of the pseudo marginal approach and leads to what is known as exact ABC. It is similar to Likelihood free MCMC (see Section 3.4.5).

Likelihood free parallel tempering

Baragatti et al (2013) proposed a parallel tempering (see Section 3.2.2) version of ABC. In the ABC version the different chains t have different tolerance values ϵ_t and move according to the ABC-MCMC algorithm described above. Because the likelihood for the higher ϵs will cover a larger area than those of a lower ϵ the proposal distribution could cover a higher area. So for tolerances

$$\epsilon_1 < \epsilon_2 < \cdots < \epsilon_T$$

the parameter in the proposal (eg. the variance if the proposal was a Gaussian distribution) will be

$$\sigma_1 < \sigma_2 < \cdots < \sigma_T.$$

The MCMC part of algorithm is just the same as the method described in Section 3.3.3 and then at each step in the algorithm T exchanges are proposed. Any chain can exchange current values with any other and to improve the acceptance probabilities chains are put into K "rings". "Rings" are fixed disjoint subsets of tolerance levels space $E_1, \ldots, E_K$. At each iteration, the ith chain (with sampled values $\boldsymbol{y}$) is associated with E_j if

$$\rho(S(\boldsymbol{y}_i), S(\boldsymbol{x})) \in E_j.$$

For each exchange move, a ring which has at least two chains associated with it is chosen at random and then two chains associated with it are chosen at random (i, j with $\epsilon_i < \epsilon_j$). The chosen chains are exchanged if

$$\rho(S(\boldsymbol{y}_j), S(\boldsymbol{x})) < \epsilon_i.$$

This is algorithm is summed up in Algorithm 11.

3.3.4 Discussion

ABC is rarely used to estimate parameters in higher dimensions. One of the main reasons for this is that as the number of dimensions goes up, the size of the prior also increases and therefore the size of the evidence, or $P(R)$ as in equation 3.1, gets smaller. For this reason rejection methods such as ABC-PMC and SMC-ABC may find it difficult to even locate this space let alone find a number of points in it (Lee et al, 2011).

We believe that an MCMC version of ABC is the way forward to perform ABC in higher dimensions. One of the good things about MCMC in higher dimensions, both when the likelihood is known and it is intractable, is that when we are in the region of the posterior distribution then we stay in this region. This means that it doesn't matter how many parameters there are we will always stay in the right region. Having said this there still needs to be work done on ABC-MCMC as it is poor in the tails of the distribution

Algorithm 11 Likelihood free parallel tempering

```
Set a starting value $[\theta_0^k]_{k=1}^T$ and rings $[E_k]_{k=1}^K$
for $i = 1 : N$ do
    for k=1:T do
        $\theta_i^k \leftarrow$ One run of Algorithm 9 with tolerance $\epsilon_k$ and proposal $q(\cdot|\theta_{i-1}^k, \sigma_k)$
    end for
    for $k = 1 : T$ do
        Associate chain $k$ with the correct ring $[E_j]_{j=1}^K$ such that
            $$\rho(S(\boldsymbol{y}_k), S(\boldsymbol{x})) \in E_j.$$
    end for
    for $k = 1 : T$ do
        Sample a subset $E_j$ (if it exists), with at least two chains associated with it. Then sample two chains, $l, m$ with $\epsilon_l < \epsilon_m$, associated with $E_j$.
        if $\rho(S(\boldsymbol{y}_m), S(\boldsymbol{x})) < \epsilon_l$ then
            $\phi \leftarrow \theta_l^i$
            $\theta_m^i \leftarrow \theta_l^i$
            $\theta_l^i \leftarrow \phi$
            $\phi \leftarrow \boldsymbol{y}_l^i$
            $\boldsymbol{y}_m^i \leftarrow \boldsymbol{y}_l^i$
            $\boldsymbol{y}_l^i \leftarrow \phi$
        end if
    end for
end for
```

(Sisson et al, 2007) which we also found from early experiments. In Chapter 4 we develop a version of ABC-MCMC that is similar to coupled-ABC as well as some Gibbs steps that are used in order to improve the mixing of the ABC-MCMC algorithm.

3.4 Other methods

There are many other methods that could be used to estimate the parameters. Below we describe a few of them.

3.4.1 Inverse Modelling Technique

Using the pattern orientated modelling method (Grimm et al, 1996) parameter values can be found indirectly by changing a number of parameters at once by calibration and seeing if the model output matches some observed data (Wiegand et al, 2003, 2004). The inverse modelling technique

(Grimm and Railsback, 2005) involves simulating the model over a wide range of parameter values in order to find suitable values for the parameters that reproduce the patterns observed in the ecological system. Grimm and Railsback (2012) describe the idea of calibration by first identifying the parameters that need to be estimated and identifying the the patterns that they want the model to replicate and deciding on an acceptance criteria before running the model for many parameter sets and seeing which ones pass the acceptance criteria. This seems to resemble a less formal version of ABC (Section 3.3).

3.4.2 Kernel Density Estimate of the likelihood

A popular method of estimating the likelihood when the likelihood is intractable is to build a kernel density estimate (Silverman, 1986) of the model output. That is if $y \sim M(\cdot|\theta)$, then a kernel density estimate $\tilde{g}(y|\theta)$ is built by sampling from $M(\cdot|\theta)$ and then the log-likelihood is estimated by $\sum_{i=1}^{T} \log\left(\tilde{g}(x_i|\theta)\right)$. This approach was first suggested by Diggle and Gratton (1984).

Another attempt at doing this was by Piou et al (2009). Piou et al developed an information citation based on the AIC (Akaike, 1974) and DIC (Spiegelhalter et al, 1998, 2002) where the log likelihood is estimated using a kernel density estimation. The authors then say that this method could then be used to perform parameter estimates and describe two algorithms, one resembling ABC-MCMC that has a value of their likelihood as the tolerance and another that is similar to MCMC. The idea of using kernel density estimates to estimate likelihoods was also suggested by Tian et al (2007).

This method can only really be used when the data are iid and when the data types are only in a few dimensions. This is because the number of simulations in order to accurately estimate the likelihood quickly becomes large as the number of dimensions increases. Silverman (1986) gives a table that shows the sample size required to ensure that the relative mean squared error of a kernel density estimate of a Gaussian distribution is less than 0.1 (Table 3.1).

In addition to this, kernel density estimates are biased by their nature (Silverman, 1986) and the bias is not necessarily proportional to the true likelihood hence performing MCMC with this as an estimate to the likelihood will not lead to the sample coming from the correct distribution (Andrieu and Roberts, 2009).

Martínez et al (2011) used an approximate likelihood based on an estimate of the distribution of the data. They fit a kernel density estimate, $\tilde{g}(\boldsymbol{x})$, constructed from empirical data $\boldsymbol{x}$. Given a parameter value θ, their likelihood function depends on pseudo observations simulated from the model

Dimensionality	Required sample size
1	4
2	19
3	67
4	223
5	768
6	2,790
7	10,700
8	43,700
9	187,000
10	842,000

Table 3.1: The sample size required to ensure that the kernel density estimate of a Gaussian distribution has relative mean squared error is less than 0.1.

$f(\cdot|\theta) = \boldsymbol{y}(\theta) = (y_1, y_2, \ldots, y_m)$ and is given by

$$l_M(\theta|\boldsymbol{x}, \boldsymbol{y}(\theta)) = \prod_{i=1}^{m} \tilde{g}(y_i|\boldsymbol{x}).$$

We are now going to show that this is not always valid and the the reader can skip the rest of this subsection if required. If we write Martinez et al.'s likelihood as $l_M(\theta|\boldsymbol{x}, \boldsymbol{y}(\theta))$ and the traditional likelihood as $l(\theta|\boldsymbol{x}, \boldsymbol{y}(\theta))$ we know that

$$\begin{aligned} l_M(\theta|\boldsymbol{x}, \boldsymbol{y}(\theta)) &= \prod_{i=1}^{m} \tilde{g}(y_i|\boldsymbol{x}) \\ l(\theta|\boldsymbol{x}, \boldsymbol{y}(\theta)) &= \prod_{i=1}^{n} \tilde{f}(x_i|\boldsymbol{y}(\theta)), \end{aligned}$$

where $\tilde{f}(x_i|\boldsymbol{y}(\theta))$ is a kernel density estimate build from $\boldsymbol{y}(\theta)$, and by taking logs

$$\begin{aligned} \mathcal{L}_M(\theta|\boldsymbol{x}, \boldsymbol{y}) &= \sum_{i=1}^{m} \log \tilde{g}(y_i|\boldsymbol{x}) \\ \mathcal{L}(\theta|\boldsymbol{x}, \boldsymbol{y}) &= \sum_{i=1}^{n} \log \tilde{f}(x_i|\boldsymbol{y}). \end{aligned}$$

Each value z is expected to occur $mf(z|\theta)$ times in the simulation or $ng(z)$ times in the data as the number of samples approaches infinity where $f(\cdot|\theta)$ is the true distribution of the model output and $g(\cdot)$ is the true distribution of the data. So taking the limit as $m, n \to \infty$.

$$\begin{aligned} \mathcal{L}_M(\theta) &= \int_{R_1} f(z|\theta) \log g(z) dz \qquad (3.5) \\ \mathcal{L}(\theta) &= \int_{R_2} g(z) \log f(z|\theta) dz \qquad (3.6) \end{aligned}$$

where R_1 is the region covered by $\log g(z|\boldsymbol{x})$ and R_2 is the region covered by $\log f(z|\theta)$. If equation 3.5 equals equation 3.6 then

$$l_M(\theta) = l(\theta). \qquad (3.7)$$

Example 1. *If the simulated data and the true data are both normally distributed with the same variance but different means (θ and θ^* respectively) then equation 3.7 is satisfied.*

Proof. Let θ^* be the true value of the parameter then

$$\begin{aligned} \mathcal{L}(\theta) &= \int_{R_2} g(z) \log f(z|\theta) dz \\ &= \int_{-\infty}^{\infty} N(z|\theta^*, \sigma^2) \log N(z|\theta, \sigma^2) dz \\ &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(z-\theta^*)^2\right) \\ &\qquad \times \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(z-\theta)^2\right)\right) dz \end{aligned}$$

which simplifies to be

$$\begin{aligned} \mathcal{L}(\theta) &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(z-\theta^*)^2\right) \log \frac{1}{\sqrt{2\pi\sigma^2}} dz \\ &\qquad + \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(z-\theta^*)^2\right) \left(-\frac{1}{2\sigma^2}(z-\theta)^2\right) dz \\ &= \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \\ &\qquad \int_{-\infty}^{\infty} (z-\theta)^2 \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(z-\theta^*)^2\right) dz. \qquad (3.8) \end{aligned}$$

Looking at the integral and expanding the brackets

$$\int_{-\infty}^{\infty} (z^2 - 2\theta z + \theta^2) \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(z-\theta^*)^2\right) dz$$

which becomes

$$\mathbb{E}(z^2) - 2\theta\mathbb{E}(z) + \theta^2. \qquad (3.9)$$

It is known using a simple identity that

$$\mathbb{E}(z^2) = \sigma^2 + \theta^{*2}$$

so equation 3.9 becomes

$$\sigma^2 + (\theta^* - \theta)^2$$

and plugging this into equation 3.8

$$\mathcal{L}(\theta) = \log \frac{1}{\sqrt{2\sigma^2\pi}} - \frac{\sigma^2 + (\theta^* - \theta)^2}{2\sigma^2}.$$

Similarly using the same method it can be shown that

$$\mathcal{L}_M(\theta) = \log \frac{1}{\sqrt{2\sigma^2\pi}} - \frac{\sigma^2 + (\theta - \theta^*)^2}{2\sigma^2}.$$

As $(\theta - \theta^*)^2 = (\theta^* - \theta)^2$, we can say that

$$\mathcal{L}(\theta) = \mathcal{L}_M(\theta).$$

□

Using a similar method two normal distributions with a simulated variance σ_{sim} and empirical variance σ_{obs} then the likelihoods are

$$\begin{aligned}
\mathcal{L}(\theta) &= \log \frac{1}{\sqrt{2\sigma_{sim}^2\pi}} - \frac{\sigma_{obs}^2 + (\theta^* - \theta)^2}{2\sigma_{sim}^2} \\
\mathcal{L}_M(\theta) &= \log \frac{1}{\sqrt{2\sigma_{obs}^2\pi}} - \frac{\sigma_{sim}^2 + (\theta - \theta^*)^2}{2\sigma_{obs}^2}
\end{aligned}$$

and hence $\mathcal{L}(\theta) \neq \mathcal{L}(\theta)$.

Example 2. *If both the simulated and real distributions are distributed exponentially with parameter λ and λ^* respectively then $\mathcal{L}_M(\lambda) \neq \mathcal{L}(\lambda)$.*

Proof. From equation 3.5

$$\begin{aligned}
\mathcal{L}_M(\lambda) &= \int_{R_1} f(z|\lambda) \log g(z) dz \\
&= \int_0^\infty \lambda \exp(-\lambda z) \log\left(\lambda^* \exp(-\lambda^* z)\right) dz \\
&= \int_0^\infty \lambda \exp(-\lambda z) \log \lambda^* dz - \int_0^\infty \lambda^* z \lambda \exp(-\lambda z) dz \\
&= \log \lambda^* - \lambda^* \mathbb{E}(z) \\
&= \log \lambda^* - \frac{\lambda^*}{\lambda}. \qquad (3.10)
\end{aligned}$$

Using a similar method

$$\mathcal{L}(\lambda) = \log \lambda - \frac{\lambda}{\lambda^*} \tag{3.11}$$

and therefore $\mathcal{L}_M(\lambda) \neq \mathcal{L}(\lambda)$ □

Another way of looking at example 2 is by working out the MLEs of both equation 3.10 and 3.11. The MLE of the true likelihood is λ^*. Differentiating equation 3.10 and 3.11

$$\begin{aligned} \frac{d(\mathcal{L}_M(\lambda))}{d\lambda} &= \frac{\lambda^*}{\lambda^2} \\ \frac{d(\mathcal{L}(\lambda))}{d\lambda} &= \frac{1}{\lambda} - \frac{1}{\lambda^*}; \quad \hat{\lambda} = \lambda^* \end{aligned}$$

respectively we can see that equation 3.10 does not have an MLE. This means that this method does not always give the correct likelihood.

3.4.3 Emulation

If the simulation model, $M(\cdot)$, is relatively smooth, meaning that $M(\theta_0)$ is similar to $M(\theta_0+\delta)$ for small δ, then it is possible to build a Gaussian process emulator (Kennedy and O'Hagan, 2001) in order to model the simulation model. The Gaussian process emulator is a stochastic representation of $M(\cdot)$ (Vernon et al, 2010; Oakley and O'Hagan, 2002; O'Hagan, 2006) and is often used when the model is computationally expensive to run.

A Gaussian process is fitted to some model evaluations $\boldsymbol{y}$ at some training parameters $\boldsymbol{\phi}$ and then for any point in parameter space the model output is

$$M(\cdot)|\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{\phi} \sim GP(\cdot|\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{\phi}).$$

Effectively a regression, with parameters $\boldsymbol{\beta}$, is fitted to the training parameters and then the Gaussian process just smoothes out the residuals (O'Hagan, 2006). The idea is that the Gaussian process is quick to evaluate and therefore can evaluate $M(\cdot)$ at any input and give an output with a measurable degree of uncertainty. Gaussian process emulators have been used on many different models from many different disciplines including climate science (Holden et al, 2010), cosmology (Vernon et al, 2010) and medical science (Strong, 2011). For a more thorough description of Gaussian Process emulation see Wilkinson (2010b).

3.4.4 Synthetic likelihood

Wood (2010) described a method of taking summary statistics, $\boldsymbol{s(y)}$, of $M(\cdot)$ that are asymptotically normally distributed such that

$$\boldsymbol{s(y)} \sim \mathcal{N}(\cdot|\boldsymbol{\mu_\theta}, \Sigma_\theta).$$

This means that if we know $\boldsymbol{\mu_\theta}$ and $\Sigma_{\boldsymbol{\theta}}$ then

$$l(\theta|\boldsymbol{s(x)}) = \mathcal{N}(\boldsymbol{s(x)}|\boldsymbol{\mu_\theta}, \Sigma_\theta)$$

which is the multivariate normal distribution evaluated at $\boldsymbol{s(x)}$. Having said this, we generally do not know $\boldsymbol{\mu_\theta}$ and $\Sigma_{\boldsymbol{\theta}}$ and therefore have to estimate them. Wood suggests repeatedly sampling from $M(\theta)$ in order to estimate these values.

The summary statistics are not usually sufficient but this is not always the aim of the inference. Often the aim of the inference is to find parameter values that recreate features of the data. This could be very useful for noisy models and individual based models where we are often only interested in patterns. Hartig et al (2011) used this approach to perform parameter estimation of an individual based tree model. Wilkinson (2014) built a Gaussian Process emulator for the synthetic likelihood where, rather than run the model every time to estimate parameters, all of the computational effort is used to build the emulator and this saves many of the runs.

3.4.5 Particle MCMC

Suppose we have a dynamic state space model with, at time t, a hidden state y_t defined by

$$Y_t \sim g(\cdot|y_{t-1}, \theta)$$

and an observed state X_t defined by

$$X_t \sim f(\cdot|y_t, \theta)$$

for $t = 1 \dots T$. Suppose also that we have observations $x_{1:T}$ and we are interested in the distribution of the latent states $p(y_{1:T}|x_{1:T})$. If the model is linear and the errors are Gaussian it is possible to use the Kalman filter (Kalman, 1960) in order to find this exactly. However for nonlinear and non-Gaussian models another approach needs to be made. One possible approach is Sequential Monte Carlo or Particle Filters (Gordon et al, 1993).

Bootstrap particle filtering is performed by first sampling n particles from the initial distribution $p(y_0)$ and setting each particle weight $w_1^i = 1/n$. If at time $t-1$ we have $\{y_{t-1}^i, w_{t-1}^i\}_{i=1}^n$, which is an estimate of $p(y_{1:t-1}|x_{1:t-1})$, then we can sample n particles $y_{t-1}^{'i}$ from y_{t-1}^i with probability proportional to w_{t-1}^i and simulate them forward so that $y_t^i \sim g(\cdot|y_{t-1}^{'i}, \theta)$. We then adjust the weight

$$w_t^i = f(x_t|y_t^i, \theta)$$

which means that $\{y_t^i, w_t^i\}_{i=1}^n$ is an estimate of $p(y_{1:t}|x_{1:t})$. Furthermore $\tilde{p}(x_t|x_{1:t-1}) = \sum_{i=1}^n w_i$ is an estimate of $p(x_t|x_{1:t-1}, \theta)$. This is just the description of one type of particle filter; for a more complete description of particle filters see Doucet and Johansen (2011).

Del Moral (2004) found that

$$\tilde{p}(x_{1:T}|\theta) = \prod_{t=1}^{T} \tilde{p}(x_t|x_{1:t-1}),$$

estimated from the particle filter, to be an unbiassed estimate of $l(\theta|x_{1:T})$ and this, coupled with the pseudo marginal MCMC algorithm described in Section 3.2, allows us to create a Markov Chain that samples from $p(\theta|x_{1:T})$ known as the Particle Independent Metropolis-Hastings (PIMH) algorithm (Andrieu et al, 2010). This is just Algorithm 3 with Z being equal to $\tilde{p}(x_{1:T}|\theta)$. Andrieu et al also developed the Particle Marginal Metropolis-Hastings (PMMH) algorithm and the particle Gibbs algorithms that also enabled us to sample from $p(\theta, y_{1:T}|x_{1:T})$. The PMMH algorithm is described in Algorithm 12.

Algorithm 12 Particle marginal Metropolis-Hastings

```
1: Propose a move to $\theta'$
2: $Z \leftarrow \tilde{p}(\boldsymbol{x}_{1:T}|\theta')$ {The estimate of the likelihood from the particle filter.}
3: $\boldsymbol{y}'_{1:T} \sim \tilde{p}(\cdot|\theta', \boldsymbol{x})$ {Sample one of the particle paths.}
4: Accept $\theta'$, $Z$ and $\boldsymbol{y}'_{1:T}$ with probability
       $\min\left\{1, \frac{q(\theta|\theta')p(\theta')Z'}{q(\theta'|\theta)p(\theta)Z}\right\}$
```

A similar algorithm is the likelihood-free MCMC (LF-MCMC) (Wilkinson, 2010a). In this algorithm $g(\cdot|y_{t-1}, \theta)$ is intractable but we are able to simulate from it and $f(x_t|y_t, \theta)$ is quite simple to calculate, then if the whole model is simulated forward so that $\boldsymbol{y} = y_{1:T}$ then it is possible to write the likelihood as

$$l(\theta, \boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{T} f(x_i|y_i).$$

This algorithm simulates the joint distribution $p(\theta, \boldsymbol{y}|x_{1:T})$ which when marginalised is $p(\theta|x_{1:T})$. LF-MCMC is the same as the PMMH algorithm with just one particle in the particle filter.

We used PMCMC with the woodhoopoe model and tried to estimate the parameters from that; details are omitted. In order to do this we ran the model up until the last submodel which is the death of the woodhoopoes and used this as an estimate of the likelihood. We found that this didn't really work because the stochastic parts are too small in the woodhoopoe submodel which meant that in order for a particle to get a non-zero weight it has to almost recreate the data. Having said this we believe that PMCMC is still a good way to estimate parameters in some complex ecological models, where the final submodel doesn't have this property.

Chapter 4

Coupling random inputs in order to perform parameter estimation

4.1 Introduction

Approximate Bayesian Computation (ABC, Beaumont, 2010; Tavaré et al, 1997, Section 3.3) has become a common method of performing inference on complex models when the likelihood is intractable and when simulating from the model is not computationally expensive. In higher dimensions, or when the prior is not very informative, this ABC-rejection algorithm (Algorithm 7 of Section 3.3) can be very inefficient. This has led to the development more efficient algorithms: ABC-MCMC (Marjoram et al, 2003), ABC-PMC (Beaumont et al, 2009) and likelihood free parallel tempering (Baragatti et al, 2013) just to name a few, which are described in Section 3.3.3. ABC-MCMC (Marjoram et al, 2003) is an ABC version of the Metropolis-Hastings algorithm. Sisson et al (2007) criticised the ABC-MCMC algorithm as the chain often gets stuck in the tails of the posterior distribution causing the Markov chain to mix poorly.

Neal (2012) developed a method of performing ABC by non-centered parameterisation where the observed data can be given by some function of this non-centered parameterisation and then it is possible to find an inverse of this function conditional on the random inputs. Andrieu et al (2012) also suggested using the random inputs with importance sampling in an ABC framework.

If we look at the ABC-MCMC, when the chain is in the tails of the distribution, for a point to be accepted the correct parameters and the correct random inputs, the stochastic elements of the complex model, need to be proposed so that the move will be accepted which has a small probability by the very nature of being in the tails of the distribution. By controlling these random inputs and moving around more strategically in the augmented space we could increase the chances of a move and get the Markov Chain to mix better.

In this chapter we will present a method of exploring the ABC posterior by exploring the joint parameter and random input space. We use information about the joint space in order to propose better moves as well as introducing a Gibbs step that conditions on the currently accepted model in order to move around. Neal and Huang (2014) took a similar approach to perform parameter on stochastic epidemic models by iteratively updating subsets of random inputs and then parameters. Our method improves the mixing of the ABC-MCMC algorithm and reduces the Monte Carlo variance of the estimate of the ABC posterior distribution especially in the tails of the distribution.

We will introduce what we mean by coupling the random inputs in Section 4.2, describe the inference problem in Section 4.3 before extending ABC-MCMC (Section 3.3.3) by coupling the random inputs and introducing Gibbs. We call this algorithm Coupled Gibbs ABC (CG-ABC). We will give a few of examples of CG-ABC in Section 4.4 and then conclude with a discussion.

4.2 Coupling the random inputs

Let $(\Omega, \mathcal{F}, P)$ be a probability triple and $X : \Omega \to \mathbb{R}$ be a random variable that we want to simulate. Write $F(\cdot)$ as the usual cumulative distribution function that is $F : \mathbb{R} \to [0, 1]$ defined by

$$F(x) = Pr(X \leq x) = P(X^{-1}((-\infty, x])).$$

If X is continuous then $F(\cdot)$ has a well defined inverse $F^{-1}(\cdot)$, and if $U \sim U[0, 1]$ then $F^{-1}(U)$ has the same distribution as X. If X is discrete, taking ordered values x_j with probabilities p_j, that is

$$\begin{aligned} w_j &= \{w : X(w) = x_j\} \\ P(W_j) &= p_j, \end{aligned}$$

then $F(\cdot)$ is not injective, and so not invertible but we can define $F^{-1}(\cdot)$ by

$$F^{-1}(u) = \inf_k \left\{ x_k : \sum_{j \leq k} p_j \geq u \right\}$$

and still have the property that $F^{-1}(U)$ has the same distribution as X. More generally, the property holds if we define

$$F^{-1}(u) = \inf_x \{x : F(x) \geq u\}.$$

Proposition 1. *Any set of random variables can be sampled from a set of independent random variables.*

Proof. Suppose we are trying to simulate

$$X_1, \dots, X_n \sim p(X_1, \dots, X_n).$$

Now if $X_1, \dots, X_n$ are independent this is satisfied trivially however if they are dependent then we can expand the distribution such that

$$p(X_1, \dots, X_n) = p_1(X_1)p_2(X_2|X_1) \dots p_n(X_n|X_{n-1}, \dots, X_1)$$

each of which is a univariate distribution. We can sample from each of the n univariate distributions, p_i, using Inverse Transform Sampling from U_i, for $i = 1, \dots, n$. The U_is will then be independent and identically distributed such that

$$U_i \sim U(\cdot|0, 1).$$

□

For example if we take a bivariate normal such that

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim N\left(\cdot \middle| \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \rho \\ \rho & \sigma_2^2 \end{pmatrix}\right)$$

then

$$x_1 \sim N(\cdot|\mu_1, \sigma_1^2)$$

or in terms of the random uniform u_1:

$$x_1 = \mu_1 + \phi^{-1}(u_1)\sigma_1^2.$$

Now we can condition on this and say that

$$x_2|x_1 \sim N\left(\cdot \middle| \mu_2 + \frac{\sigma_2}{\sigma_1}\rho(x_1 - \mu_1), (1 - \rho^2)\sigma_2^2\right)$$

or in terms of a random uniform u_2:

$$x_2 = \mu_2 + \frac{\sigma_2}{\sigma_1}\rho(x_1 - \mu_1) + \phi^{-1}(u_2)(1 - \rho^2)\sigma_2^2$$

hence the dependent random variables, x_1 and x_2 have been sampled from two independent random uniform distributions u_1 and u_2.

So therefore every random variable X_i used in the model $M(\cdot|\theta)$ can be mapped to a point in $[0, 1]^d$ space and recreated using Inverse Transform Sampling. Let the point in $[0, 1]^d$ space be defined by $\boldsymbol{u}$, then we can write as

$$Y = M(\theta, \boldsymbol{u})$$

as the model is now deterministic.

4.2.1 Random inputs

By coupling, we mean linking together realisations of stochastic models by controlling their random inputs. For example, we can ensure that small changes in the parameters result in small changes in the model output by using identical values of the random inputs. Using this idea, we are going to couple successive realisations of a stochastic model within an ABC-MCMC algorithm in order to improve its performance. However, depending on how the inputs are used, a change in a parameter may cause a submodel to require a different number of random inputs from what it required before which could result in a large change in the output.

For example let parameter θ control the number of births at a given time, N_t, ψ control the weights of these births, and X_{tk} and $u_1, u_2, \ldots$ be the random inputs. The two simple stochastic submodels

$$\begin{aligned} N_t &\sim B(\cdot|\theta) \\ X_{tk} &\sim W(\cdot|\psi) \end{aligned}$$

can be written as deterministic functions

$$\begin{aligned} N_t &= B(\theta, u_i) \\ X_{tk} &= W(\psi, u_j) \end{aligned}$$

where the values of the indices i and j will depend on the order of calculations. Suppose that the process is simulated in the obvious order and that for a particular parameter value, $\theta = \theta_0$ say, we obtain a single birth at time 1,

$$N_1 = B(\theta_0, U_1) = 1.$$

The weight of the individual born will be given by

$$X_{11} = W(\psi, U_2),$$

and the number of births the next year will be

$$N_2 = B(\theta_0, U_3).$$

Now suppose that we use the same sequence of random numbers, but change the parameter slightly to $\theta = \theta_0 + \epsilon$, giving

$$N_1 = B(\theta_0 + \epsilon, U_1) = 2$$

say. The weights of these two births will be given by

$$\begin{aligned} X_{11} &= W(\psi, U_2) \\ X_{12} &= W(\psi, U_3), \end{aligned}$$

and the number of births given in the second year will be given by

$$N_1 = B(\theta_0 + \epsilon, U_4).$$

Thus the value of N_2 may change completely, since it is based on a different random input, U_4 instead of U_3. The same is likely to apply to later numbers of births and weights, with widespread 'relabelling' of the random inputs.

One way to get round this is to control all of the inputs individually so that each process will have its own inputs and then the inputs for each submodel will remain the same regardless of what happened in earlier submodels. In the example above this could be solved by having two sets of random inputs and the deterministic functions being

$$\begin{aligned} N_t &= B(\theta, u_t) \\ X_{tk} &= W(\psi, w_{tk}). \end{aligned}$$

In the next few sections we will give some examples of how a few different models can be coupled.

Mixed normals

Sisson et al. (Sisson et al, 2007) describe the toy model

$$X|\theta \sim \begin{cases} N(\cdot|\theta, 1) & \text{with probability } \frac{1}{2} \\ N\left(\cdot|\theta, \frac{1}{100}\right) & \text{with probability } \frac{1}{2}. \end{cases}$$

This model has two random inputs u and w with

$$u, w \sim U(\cdot|0, 1).$$

Let

$$Z = \Phi^{-1}(w)$$

where Φ^{-1} is the inverse cumulative distribution of a standard normal, then the model output will be

$$X|\theta, u, Z \sim \begin{cases} \theta + Z & \text{if } u < \frac{1}{2} \\ \theta + \frac{Z}{10} & \text{otherwise.} \end{cases}$$

Hence, given u and w, the model is deterministic.

Ricker Model

The Ricker Model (Ricker, 1954) is used to describe the expected number of individuals in a generation t conditional on the previous generation $t-1$. Wood (2010) then used a stochastic version of this model to create a hidden Markov model with observations

$$Y_t \sim Pois(\lambda N_t)$$

and a continuous hidden state

$$N_t = rN_{t-1} \exp\left(-N_{t-1} + e_t\right)$$

where

$$e_t \sim N(\cdot|0, \sigma_e^2).$$

The model has random inputs $\boldsymbol{u}$ and $\boldsymbol{w}$ where

$$u_t, w_t \sim U(\cdot|0, 1)$$

for $t = 1, \ldots, T$. If we let

$$Z_t = \Phi^{-1}(w_t)$$

where Φ^{-1} is the inverse cumulative distribution of a standard normal, then

$$Y_t = \min\left\{ n : e^{-\lambda N_t} \sum_{i=0}^{n} \frac{(\lambda N_t)^i}{i!} \geq u_t, n \in \mathbb{N}_0 \right\}$$

where

$$N_t = rN_{t-1} \exp\left(-N_{t-1} + e_t\right)$$

and

$$e_t = Z_t \sigma_e.$$

Hence, given $\boldsymbol{u}$ and $\boldsymbol{w}$, the model is deterministic.

Queuing model

Fearnhead and Prangle (2012), Blum and François (2010) and Heggland and Frigessi (2002) looked at performing inference on a M/G/1 queuing model. In this model the customers from a Poisson (θ_3) process i.e. they arrive at intervals given by an exponential distribution with parameter θ_3 and are served one at a time with the time taken following a uniform distribution on the interval $[\theta_1, \theta_1 + \theta_2]$. The output of the model is the inter-service times for the first 50 customers.

The stochastic inputs are for the arrival time and the service time denoted $\boldsymbol{u}$ and $\boldsymbol{w}$ respectively with

$$u_i, w_j \sim U(\cdot|0, 1)$$

for $i = 1, 2, \ldots$ and $j = 1, \ldots, 50$. The ith arrival takes place at an interval

$$-\frac{\log(1 - u_i)}{\theta_3}$$

after the $(i - 1)$th arrival and the jth service time is

$$\theta_1 + \theta_2 w_j.$$

Given $\boldsymbol{u}$ and $\boldsymbol{w}$, the model is then deterministic. In practice this could be difficult as before running the model it is unknown how many u_is need to be generated so in practice a method of generating random variable needs to be included in the model. Having said this, in this case only u_i and w_i for $i = 1, \ldots, 50$ affect the output of the model because it is defined to be based on a fixed number of customers, rather than a fixed duration.

4.2.2 Infinite random fields

In the queuing model (Section 4.2.1), there is no limit to the number of customers that arrive before 50 of them have been served. However, only the first 50 customers affect the model output so we only have to worry about the first 50 inputs. The inputs after these could be generated in any way and will have no effect on the model output. However for some models it is not known how many random inputs are needed for one model run. For example, if the model was run for a finite interval of time rather than for a specific number of customers, then we would be unsure how many random inputs were required to run the model.

One way of generating an unknown number of random numbers would be to generate too many random inputs before running the model. However this could be wasteful as we will be generating random inputs that are not used and may be computationally impossible due to memory. There is also no guarantee that the number of random inputs will be sufficient to run the model.

Another way of doing this is to generate the pseudo-random inputs whilst running the model. It is possible to generate a deterministic sequence of numbers between 0 and 1 that appear to be random by an auto-regressive process of order 1 (Devroye, 1986). This means that a whole sequence of pseudo random numbers can be generated from one number between 0 and 1. For example, in the queueing model, the model is deterministic conditional on the parameters, u_1, w_1 and pseudo random number generator. Algorithm 13 shows how this would work. In line 1 you can input more than just the first random inputs for each process if you wish and these random inputs will be used before generating new ones.

Algorithm 13 Generating random inputs on the fly

```
1: Input the parameters $\boldsymbol{\theta}$ and the first random inputs $u_1^j$ for $j = 1, \ldots, q$
   where $q$ is the number of processes that require random numbers.
2: repeat
3:    Run the model until a random number needs to be generated
4:    $u_i^j = f(u_{i-1}^j)$ {for the $j$th process and the $i$th random input in that
      process where $f$ is a pseudo random number generator}
5: until Model has finished running
```

Woodhoopoe model

Woodhoopoes are birds that can be found in sub-Saharan Africa (du Plessis, 1992). They live in groups with one dominant pair which are the only ones that breed. Neuert et al (1995) used an individual-based approach in order to model the population and group dynamics of the woodhoopoes. Railsback and Grimm (2012) simplified this model for use as an example in their textbook.

The simplified woodhoopoe model is an agent-based model where the agents are the woodhoopoes themselves and they live in groups with one dominant male and one dominant female which breed once a year. There are 25 groups in the model with the groups laid out in a circle and each group having two neighbours. Each step of the model represents one month and every month each woodhoopoe dies with a probability θ_1. The aim of a woodhoopoe is to become a dominant in a group so each subordinate will leave its group with probability θ_2 in order to try and become a dominant in another group. However leaving the group will leave the subordinate vulnerable to predators and it will be killed with probability θ_3. A full description of the model can be found in Section 2.2 and in Railsback and Grimm (2012).

There are a number of stochastic parts. Initially there are two adult males and females placed in each territory with the initial ages determined from the random inputs $\boldsymbol{v}$.

In the original model the order in which the subordinates leave their group in order to become a dominant elsewhere is random. In the coupled model each subordinate i will have a random input w_{it} at time t and then these will be ordered and this is the order the subordinates will attempt to leave their respective group.

Each woodhoopoe will have random inputs that determine whether or not they will leave the safety of the group as a subordinate, one that determines whether or not they die leaving the group and one that decides whether, if they have successfully left their group, they go left or right in order to search for a vacant suitable dominant position. Each woodhoopoe will also have a random input every month that determines whether or not they die naturally.

Each group will have its own random inputs that decide whether a newborn woodhoopoe will be male or female.

Conditional on these random inputs the model will become deterministic. We also describe coupling the woodhoopoe model in Section 7.4.2.

4.3 Inference

If we look at the joint posterior distribution of the parameters θ and the random inputs u conditional on the observed data $\boldsymbol{x}$, then if we can generate samples from

$$\pi(\theta, u|\boldsymbol{x}) = \frac{\pi(\theta)p(u)p(\boldsymbol{x}|\theta, u)}{p(\boldsymbol{x})} \tag{4.1}$$

and we marginalize over u we will find that

$$\begin{aligned}\int_U \pi(\theta, u|\boldsymbol{x})du &= \int_U \frac{\pi(\theta)p(u)p(\boldsymbol{x}|\theta, u)}{p(\boldsymbol{x})}du \\ &= \frac{\pi(\theta)p(\theta|\boldsymbol{x})}{p(\boldsymbol{x})} \\ &= \pi(\theta|\boldsymbol{x}) \end{aligned} \tag{4.2}$$

where U is the space $[0,1]^d$ and where d is the number of dimensions of u. Equation 4.2 is just the posterior for θ alone. In equation 4.1 the likelihood, $p(\boldsymbol{x}|\theta, u)$ is either 1 if

$$M(\theta, u) = \boldsymbol{x}$$

or 0 otherwise. It is often very difficult to find values of θ and u that will make the likelihood 1 and often has probability zero. However if we look at this and change the likelihood so instead of it being the likelihood of the data it is

$$p(\rho(s(M(\theta, u)), \boldsymbol{s}(\boldsymbol{x})) < \epsilon),$$

or in other words the ABC likelihood, then equation 4.2 will become the ABC posterior.

We could use ABC-MCMC (Marjoram et al, 2003) as a method of sampling from $\pi(\theta, u|x)$ by treating the random inputs u as additional parameters. In addition to this we are going to introduce Gibbs steps that should also improve the mixing of the Markov Chain. We will describe these methods in the next two subsections. We call this algorithm Coupled Gibbs ABC (CG-ABC).

4.3.1 Gibbs move

If we look at a typical model with parameters θ and random inputs u and w shown in Figure 4.1, we can see that conditional on the latent variables, z, the model output, Y, and the parameters, θ, are independent. If we condition on the latent variables z we can often move the parameters θ using Gibbs steps and then change the random inputs u so that z remains the same. The Gibbs step is shown in Algorithm 14.

Proposition 2. *The Gibbs move suggested above keeps the Markov Chain in its stationary distribution.*

Proof. The target distribution of the CG-ABC algorithm ABC posterior distribution of θ, u and w is

$$\pi(\theta, u, w)p(z|\theta, u)\mathbb{I}_{(\rho(s(M(z,w)),\boldsymbol{s}(\boldsymbol{x}))<\epsilon)}$$

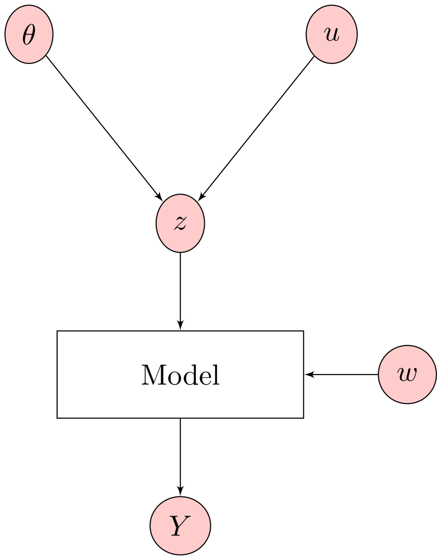

Figure 4.1: A general model. We have inputs, parameters θ and u that generate latent variables z, which is often tractable, and inputs w which along with the model and latent variables z generate a deterministic output Y.

where $M(z, w)$ is the model run a with the latent variable z and random inputs w, $p(z|\theta, u)$ is a Dirac mass at z and $\mathbb{I}_{(\rho(s(M(z,w)),\boldsymbol{s}(\boldsymbol{x}))<\epsilon)} = 1$ if the model is accepted and 0 otherwise. If we assume that the MCMC algorithm is in its stationary distribution, i.e. the ABC posterior, then we just have to satisfy detailed balance. We are going to propose a new point θ' and u' by first sampling $\theta' \sim p(\cdot|z)$ and then setting u' so that $p(z|\theta', u') = 1$. Hence

$$q(\theta', u'|z) = p(\theta'|z)p(u'|\theta', z) = p(\theta', u'|z).$$

The Metropolis-Hastings ratio then becomes

$$\begin{aligned} & \min\left(1, \frac{\pi(\theta', u', w)p(z|\theta', u')\mathbb{I}_{(\rho(s(M(z,w)),\boldsymbol{s}(\boldsymbol{x}))<\epsilon)}p(\theta, u|z)}{\pi(\theta, u, w)p(z|\theta, u)\mathbb{I}_{(\rho(s(M(z,w)),\boldsymbol{s}(\boldsymbol{x}))<\epsilon)}p(\theta', u'|z)}\right) \\ = & \min\left(1, \frac{\pi(\theta', u', w)p(z|\theta', u')\mathbb{I}_{(\rho(s(M(z,w)),\boldsymbol{s}(\boldsymbol{x}))<\epsilon)}\pi(\theta, u)p(z|\theta, u)p(z)}{\pi(\theta, u, w)p(z|\theta, u)\mathbb{I}_{(\rho(s(M(z,w)),\boldsymbol{s}(\boldsymbol{x}))<\epsilon)}\pi(\theta', u')p(z|\theta', u')p(z)}\right) \\ = & \min\left(1, \frac{\mathbb{I}_{(\rho(s(M(z,w)),\boldsymbol{s}(\boldsymbol{x}))<\epsilon)}}{\mathbb{I}_{(\rho(s(M(z,w)),\boldsymbol{s}(\boldsymbol{x}))<\epsilon)}}\right) \end{aligned}$$

which is 1 as $M(z, w)$ remains unchanged. □

Although the Gibbs step could be used with usual ABC-MCMC in the same way as described above, the addition of the coupling means that we can

Algorithm 14 The Gibbs step given latent variables z generated using the model $z = p(\theta, u)$ and random inputs u.

```
$\theta' \sim p(\cdot|z)$
$u' \leftarrow u$ such that $z = p(\theta', u')$
```

be more flexible and mix both Gibbs steps and Metropolis-Hastings (MH) moves. In the ABC-MCMC algorithm, if one wanted to move just one parameter then that would take one run of the model and will, if accepted, only move the parameter in one direction. With the addition of the Gibbs step, it would allow us to make a large movement in one direction by holding the other parameters without being wasteful in terms of model runs. For example suppose

$$Y = M(u_4, z_1, z_2, z_3)$$

with

$$\begin{aligned} z_1 &= f(\theta_1, u_1) \\ z_2 &= g(\theta_2, u_2) \\ z_3 &= h(\theta_3, u_3) \end{aligned}$$

then once we find a region of the parameter space where we get an accepted model

$$Y' = M(u'_4, z'_1, z'_2, z'_3)$$

then we could perform Gibbs steps

$$\begin{gathered} \theta''_1 \sim p(\cdot|z'_1) \\ \theta''_2 \sim p(\cdot|z'_2) \end{gathered}$$

and update u'_1 and u'_2 to u''_1 and u''_2 such that

$$\begin{aligned} z'_1 &= f(\theta''_1, u''_1) \\ z'_2 &= g(\theta''_2, u''_2). \end{aligned}$$

The model output Y' will not change as it is conditionally independent of the parameters given z_1, z_2 and z_3 and therefore will still be accepted. We can then propose θ''_3 using a Metropolis-Hastings step and conditioning on the inputs will allow us to make better moves in this direction with a reasonable acceptance rate. We are essentially just changing a handful of latent variables at a time here.

4.3.2 CG-ABC

In order to make sure that we have sampled from $p(u)p(x|\theta, u)$ we need to be sure that we have explored the whole of the random input space $[0, 1]^d$.

This means that as well as making small moves in the random input space we have to also make large moves in this space. There are a number of possible moves.

a) We could keep the random inputs the same and only new parameters could be proposed. This would mean that latent variables (z in Figure 4.1) will change and then the accepted model output, Y, will change. However we want to ensure that, if the random inputs are the same a small change in the parameters results in a small change in the model output. This move is essentially exploring an island in parameter space where the ABC likelihood is one rather than zero.

b) All of the random inputs could be re-sampled with either the parameters being changed or not. This will attempt to change the random inputs (u and w in Figure 4.1), the latent variables (z in Figure 4.1) and the accepted model. When the parameters move as well, it is the same as a step of the ABC-MCMC algorithm described by Algorithm 9. If the parameters don't move it is just a step to try and change the model and the latent variables in order to make other moves easier, for example finding another ABC-likelihood island to explore. This could be used with Gibbs steps described in Section 4.3.1.

c) A block of the random inputs could be changed. Again this could be done by the changing the parameters or not. This changes the model and subset of the latent variables (z in Figure 4.1). These moves in the random input space can either be coupled with moves in the parameter space which then increases the chances of the move being accepted or they could be re-sampled completely.

We are going to mix all of these moves in order to try and explore the joint space of the parameters and the random inputs.

4.4 Examples

We are going to demonstrate this method on some the models described in Section 4.2.1.

4.4.1 Ricker Model

We tried to perform inference on the Ricker model (described in Section 4.2.1). We simulated some data using $\log r = 3.8$, $\lambda = 10$ and $\sigma_e^2 = 0.3$ and used the summary statistics suggested by Wood (2010).

We found that the the model was very sensitive to small changes in $\log r$ and σ_e^2 when the inputs are fixed and that given these inputs there is no smooth acceptance region. This is shown in Figures 4.2 and 4.3 where we kept the random inputs constant but varied the parameters. The summary statistics are the summary statistics suggested by Wood (2010):

- the ordered differences of the simulated and observed values.
- the coefficients, β_1 and β_2, of autoregression, $y_{t+1}^{0.3} = \beta_1 y_t^{0.3} + \beta_2 y_t^{0.6}$.
- the mean of the outputs and the number of outputs that are equal zero.
- the autocovariances up to lag 5.

In this case we do not believe that it is useful to perform CG-ABC on this model because it is difficult to anticipate what a move in either the parameter space or the random input space would cause to the model output.

However it is possible to create Gibbs steps for each of the parameters conditional on an accepted model. We can re-write the latent state, N_t, as

$$\begin{aligned} \log N_t - \log N_{t-1} + N_{t-1} &= \log r + e_t \\ &\sim N(\cdot|\log r, \sigma_e^2) \end{aligned}$$

so therefore

$$\log r, \sigma_e^2 | N_{1:t} \sim NIG(d, a, m, v)$$

which is the Normal-inverse-gamma distribution with parameters

$$\begin{aligned} v &= \frac{1}{T-1}, \\ d &= \frac{T-1}{2} - \frac{3}{2}, \\ m &= \bar{X}, \\ a &= \frac{S}{2} \end{aligned}$$

where

$$\begin{aligned} X_t &= \log N_t - \log N_{t-1} + N_{t-1}, \\ \bar{X} &= \sum_{t=2}^{T} X_t \end{aligned}$$

and

$$S = \sum_{t=2}^{T} (X - \bar{X})^2.$$

We can condition on the hidden states, $N_{1:t}$, in order to find a closed state for ϕ. The likelihood

$$\begin{aligned} l(\phi|Y_{1:T}, N_{1:T}) &\propto (\phi N_1)^{Y_1} \exp(-\phi N_1) \cdots (\phi N_T)^{Y_T} \exp(-\phi N_T) \\ &\propto \phi^{\sum_{t=1}^{T} Y_t} \exp\left(-\phi \left(\sum_{t=1}^{T} N_t\right)\right) \end{aligned}$$

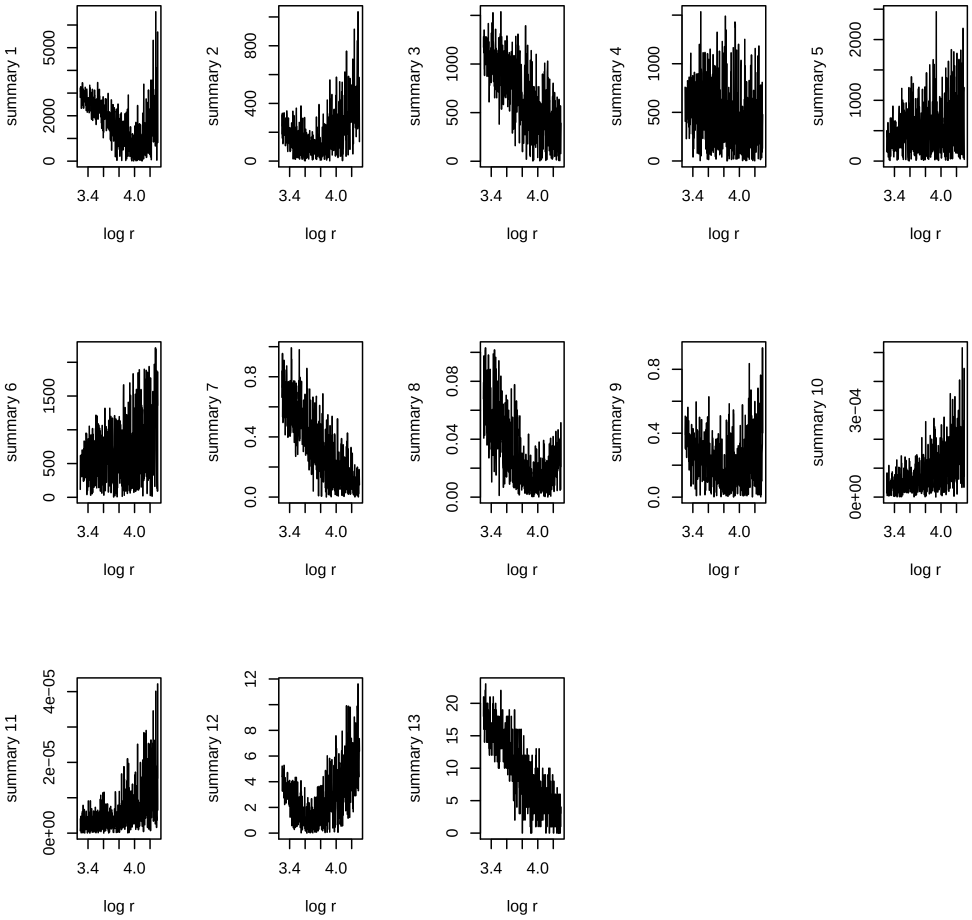

Figure 4.2: The summaries for various values of $\log r$ with the same random inputs.

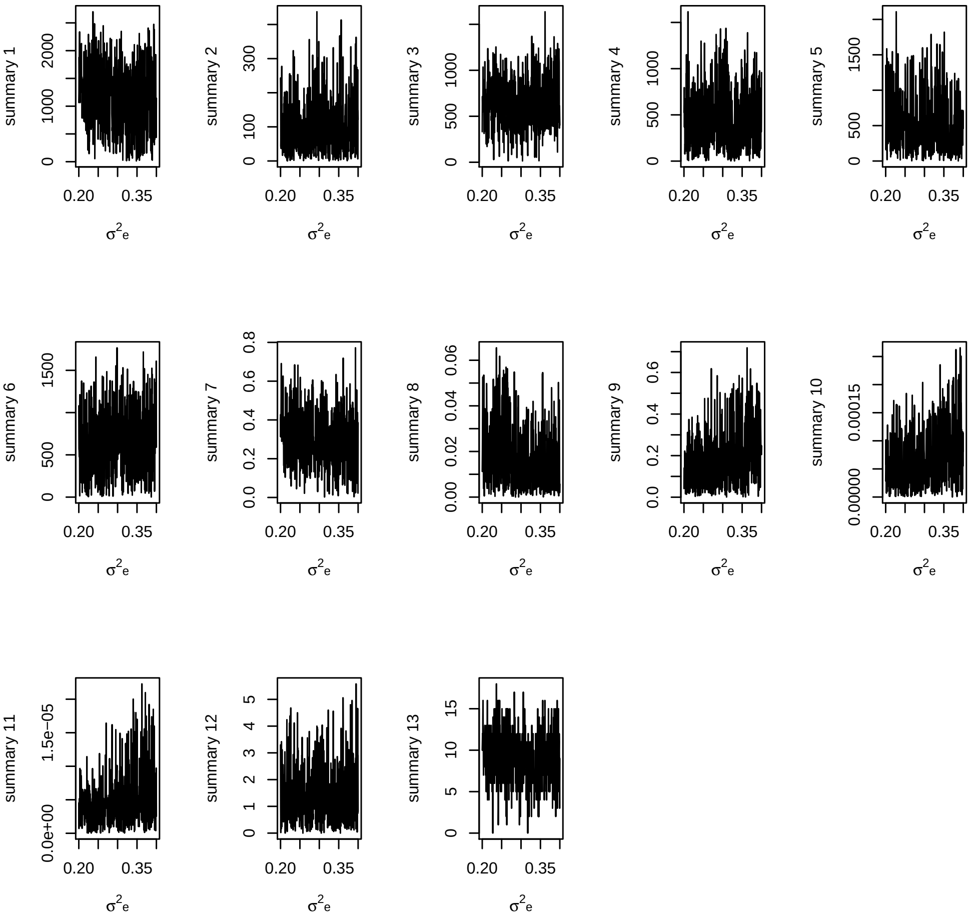

Figure 4.3: The summaries for various values of σ_e^2 with the same random inputs.

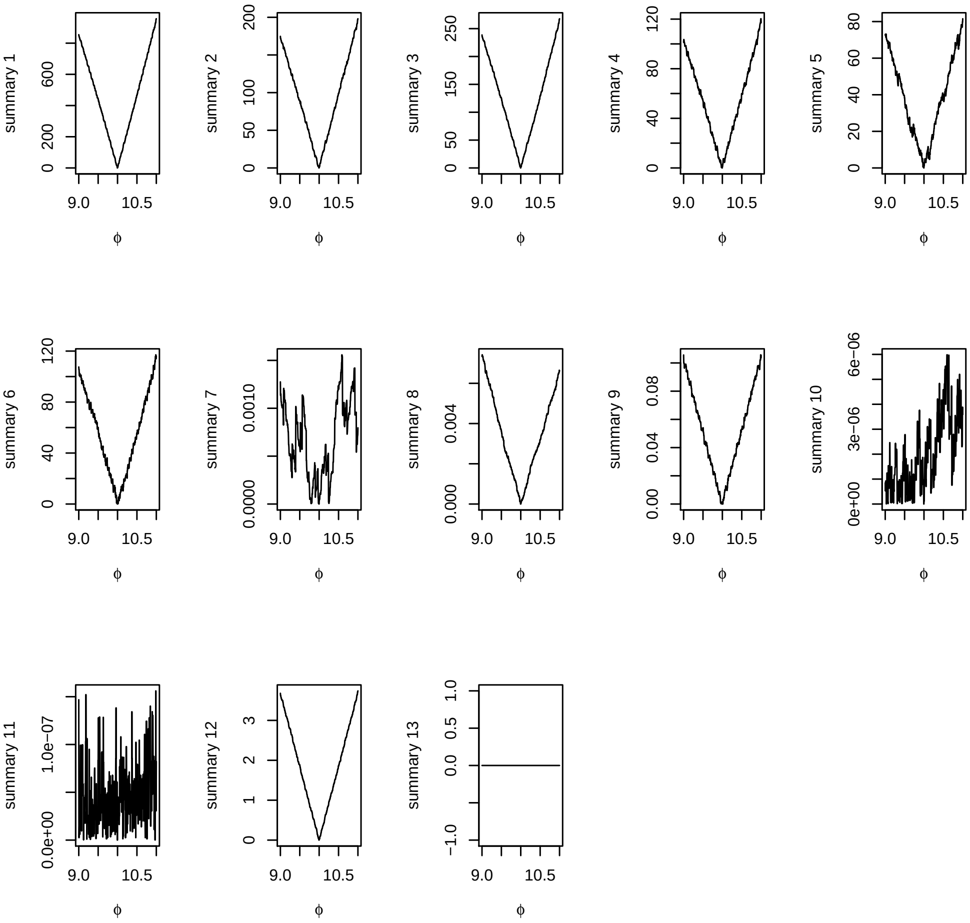

Figure 4.4: The summaries for various values of ϕ with the same random inputs.

and therefore

$$\phi|Y_{1:T}, N_{1:T} \sim Gamma\left(\cdot|1+\sum_{t=1}^{T} Y_t, \sum_{t=1}^{T} N_t\right).$$

These Gibbs steps could be used to improve the mixing of ABC-MCMC but we do not demonstrate it here.

4.4.2 Queuing model

We simulated some data with $\theta_1 = 1$, $\theta_2 = 5$ and $\theta_3 = 0.2$ and used summary statistics suggested by Blum and François (2010) to perform parameter estimation on θ_1, θ_2 and θ_3. The priors were uniformly distributed on $[0, 10]$ for θ_1 and θ_2 and on $\left[0, \frac{1}{3}\right]$ for θ_3.

Before describing the algorithms used for inference, it is useful to explore the structure of the region with likelihood 1 in the input space, in the vicinity of an arbitrary acceptable point. More precisely, we look at the effect of moving away from that initial point in various pairs of directions chosen from the θ_1, θ_2 and θ_3 axes, a random direction in $\boldsymbol{u}$ space and a random direction in $\boldsymbol{w}$ space. The results are shown in Figure 4.5. We found that the region of parameter space that will be accepted is smooth in all directions which allowed us to use CG-ABC in order to explore the ABC posterior distribution. We can also see that, when the random inputs change, the likelihood changes smoothly. This suggests that we can move around in the random input space in a similar way.

ϵ, the tolerance, was selected so that approximately a proportion of 0.00018 of the parameter sets sampled from the prior would be accepted. This is equivalent of setting the normalising constant, $p(\rho(s(M(\theta, u)), s(x)) < \epsilon) = 0.00018$. Each algorithm will be set up so that the model, which usually is the most computationally expensive part of the algorithm, is run the same number of times.

Gibbs Steps

It is possible to perform a Gibbs step by conditioning on the current accepted model. Let $\boldsymbol{s}$ be the service times of the customers with $\max(\boldsymbol{s}) = \check{s}$ and $\min(\boldsymbol{s}) = \hat{s}$. It is dependent on θ_1, θ_2 and u. The full conditional distribution is

$$f(\theta_1, \theta_2|\boldsymbol{s}) \propto \frac{1}{\theta_2^n} \times \pi(\theta_1, \theta_2) \tag{4.3}$$

provided $\theta_1 \leq \hat{s}$ and $\theta_2 \geq \check{s} - \hat{s}$ and 0 otherwise. The prior in this case is a uniform distribution so we can sample θ_1 and θ_2 from the above distributed truncated at 0 and 10 in both directions.

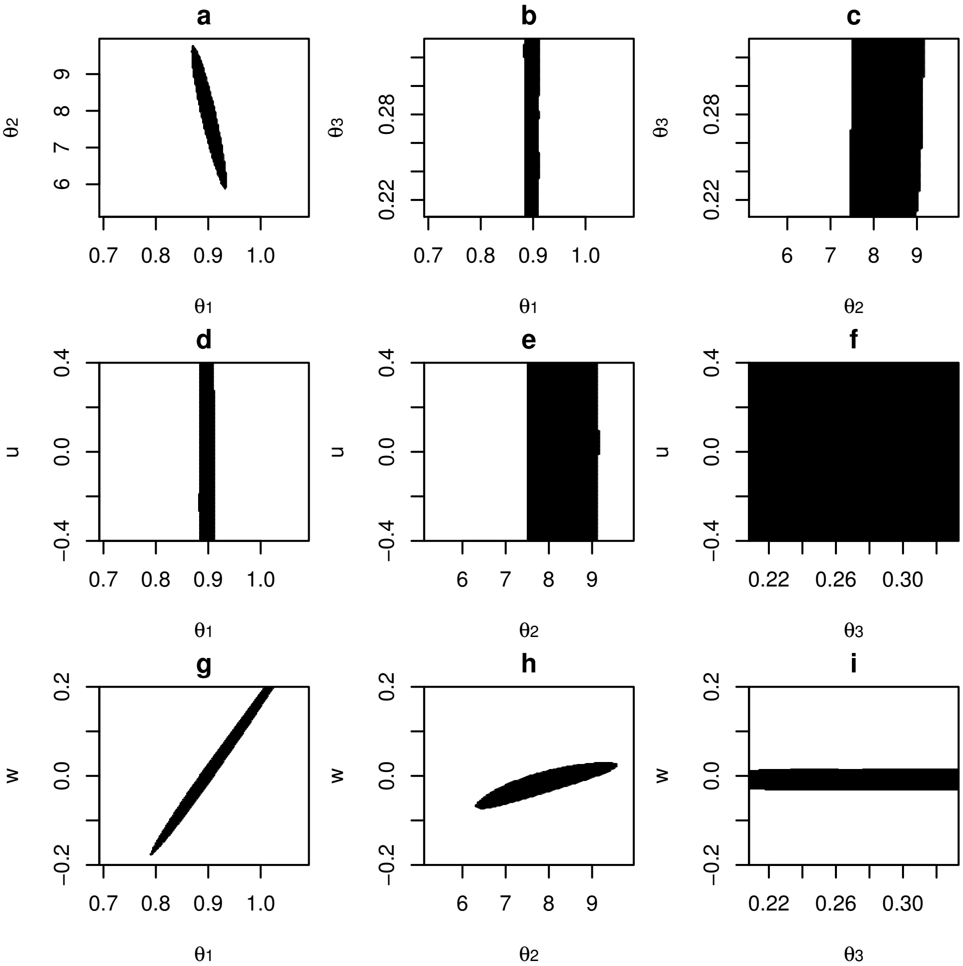

Figure 4.5: The movement along the different parameter and random input axes in the queuing model. **a**-**c** are movements along the parameters, **d**-**f** are movements in the $\boldsymbol{u}$ direction and **g**-**i** in the $\boldsymbol{w}$. It is black where the point would be accepted ($p(R|\theta) = 1$). Note that in f all of the plot is accepted which is not the usual case for all of the space but just the region shown in the plot.

If we let $\boldsymbol{a}$ be the arrival intervals then the full conditional distribution is

$$f(\theta_3|\boldsymbol{a}) \propto Gamma(\theta_3|n+1, \sum \boldsymbol{a}) \times \pi(\theta_3). \tag{4.4}$$

The prior for this parameter is a uniform distribution with parameters 0 and $\frac{1}{3}$ so we can sample θ_3 from the gamma distribution shown truncated at $\frac{1}{3}$.

Inference

We ran ABC-MCMC with optimal proposal distributions and found that the Markov Chain did not mix well (Figure 4.6). In order to improve the mixing

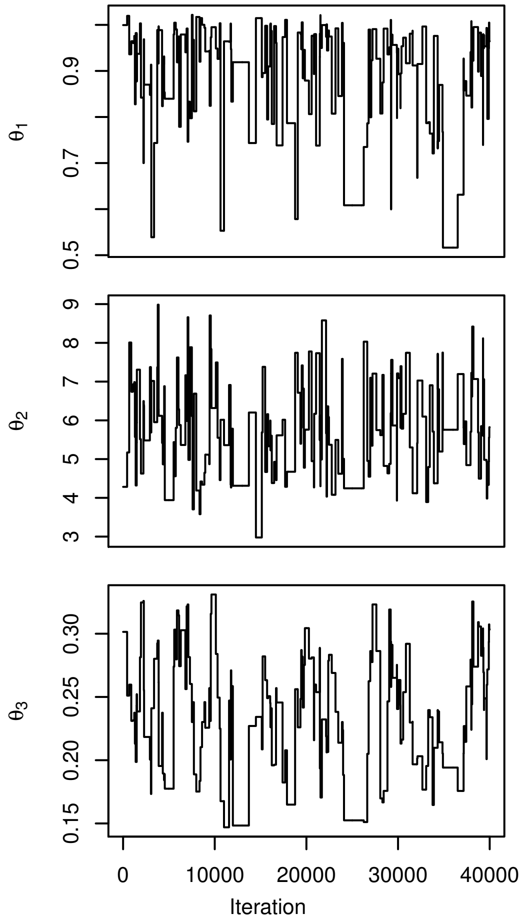

Figure 4.6: The trace plot of the ABC-MCMC version of queuing model.

we performed CG-ABC. At each iteration in the algorithm we performed one of several moves:

a) We only moved the parameters keeping the random inputs the same. This move aimed to explore the islands where the likelihood is 1; in other words, exploring the region in the top three graphs in Figure 4.5 (a,b,c). More formally, $\theta'_{1:3} \sim q_\theta(\cdot|\theta_{1:3}, \Sigma_a)$, where Σ_a is a tuning parameter, $\boldsymbol{u'} = \boldsymbol{u}$ and $\boldsymbol{w'} = \boldsymbol{w}$.

b) We moved the random inputs $\boldsymbol{u}$ a large amount in $\boldsymbol{u}$ space as well as all of the parameters. Looking at Figure 4.5, we can see that $\boldsymbol{u}$ is not very sensitive to the ABC likelihood and so this method was very similar to exploring the top three graphs in Figure 4.5 (a,b,c) but changed the arrival times in the model, thus exploring the latent variable space better. More formally, $\theta'_{1:3} \sim q_\theta(\cdot|\theta_{1:3}, \Sigma_b)$, where Σ_b is a tuning parameter, $\boldsymbol{u'} = U(\cdot|0, 1)$ and $\boldsymbol{w'} = \boldsymbol{w}$.

c) We moved the parameters and the random inputs $\boldsymbol{w}$ by a small amount. More formally, $\theta'_{1:3} \sim q_\theta(\cdot|\theta_{1:3}, \Sigma_c)$, where Σ_a is a tuning parameter, $\boldsymbol{u'} = \boldsymbol{u}$ and $\boldsymbol{w'} = q_u(\cdot|\boldsymbol{w}, \sigma_w)$ with σ_w being a tuning parameter.

d) We moved the random inputs $\boldsymbol{w}$ by a small amount and then moved θ_1. The two moves were correlated. This method changed the latent service times of the accepted model. More formally, $\theta'_1, \boldsymbol{w'} \sim q_{\theta u}(\cdot|\theta_1, \boldsymbol{w}, \sigma_w, b_{\theta_1 w}, \sigma_{\theta_1 w})$ with $b_{\theta_1 w}$ and $\sigma_{\theta_1 w}$ being tuning parameters, $\theta'_2 = \theta_2$, $\theta'_3 = \theta_3$ and $\boldsymbol{u'} = \boldsymbol{u}$.

e) We had a move the same as the previous one where θ_2 moves instead of θ_1. More formally, $\theta'_2, \boldsymbol{w'} \sim q_{\theta u}(\cdot|\theta_2, \boldsymbol{w}, \sigma_w, b_{\theta_2 w}, \sigma_{\theta_2 w})$ with $b_{\theta_2 w}$ and $\sigma_{\theta_2 w}$ being tuning parameters, $\theta'_1 = \theta_1$, $\theta'_3 = \theta_3$ and $\boldsymbol{u'} = \boldsymbol{u}$.

f) We had two additional moves that completely re-sampled the random inputs $\boldsymbol{u}$ and $\boldsymbol{w}$ as well as moving all three parameters, one by a small amount and the other by a larger amount. These moves are essentially the moves used in ABC-MCMC. The proposal distributions for these moves were the same as those for the ABC-MCMC algorithm we are comparing the results to. More formally, $\theta'_{1:3} \sim q(\cdot|\theta_{1:3}, \Sigma_{f_1})$ with probability 1/2 or $\theta'_{1:3} \sim q(\cdot|\theta_{1:3}, \Sigma_{f_2})$ with probability 1/2 with Σ_{f_1} and Σ_{f_2} being tuning parameters. On both occasions $\boldsymbol{u'}, \boldsymbol{w'} \sim U(\cdot|0, 1)$.

After one of these moves has taken place, all three of the parameters were updated by Gibbs steps.

We used truncated normals, truncated at 0 and 1, to move the random inputs (q_u) and multivariate normals to move the parameters (q_θ). For the joint moves involving the random inputs and parameters we first proposed the random inputs and then conditional on the proposed inputs proposed new parameters. For example, conditional on $\boldsymbol{w}$ and θ_1 we first proposed $\boldsymbol{w'}$ from a truncated normal then proposed θ_1' such that

$$\theta_1'|\boldsymbol{w}, \boldsymbol{w'}\theta_1 = \begin{cases} b_{\theta_1 w}||\boldsymbol{w} - \boldsymbol{w'}|| + \theta_1 + \sigma_{\theta_1 w}\zeta & \text{wp } \frac{1}{2} \\ -b_{\theta_1 w}||\boldsymbol{w} - \boldsymbol{w'}|| + \theta_1 + \sigma_{\theta_1 w}\zeta & \text{wp } \frac{1}{2} \end{cases}$$

where $||\boldsymbol{w} - \boldsymbol{w'}||$ is the Euclidean distance between $\boldsymbol{w}$ and $\boldsymbol{w'}$ and ζ is sampled from a standard normal. Algorithm 15 sums up the CG-ABC algorithm used to sample from the ABC posterior distribution. The initial value for $\theta_{1:3}$, $\boldsymbol{u}$ and $\boldsymbol{w}$ used in the CG-ABC algorithm was found by running the Rejection-ABC (Algorithm 7) to find one accepted parameter set.

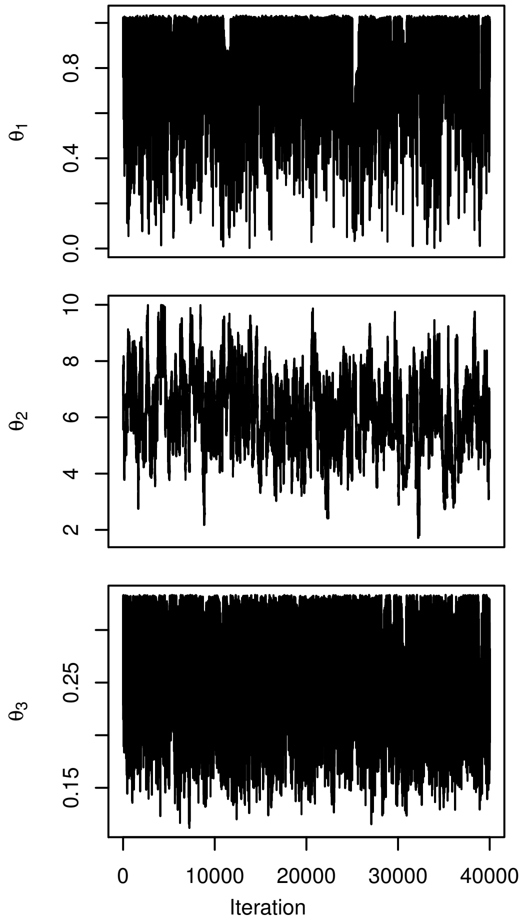

Figure 4.7: The trace plot of the coupled version of queuing model.

This method shows improved mixing and has high Effective Sample Sizes for the parameters, especially θ_1 and θ_3, as shown in Figure 4.7.

Algorithm 15 At each iteration of the CG-ABC algorithm for the queuing model, $\theta_{1:3}$, $\boldsymbol{u}$ and $\boldsymbol{w}$ are updated as follows.

```
$q \sim \{a), b), c), d), e), f)\}$ {The proposal function is sampled from the methods a) – f) described in Section 4.4.2 with equal probability.}
$\theta'_{1:3}, \boldsymbol{u'}, \boldsymbol{w'} \sim q(\cdot|\theta_{1:3}, \boldsymbol{u}, \boldsymbol{w})$ {Note that, dependent on $q$ sampled above, some of the elements, $\theta'_{1:3}, \boldsymbol{u'}, \boldsymbol{w'}$ may be equal to $\theta_{1:3}, \boldsymbol{u}, \boldsymbol{w}$)}
$\boldsymbol{y} \leftarrow M(\theta'_{1:3}, \boldsymbol{u'}, \boldsymbol{w'})$
if $\rho(S(\boldsymbol{y}), S(\boldsymbol{x})) < \epsilon$ then
    $\alpha \leftarrow \min\left(1, \frac{\pi(\theta'_{1:3}, \boldsymbol{u'}, \boldsymbol{w'}) q(\theta_{1:3}, \boldsymbol{u}, \boldsymbol{w} | \theta'_{1:3}, \boldsymbol{u'}, \boldsymbol{w'})}{\pi(\theta_{1:3}, \boldsymbol{u}, \boldsymbol{w}) q(\theta'_{1:3}, \boldsymbol{u'}, \boldsymbol{w'} | \theta_{1:3}, \boldsymbol{u}, \boldsymbol{w})}\right)$
    Sample $u \sim \mathrm{U}(\cdot|0, 1)$
    if $u < \alpha$ then
        $\theta_{1:3}, \boldsymbol{u}, \boldsymbol{w} \leftarrow \theta'_{1:3}, \boldsymbol{u'}, \boldsymbol{w'}$
    end if
end if
$\theta_1, \theta_2 \sim f(\cdot|\boldsymbol{s})$ {$\theta_1$ and $\theta_2$ are sampled from the density described in equation 4.3.}
$\boldsymbol{w} \leftarrow \frac{\boldsymbol{s} - \theta_1}{\theta_2}$
$\theta_3 \sim f(\cdot|\boldsymbol{a})$ {$\theta_3$ is sampled from the density described in equation 4.4.}
$\boldsymbol{u} \leftarrow 1 - \exp(-\theta_3 \boldsymbol{a})$
```

Results

We optimised the proposal distributions for both ABC-MCMC and CG-ABC and ran each algorithm 200 times and compared the empirical cumulative distribution functions of the parameters. We inspected the 5%, 25%, 50%, 75% and 95% percentiles of the samples and looked at the standard deviation of these estimates across the 200 runs. Table 4.1 shows that the CG-ABC gives a much smaller Monte Carlo error on the estimate of the distribution than ABC-MCMC. Moreover, it significantly improves the estimates of θ_1 and θ_3. As mentioned earlier, CG-ABC mixes better for θ_1 and θ_3 and this is demonstrated in the results. CG-ABC also improves the mixing of the chain as the ABC-MCMC algorithm has an acceptance rate of around 0.004 whereas the coupled version moves, with the exception of Gibbs steps, at a rate of around 0.08.

4.4.3 Woodhoopoe model

We simulated data from the Woodhoopoe model with $\theta_1 = 0.01$, $\theta_2 = 0.5$ and $\theta_3 = 0.2$ as suggested by Railsback and Grimm (2012). We simulated for 25 years, and treated the first 5 years as initialisation; the output consisted of the number of vacant dominant positions and the number of adult (subordinate or dominant) woodhoopoe in each group, at the end of each year for the remaining 20 years.

	Percentile	ABC-MCMC	CG-ABC	Ratio
θ_1	0.05	0.145	0.060	2.420
	0.25	0.112	0.020	5.757
	0.5	0.010	0.007	8.881
	0.75	0.036	0.003	13.851
	0.95	0.005	0.001	8.117
θ_2	0.05	0.373	0.379	0.984
	0.25	0.243	0.174	1.401
	0.5	0.228	0.156	1.596
	0.75	0.273	0.168	1.621
	0.95	0.411	0.228	1.799
θ_3	0.05	0.012	0.002	5.667
	0.25	0.010	0.003	3.941
	0.5	0.006	0.003	3.553
	0.75	0.011	0.003	4.382
	0.95	0.007	0.001	8.670

Table 4.1: The standard deviations of selected percentiles of the empirical cumulative distribution functions from 200 runs of the ABC-MCMC and CG-ABC with 40,000 points ran for the queuing model. The ratio of the standard deviation of the ABC-MCMC method and the CG-ABC method is shown in the final column.

The summary statistics for the ABC were the 0.25, 0.5 and 0.75 quantiles of the population; the minimum, mean and maximum of the number of groups with free alpha positions and the 0.25 and 0.75 quantiles as well as the mode of the group sizes. We also check that younger subordinates leave their groups more often than older subordinates by looking at the mean age of subordinates that leave and comparing it with the mean age of those that do not, as well the qualitative fact that more woodhoopoe leave their groups earlier in the year than later and rejected if the qualitative facts did not happen in the simulation. If the model reached its absorbing state all of the summaries are still defined and, for all values of ϵ used here, the parameter set would be rejected. When implementing this example we did not find a single model run where the model reached its absorbing state.

Gibbs steps

As with the queuing model, it is possible to perform Gibbs steps on the current accepted Woodhoopoe model. Instead of viewing θ_1 as the probability of dying each month, we can simulate when an individual will die due to natural causes at the beginning of its life. We can simulate the time (in months) until the death of woodhoopoe i, d_i, as

$$d_i \sim Geom(\cdot|\theta_1).$$

If we have n woodhoopoe born in the model, then the full conditional distribution is

$$f(\theta_1|\{d_i\}) = Beta\left(\theta_1|n, \sum (d_i - 1)\right) \times \pi(\theta_1)$$

where $\pi(\theta_1)$ is the prior distribution of θ_1, $U(\cdot|0, 0.05)$. Hence θ_1 is sampled from a truncated Beta distribution. Once θ_1 is sampled, the random inputs that are used to create these parameters are then moved so that the observations, d_is, remain the same.

Whether or not a subordinate leaves the group in order to attempt to become a dominant somewhere else is a Bernoulli process with parameter θ_2. If we look at the number of subordinates that left their group, l, compared to the number that didn't, s, then the full conditional distribution is

$$f(\theta_2|l, s) \sim Beta(\theta_2|l, s) \times \pi(\theta_2)$$

where $\pi(\theta_2)$ is the prior on θ_2 which is $Beta(\cdot|1, 1)$. We are just treating the stay/leave indicators as augmenting the data. We are going to move θ_2 as above and the random inputs so that the stay/leave indicators remain the same.

We can also say that when woodhoopoe i leaves their group, they will be killed by predation in its p_ith outing. We sample

$$p_i \sim Geom(\cdot|\theta_3).$$

If we have n woodhoopoe born in the model, then the full conditional distribution is

$$f(\theta_3|\{p_i\}) \sim Beta\left(\theta_3|n, \sum (p_i - 1)\right) \times \pi(\theta_3)$$

where $\pi(\theta_3)$ is the prior on θ_3, $U(\cdot|0, 0.5)$. Hence θ_3 is sampled from a truncated Beta distribution. Once θ_3 is sampled, the random inputs that are used to create these parameters are then moved so that the observations, p_is, remain the same.

Some of d_i and p_i may become irrelevant if either woodhoopoe i is still alive at the end of the model run or has died from the other cause. For example, suppose woodhoopoe j died because $p_j = 2$ and it took 25 months for this woodhoopoe to leave their group twice, then in order to keep the same model run the true value of d_j is not that important, just that $d_j > 25$. In this example we ignore this and keep all d_is and p_is the same for all woodhoopoe that were alive when the model was run.

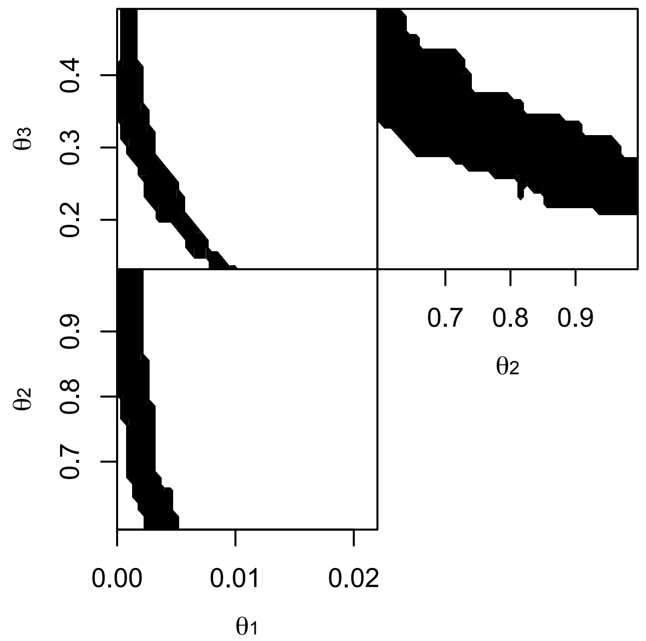

Figure 4.8: The accepted regions for each of the parameters in the woodhoopoe model. Regions where the points would be accepted are shown in black.

Inference moves

We are going to run the Markov Chain with mixed rules regarding both the proposal of parameters and the random inputs. At each stage of the algorithm we will perform one of several moves:

a) We move the parameters and none of the random inputs. Figure 4.8 shows that the regions that are to be accepted form a mass in the parameter space so this means that we can explore these regions and that is what this move does.

b) We move θ_1 and θ_3 according to Gibbs steps. The random inputs change so that the model doesn't change and then θ_2 is moved according to a Metropolis step fixing all of the other parameters and random inputs. This searches the accepted region in θ_2 conditional on θ_1 and θ_3. If the Metropolis step is accepted all three parameters and the model have moved but if it isn't only θ_1 and θ_3 will have moved. We do not move θ_2 according to a Gibbs step as $p(\theta_2|l)$ is very narrow and θ_2 doesn't move very much.

c) We had a move that completely re-sampled the random inputs, as well as moving the parameters. This move is essentially the move used in

ABC-MCMC.

We move a) twice as often as b) and c). We sum one iteration of the CG-ABC up in Algorithm **??**. The initial value for $\theta_{1:3}$ and $\boldsymbol{u}$ (all of random inputs) for the CG-ABC algorithm was found by running the Rejection-ABC (Algorithm 7) to find one accepted parameter set.

Algorithm 16 At each iteration of the CG-ABC algorithm for the woodhoopoe model, $\theta_{1:3}$ and $\boldsymbol{u}$, where $\boldsymbol{u}$ is all of the random inputs, are updated as follows.

```
$\gamma \sim Mult(3, (1/2, 1/4, 1/4)')$ {$\alpha = 1$ with probability 1/2, $\alpha = 2$,with probability 1/4 and $\alpha = 3$ with probability 1/4.}
if $\gamma = 1$ then
    $\theta'_{1:3} \sim N(\cdot|\theta_{1:3}, \Sigma_a)$
    $\boldsymbol{u'} = \boldsymbol{u}$
else
    if $\gamma = 2$ then
        $\theta_1 \sim f(\theta_1|\{d_i\})$
        $\theta_3 \sim f(\theta_1|\{p_i\})$
        $\boldsymbol{u} = f(\theta_1, \theta_3, \{d_i\}, \{p_i\})$ {Update $\boldsymbol{u}$ so that $\{d_i\}$, and $\{p_i\}$ remain the same.}
        $\theta'_2 \sim N(\cdot|\theta_2, \sigma_c^2)$
        $\theta'_1 = \theta_1$
        $\theta'_3 = \theta_3$
        $\boldsymbol{u'} = \boldsymbol{u}$
    else
        $\theta'_{1:3} \sim N(\cdot|\theta_{1:3}, \Sigma_c)$
        $\boldsymbol{u'} \sim U(\cdot|0, 1)$
    end if
end if
$\boldsymbol{y} \leftarrow M(\theta'_{1:3}, \boldsymbol{u'}, \boldsymbol{w'})$
if $\rho(S(\boldsymbol{y}), S(\boldsymbol{x})) < \epsilon$ then
    $\alpha \leftarrow \min\left(1, \frac{\pi(\theta'_{1:3}, \boldsymbol{u'})}{\pi(\theta_{1:3}, \boldsymbol{u})}\right)$
    Sample $u \sim \mathrm{U}(\cdot|0, 1)$
    if $u < \alpha$ then
        $\theta_{1:3}, \boldsymbol{u} \leftarrow \theta'_{1:3}, \boldsymbol{u'}$
    end if
end if
```

Results

We found that the CG-ABC (Figure 4.9) improved the mixing of the Markov chain compared to ABC-MCMC (Figure 4.10).

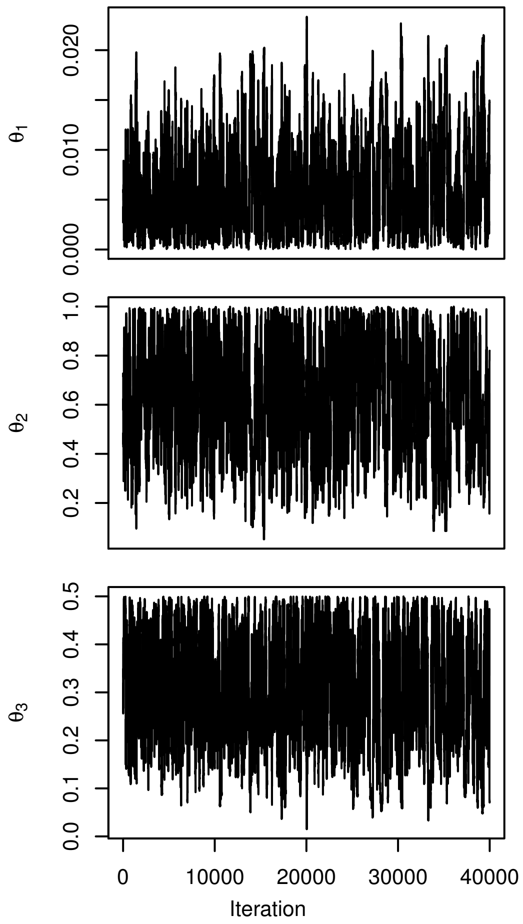

Figure 4.9: The trace plot of the CG-ABC version of the Woodhoopoe model.

Although these facts show that there is an improvement in the mixing and results are much smoother, they do not show that the method reduces the Monte Carlo error. In order to compare Monte Carlo error, we examined the standard deviations of the quantiles of the estimations of θ_1, θ_2 and θ_3. Table 4.2 shows that in most cases CG-ABC shows an improvement over ABC-MCMC for estimates of the three parameters.

4.5 Discussion

In this chapter we have proposed a method of exploring the random inputs and the parameters in order to improve the mixing of Monte Carlo algorithms for complex stochastic models. We have shown empirically that

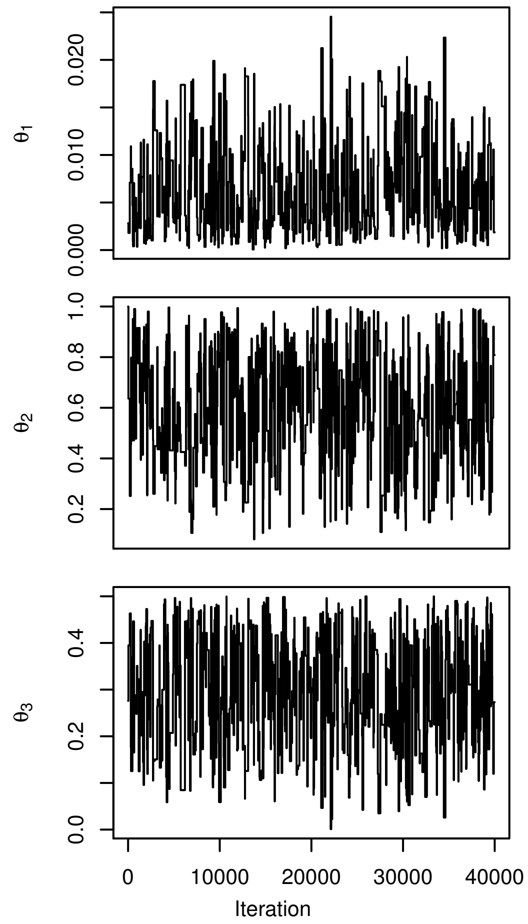

Figure 4.10: The trace plot of the ABC-MCMC version of the Woodhoopoe model.

by controlling the random inputs, we can improve the mixing of the ABC-MCMC algorithm with the greatest improvements being in the tails of the distribution, which is one of the major criticisms of that algorithm (Sisson et al, 2007; Lee et al, 2011). Like ABC-MCMC, CG-ABC takes advantage of the pseudo-marginal MCMC (Andrieu and Roberts, 2009) but the proposal method is changed in a much more strategic way. This does not effect the validity of the likelihood estimate though.

One criticism of CG-ABC could be that the dimension of the state space for the MCMC is increased, and that we may fail to explore the random input space completely, unlike the conventional case where new random inputs are generated for every run. Note that this cannot simply be checked by looking at the marginal posterior distributions of the random inputs, since it is not

	Percentile	ABC-MCMC	CG-ABC	Ratio
θ_1	0.05	1.18×10^{-4}	6.13×10^{-5}	1.931
	0.25	2.70×10^{-4}	1.66×10^{-4}	1.630
	0.5	4.90×10^{-4}	3.63×10^{-4}	1.351
	0.75	6.21×10^{-4}	5.76×10^{-4}	1.078
	0.95	9.21×10^{-4}	9.13×10^{-4}	1.008
θ_2	0.05	2.19×10^{-2}	1.76×10^{-2}	1.246
	0.25	2.19×10^{-2}	1.80×10^{-2}	1.218
	0.5	2.34×10^{-2}	1.79×10^{-2}	1.304
	0.75	2.00×10^{-2}	1.49×10^{-2}	1.341
	0.95	09.51×10^{-2}	5.23×10^{-3}	1.816
θ_3	0.05	1.35×10^{-2}	1.55×10^{-2}	0.873
	0.25	1.11×10^{-2}	1.14×10^{-2}	0.968
	0.5	1.08×10^{-2}	9.43×10^{-3}	1.142
	0.75	9.89×10^{-3}	7.43×10^{-3}	1.329
	0.95	5.19×10^{-3}	2.70×10^{-3}	1.918

Table 4.2: The standard deviation of the empirical cumulative distribution function for 200 runs of the ABC-MCMC and CG-ABC and the ratio of the two standard deviations.

generally the case that these are uniform on $[0, 1]^d$. For example, in the woodhoopoe model, in order for a model run to be accepted, some of the random inputs have to be larger than others so the posterior of the random inputs is certainly not uniformly distributed. In order to try to explore this, we looked at the effective sample size. We found that the effective sample size of the random inputs was considerably higher for CG-ABC than ABC-MCMC so this suggests that we are exploring the random input space. Furthermore if we were not exploring the random inputs space, the Monte Carlo error of the estimates in the two examples would be higher. Given this information we are convinced that we are exploring the random input space as well as the ABC-MCMC algorithm does.

An important part of this algorithm is the Gibbs steps. When performing Gibbs steps, the accepted version of the model remains the same and only the parameters change. This means that these moves can be done with standard ABC-MCMC. If the chain is stuck in a region, the Gibbs steps make the chain move and could cause a chance of getting out of that region and the model moving. We found that the addition of Gibbs steps improved the ABC-MCMC but not as much as CG-ABC did.

If we use Gibbs steps with the coupling we can update the latent variables in blocks whilst still updating all of the parameters, like in move b) in the woodhoopoe example. One of the advantages of coupling is that we can make Gibbs moves in some directions and Metropolis moves in other directions.

This would be the same as moving a subset of latent variables z like in the example in Section 4.3.1. In order to do this the parameters are moved according to Gibbs steps and the random inputs are changed so that if the model was run, the latent variable generated by the parameter remains the same even though the parameters are different.

Moving in one direction in parameter space allows a more controlled move and therefore allows a larger move. This can be wasteful in ABC-MCMC as every move requires one run of the model. However with the coupled Gibbs steps all of the parameters can be moved in one step at the expense of one model run and with the same chance of being accepted as moving in one direction. This is allowed only because of the coupling keeping the accepted model the same. Therefore it would be interesting to see how well CG-ABC works in higher dimensions.

CG-ABC allows us to control the latent variables much more easily by only changing a handful of them at a time and often we can change the parameter and the random inputs together in order to increase the chances of moving to an accepted set of latent variables as shown in the queuing example in Section 4.4.2. This allows us to use CG-ABC in order to move around when the tolerance ϵ is lower.

One problem when ϵ is lower or when the prior space is large is finding the posterior region in the first place. Using standard MCMC, when the likelihood is known, the chain will move towards the region where there is non-negligible posterior density, however in ABC-MCMC the likelihood is either one or zero and it is the region where the likelihood is one that we are looking for so either the Markov Chain starts in the target region or it will move around as a random walk looking until it finds the region. The problems with these is that they are both very inefficient. The random walk may never find the target region (Lee and Łatuszyński, 2014) and finding the region using other methods, eg. SMC-ABC or rejection-ABC could be very inefficient. One feature that could be exploited is that the model is deterministic conditional on the random inputs. This means that optimisation methods could be used in order to find a point to start the chain. Instead of finding the minimal difference between the simulated output and the data we just need to run the optimisation algorithm until we find a set of inputs that generate output that is within ϵ of the data. This point could then be used as a starting point for CG-ABC.

CG-ABC is not just limited to use with ABC-MCMC. Regression-Based Conditional Density Estimation (Beaumont, 2010) (Section 3.3.1) could be performed after ABC-MCMC. This can only be done by marginalising over the random inputs and then performing weighted regression on the parameters.

There are a number of variations of the ABC-MCMC algorithm that have been described in Section 3.3.3 and not mentioned in this chapter. The coupling and the ABC-Gibbs steps can be applied to the method devised by

Wegmann et al (2009) and it would be interesting how much this improves the mixing of this algorithm. Both the Gibbs sampler and the coupled version will transfer directly to the GIMH-ABC algorithm of Lee et al (2012). However we believe that coupling this algorithm will not have a great effect on its efficiency because there are a lot of random inputs and controlling them may take a lot of computational effort and due to the large number of accepted models, the Gibbs step will make smaller moves. However it is difficult to see how a coupled version of the one-hit MCMC-ABC would work.

Chapter 5

Parameter uncertainty of a dynamic North Sea size spectrum model

5.1 Size-spectrum models

There are a number of ways of modelling a marine ecosystem and how species react to one another through predator-prey relationships (Polovina, 1984; Walters et al, 1997; Pauly et al, 2000). These models set a fixed trophic level (position in a food chain) throughout an individual's life. Although it would seem natural to build a model in this way, marine ecosystems often violate this assumption (Werner and Gilliam, 1984; Hartvig et al, 2011) because an individual moves up trophic levels as it grows and thus eats different things throughout its life (Pimm and Rice, 1987).

An alternative is to model the ecosystem by examining the size of individuals which is a much more appealing method as size is a very important driver of food web interactions and the abundance of organisms scale with body size (Andersen and Beyer, 2006; White et al, 2007). Empirical evidence has shown that the distribution of abundance for all species $N(w)$ follow a power law,

$$N(w) = aw^b$$

where w is the body size (White et al, 2007; Reuman et al, 2008) and is known as the size spectra (Sheldon and Parsons, 1966; Andersen and Beyer, 2006). In marine ecosystems there is a negative relation between the size of individuals and their abundance at that size which is linear on the log-log scale as shown in Figure 5.1. Damuth (1981) says that $b = -0.75$ which has become known as Damuth's rule (White et al, 2007) for the community size structure. Andersen and Beyer (2006) found that this is more like -0.5 for individual species. This approach has been used to show that the size spectrum becomes steeper following exploitation from fishing (Petchey and

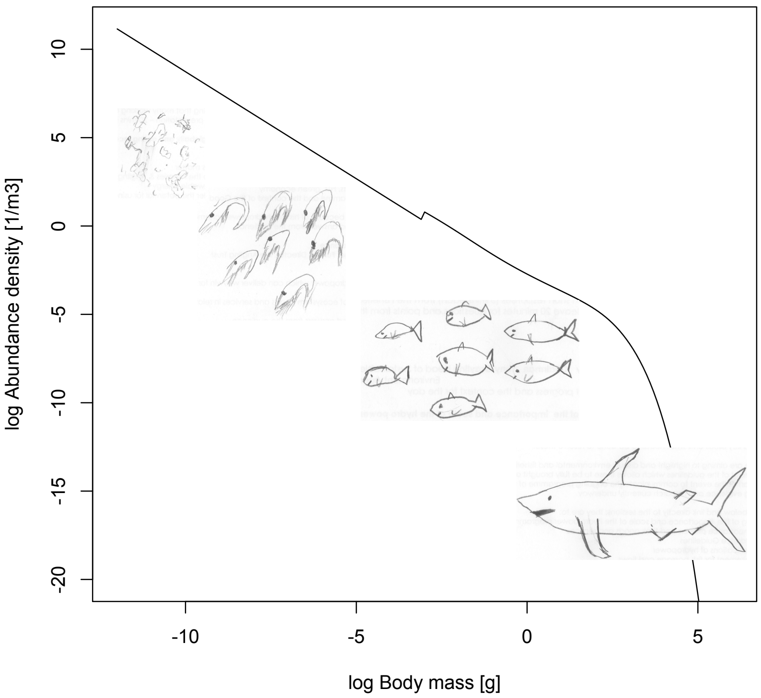

Figure 5.1: The size spectrum

Belgrano, 2010; Rice and Gislason, 1996; Jennings and Blanchard, 2004) as well as to calculate the number of monsters in Loch Ness (Sheldon and Kerr, 1972).

Benoît and Rochet (2004) developed a size-based model that examined the community as a whole based on the idea that "big things eat little things". This meant that the whole ecosystem was modelled as one species and a background resource and an individual could be identified by its weight w. This model was further developed to include different species by having a number of traits with different asymptotic sizes, W_i (Andersen and Beyer, 2006; Andersen and Pedersen, 2009; Hall et al, 2006; Rossberg et al, 2013; Pope et al, 2006) and to model different species with species-specific parameters (Hartvig et al, 2011; Blanchard et al, 2014).

Blanchard et al (2014) applied a trait-based model to the North Sea and set all of the other species-specific parameters as well as the asymptotic size according to values found in the literature from other studies to determine whether meeting management targets for exploited North Sea populations would be sufficient to meet proposed Marine Strategy Framework Directive targets for biodiversity and food web functioning (including the "Large fish indicator"). The model explicitly modelled 12 species as well as a general background resource with parameter values determined from other literature. The maximum recruitment parameters, R_0, and the background resource, κ, were unknown and so the model was then 'calibrated' to spawning stock biomass (SSB) and landings data by minimising the sum of squares, obtaining a point estimate for R_0 to time averaged data. However Blanchard et al (2014) said that the results are sensitive to the R_0 values. This suggests that rather than a point estimate, formal parameter uncertainty would be required as there could be some parameter uncertainty that has been overlooked in the solution.

There are three main types of uncertainty: parameter uncertainty, structural or model uncertainty and data uncertainty. Parameter uncertainty is uncertainty that is caused by uncertainty in parameters. This can be reduced by increasing the amount of data we fit the parameters to: Structural uncertainty is uncertainty in the model itself. This could be because of simplifications, stochastic elements, uncertain processes or even numerical approximations and data uncertainty is the uncertainty in the data. This can be transferred to the parameters but can also be shown in the model output. This could have been caused by sampling biases or errors in data collection (Harwood and Stokes, 2003).

Although parameter uncertainty is important it has hardly been considered on size structured models. Thorpe et al (2015) estimated the parameters of a length-based multi species model by performing an algorithm similar to Approximate Bayesian Computation (Beaumont, 2010; Tavaré et al, 1997) (Section 3.3) which resulted in the parameter values having a distribution rather than a point estimate. This shows all plausible parameter values and their probabilities that could have generated the data.

In order to use these methods to predict future events, such as how management strategies affect the population (Blanchard et al, 2014), the parameter, model and data uncertainty need to be quantified as when advising policy makers and environmental managers it is important to report uncertainty (Harwood and Stokes, 2003).

There are many challenges with fitting a size spectrum model in a Bayesian way. As the models are written as a series of algorithms and ecological rules, the model output at a particular parameter set is uncertain unless the model is run at these parameters meaning that getting posterior distributions in closed form is not possible. The model is actually quite slow to run, taking up to a minute at times, and this means exploring the parameter space is difficult which is often the reason why uncertainty has been overlooked. Additionally some parameter combinations result in a cyclic steady state (Law et al, 2009) which means that when in the cycle you take outputs has an effect on the reported output of the model.

We have parameterised the multi species size spectrum model and performed a Bayesian parameter estimate in this Chapter by first exploring the parameter space to find regions that fit the model reasonably well and then from these regions explored the posterior distribution using a common algorithm from the Bayesian literature, MCMC (Section 3.2).

In Section 5.2, we describe what we are trying to estimate and description the model. In Section 5.3 we describe how we performed the inference. We will show the results of the inference in Sections 5.4 and 5.5 and discuss the findings in Section 5.6.

5.2 Model

i	Species	SSB	Landings
1	Sprat	N/A	1967-2010
2	Sandeel	1983-2011	1983-2010
3	N. Pout	1983-2011	1983-2010
4	Herring	1960-2011	1967-2010
5	Dab	N/A	1967-2010
6	Whiting	1990-2011	1990-2010
7	Sole	1957-2011	1967-2010
8	Gurnard	N/A	1967-2010
9	Plaice	1957-2011	1967-2010
10	Haddock	1963-2011	1967-2010
11	Cod	1963-2011	1967-2010
12	Saithe	1967-2011	1967-2010

Table 5.1: Species data sets.

Blanchard et al (2014) developed a multi species size spectrum model. The model is based on the equations of Hartvig et al (2011) and Andersen and Pedersen (2009) but with explicit representation of species specific traits for the species shown in Table 5.1 and not just its asymptotic size as in trait based models (Scott et al, 2014).

The sized-based model is used to calculate the size- and trait-spectrum $N_i(w)$ which is the density of individuals such that $N_i(w)dw$ is the number of individuals of species i in the size interval $[w, w + dw]$.

The model works on the assumptions that an individual can be characterised by its weight and species number only which implies that the preference of food is determined by its weight combined with its species preference.

The model is a system of partial differential equations (PDE) and so is solved simultaneously for all of the species. The model is based on a prey selection principle where “big individuals eat smaller individuals” (Hartvig et al, 2011). In order to run the model a number of processes occur and are described here.

The process of growth and mortality to the size spectrum of each trait group is achieved by the McKendrik-von Foerster conservation equation

$$\frac{\partial N_i(w)}{\partial t} + \frac{\partial\left(g_i(w)N_i(w)\right)}{\partial w} = -\mu_i(w)N_i(w), \tag{5.1}$$

where $g_i(w)$ is the individual growth and μ is the mortality. Both growth and mortality are determined by the availability of food from other species plus the background resource, $N_R(w)$, and predation by the other species, as explained below.

Equation 5.1 is supplemented by a boundary condition at w_0, the weight of an egg, so that the increase of the number of individuals, $g_i(w_0)N_i(w_0)$ is determined by

$$g_i(w_0)N_i(w_0) = R_i$$

where R_i is the reproduction of the offspring by mature individuals in population i (Andersen and Zhang, 2011). Each species has its own preference in terms of the weights of prey that it will consume, expressed in terms of the ratio of weights between predator and prey, w and w_p respectively. This is calculated by an unnormalised log-normal distribution

$$\phi_i(w_p/w) = \exp\left[-\frac{(\log(w/\beta_i w_p))^2}{2\sigma^2}\right].$$

Predator-prey encounter

The predator prey encounter is based on the Anderson-Ursin encounter model (Andersen and Ursin, 1977; Andersen and Beyer, 2006). The amount of food potentially encountered is calculated and actual food encountered is calculated from this.

The food available for a predator of species i of weight w is

$$E_{a.i}(w) = \int \left(N_R(w_p) + \sum_{j=1}^{s} \theta_{ij} N_j(w_p) \right) \phi_i(w_p/w) w dw_p \tag{5.2}$$

where θ_{ij} is the preference of species i for species j and s is the number of species in the model. However the food encountered is dependent on the search rate $\gamma_i w^q$, where q is positive so that larger individuals search a larger region (Ware, 1978; Hartvig et al, 2011), and is determined by

$$E_{e.i} = \gamma_i w^q E_{a.i}.$$

Consumption

An individual doesn't consume all of the food it encounters. The consumption process determines the feeding level with a number between 0 (no food) and 1 (fully fed)(Scott et al, 2014).

The feeding level,

$$f_i(w) = \frac{E_{e.i}}{E_{e.i} + h_i w^n},$$

is a number between 0, where the individual gets no food, and 1, where the individual is fully satiated. h is the maximum food intake and n is the exponent for the maximum food intake (Kitchell and Stewart, 1977; Andersen and Ursin, 1977) which corresponds to a type II functional response (Scott et al, 2014; Hartvig et al, 2011).

Somatic Growth

Some of the energy gained from consumption is used for standard metabolism and activity and the rest is divided between growth and reproduction. When an individual is below the weight of maturity it will use all of its energy on growth whereas when an individual is close to its asymptotic weight, most of the energy will be used for reproduction.

If the intake of food is less than that required for the standard metabolism and activity, then the body size does not shrink. The individual would experience starvation mortality (Hartvig et al, 2011). This is described in the submodel "Mortality".

The energy available for growth and reproduction gained from consumption is

$$E_i(w) = \alpha f_i(w) h_i w^n - k_{s.i} w^p.$$

α is the efficiency parameter, $k_{s.i}$ is the energy required for standard metabolism and activity and p is the exponent of standard metabolism. The energy used for reproduction is

$$\psi_i(w) = \left[1 + \left(\frac{w}{w_{m.i}}\right)^{-u}\right]^{-1} \left(\frac{w}{W_i}\right)^{1-n},$$

where $w_{m.i}$ is the weight at maturation and W_i is the asymptotic size of species i. This means that when an individual is small it spends more energy growing and less when it is larger with u being the transition width (Hartvig et al, 2011). The somatic growth is

$$g_i(w) = E_i(w)(1 - \psi_i(w)).$$

Reproduction

Females produce eggs dependent on their efficiency of reproduction and energy allocated to reproduction in the Growth submodel.

The total production of eggs is

$$R_{p.i} = \frac{\epsilon}{2w_0} \int N_i(w) E_i(w) \psi_i(w) dw,$$

where w_0 is the egg weight, ϵ is the efficiency of reproduction and the $1/2$ takes into account that only females reproduce (Hartvig et al, 2011; Scott et al, 2014).

Recruitment

The recruitment is density dependent and also depends on the maximum recruitment for each species, $R_{0.i}$. This determines how many new individuals are put into the system.

The recruitment is defined by

$$R_i = R_{0.i} \frac{R_{p.i}}{R_{p.i} + R_{0.i}}.$$

Mortality

There are four types of mortality. There is predation mortality, where individuals are predated by other species, starvation mortality, the mortality of individuals that didn't consume enough energy for standard metabolism and activity, background mortality and fishing mortality, mortality is caused by human beings removing fish from the system. The background mortality is needed so that the largest individuals in the community experience mortality as they are not predated upon by any individuals from the community spectrum.

Mortality, $\mu_i(w)$, is the sum of the four types of mortality

$$\mu_i(w) = \mu_{p.i}(w) + \mu_{s.i}(w) + \mu_{b.i}(w) + \mu_{f.i}(w).$$

The predation mortality is defined as

$$\mu_{p.i}(w_p) = \sum_{j=1}^{s} \int \phi_j(w_p/w)(1 - f_j(w)) \gamma_j w^q \theta_{ji} N_j(w) dw$$

which is the sum over all of the species that have preyed on species i. This follows from the predation in equation 5.2. If $E_i(w) < 0$ then an individual doesn't receive enough energy to perform standard metabolism and activity and therefore has a starvation mortality, $\mu_{s.i}(w)$, such that

$$\mu_{s.i}(w) = \begin{cases} 0 & \text{if } E_i(w) > 0 \\ \frac{-E_i(w)}{\chi w} & \text{otherwise} \end{cases}$$

where χ is the fraction of energy reserves (Hartvig et al, 2011). The background mortality is

$$\mu_{b.i}(w) = \mu_0 W_i^{n-1}.$$

The fourth mortality is through fishing. The catch or yield for species i, Y_i, is dependent on the stock abundance or biomass by

$$Y_i = f_i \int q_i(w) N_i(w) dw$$

where f_i is the fishing effort or intensity and $q_i(w)$ is the catchability coefficient for species i. This is used to estimate the fishing mortality

$$\mu_{f.i}(w) = q_i(w) f_i.$$

Resource dynamics

The smallest of individuals eat a background resource, as seen in equation 5.2. This resource is described using semi-chemostatic growth, meaning that it gets replenished semi automatically. This is

$$\frac{\partial N_R(w,t)}{\partial t} = r_0 w^{p-1} \left[\kappa w^{-\lambda} - N_R(w,t) \right] - \mu_p(w) N_R(w,t),$$

where $r_0 w^{p-1}$ is the population regeneration rate (Fenchel, 1974; Savage et al, 2004) and the carrying capacity is

$$\kappa w^{-2-q+n}.$$

5.2.1 Model applied to the North Sea

Blanchard et al (2014) fitted the model described above to the North Sea by fixing the parameters from other studies and sources. They were unable to get values for $R_{0.i}$, the maximum carrying capacity for each species, and κ, the background food resource's carrying capacity, so they ran an optimisation algorithm that estimated these parameters.

The authors varied the fishing effort within the model from data from single species stock assessments (www.isec.dk). However, the $R_{0.i}$ values were fitted to time averaged fishing efforts from 1985-1995, B_i, and then the dynamics of the system were added to forecast future effects of different fishing scenarios. In addition to this, uncertainty was added by making the recruitment stage stochastic.

In what follows we are going to fit $R_{0.i}$ and κ to the model with the dynamics on using a Bayesian framework in order to examine the uncertainty from uncertainty in parameters and the data rather than adding stochasticity to the model.

5.2.2 Errors

We are going to fit the model to SSB and landings data, $\boldsymbol{Y}$, from stock assessments (www.ises.dk) for the years shown in table 5.1. If the landings were simulated from the model, $M(\boldsymbol{\mu})$, where the unknown parameters are defined $\boldsymbol{\mu}$ and the other inputs are implicit in $M(\cdot)$, then it is usual to say

$$\boldsymbol{Y} = M(\boldsymbol{\mu}) + \delta(\boldsymbol{\mu}) + \Sigma^{\frac{1}{2}}\boldsymbol{\epsilon},$$

where $\delta(\boldsymbol{\mu})$ is the model discrepancy and $\boldsymbol{\epsilon}$ are independent errors with mean 0 and variance 1 (Kennedy and O'Hagan, 2001). For this model we will assume that

$$\log \boldsymbol{Y} = \log M(\boldsymbol{\mu}) + \Sigma^{\frac{1}{2}}\boldsymbol{\epsilon}$$

where Σ's off diagonal elements are 0 and the diagonal elements are σ_i^2 ($i = 1 \ldots 9$ for SSB and $i = 1 \ldots 12$ for landings) and $\boldsymbol{\epsilon}$ is a vector of standard normals like in Nielsen and Berg (2014).

5.2.3 Numerical solution of the PDE

The model is a solution of partial differential equations (PDEs) which is intractable so we will have to estimate it by discretising the time points and size (Hartvig et al, 2011). Essentially the year is divided into δt amounts and the PDEs are estimated at these parts. Initially we experimented with $\delta t = 1$, the same value used by Blanchard et al (2014), i.e. the PDEs were estimated every year, and we found that the likelihood surface was very unstable and that often made a large difference to the results. As δt decreases the numerical estimation becomes more accurate. Changing δt we found that the estimate stabilised at around $\delta t = \frac{1}{4}$. However as we decrease δt, the model takes longer to run so we have the classic problem of efficiency verses accuracy. Table 5.2 shows what happens as we vary δt. It shows how as δt decreases the likelihood becomes more stable but the amount of time in order to calculate the likelihood increases.

$\delta t = 1$		$\delta t = \frac{1}{2}$		$\delta t = \frac{1}{3}$		$\delta t = \frac{1}{4}$		$\delta t = \frac{1}{5}$		$\delta t = \frac{1}{6}$	
-2×10^4	12	-760	22	-742	23	-742	33	-742	41	-742	49
-8×10^3	11	-748	22	-734	23	-736	32	-738	40	-739	49
-2×10^5	16	-1063	24	-753	34	-740	48	-741	59	-741	73
-3×10^5	15	-738	24	-735	24	-735	32	-736	41	-736	48

Table 5.2: The log likelihood and time running, in seconds, for four different parameter sets close to the point found by (Blanchard et al, 2014) and different values of δt. We found similar results across the whole of the prior space.

5.2.4 Burn-in

The model requires a burn-in period, where the fishing effort is fixed, so that the model reaches a steady state before the fishing effort is varied and output is collected. It is also not obvious what the fishing effort should be whilst the model in burning in so we have added the burn-in fishing mortality as an additional parameter to estimate for each of the 12 species, $[\xi_i]_{i=1}^{12}$.

Sometimes the steady state of the model results in the dynamics being constant, as in Figure 5.2a, however sometimes the dynamics in the steady state have some cyclic behaviour in the burn-in period of the model, as in Figure 5.2b.

Looking at Figure 5.2c, where we have runs of the model with the same parameters that generate a cyclic steady state but started the model dynamics at different points in the cycle, we can see that the biomass of sprat changes. This means that where in the cycle the run is very important. This is further shown in Figure 5.2d. It shows that as we start the model in different phases the log-likelihood changes by as much as 4.

If we let $N_{i,t}(w)$ be $N_i(w)$ at time t when

$$\frac{\partial N_i(w)}{\partial t} = 0 \tag{5.3}$$

then the phase of the system is

$$\inf\{|t - t'| : N_i(w)_t = N_i(w)_{t'} \forall i, t \neq t'\}.$$

We will treat the phase that we begin the model in as an additional parameter ω. ω will take a value between 0 and 1 with 0 being at the beginning of the of the phase, t such that equation 5.3 is satisfied for all i. Note that if $\omega = 0$ or $\omega = 1$, the model will begin at the same point, the beginning of the cycle. Furthermore, when we have more than one element in the set

$$\{t : N_i(w)_t = A_i \forall i\}$$

for any $\boldsymbol{A}$, then the model run has reached its steady state.

5.2.5 Fishing mortality

The model enables the fishing effort to be varied over time. The fishing effort is the amount of effort put into fishing and can be measured by the number of boats, man hours fishing or nets put in the water and the catch ability coefficient is the effectiveness of the gear used (Jul-Larsen et al, 2003). Blanchard et al (2014) estimated the fishing effort using stock assessments (www.ices.dk) for the 12 species from 1967 to 2010. According to these inputs, the fishing effort for Norway Pout in 2005 is 0. This is inconsistent with the landings data as we have landings data for Norway Pout in this year. In order to try and estimate this we have added the fishing effort of Norway Pout in 2005 as another parameter, ρ.

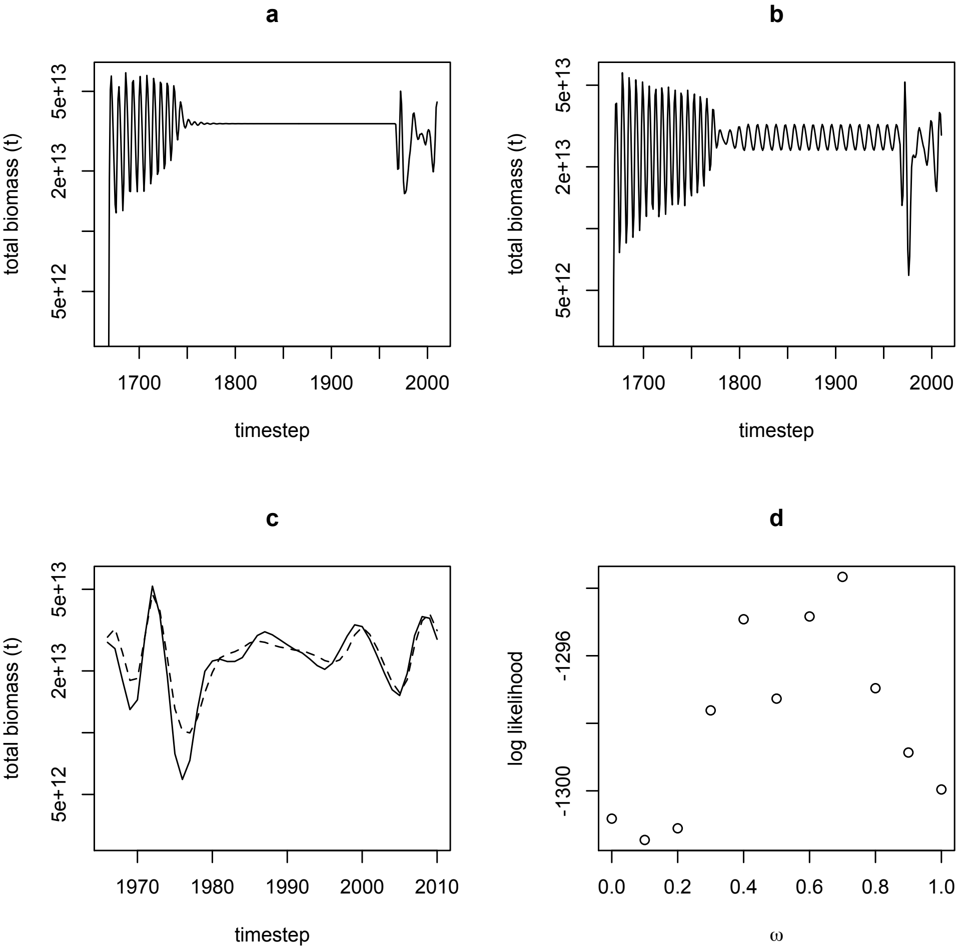

Figure 5.2: Images a, b and c show the total biomass for sprat whilst running the model. The time varying inputs start in 1967. In a, the parameters create a steady state that is constant whereas the parameters used to in b generate a steady state that has some cyclic behaviour. In c, the model is run with the same parameters as in b, but is started at two different values of ω. In a, all of the species have a constant steady state whereas in b all of the species have cycles of the same length. d shows how the log likelihood varies with ω.

5.3 Methods

We are going to use an MCMC algorithm to explore the posterior distribution (Section 3.2). The model is very sensitive to small changes in the parameters and the output is high dimensional so we were unable to build an emulator (Oakley and O'Hagan, 2004) (Section 3.4.3). We can however run the model in order to find the value of the likelihood at a given point in parameter space. This takes, with $\delta t = \frac{1}{4}$, about one minute per parameter evaluation so we need a way of speeding this up by taking advantage of parallel computing. The posterior distribution is also multimodal which means that we are going to use parallel tempering to explore the posterior distribution. Furthermore, we are going to couple the version of Cui et al (2011) with parallel tempering and use the doubly parallel tempering algorithm to explore the posterior (Section 3.2.3).

5.3.1 Priors

For the inputs $R_{0.i}$, we specify priors in terms of $\psi_i = \log R_{0.i}$ taking $\psi_i \sim U(\cdot|\alpha_i, \beta_i)$, where $\alpha_i < \beta_i$. So the prior density for $R_{0.i}$ is

$$f_{R_{0.i}}(r) = \begin{cases} \frac{\exp(r)}{\beta_i - \alpha_i} & \text{if } \exp(\alpha_i) \leq r \leq \exp(\beta_i) \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1 \dots 12$. Similarly we specify the prior κ in terms of $b_0 = \log \kappa$ taking $b_0 \sim U(\cdot|\gamma, \zeta)$, where $\gamma < \zeta$. So the prior density for κ is

$$f(\kappa) = \begin{cases} \frac{\exp(\kappa)}{\zeta - \gamma} & \text{if } \exp(\gamma) \leq \kappa \leq \exp(\zeta) \\ 0 & \text{otherwise.} \end{cases}$$

For this inference

$$\alpha_i = \gamma = 0$$

and

$$\beta_i = \zeta = 50$$

for $i = 1 \dots 12$.

The additional latent parameters, $[\xi_i]_{i=1}^{12}$ are used to make sure that the model is in the correct state when the model is run. The parameters represent the fishing effort over the period before the model is run. Although they are nuisance parameters, it does not make sense for them to be negative. We decided on

$$\xi_i B_i \sim \text{Half normal}\left(\cdot\,|0, (1.824)^2\right)$$

where B_i is the mean fishing effort between 1985 and 1995 for species i. This ensures that this value is positive as if it was negative it would not have an interpretation in the model. This value was agreed upon after communications with Julia Blanchard.

We assumed that the yield for Norway Pout was just down to rounding error so we put quite an informative prior on ρ such that

$$\rho \sim Exp\left(\cdot\left|\frac{1}{0.23}\right.\right),$$

the mean value 0.23 being agreed upon after personal communication with Julia Blanchard. The phase parameter ω does not have a real life representation and is just a nuisance parameter so we gave it a flat prior

$$\omega \sim U(\cdot|0,1).$$

The variance parameters will all have inverse-gamma prior distributions with parameters and will be defined

$$\sigma_i^2 \sim Inv - Gamma(\cdot|0.0001, 0.0001)$$

for $i = 1, \ldots, 21$. This is a fairly uninformative prior as we do not have much prior knowledge of the errors.

5.3.2 Exploration

The output from the 27 dimensional input space is not smooth and full of peaks and troughs that an MCMC chain will get stuck in. So in order to find a region with non-negligible posterior mass and to keep the computations as efficient as possible we initially performed a Latin hypercube sample (McKay et al, 1979) with $\delta t = \frac{1}{2}$. We also decided that the phase in which the model started in was not that important for determining where in the parameter space had non-negligible mass as it doesn't effect the likelihood that much (Figure 5.2d) so we fixed $\omega = 0$ whilst exploring the space.

Latin hypercube sampling (LHS) (McKay et al, 1979; Rose, 1989) is a method that assembles random values of the parameters that represent the true variability of the system. It allows samples to be drawn from different areas of the parameter space (Cunningham, 2007). Each parameter's prior probability density function is divided into M equally probable regions. In each round R regions are selected with equal probability from the M regions without replacement and then a point is sampled from each selected region using the probability distribution within that region. The model is then run using that point.

In the first round we used LHS to sample 50,000 parameter sets and evaluated the model at each of the parameter sets, setting all of the σ^2's to 1, which is effectively using the mean squared error as a measure of how good a parameter set was.

We then performed a second round of LHS around each of the ten best points found in round 1. For each of top-ten point $(\mu_1, \ldots, \mu_{26})$ (the 27th parameter being ω), we applied LHS on the Cartesian product, $j = 1, \ldots, 26$, of the parameter intervals

$$\left(F_j^{-1}\left(\max\left\{F_j(\mu_j) - \epsilon, 0\right\}\right), F_j^{-1}\left(\min\left\{F_j(\mu_j) + \epsilon, 1\right\}\right)\right)$$

where $F_j(\cdot)$ is the prior cumulative distribution function of parameter j, and we take $\epsilon = 0.025$.

From the best 49 points from the second round, plus the point representing the parameters that Blanchard et al (2014) found, we optimised using a Nelder-Mead algorithm (Nelder and Mead, 1965), capping the number of model runs in order to keep the computational effort down.

5.3.3 Implementation of MCMC

We ran an MCMC algorithm starting from each of the 50 points found from the optimisation step with $\delta t = \frac{1}{4}$ and found that, although only 3 of them found regions that had non-negligible posterior mass, these chains were exploring disjoint regions and a standard MCMC chain would find it difficult to move between these regions. We used a parallel tempering algorithm that allowed us to move between the regions and explore the posterior distribution. We set it up so that we had 5 temperatures with $\tau_1 = 1$ and $\tau_5 = 0.01$ so that we did not move too far away from the non-negligible mass. Each temperature had a stationary distribution

$$\frac{1}{C_i}\pi(\boldsymbol{\mu}, \sigma^2) l(\boldsymbol{\mu}, \sigma^2 | \boldsymbol{x})^{\tau_i}$$

where C_i is the normalising constant for temperature i and $\boldsymbol{\mu}$ is the set of $\psi, b_0, \xi, \rho, \omega$.

We used doubly parallel tempering, described in Section 3.2.3, in order to explore the posterior distribution. This enabled us to explore the space quite well whilst taking advantage of parallel computing. In order to improve the mixing of the algorithm, when proposing an exchange we performed a Gibbs step assuming the swap would be accepted, so that σ^2 would move to $\sigma^{2'}$, and then accepted or rejected the swap and the Gibbs step in one. This meant that the acceptance probability of a swap was

$$\min\left(1, \frac{\pi(\boldsymbol{x}|\boldsymbol{\mu}_j, \sigma_j^{2'})^{\tau_i}\pi(\boldsymbol{x}|\boldsymbol{\mu}_i, \sigma_i^{2'})^{\tau_j} p(i,j) p(\sigma_i^2|\boldsymbol{\mu}_i, \boldsymbol{x}, \tau_i) p(\sigma_j^2|\boldsymbol{\mu}_j, \boldsymbol{x}, \tau_j)}{\pi(\boldsymbol{x}|\boldsymbol{\mu}_i, \sigma_i^2)^{\tau_i}\pi(\boldsymbol{x}|\boldsymbol{\mu}_j, \sigma_j^2)^{\tau_j} p(j,i) p(\sigma_i^{2'}|\boldsymbol{\mu}_i, \boldsymbol{x}, \tau_j) p(\sigma_j^{2'}|\boldsymbol{\mu}_j, \boldsymbol{x}, \tau_i)}\right).$$

The MCMC algorithm is summed up in Algorithm 17.

Algorithm 17 One iteration of the doubly parallel tempering algorithm used to sample from the posterior distribution.

```
Given current values $[\boldsymbol{\mu}_k]_{k=1}^5$, $\left[\sigma_k^2\right]_{k=1}^5$, temperatures $[\tau_k]_{k=1}^5$, exchange rates $[s_k]_{k=1}^5$ and the acceptance rates, $a_k$, over the last $n$ Metropolis Hastings updates.
if $\sum_{k=1}^5 s_k \leq 20$ then
    $P_k \leftarrow s_k \forall k$
else
    for $k = 1 \ldots M$ do
        $\chi_1 \leftarrow \sum_{k=1}^5 \mathbb{I}(s_k \geq 1)$
        $P_k \leftarrow$
            $$\mathbb{I}(s_k \geq 1) + \left\lfloor \left( \chi_1 \times \frac{\frac{1}{a_k}}{\sum_{i=1}^5 \frac{1}{a_k}} + \alpha \right) \right\rfloor$$
    {$\alpha$ is adjusted so that $\sum_{k=1}^5 P_k = 20$.}
    end for
end if
for $k = 1 \ldots 5$ do
    Generate $P_k$ candidate points $\boldsymbol{\mu}'_{1:P_k} \sim q_k(\cdot|\boldsymbol{\mu}_k)$
    Generate $P_k - 1$ candidate points $\sigma^{2'}_{1:P_k-1} \sim p(\cdot|\boldsymbol{\mu}_j, \boldsymbol{x}, \tau_k)$
    $\sigma_0^{2'} \leftarrow \sigma_k^2$
    $j \leftarrow 0$
    $t \leftarrow 1$
    while $j < P_k$ do
        $j \leftarrow j + 1$
        $\alpha' \leftarrow \min\left(1, \frac{\pi(\boldsymbol{x}|\boldsymbol{\mu}'_j\sigma^{2'}_{j-1})^{\tau_k} q(\boldsymbol{\mu}_k|\boldsymbol{\mu}'_j)}{\pi(\boldsymbol{x}|\boldsymbol{\mu}_k,\sigma^{2'}_{j-1})^{\tau_k} q(\boldsymbol{\mu}'_j|\boldsymbol{\mu}_k)}\right)$
        Sample $u \sim \mathrm{U}(0, 1)$
        if $u < \alpha'$ then
            $\boldsymbol{\mu}_k \leftarrow \boldsymbol{\mu}'_j$
            $j \leftarrow P_k$
        else
            $t \leftarrow t + 1$
        end if
    end while
    $\sigma_j^2 \sim p(\cdot|\boldsymbol{\mu}_j, \boldsymbol{x}, \tau_k)$
    Update $a_k$ so that it is the proportion of the last $n$ proposed Metropolis Hastings moves accepted moves for chain $k$.
    $s_k \leftarrow s_k - t$
end for
```

```
while $\sum_{k=1}^{M} \mathbb{I}_{s_k=0} > 1$ do
    $i, j \sim h(\cdot, \cdot)$ {Sample $i$ and $j$ from the distribution $h(i, j)$ where $s_i = s_j = 0$.}
    $\sigma_i^{2'} \sim p(\cdot|\boldsymbol{\mu}_i, \boldsymbol{x}, \tau_j)$
    $\sigma_j^{2'} \sim p(\cdot|\boldsymbol{\mu}_j, \boldsymbol{x}, \tau_i)$
    $g \leftarrow \min\left(1, \frac{\pi(\boldsymbol{x}|\boldsymbol{\mu}_j, \sigma_j^{2'})^{\tau_i} \pi(\boldsymbol{x}|\boldsymbol{\mu}_i, \sigma_i^{2'})^{\tau_j} p(i, j) p(\sigma_i^2|\boldsymbol{\mu}_i, \boldsymbol{x}, \tau_i) p(\sigma_j^2|\boldsymbol{\mu}_j, \boldsymbol{x}, \tau_j)}{\pi(\boldsymbol{x}|\boldsymbol{\mu}_i, \sigma_i^2)^{\tau_i} \pi(\boldsymbol{x}|\boldsymbol{\mu}_j, \sigma_j^2)^{\tau_j} p(j, i) p(\sigma_i^{2'}|\boldsymbol{\mu}_i, \boldsymbol{x}, \tau_j) p(\sigma_j^{2'}|\boldsymbol{\mu}_j, \boldsymbol{x}, \tau_i)}\right)$
    Sample $u \sim U(\cdot|0, 1)$
    if $u < g$ then
        $\phi \leftarrow \boldsymbol{\mu}_t^i$
        $\boldsymbol{\mu}_t^i \leftarrow \boldsymbol{\mu}_t^j$
        $\boldsymbol{\mu}_t^j \leftarrow \phi$
        $\sigma_i^2 \leftarrow \sigma_i^{2'}$
        $\sigma_j^2 \leftarrow \sigma_j^{2'}$
    end if
end while
```

5.4 Results

We found that the posterior was divided into three different modes with only the best three points from the optimisation round leading to regions where there was non-negligible posterior density. This suggests that the parallel tempering algorithm worked quite well. We had an acceptance rate of 0.05 for the temperature with $t_i = 1$ and we found that it mixed quite well. We also had desired acceptances for other temperatures that also mixed well. The algorithm took a week on 20 cores to run 42,000 iterations. We discarded the first 10,000 iterations as burn-in.

5.4.1 Violin plots

The figures that follow show the sampled posterior distributions plotted using violin plots (Hintze and Nelson, 1998). A violin plot a is a mix between a density plot and a box plot. If we have a sample $w_{1:n}$ from the distribution $f(\cdot)$, then

$$\tilde{f}(u) = \frac{|\Omega|}{nh}$$

where $\Omega = \{w : |w - u| \leq h/2\}$ and h is known as the interval width is an estimate of the density of $f(\cdot)$. $\tilde{f}(\cdot)$ is then used as the width of the violin in the plot.

5.4.2 Recruitment parameters

The marginal posterior densities were multi-modal are shown in Figure 5.3. There does not seem to be a relationship between R_0 and size as suggested by Andersen and Pedersen (2009) and Andersen and Beyer (2013).

5.4.3 Burn in parameters

Some of the posterior distributions of the burn in parameters are not too dissimilar to their respective prior distributions, suggesting that the fishing effort in the burn in period is not that important to the model output. However some posterior distributions are very different, namely those of Sandeel and of Norway Pout.

5.4.4 Variance parameters

We found that the SSB estimates from the model for Sandeel and Whiting were poor as they had reasonably high σ^2 values. Whereas the SSB for the others was reasonably good which is shown in Figure 5.4. The landings σ^2 values for Sprat, Norway pout, Gurnard and Haddock are also quite high. The variance parameters and their uncertainties for the landings are shown in Figure 5.5.

5.4.5 Norway pout fishing in 2005 and Phase parameter

We found that the fishing effort for Norway pout in 2005 is very much multimodal as shown in Figure 5.6. It shows that three distinct regions of posterior density. The parameter, ω, does not feature in the posterior distribution as the model runs with parameters sampled from the posterior distribution do not have any cyclic behaviour in their stationary distributions hence the posterior distribution of ω is its prior distribution.

5.4.6 Residuals

We re-ran the model for 2,500 parameter sets sampled from the posterior distribution and examined their standardised residuals. We found that for all of the outputs as a whole, the residuals are roughly Gaussian. Which suggests that the model is able to recreate the average of the system.

Figure 5.7 shows the range of residuals of each of the 9 SSB outputs with the point representing the mean. If the model was a good estimate of the data there would be no visible patterns in the ranges. However there does seem to be a trend in the results. This trend continues in Figure 5.8 for the residuals of the landings. It is interesting to see that in both outputs the dynamic trends in residuals are similar. This suggests that treating SSB and landings as independent is not the right thing to do, as this means that

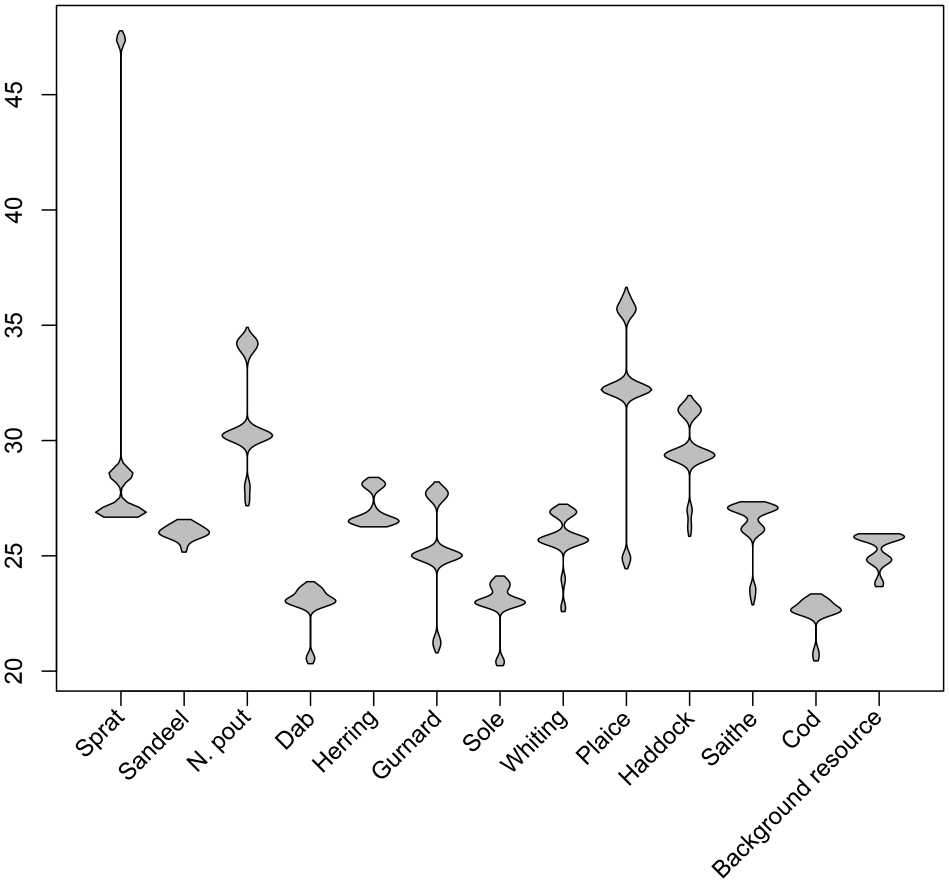

Figure 5.3: The posterior distribution of $\psi_{1:12}$ and b_0.

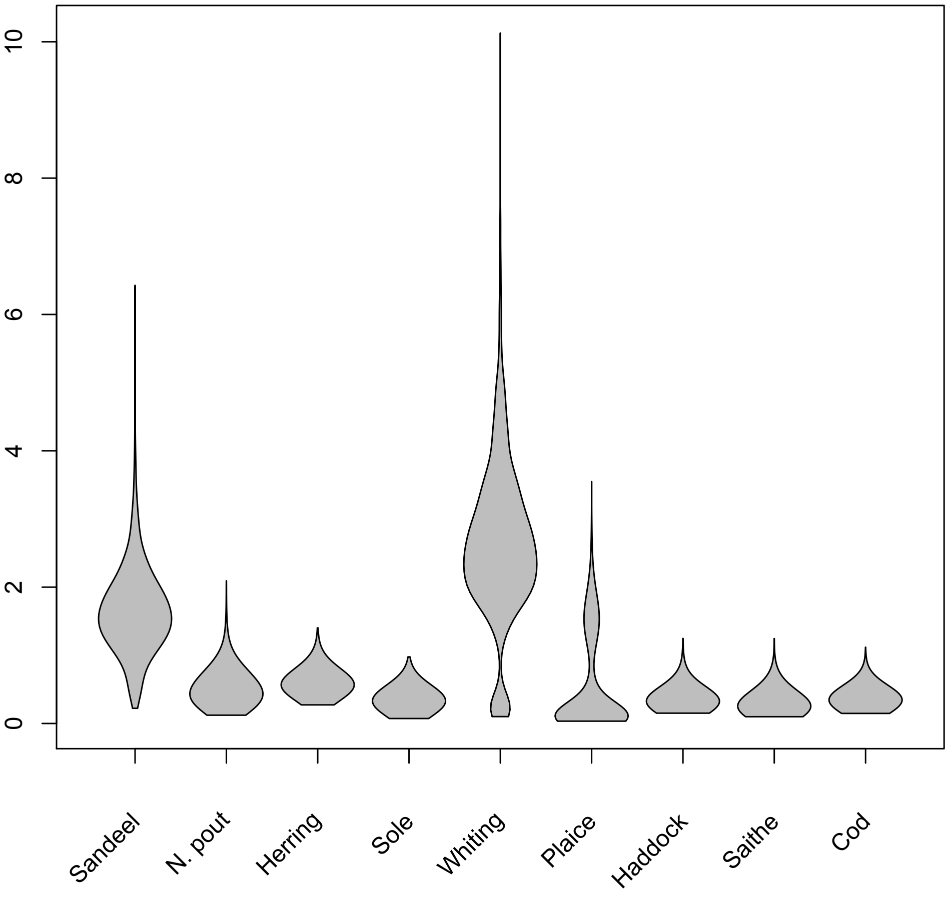

Figure 5.4: The variance parameters for the SSB output.

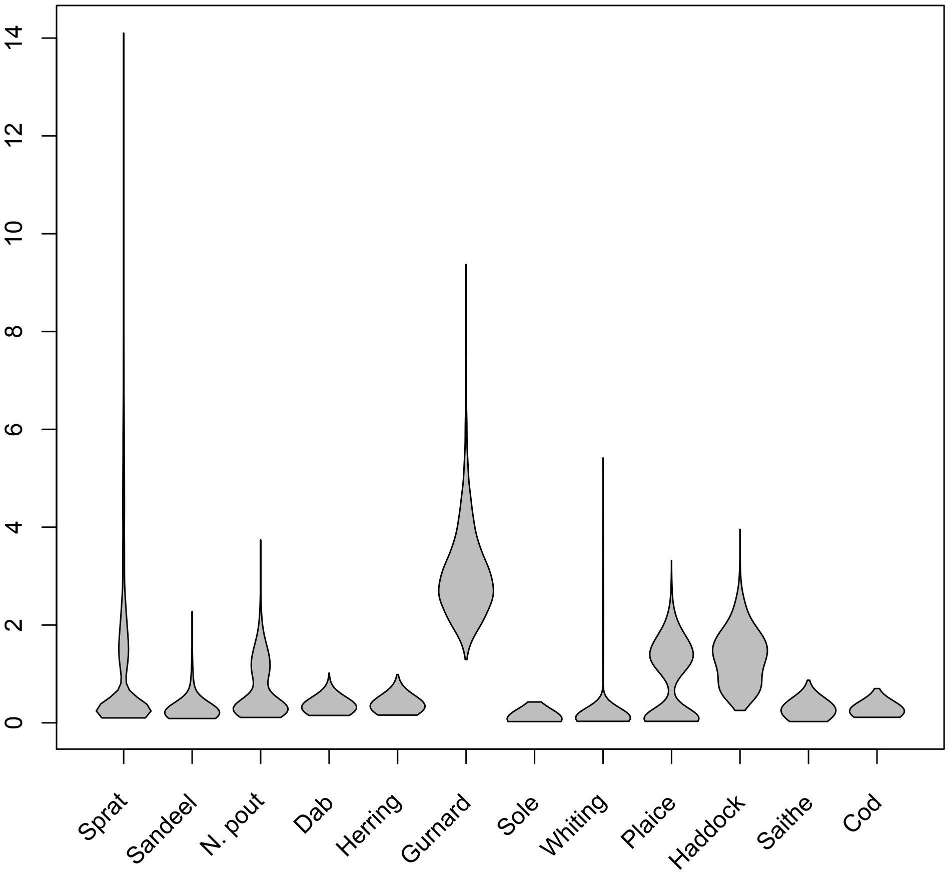

Figure 5.5: The variance parameters for the landings output.

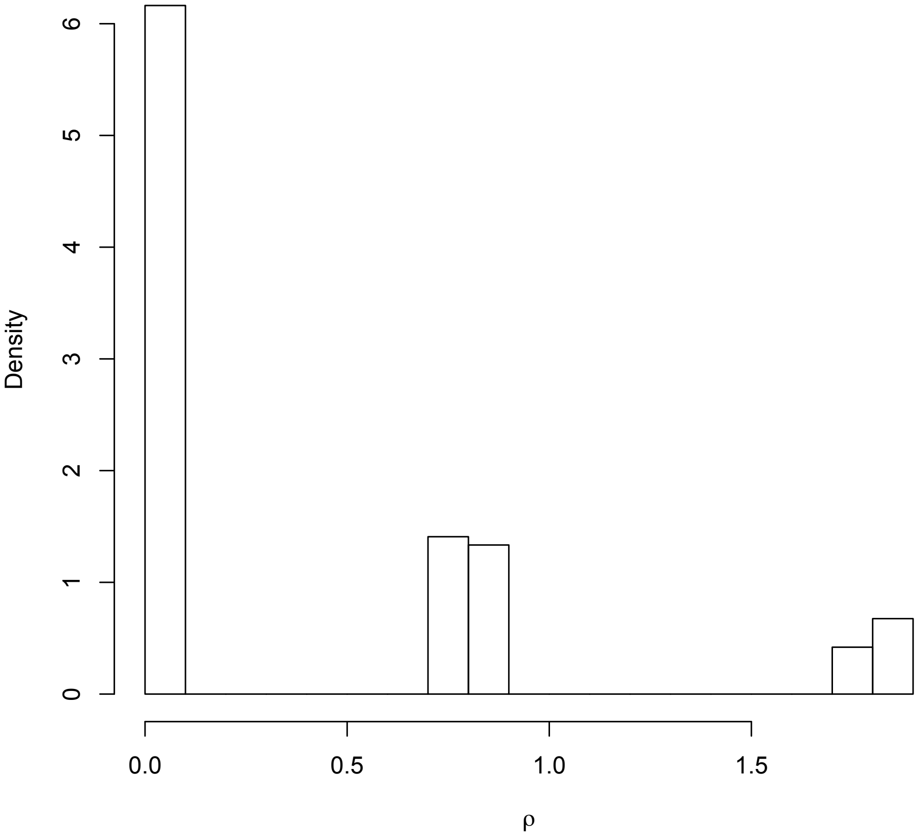

Figure 5.6: The marginal posterior distribution for ρ.

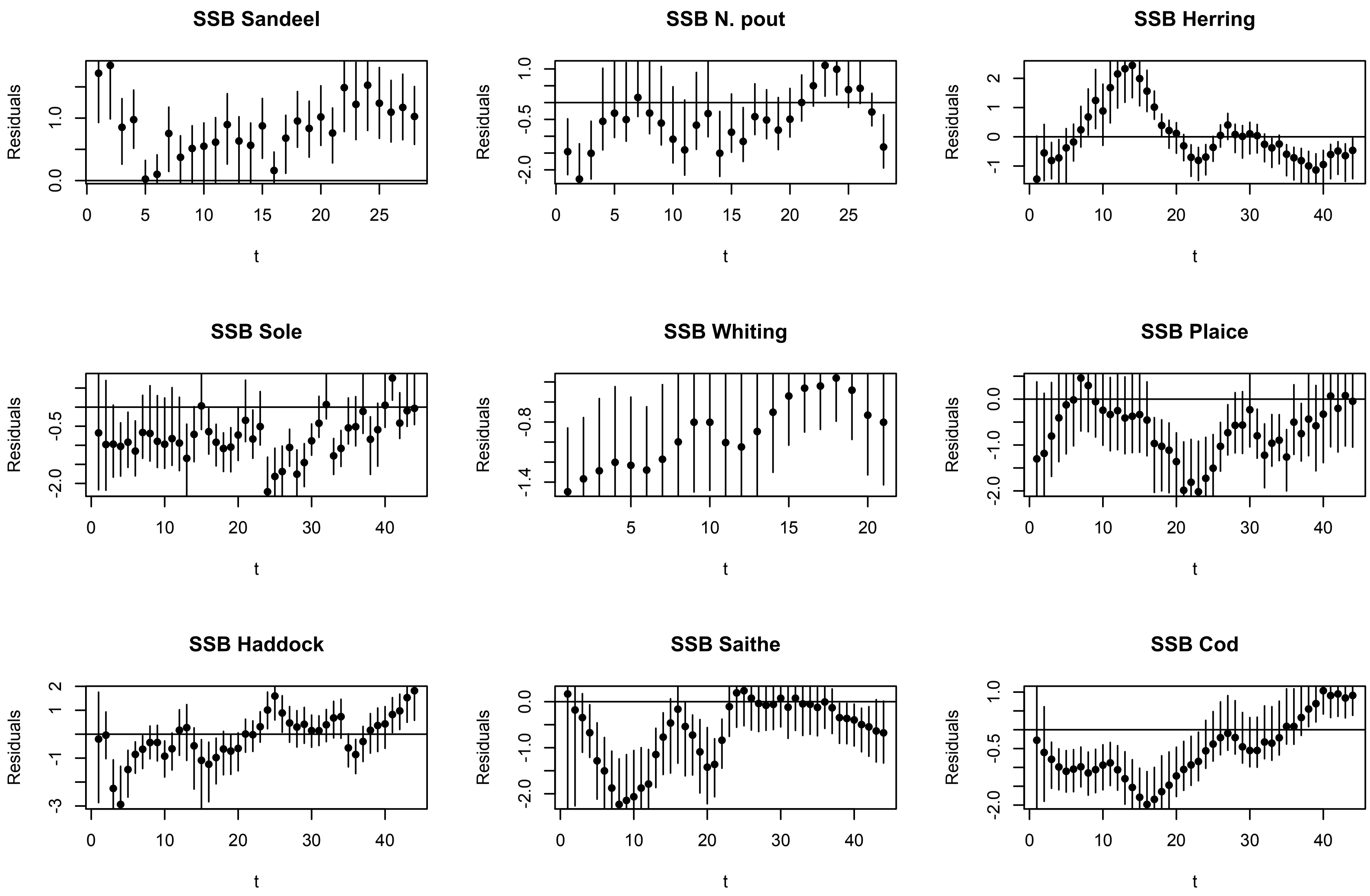

Figure 5.7: The ranges of residuals for the SSB. There does seem to be some dynamic behaviour in the errors.

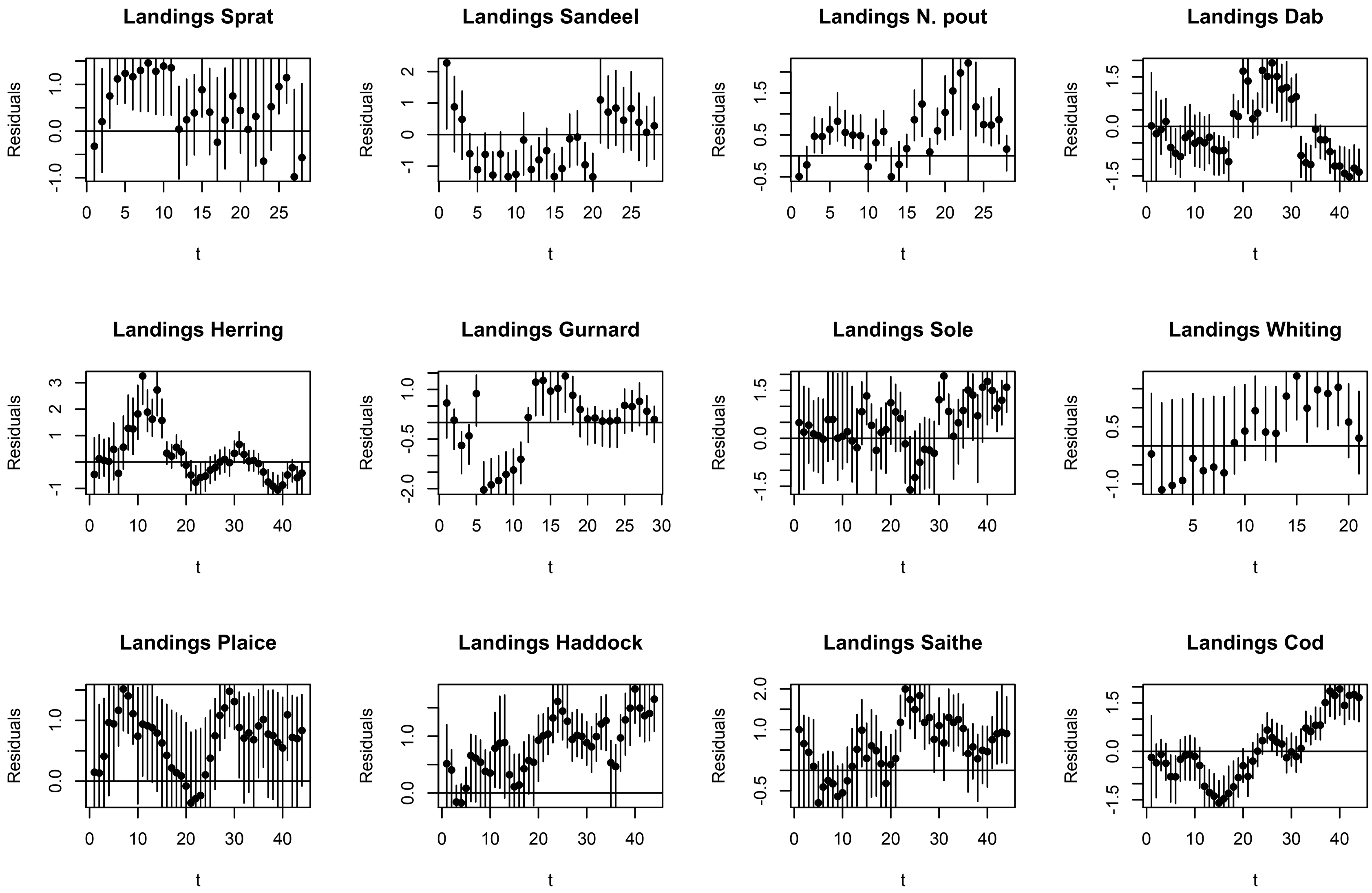

Figure 5.8: The ranges of residuals for the landings. It can be seen that although the overall errors seem normally distributed and therefore estimate the mean well, there does seem to be some dynamic behaviour in the errors.

we are underestimating the uncertainty in the parameters. We decided not to fit to the SSB data set and to just fit to the landings data set.

5.5 Analysis with landings only

We assumed that the best points at the end of the optimisation round were the similar with and without SSB as one implied the other so we returned to these points. We ran MCMC chains from these points and found that only one point had non-negligible posterior density.

We also returned to the end of the second round and re-ran the optimisation algorithm using only the sum of squared errors for the landings as a measure of how good a parameter set was. This resulted in finding 50 new points. We ran MCMC chains from these points and found that only three points lead to regions where there was non-negligible posterior density.

We ran doubly parallel tempering as before with these 4 points plus the 5th best point from the previous round being starting points two of which seemed to be in the same mode of the posterior. We had an acceptance rate of 0.1 for the target temperature and we found that it mixed quite well. We also had desired acceptances for other temperatures that also mixed well. The algorithm took a week on 20 cores to run 60,000 iterations. We discarded the first 10,000 as burn in.

5.5.1 Recruitment parameters

We found that the marginal posteriors for the recruitment parameters are unimodal; summaries are shown in Figure 5.9. Once again, the ranked maximum recruitment does not seem to have a relation with the asymptotic size as suggested in Andersen and Pedersen (2009) and Andersen and Beyer (2013).

5.5.2 Burn-in parameters

As in Section 5.4.3, some of the posterior distributions of the burn-in parameters are not too dissimilar to their respective prior distributions, suggesting that the fishing effort in the burn-in period is not that important to the model output. This is definitely the case with Gurnard.

5.5.3 Variance parameters

We found that the variance parameters are reasonably small, as shown in Figure 5.10, which suggests that the model is doing a reasonable job of recreating reality. We found that the model did a good job of estimating the yield for Sole, Whiting, Plaice and Saithe. It is particularly bad at

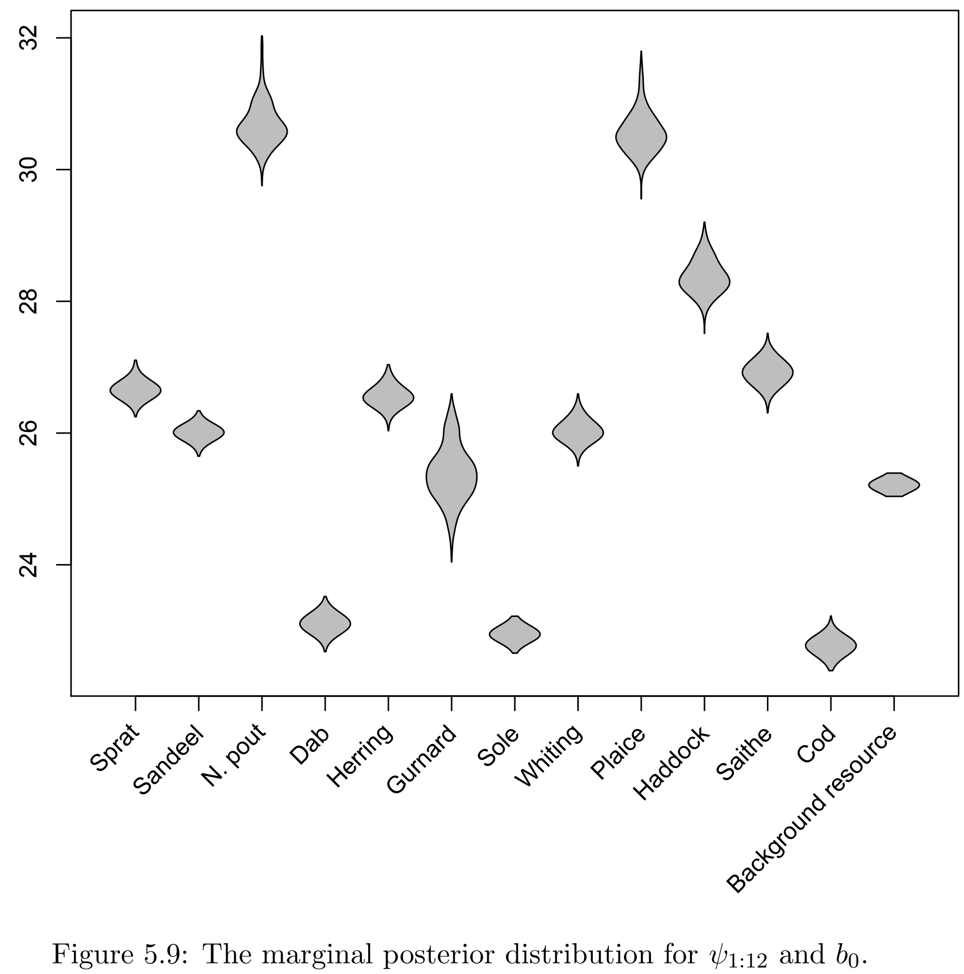

Figure 5.9: The marginal posterior distribution for $\psi_{1:12}$ and b_0.

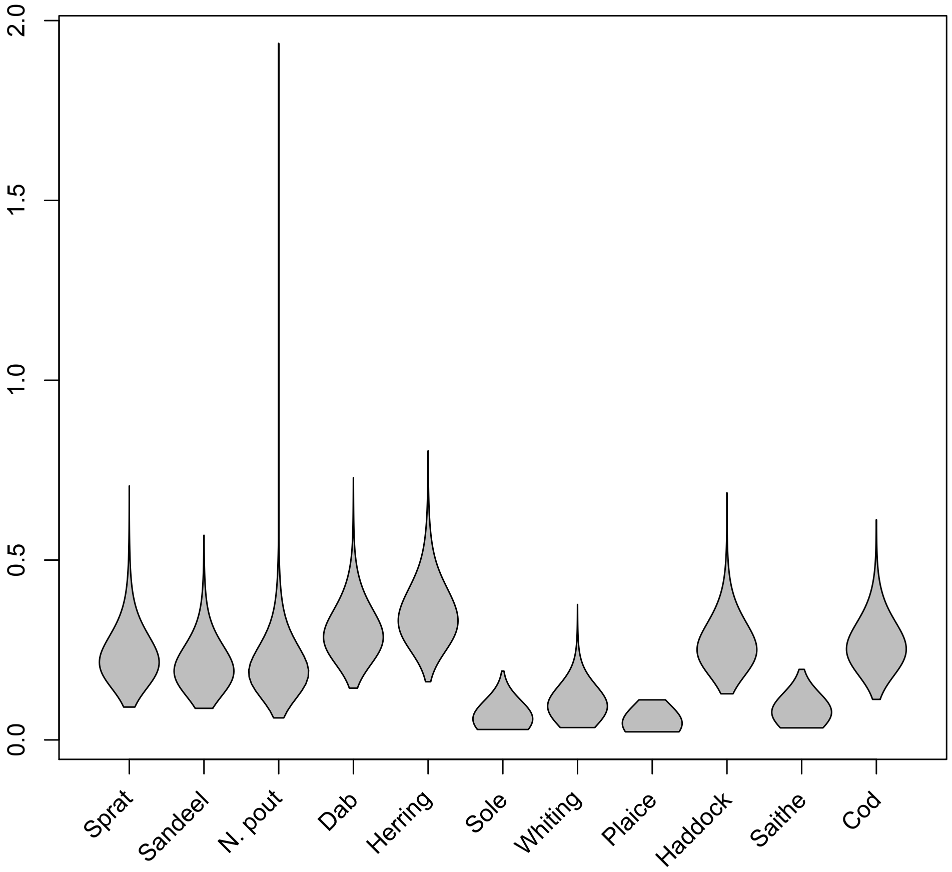

Figure 5.10: The marginal posterior for the error parameters for all but Gurnard.

estimating Gurnard which is omitted from Figure 5.10 because it is too big to plot on the same scale.

5.5.4 Norway pout fishing in 2005 and Phase parameter

We found that the posterior mean fishing effort for Norway pout in 2005 is about 0.013. This confirms our suspicion that the fishing effort for Norway Pout in 2005 is a rounding error. The posterior of the phase parameter, ω, is very similar to the prior of this distribution. It turns out that we didn't get any cyclic behaviour for any of the parameter sets sampled in the posterior distribution.

5.5.5 Model output

We have examined the data and how they compare to the output in Figure 5.11. For many of the species, the model does a good job of fitting the dynamics of the data. This is definitely the case for Norway Pout, Herring and Sandeel. However for others it only seems to get the average output namely for Cod, Gurnard and Dab.

5.5.6 Residuals

Figure 5.12 shows a histogram of the standardised residuals of the model output with the curve showing the standard normal density. The comparison of the residuals and the curve itself would match if our assumption, about the residuals being Gaussian on the log scale, were reasonable which does seem to be the case. However if we look at the residuals over time, as in Figure 5.13, we can see that there seems to be some structure to the residuals. Looking at Cod, at the beginning of the simulation we seem to under-estimate Cod and as time goes on we over-estimate it. The correlation between residuals can be seen in almost all of the species.

5.6 Discussion

Uncertainty is very important for reporting to policy makers (Harwood and Stokes, 2003) as we may need to assess the risk of an event should some policy be made. Therefore it is important to be able to quantify this risk in a robust and accurate way. We have demonstrated a way of performing parameter estimation and uncertainty that would enable us to forward simulate and make predictions of what might happen in the future. We have shown a way of probabilistically relating the model to reality and have estimated parameters that represent the uncertainty in the model.

The results presented here are an estimate of the posterior distribution of the North Sea fishing model. We are unable to say that what we have

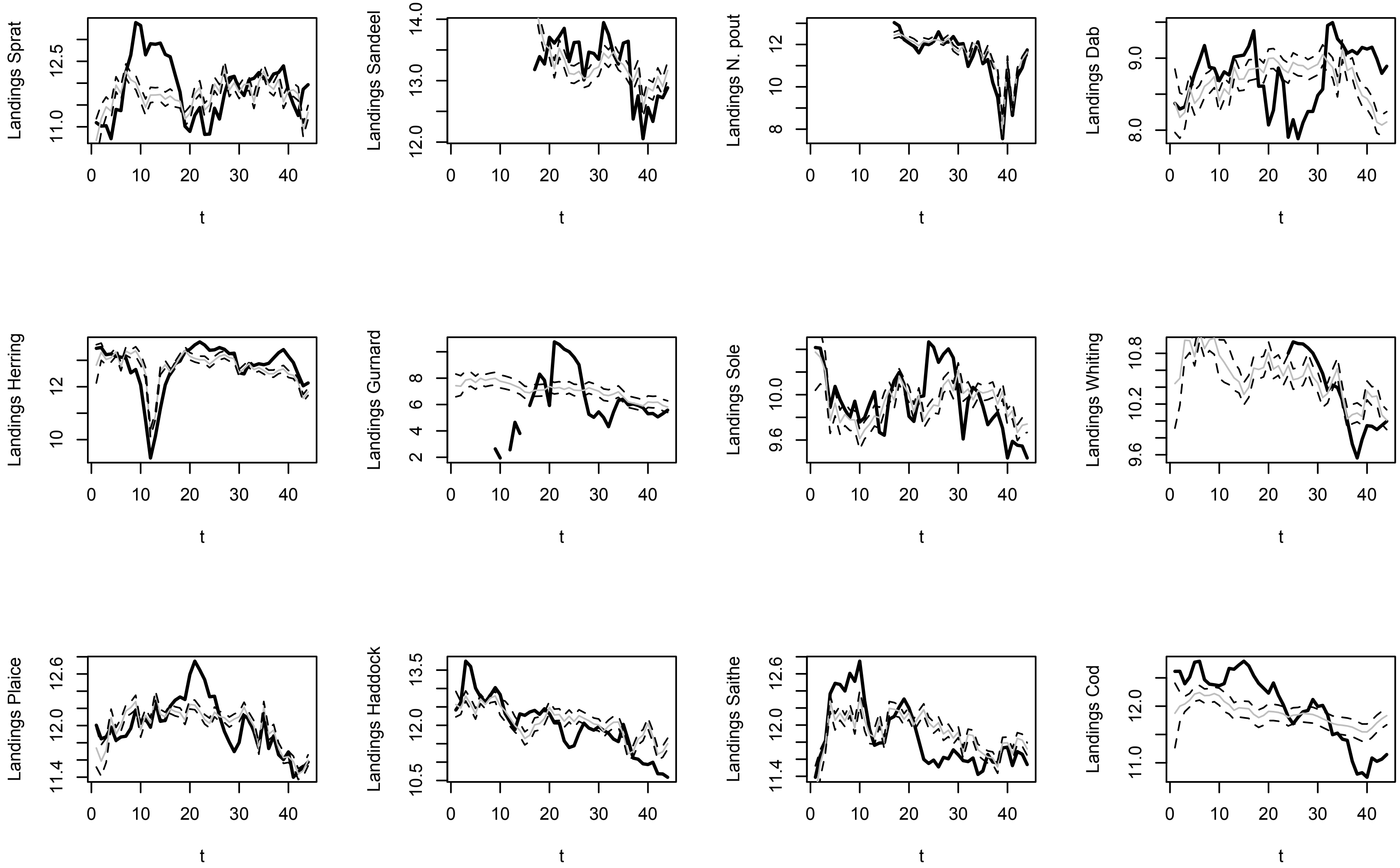

Figure 5.11: Runs of the model with parameters sampled from the posterior distribution. The grey line shows the median model output, the dotted lines are the 5th and 95th percentiles for the model output and the thick black line is the observed landings.

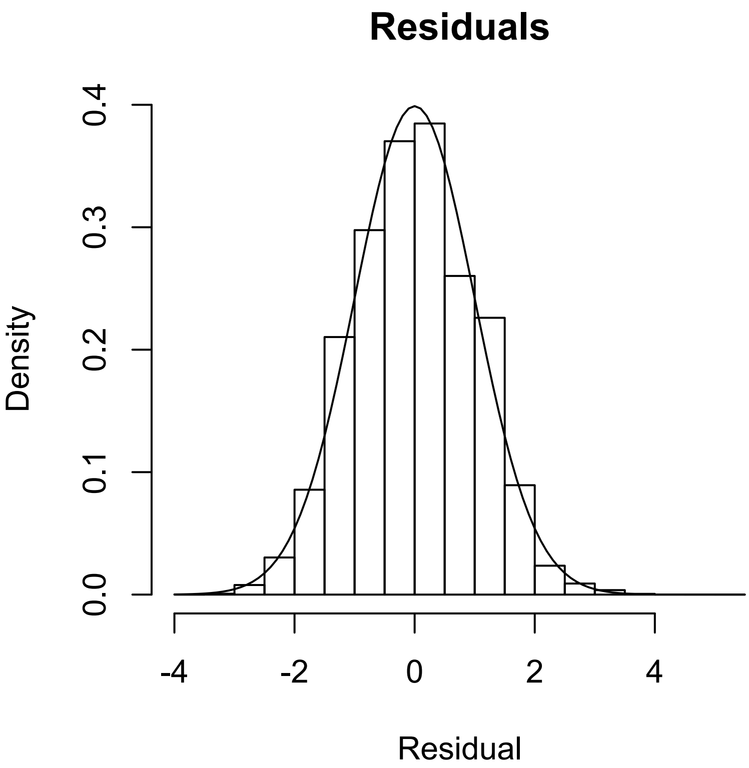

Figure 5.12: Histogram of standardised residuals.

found is the true posterior distribution but only part of it. We suspect that the true posterior is very multimodal and some of these modes were not even explored because it is difficult to explore the whole of the space. Initially we were exploring 26 inputs. If we took the maximum and the minimum of each input and searched each combination of these, essentially exploring the corners of the prior distribution[1] there would be 67,108,864 points. This is just searching the extremes of the distribution which is probably not the regions that we are interested in. One way of improving the posterior distribution would be to uses more informative priors. This could be done by eliciting the parameters (O'Hagan et al, 2006) or using simpler, more tractable models in order to produce priors for example single species models (Andersen and Beyer, 2013).

Having said all of this we have explored the parameter space in a strategic

[1] Something that is actually impossible due to the fact that in some directions, the priors are unbounded.

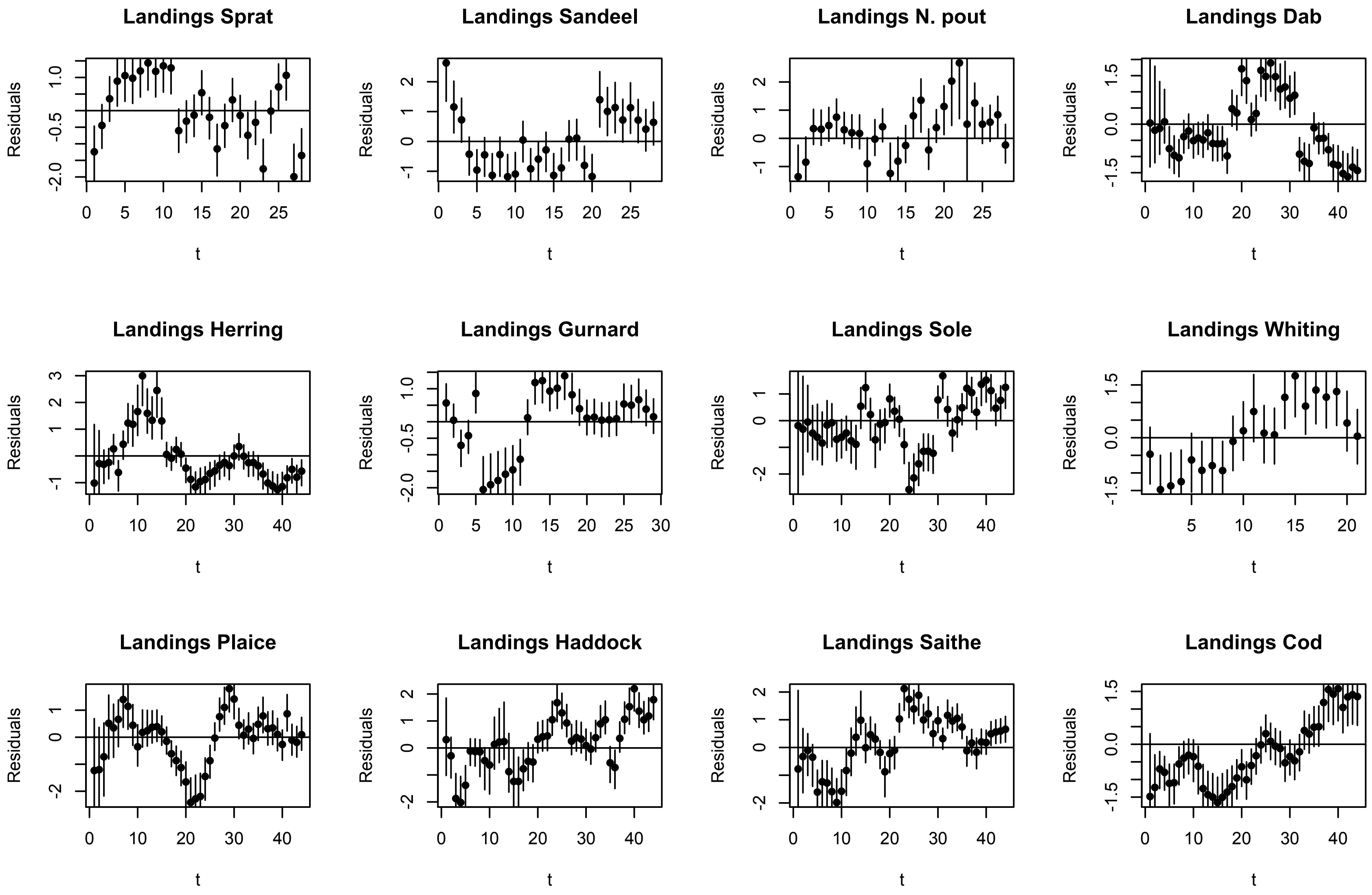

Figure 5.13: Runs of the model with the residuals shown. The mean residual is shown by the point and the 95th and 5th percentiles are shown by the length of each line.

way in order to try and explore as much of the space as possible. Our method has greatly improved things. The best point found in the first round, the initial latin hypercube search, had a log likelihood of -13790.19. In the MCMC round, the best point from the sampled posterior had a log likelihood of -322.08. We are quite happy that the method described here gives a good estimate of the posterior distribution however we are not ruling out the possibility that if we ran the method again there may turn out to be a much better region.

The work presented here gives a way of relating the model output to reality. We have taken into account parameter uncertainty and shown that the model does well at estimating the long term average of the real world. However it doesn't estimate the dynamics of the system in the short term as well. This was mentioned in Blanchard et al (2014) where they used stochasticity in their model to try and estimate the model uncertainty whereas we are using noise on the model output and parameter uncertainty to estimate the error in the model.

This leads to asking the question, what can we do to try and make the model a good estimate of reality? If we look at cod, we can see that at first we underestimate the landings and then over time we over estimate the landings. We believe either that the estimates of the fishing effort are poor or that $R_{0.i}$ changes over time, possibly as a function of temperature (Bigg et al, 2008). Both these could be estimated by taking a time varying latent state approach.

Furthermore, the assumption that $\delta(\theta) = 0$, i.e. the model discrepancy equals 0, could be incorrect. This is suggested by the fact that the residuals do suggest normality but we are able to see a structure in them over time which suggests some bias in the model. Additionally, the fact that all of the modes found in Section 5.4 result in similar patterns of residuals suggests that this could be corrected by a discrepancy term and is not just a feature of this region in parameter space.

Although there were no oscillations found in the posterior distribution, we did find them in earlier runs and they may be encountered at other regions in the posterior distribution. We presented here a method of removing the arbitrariness of the burn-in period. We demonstrated a way of identifying the phase of the cycle and then treating where with in the cycle a model run begins as an additional parameter.

The variance parameters, σ^2, are describing the model error and the data error. We could interpret it as how reliable the data are. Before beginning this work we weren't overly confident with the data for a number of species, Landings for Sprat, Dab and Gurnard just to name a few. The results did show large variance parameters for the landings of Sprat and Gurnard. This does not necessarily mean that the data are wrong for these sets, but does indicate that the model and the data together are more poor estimates of reality for these species compared to the other species.

When examining complex models some of the uncertainty in the model is down to uncertainty in the data (Wilkinson, 2013; Harwood and Stokes, 2003). In this article we have assumed that at each time step the true catch is the data plus some Gaussian noise with zero mean. Although this assumption does not seem to be fully met, evident by the correlated residuals, it does lead to the question what is the correct error structure? From a pure statistical point, it was not the right thing to do to assume that the errors for one species were independent and identically distributed in this work. This could be counteracted by modelling the residuals as an autoregressive model of order 1 (AR(1)). In reality there are many reasons that there could be errors in landings, miss reporting, measurement or collecting error, just to name a few, and we do believe that for each year some of these effects are independently distributed.

The model took a long time to run and it might therefore be better to build an emulator (Kennedy and O'Hagan, 2001) for the model. However we have a lot of outputs, at least 21, and the model is very sensitive to changes in the parameters which means that a Gaussian process emulator would be difficult, if not impossible to build. It would be interesting though if we fitted the model to some summaries of the data, in a similar way to Wood's synthetic likelihood (Wood, 2010), rather than the data itself, whether or not we would find that these summaries would be much smoother to small changes in the parameters which may enable us to build an emulator and speed up the inference.

The doubly parallel tempering algorithm, described in Section 3.2.3, allows the model to be run with every parameter set. In the final inference, the parallel tempering, we ran the model 88,000 times. If run on a single core, this would mean that it would take roughly 20 weeks. However we could have parallelised it by running one temperature on one core, ie. 5 cores in total. This would take about 4 weeks but the doubly parallel tempering allowed the inference to be done in one week and was key to this work.

Chapter 6

Global Sensitivity Analysis

6.1 Introduction

A simulation model will never be able to fully represent the true system (Grimm and Railsback, 2005) as there will be a degree of uncertainty in the structure and the inputs of the model. This can be caused by simplifications of the model by the modeller, either because of a lack of knowledge or in order to keep the complexity of the model at a manageable level, or uncertainties in the inputs to the model (Beck, 1987; Berk et al, 2002; Clark et al, 2001; Katz, 2002; Li and Wu, 2006; Reckhow, 1994; Stott and Kettleborough, 2002; Grimm et al, 1996). Sensitivity analysis evaluates how the uncertainties in the model and its inputs affect the uncertainty in the output (Saltelli et al, 2004). Some of the uncertainties can be quantified and reduced (Katz, 2002) and some uncertainties are so small that they don't need to be worried about. If a modeller can find out which uncertainties have a large effect on the output of the model then they can spend more time and effort learning about these. On the other hand if a modeller can find out which inputs have a very little effect on the output then they can fix the inputs (Oakley and O'Hagan, 2004).

Suppose we have a model with uncertain inputs $x_1, x_2, \ldots, x_n$ and output y defined

$$y = f(x_{1:n}).$$

Local sensitivity analysis is used to see how small changes in individual inputs affect the output of the model (Railsback and Grimm, 2012). When the model can be explicitly written out, local sensitivity analysis can be performed by computing the partial derivatives (Saltelli et al, 2000)

$$\frac{\partial f(x_{1:n})}{\partial x_i}.$$

When this cannot be performed, which is often the case when it comes to complex computer models, an input is changed by a small amount around a nominal value, $\boldsymbol{x_0}$, whilst the other parameters keep the same values in order to estimate the partial derivatives (Railsback and Grimm, 2012). This can be seen as a particular case of a one-factor-at-a-time (OAT) approach (Saltelli et al, 2000).

The idea of fixing x_{-i} (all of the inputs except x_i) based on $\boldsymbol{x_0}$ and changing x_i in order to estimate local sensitivity to inputs seems very dependent on the value of $\boldsymbol{x_0}$ and doesn't consider the whole of the uncertain space. Instead it gives the sensitivity along the axis of the inputs, with $\boldsymbol{x_0}$ being the origin, which, except for linear models, is not generally typical of the whole of the input space.

There are global methods that assume the function $f(\cdot)$ is differentiable at all points in the input space, $p(x)$, and look at partial derivatives (Section 6.2) which estimate the partial derivatives with the nominal point being various points around the whole of the input space or

$$\int \left| \frac{\partial f(x_{1:n})}{\partial x_i} \right| dP(x)$$

where $dP(x)$ is integrated over the whole input distribution of $x_{1:n}$.

Another common global method, which has been used with IBMs, is that of Standardised Regression Coefficients (van Nes et al, 2002; Drechsler, 1998; Railsback et al, 2006; Maes et al, 2005). This is done by sampling from the input distributions and then running the model at these values. A regression model is fitted to the model outputs and inputs and then the standardised coefficients are measures of the sensitivity of the inputs. This works well when the model is approximately linear and monotonic but struggles if not (Cariboni et al, 2007).

Global methods estimate the effect on the output of the model when all of the inputs are varied across the whole of the uncertain input space (Cariboni et al, 2007). Typically methods of global sensitivity analysis are independent of their model and do not rely on the assumptions of linearity or additivity but are computationally expensive. One such method is variance-based sensitivity analysis where the variance is decomposed which enables us to calculate how much each input contributes to the uncertainty of the output.

The rest of the chapter is broken down by first describing screening methods that are usually done before more quantitive methods are used, in Section 6.2, and then variance-based methods are reviewed in Section 6.3. This chapter is a review of current methods and can be skipped for an expert in the area. The methods described here are used and extended in Chapter 7.

6.2 Screening methods

Often it is the case that there are a large number of inputs of which only a small subset have a non-negligible effect. It is then important to screen out these parameters that have negligible effect on the input. This is often done before more accurate but more expensive sensitivity methods are used (Cariboni et al, 2007; Morris et al, 2014).

6.2.1 Morris Method

The Morris method (Morris, 1991; Campolongo et al, 2007) is a OAT method but is done over the whole space and therefore can be described as a method of global sensitivity analysis. It enables classification of the inputs into inputs with negligible effects, inputs with linear effects and inputs with non-linear and/or interaction effects (Iooss and Lemaître, 2015). r points in the input space are evaluated across the whole input distribution and then the elementary effect of the ith input at the jth point are

$$E_i^{(j)} = \frac{f(x_{-i}^j, x_i^j + \Delta) - f(x^j)}{\Delta},$$

The mean absolute value and standard deviation of the elementary effects are estimated by

$$\begin{aligned} \mu_i^* &= \frac{1}{r}\sum_{j=1}^{r} |E_i^{(j)}| \quad \text{and} \\ \sigma_i^* &= \sqrt{\frac{1}{r}\sum_{j=1}^{r}\left(E_i^{(j)} - \frac{1}{r}\sum_{j=1}^{r} E_i^{(j)}\right)^2}. \end{aligned}$$

μ_i^* gives the effect of the input i and σ_i^* is a measure of non-linearity and/or interactions with other parameters.

6.2.2 Derivative based global sensitivity

A generalisation of this method is derivative based global sensitivity (Sobol' and Kucherenko, 2009, 2010) defined by

$$\nu_i = \int \left(\frac{\partial f}{\partial x_i}\right)^2 dP(x).$$

It can be seen that if $c \leq |\partial f / \partial x_i| \leq C$ then

$$\frac{\sigma_i c^2}{Var(y)} \leq \overline{\tau}_i^2 \leq \frac{\sigma_i C^2}{Var(y)}$$

where $\overline{\tau}_i^2$ is the total sensitivity index and is defined in equation 6.8 below and $\sigma_i = Var(x_i)$ (Sobol' and Kucherenko, 2009). If $f(x)$ is non-monotonic then $c = 0$ as somewhere $\partial f / \partial x_i = 0$ hence

$$\overline{\tau}_i^2 \leq \frac{\sigma_i C^2}{Var(y)}$$

and therefore if $\frac{\sigma_i C^2}{Var(y)}$ is small, the total sensitivity caused by input i is also small. Sobol' and Kucherenko (Sobol' and Kucherenko, 2009) also showed that if the input distributions of $x_1, \ldots, x_n$ were a n dimensional hypercube (which can be all distributions as shown in Proposition 1 in Chapter 4), then

$$\overline{\tau}_i^2 \leq \frac{\nu_i}{\pi^2 Var(y)}.$$

These can be seen as indices in there own right and are justified by their relation to variance-based sensitivity as described in the next section. In practise people use the sample maximum as a value of C found when evaluating $|\partial f / \partial x_i|$ numerically.

6.3 Variance-based sensitivity

Variance-based global sensitivity is based on the law of total variance

$$Var(y) = Var_{x_i}\left(E_{x_{-i}}(y|x_i)\right) + E_{x_i}\left(Var_{x_{-i}}(y|x_i)\right).$$

This can be rearranged so that

$$V_i \equiv Var_{x_i}\left(E_{x_{-i}}(y|x_i)\right)$$

is the expected reduction in the variance of the output if you learned the exact value of X_i. Often this is standardised by

$$D_i \equiv \frac{Var_{x_i}\left(E_{x_{-i}}(y|x_i)\right)}{Var(y)}. \tag{6.1}$$

This is known as the first order sensitivity index (Saltelli et al, 2004), importance measure (Horan and Iman, 2008; Homma and Saltelli, 1996), correlation ratio (Krzykacz, 1990; McKay, 1996) or main effect index (Oakley and O'Hagan, 2004). The second order sensitivity index is defined

$$V_{ij} \equiv Var_{x_i,x_j}\left(E_{x_{-i,j}}(y|x_i,x_j)\right) - Var_{x_i}\left(E_{x_{-i}}(y|x_i)\right) - Var_{x_j}\left(E_{x_{-j}}(y|x_j)\right)$$

or in standardised form by

$$D_{ij} \equiv \frac{V_{ij}}{Var(y)}. \tag{6.2}$$

V_{ij} is the expected reduction in variance by learning both the true value of x_i and x_j and not the expected reduction of the first order effects i.e. learning x_i or x_j on their own. If the inputs, $x_{1:n}$ are independent then

$$Var(y) = \sum_i V_i + \sum_i \sum_{j>i} V_{ij} + \ldots + V_{123\ldots n}. \tag{6.3}$$

Homma and Saltelli (1996) introduced the total effect index, D_{Ti}, which is the proportion of the variance that is left if we learnt everything except x_i (Oakley and O'Hagan, 2004)

$$D_{Ti} \equiv 1 - \frac{Var_{-x_i}\left(E_{x_i}(y|x_{-i})\right)}{Var(y)}. \tag{6.4}$$

6.3.1 Sobol' Indices

Sobol' (1993) decomposed the function rather than the variance as

$$f(x) = f_0 + \sum_i f_i(x_i) + \sum_i \sum_{i<j} f_{ij}(x_i, x_j) + \ldots + f_{12\ldots k}(x_1, x_2, \ldots, x_n) \tag{6.5}$$

where

$$\int f_{i_1,\ldots i_s}(x_{i_1}, \ldots x_{i_s}) dP(x_{i_1}) \ldots dP(x_{i_s}) = 0.$$

In other words

$$E(f_{i_1,\ldots i_s}(x_{i_1}, \ldots x_{i_s})) = 0$$

and

$$\begin{aligned} E(f(x)) &= \int f(x) dP(x) \\ &= f_0. \end{aligned}$$

This means every term, apart from the first one, in equation 6.5 will have expectation 0.

The decomposition equations are unique (Efron and Stein, 1981), provided that the input parameters are orthogonal, and that the individual terms $f_{i_1,i_2,\ldots,i_s}$ are square integrable, which if $f(x)$ is then all of the individual terms are. Each individual term is orthogonal such that

$$\int f_{i_1,i_2,\ldots,i_k}(x_{i_1}, x_{i_2}, \ldots, x_{i_s}) f_{j_1,j_2,\ldots,j_l}(x_{j_1}, x_{j_2}, \ldots, x_{j_l}) dP(x) = 0 \tag{6.6}$$

which allows the decomposition to be found by

$$\begin{aligned} f_i(x_i) &= \int f(x) dP(x)_{-i} - f_0 \\ f_{ij}(x_{ij}) &= \int f(x) dP(x)_{-ij} - f_i(x_i) - f_j(x_j) - f_0. \end{aligned}$$

The variance of Y can be found by

$$\begin{aligned} Var(Y) &= E(Y^2) - (E(Y))^2 \\ &= \int (f(x))^2 dP(x) - \left(\int f(x) dP(x) \right)^2 \\ &= \int (f(x))^2 dP(x) - f_0^2. \qquad (6.7) \end{aligned}$$

Now from equation 6.5

$$\begin{aligned} (f(x))^2 = f_0^2 &+ f_0 \left(\sum_i f_i(x_i) + \sum_i \sum_{i<j} f_{ij}(x_i, x_j) + \ldots + f_{12\ldots k}(x_1, x_2, \ldots, x_k) \right) \\ &+ \ldots + f_{12\ldots k}(x_1, x_2, \ldots, x_k)^2. \end{aligned}$$

Plugging this into equation 6.7 the terms that are orthogonal will integrate to 0 because of equation 6.6 and then

$$Var(y) = \sum_i \int f_i(x_i)^2 dP(x) + \sum_i \sum_{i<j} \int f_{ij}(x_i, x_j)^2 dP(x) + \ldots + \int f_{12\ldots k}(x_1, x_2, \ldots, x_k)^2 dP(x).$$

This means that the variance is broken into the variance of the decomposition terms. Sobol' (1993) defined the total variance

$$V \equiv Var(y)$$

and the variance of the decomposition terms

$$V_{i_1,\ldots,i_s} = \int f_{i_1,\ldots,1_s}(x_{i_1}, \ldots, x_{i_s})^2 dP(x).$$

The ratios

$$D_{i_1,\ldots,i_s} = \frac{V_{i_1,\ldots,i_s}}{V}$$

are called global sensitivity indices or Sobol' indices. These indices are non-negative and sum to 1 and include as special cases the indices described in equations 6.1 and 6.2.

6.3.2 Variable importance

Sobol' (1993) defines two measures of the importance of subsets of variables. If now we let $i \subseteq \{1, 2, \ldots, k\}$, i.e. i (and also j) represents a subset of inputs, the closed sensitivity index is defined as,

$$\underline{\tau}_i^2 = \sum_{j \subseteq i} V_j,$$

and the total sensitivity index

$$\overline{\tau}_i^2 = \sum_{j \cap i \neq \emptyset} V_j. \tag{6.8}$$

It can be shown that $0 \leq \underline{\tau}_i^2 \leq \overline{\tau}_i^2 \leq V$ and that

$$\underline{\tau}_i^2 + \overline{\tau}_{-i}^2 = V$$

which can be useful for estimating these values. Sobol' (1993) shows that

$$\underline{\tau}_i^2 = \int f(x) f(x_i, x'_{-i}) dP(x) dP(x') - f_0^2$$

which can be written as

$$\begin{aligned} \underline{\tau}_i^2 &= E(f(x) f(x_i, x'_{-i})) - E(f(x))^2 \\ &= Cov(f(x), f(x_i, x'_{-i})); \end{aligned}$$

as well as

$$\overline{\tau}_i^2 = \frac{1}{2} \int (f(x) - f(x_{-i}, x'_i))^2 dP(x) dP(x'),$$

which can be written as

$$Var(f(x)) - Cov(f(x), f(x_{-i}, x'_i)).$$

Liu and Owen (2006) showed that we can calculate

$$V_i = \sum_{j \subseteq i} (-1)^{|j-i|} \underline{\tau}_j^2$$

and

$$V_i = \sum_{j \subseteq i} (-1)^{|j-i|} (V - \overline{\tau}_{-j}^2)$$

from these values. The closed sensitivity index, $\underline{\tau}_i^2$, describes the total contribution of i and all of the subsets of i to the variance; whereas the total sensitivity index, $\overline{\tau}_i^2$, describes the total contribution of the set of inputs i and all of their interactions with themselves and the other inputs to the variance (Fruth et al, 2014). Liu and Owen (2006) also define a superset importance as

$$\Upsilon_i^2 = \sum_{j \supseteq i} V_j,$$

which can be interpreted as the amount of variance that the higher order interactions of the set i contribute to the variance. It is possible to get the sensitivity index of the set i from the superset

$$V_i = \sum_{j \supseteq i} (-1)^{|j-i|} \Upsilon_j^2.$$

6.3.3 Estimation

The integrals above are often intractable so numerical methods are required in order to estimate the Sobol' indices. In this subsection we describe a number of different methods.

Pick and Freeze methods

These problems all have in common that they are trying to estimate the covariance of two random samples, $f(x)$ and $f(x_i, x'_{-i})$. One way of doing this would be

$$\widetilde{\underline{\tau}_i^2} = \frac{1}{n-1}\left(\sum_{m=1}^{n}(f(x^m) - \widetilde{f_0})(f(x_i^m, x'^m_{-i}) - \widetilde{f'_0})\right)$$

where we draw two samples x^m and x'^m from the inputs with $m = 1 \dots n$. This would lead to an unbiassed estimate of $\underline{\tau}_i^2$. However the variance of this estimate is much higher than it could be. Here we know that $E(f(x_i)) = E(f(x_i, x'_{-i}))$ and we could use this in order to better estimate the indices.

Sobol' (2001) suggested estimating the closed sensitivity index by

$$\widetilde{\underline{\tau}_i^2} = \frac{1}{n}\sum_{m=1}^{n} f(x^m) f(x_i^m, x'^m_{-i}) - \widetilde{f_0}^2 \qquad (6.9)$$

where

$$\widetilde{f_0} = \frac{1}{n}\sum_{m=1}^{n} f(x^m)$$

and total sensitivity index

$$\widetilde{\overline{\tau}_i^2} = \frac{1}{2n}\sum_{m=1}^{n}(f(x^m) - f(x^m_{-i}, x'^m_i))^2.$$

The estimate of $\widetilde{\underline{\tau}_i^2}$ is biased by $-var(\widetilde{f_0})$ (Owen, 2013c). Owen (2013c) gave an alternative method of estimating $\underline{\tau}_i^2$ that doesn't require any more runs of the model than the method described by Sobol' (2001). Owen proposed estimating

$$\begin{aligned}
\widetilde{f_0} &= \frac{1}{n}\sum_{m=1}^{n} f(x^m), \\
\widetilde{f_0}' &= \frac{1}{n}\sum_{m=1}^{n} f(x_i^m, x'^m_{-i}), \\
s^2 &= \frac{1}{n-1}\sum_{m=1}^{n}(f(x^m) - \widetilde{f_0})^2 \quad \text{and} \\
s^{2\prime} &= \frac{1}{n-1}\sum_{m=1}^{n}(f(x_i^m, x'^m_{-i}) - \widetilde{f_0}')^2,
\end{aligned}$$

and then

$$\widetilde{\underline{\tau}_i^2} = \frac{2n}{2n-1}\left(\frac{1}{n}\sum_{m=1}^{n} f(x^m)f(x_i^m, x_{-i}'^m) - \left(\frac{\widetilde{f_0} + \widetilde{f_0}'}{2}\right)^2 + \left(\frac{s^2 + s^{2\prime}}{4n}\right)\right)$$

is an unbiased estimator of $\underline{\tau}_i^2$.

Saltelli (2002) tried to maximise the number of estimates from a number of runs in order to minimise the model evaluations. If we sampled two input sets, x and x' as before and ran $f(x)$, $f(x')$, $f(x_i, x')$ and $f(x_i', x)$ for $i = 1 \ldots d$. From this it is possible to estimate all of the first order indices, all of the second order indices and the first order total sensitivities from the $n(2d+2)$ model runs. Furthermore the estimates of the first order indices were

$$\widetilde{\underline{\tau}_i^2}^{1} = \frac{1}{n}\sum_{m=1}^{n} f(x^m)(f(x_i^m, x_{-i}'^m) - f(x'^m)) \qquad (6.10)$$

which are an unbiased estimate (Kucherenko et al, 2011) of $\underline{\tau}_i^2$ (see also Proposition 3 in Chapter 7). The estimate of the mean squared is $\widetilde{f_0}\widetilde{f_0'}$ rather than $\widetilde{f_0}^2$ as in equation 6.9. Saltelli (2002) noticed that if you ran every possible combination there would be two estimates of $\underline{\tau}_i^2$, the one described in equation 6.10 and

$$\widetilde{\underline{\tau}_i^2}^{2} = \frac{1}{n}\sum_{m=1}^{n} f(x'^m)(f(x_i'^m, x_{-i}^m) - f(x^m)). \qquad (6.11)$$

The author suggested that the mean of these two estimates should be the estimate of $\underline{\tau}_i^2$. This is an unbiassed estimate of $\underline{\tau}_i^2$ and has a lower variance than equations 6.10 and 6.11 by half the difference of the covariance of $\widetilde{\underline{\tau}_i^2}^{1}$ and $\widetilde{\underline{\tau}_i^2}^{2}$ and the variance of the estimates (which will be the same and is shown in Proposition 3). Tarantola et al (2006b) took this one step further and realised that, as well as equations 6.10 and 6.11, $\underline{\tau}_i^2$ could be estimated by

$$\begin{aligned}
\widetilde{\underline{\tau}_i^2}^{3} &= \frac{1}{n}\sum_{m=1}^{n} f(x_i^m, x_{-i}'^m)(f(x^m) - f(x_i'^m, x_{-i}^m)) && (6.12)\\
\widetilde{\underline{\tau}_i^2}^{4} &= \frac{1}{n}\sum_{m=1}^{n} f(x_i'^m, x_{-i}^m)(f(x'^m) - f(x_i^m, x_{-i}'^m)). && (6.13)
\end{aligned}$$

The mean of all of these estimates is then an unbiassed estimate for $\underline{\tau}_i^2$ with a lower variance than when we have only two estimates. Glen and Isaacs (2012) showed that this method was better, in terms of variance of the estimate, than anything previously described in this chapter.

Owen (2013a) examined estimates of small sensitivity indices. The author found that if it were possible to sample the inputs a third time $z^{1:m}$, then

$$\widetilde{\underline{\tau}}_i^2 = \frac{1}{n}\sum_{m=1}^{n}(f(x^m) - f(z_i^m, x_{-i}^m))(f(x_i^m, x'^m_{-i}) - f(x'^m))$$

would give a much better estimate of the variance compared to the other correlation methods (described above) especially when the true index, $\underline{\tau}_i^2$, is small.

Rather than estimating the correlation, Liu and Owen (2006) estimated the superset importance and showed that

$$\Upsilon_i^2 = \frac{1}{2^{|i|}}\int\left(\sum_{j\subseteq i}(-1)^{|j-i|}f(x_j, x'_{-j})\right)^2 dP(x)_i dP(x') \tag{6.14}$$

which can be estimated by

$$\widetilde{\Upsilon}_i^2 = \frac{1}{n2^{|i|}}\sum_{k=1}^{n}\left(\sum_{j\subseteq i}(-1)^{|j-i|}f(x_j,{}^k x'^k_{-j})\right)^2.$$

This is an unbiased estimator and has the added property that it is non-negative. Furthermore, Fruth et al (2014) showed that if $\Upsilon_i^2 = 0$ then using the Liu and Owen estimator $\widetilde{\Upsilon}_i^2 = 0$. This method is useful at estimating lower order indices if d is small however this method becomes very expensive when d becomes larger as to estimate all of the first order indices you need to run the model in all $2^d n$ times.

Sampling schemes

The above methods can be implemented by using Monte Carlo sampling (Metropolis and Ulam, 1949). However there are many sampling schemes that are used rather than Monte Carlo sampling in order to estimate the Sobol' indices. Latin Hypercube sampling (McKay et al, 1979) and Quasi Monte Carlo techniques, such as Sobol' sequences (Sobol', 1967), that generate points that fill the space better than independent pseudo random numbers can improve the estimation of the sensitivity indices by a factor of ten (Saltelli et al, 2008).

Cukier et al (1978) developed a method of estimating the first order sensitivity indices named Fourier amplitude sensitivity testing (FAST) which was extended by Saltelli et al (1999) to be able to estimate total sensitivity indices. However, in practice there are a lot of sources of error and FAST could only be used in small dimensions (Tissot and Prieur, 2012). Tarantola et al (2006a) developed a version of FAST based on random balanced designs (Satterwaite, 1959), namely RBD-FAST, that over came some of the errors and enables the estimation of any order sensitivities (Mara, 2010). Tissot and Prieur (2012) then corrected for the bias of the RBD-FAST.

Meta Models methods

One method of estimating the Sobol' indices is to smooth the function and estimate $E_{-x_i}(y|x_i)$ and then use this to estimate $Var_{x_i}(E_{-x_i}(y|x_i))$ (Wainwright et al, 2014). Strong et al (2011) used a method of estimating $E_{-x_i}(y|x_i)$ using generalised additive models. The variance of the estimate of $E_{-x_i}(y|x_i)$, from the generalised additive model, is then an estimate of V_i. The advantage of this is that it just requires n evaluations in order to work out all of Sobol' indices. For a more detailed description of smoothing methods see Storlie and Helton (2008).

Sometimes models can be expensive to run or only a limited number of evaluations of the model may be available, then a less expensive meta model could be created Saltelli et al (2004). Oakley and O'Hagan (2004) built a Gaussian process emulator (see Section 3.4.3) and Kleijnen (2007) used methods of kriging to estimate the model output quickly and from these values they can estimate the sensitivity indices. In both these methods the Sobol' indices can be calculated analytically. Other methods of building meta models exist such as linear, non-linear parametric or non-parametric regression (Iooss and Lemaître, 2015) and simulation from the meta model allows estimation of the Sobol' indices (Santner et al, 2003; Iooss et al, 2006).

6.3.4 Stochastic Models

For stochastic models with inputs x and stochastic elements u it is not possible to fully decompose the variance as described in equation 6.3. Instead it is decomposed as (Iooss and Ribatet, 2009; Marrel et al, 2012)

$$Var(y) = \sum_i V_i + \sum_i \sum_{j>i} V_{ij} + \ldots + V_{123\ldots n} + E_x(Var_u(y|x))$$

where $E_x(Var_u(y|x))$ is the total sensitivity of the randomness, u. Iooss and Ribatet (2009) and Marrel et al (2012) defined the total sensitivity of u by

$$Var(y) - \widetilde{\underline{\tau}_i^2}$$

which they estimate by meta models. This use of meta models does not enable the authors to quantify the the different terms V_{iu}. In Marrel et al (2012), two estimators are built; one that models the expectation of the model output and the other the noise in the output.

In Iooss and Ribatet (2009), the authors suggested treating all of the stochastic elements, u, as inputs, which would enable us to decomposed the variance caused by these stochastic elements. In the next chapter we further develop this method of performing variance-based sensitivity analysis on stochastic simulation models in order to enable us to quantify the different terms V_{iu}. We use this method to enable us to perform robustness analysis on IBMs.

Chapter 7

Variance based sensitivity analysis of stochastic complex models

7.1 Introduction

Robustness analysis tests whether the results of a model depends on the essentials of the model or the simplifying assumptions (Weisberg, 2006). If a model's ability to reproduce patterns is sensitive to the details of the model then it probably doesn't capture the real mechanics of the system (Grimm and Railsback, 2005). This means that any features of the model that do not represent something in reality, for example scale of the model or grid, should not have much of an effect on the results of the model.

In other words any feature of the model that drives the output of the model should be theoretically correct and justified. It is possible to test this by seeing how sensitive the model is to these details. This is so important that Railsback and Grimm (2012) suggest that robustness analysis should be included in all publications of IBMs. Maclean (2010) tested a number of arbitrary decisions on Hovel and Regan's blue crab IBM (Hovel and Regan, 2008) model using local sensitivity analysis. Maclean (2010) found that a number of arbitrary parameter values had a large effect on the model output but found that the scale of the model had little effect on the results. In addition, Kloprogge et al (2011) devised a method based on asking experts how plausible the model assumptions are and Chen and Mynett (2003) examined the effects of cell size. Robustness analysis also examines what effect different submodels have on the output of the model, something that was tested by Tarantola et al (2002) and Cortés-Avizanda et al (2014).

In order to test the arbitrary features of models we need to see which inputs or parameters cause the variation of the model output. Any uncertain inputs should be given prior input distributions and then the sensitivity across the whole input space should be tested rather than at some nominal point which is often the case (Railsback and Grimm, 2012). A popular method is to partition the variance as described in the previous chapter and we hope to do this with these types of models.

One of the issues with doing variance-based sensitivity analysis is that the Sobol' decomposition cannot be made as some of the variance cannot be assigned because it is down to the stochasticity within the model. Marrel et al (2012) showed one way of decomposing the variance that treats the total effect of the stochasticity to be part of this decomposition. We demonstrate in Section 7.2 that it is possible to further decompose the variance into different types of stochastic inputs as well as estimate interaction terms of parameters and stochastic inputs. We use this partition of the variance amongst the different submodels in Section 7.3 that should enable us to test a model for arbitrary decisions and show how the model is sensitive to its submodels. This is demonstrated on the woodhoopoe and bird synchrony models in Section 7.4.

7.2 Stochastic Models

We are going to examine a general model $f(\cdot)$ with inputs x. For the time being let us assume that it is deterministic. The closed sensitivity index of x_i can be estimated by a method described by Kucherenko et al (2011) (see Section 6.3.3).

Proposition 3. *The estimator*

$$\widetilde{\underline{\tau}}_i^2 = \frac{1}{n}\sum_{m=1}^{n} f(x^m)(f(x_i^m, x'^m_{-i}) - f(x'^m)) \tag{7.1}$$

is unbiased and has variance

$$\frac{1}{n}\left(C_2 + 2\left(Var(W) + f_0^2\right)^2 - C^2 - 2E\left(W^2YZ\right)\right)$$

where $C_2 = Cov(f(x)^2, f(x_{-i}, x_i')^2)$, $C = Cov(f(x), f(x_{-i}, x_i'))$, $W = f(x)$, $Y = f(x_{-i}, x')$ and $Z = f(x')$.

Proof. Let $W = f(x)$, $Y = f(x_i, x'_{-i})$ and $Z = f(x')$. We will also drop the superscripts m. The expectation is

$$\begin{aligned} E\left(\frac{1}{n}\sum W(Y-Z)\right) &= \frac{1}{n}\left(E\left(\sum WY\right) - E\left(\sum WZ\right)\right) \\ &= \frac{1}{n}\left(nE\left(WY\right) - nE\left(WZ\right)\right) \\ &= Cov(W,Y) + E(W)E(Y) - E(W)E(Z) \\ &= Cov(W,Y) + f_0^2 - f_0^2 \\ &= Cov(W,Y) \end{aligned}$$

Hence the estimator is unbiased. The variance is

$$\begin{aligned} Var\left(\frac{1}{n}\sum W(Y-Z)\right) &= \frac{1}{n}Var\left(WY-WZ\right) \\ &= \frac{1}{n}\left(Var(WY)+Var(WZ)\right. \\ &\quad\left.-\ 2Cov(WY,WZ)\right) \end{aligned} \tag{7.2}$$

Now W and Y are not independent but do have the same mean and variance so we can say that

$$Var(WY) = Cov(W^2, Y^2) + \left(Var(W) + f_0^2\right)^2 - \left(Cov(W,Y) + f_0^2\right)^2.$$

W and Z are independent so

$$Var(WZ) = \left(Var(W) + f_0^2\right)^2 - f_0^4$$

and

$$Cov(WY, WZ) = E\left(W^2YZ\right) - f_0^2\left(Cov(W,Y) + f_0^2\right).$$

Plugging these into equation 7.2 we get

$$\begin{aligned} Var\left(\frac{1}{n}\sum W(Y-Z)\right) &= \frac{1}{n}\Big(Cov(W^2,Y^2) + \left(Var(W)+f_0^2\right)^2 \\ &\quad - \left(Cov(W,Y)+f_0^2\right)^2 + \left(Var(W)+f_0^2\right)^2 - f_0^4 \\ &\quad -2\left(E\left(W^2YZ\right) - f_0^2\left(Cov(W,Y)+f_0^2\right)\right)\Big) \\ &= \frac{1}{n}\Big(Cov(W^2,Y^2) + 2\left(Var(W)+f_0^2\right)^2 \\ &\quad - \left(Cov(W,Y)\right)^2 - 2E\left(W^2YZ\right)\Big). \end{aligned}$$

□

Now let's assume that $f(\cdot)$ is a stochastic model. We are now going to analyse the sensitivity of $E(f(x)|x)$ to the elements of x. For stochastic models, $f(x)$ is no longer known perfectly from a single run, there is an additional amount of uncertainty, called aleatory uncertainty, that is caused by the stochasticity in the model. A possible solution to this problem is to better estimate $f(x)$ by sampling from it m times each. If we want to estimate the sensitivity index of x_i then we can sample q points and sample them m times. If we fix the computation effort at n model runs then we have the constraint that

$$n = qm.$$

Proposition 4. *The variance of the estimator described in equation 7.1 estimated using the method described above is*

$$\frac{1}{q}\left(C_2 + 2\left(Var_x(W) + \frac{E_x(Var(f(\cdot|x)))}{m} + f_0^2\right)^2 \right.$$
$$\left. -C^2 - 2E\left(W^2 YZ\right)\right)$$

where $C_2 = Cov(f(x)^2, f(x_{-i}, x_i')^2)$*,* $C = Cov(E(f(x)|x), E(f(x_{-i}, x_i'))|x, x')$*,* $W = E(f(x)|x)$*,* $Y = E(f(x_{-i}, x_i')|x, x')$*,* $Z = E(f(x')|x')$ *and* $E_x(Var(f(\cdot|x)))$ *is the variance of* W *caused by the stochastic elements.*

Proof. Using the result from Proposition 3 and saying that $W' \sim \frac{1}{m}\sum_{j=1}^m f^j(\cdot|x)$, $Y' \sim \frac{1}{m}\sum_{j=1}^m f^j(\cdot|x_{-i}, x_i')$ and $Z' \sim \frac{1}{m}\sum_{j=1}^m f^j(\cdot|x')$ then we can say that the variance of the estimator is

$$\frac{1}{q}\left(C_2' + 2\left(Var(W') + f_0^2\right)^2 - C'^2 - 2E\left(W'^2 Y' Z'\right)\right).$$

with $C_2' = Cov(W'^2, Y'^2)$, $C' = Cov(W', Y')$. Furthermore

$$\begin{aligned} C_2' &= \frac{1}{m^4} Cov\left(\left(\sum_{j=1}^m f^j(\cdot|x)\right)^2, \left(\sum_{j=1}^m f^j(\cdot|x_{-i}, x_i')\right)^2\right) \\ &= Cov(W^2, Y^2) \end{aligned}$$

and

$$\begin{aligned} C' &= \frac{1}{m^2} Cov\left(\sum_{j=1}^m f^j(\cdot|x), \sum_{j=1}^m f^j(\cdot|x_{-i}, x_i')\right) \\ &= Cov(W, Y). \end{aligned}$$

Also,

$$\begin{aligned} E\left(W'^2 Y' Z'\right) &= \frac{1}{m^4} E\left(\sum_{j_1=1}^m \sum_{j_2=1}^m \sum_{j_3=1}^m \sum_{j_4=1}^m f^{j_1}(x) f^{j_2}(x) f^{j_3}(x_{-i}, x_i') f^{j_4}(x')\right) \\ &= \frac{1}{m^4} \sum_{j_1=1}^m \sum_{j_2=1}^m \sum_{j_3=1}^m \sum_{j_4=1}^m E\left(f^{j_1}(x) f^{j_2}(x) f^{j_3}(x_{-i}, x_i') f^{j_4}(x')\right) \\ &= \frac{1}{m^4} m^4 E\left(W^2 YZ\right) \\ &= E\left(W^2 YZ\right) \end{aligned}$$

and if we consider the stochastic elements as inputs u then

$$\begin{aligned} Var(W') &= Var_x(E(W')) + E_x(Var(W')) \\ &= Var_x(W) + \frac{Var(f(\cdot|x))}{m} \end{aligned}$$

and the result follows. □

If we set

$$\begin{aligned} A &= C_2 - C^2 - 2E(W^2YZ), \\ Q &= Var_x(W) + f_0^2 \end{aligned}$$

and

$$V = E_x(Var(W))$$

then the variance is minimised, by differentiating the variance of the estimator with respect to M, when

$$m^2 = \frac{2V^2}{A + 2Q^2}.$$

If the model is deterministic, ie. $E_x(Var_u(f(x))) = 0$, then we should put all of our effort into searching around the input space whereas if the variance is dominated by $E_x(Var_u(f(\cdot|x)))$ then we should aim to estimate the expectation $E(W)$ well by increasing m.

Using the queuing model described in Section 4.2.1 we performed variance based sensitivity analysis by setting $q = 10,000$ and $m = 1$ as we believe this to be the ideal settings and using the method of Tarantola et al (2006b) (Section 6.3.3) partitioned the variance for the minimum, maximum and every 1/19th quantile of the inter-departure times. We found that the numbers do not add up to 1 and can actually be a long way off. The estimates of the closed sensitivity indices are quite noisy and it is difficult to see what proportion of the variance is unaccounted for by the parameters.

It is possible to use the method of common random numbers (Owen, 2013b) in order to keep numerical stability as this will maximise $Cov(XY, XZ)$ and hence $E(X^2YZ)$. This is done by, as well as sampling x^j and x'^j, sampling the random inputs u^j for $j = 1, \ldots, q$. The estimate then becomes

$$\widetilde{\underline{\tau}}_i^2 = \frac{1}{n}\sum_{j=1}^{n} f(x^j, u^j)(f(x_i^j, x'^j_{-i}, u^j) - f(x'^j, u^j)). \tag{7.3}$$

Table 7.2 shows the estimates of the first order sensitivities when the random inputs are controlled as described in equation 7.3 and does demonstrate that the closed sensitivity indices are more stable. Having said this, we may not actually be interested in this as using this method decomposes the variance down to the inputs x_i only.

It is possible to decompose the variance into the epistemic and aleatory parts and estimate $E_x(Var_u(f(\cdot|x)))$. We will treat the stochastic part of the model as if it were an input u. This means that as well as sampling x and x', we sample u and u'. Then u acts just like another input enabling us to use it each of the estimation algorithms in order to estimate the sensitivity of each input (Iooss and Ribatet, 2009). The variance down to the aleatory parts, $E_x(Var_u(f(\cdot|x)))$, is then $\overline{\tau}_u^2$ (Marrel et al, 2012).

Quantile	θ_1	θ_2	θ_3
0	0.97	0	0.01
1/19	0.01	0	0.90
2/19	0.00	0	1.00
3/19	0.00	0	0.88
4/19	0.00	0	0.93
5/19	0.00	0	0.81
6/19	0.00	0	0.85
7/19	0.00	0	0.85
8/19	0.00	0	0.82
9/19	0.00	0	0.81
10/19	0.00	0	0.87
11/19	0.00	0	0.89
12/19	0.00	0	1.04
13/19	0.00	0	0.99
14/19	0.00	0	0.99
15/19	0.00	0	0.98
16/19	0.00	0	0.87
17/19	0.00	0	0.82
18/19	0.00	0	0.82
1	0.00	0	0.67

Table 7.1: The first order indices using the algorithm of Tarantola et al (2006b) and $q = 10,000$ and $m = 1$.

This was done by Iooss and Ribatet (2009) where they treated all the stochastic realisation as an input and calculated the related sensitivity indices. We can use the idea of treating the random inputs as conventional inputs to further break down the aleatory uncertainty to different types of random inputs by coupling the random inputs as described in Section 4.2.1. This will help us in the next section where we are going to partition the variance into different submodels. We demonstrate this method on the queuing model and the Ricker model but first we want to show that the variance and total variance of the random inputs will remain the same regardless of how the model is parameterised.

If we can write the model with inputs $\theta_{1:N}$ or $\phi_{1:N}$ such that

$$e_\theta(\theta_{1,\ldots,N}) \overset{D}{=} e_\phi(\phi_{1,\ldots,M}).$$

Both models have the same variance but have different decompositions. Now if we can partition the inputs to x in common and $\theta_{1\ldots n}$ and $\phi_{1\ldots m}$ not $(n < N, m < M)$ with

$$f_\theta(x, \theta_{1,\ldots,n}) \overset{D}{=} f_\phi(x, \phi_{1,\ldots,m})$$

Quantile	θ_1	θ_2	θ_3
0	0.97	0	0.01
1/19	0.00	0	1.00
2/19	0.00	0	1.00
3/19	0.00	0	1.00
4/19	0.00	0	1.00
5/19	0.00	0	1.00
6/19	0.00	0	1.00
7/19	0.00	0	1.00
8/19	0.00	0	1.00
9/19	0.00	0	1.00
10/19	0.00	0	1.00
11/19	0.00	0	1.00
12/19	0.00	0	1.00
13/19	0.00	0	1.00
14/19	0.00	0	1.00
15/19	0.00	0	1.00
16/19	0.00	0	1.00
17/19	0.00	0	1.00
18/19	0.00	0	1.00
1	0.00	0	1.00

Table 7.2: The first order indices using Tarantola et al's (Tarantola et al, 2006b) and using common random numbers for 10,000 samples of the inputs.

and write as

$$f(x, g(\theta_{1:n})) \overset{D}{=} f(x, h(\phi_{1:m}))$$

then

$$\underline{\tau}^2_{\theta_1,\ldots,\theta_n} = \underline{\tau}^2_{\phi_1,\ldots,\phi_m}$$

and

$$\overline{\tau}^2_{\theta_1,\ldots,\theta_n} = \overline{\tau}^2_{\phi_1,\ldots,\phi_m}.$$

This means that it does not matter how the model is parameterised, the closed sensitivity index and total effects indices of the inputs x will remain the same, as too will $E_x(Var_u(X))$. This parameterisation could include how the random inputs have been coupled which means that how the model is coupled does not effect the closed sensitivity indices and total effects indices.

7.2.1 Queueing Model

We are going to demonstrate the method of coupling the random inputs and performing variance based sensitivity analysis for the queuing model described in Section 4.2.1 with priors described in Section 4.4.2 with the outputs being the summary statistics of the inter-departure times suggested by Blum and François (2010). In order to do this, as it is quick to simulate and we are interested in all indices and not just the first order ones we will use the method of Liu and Owen (2006). We coupled the model in the way described in Section 4.2.1. The results are shown in Table 7.3 and

Quantile	θ_1	θ_3	u	$\theta_3 u$	$E_x(Var_u(f(x)))$
0	0.98	0.01	0.00	0.00	0.01
1/19	0.02	0.88	0.03	0.07	0.09
2/19	0.01	0.94	0.01	0.04	0.05
3/19	0.00	0.92	0.03	0.05	0.08
4/19	0.00	0.88	0.05	0.06	0.12
5/19	0.00	0.85	0.07	0.08	0.15
6/19	0.00	0.91	0.04	0.05	0.09
7/19	0.00	0.93	0.03	0.04	0.07
8/19	0.00	0.88	0.06	0.06	0.12
9/19	0.00	0.93	0.03	0.04	0.07
10/19	0.00	0.88	0.05	0.06	0.12
11/19	0.00	0.93	0.03	0.04	0.07
12/19	0.00	0.98	0.01	0.01	0.02
13/19	0.00	0.98	0.01	0.01	0.02
14/19	0.00	0.98	0.01	0.01	0.02
15/19	0.00	0.96	0.01	0.02	0.04
16/19	0.00	0.93	0.03	0.04	0.07
17/19	0.00	0.86	0.07	0.07	0.14
18/19	0.00	0.91	0.04	0.05	0.09
1	0.00	0.94	0.03	0.04	0.06

Table 7.3: The estimates of the variances for each of the inputs and the total sensitivity of the random inputs. Every other combination is zero.

show that a lot of the variance of the minimum service time is down to θ_1 which makes sense as this is a sufficient statistic (Fearnhead, 2004) but all other quantiles are controlled by θ_3, the arrival rate, or u, the random inputs that control the arrival times. Furthermore we found that θ_2 was very insensitive to all of the summary statistics. This may suggest a reason why θ_2 was difficult to estimate when performing parameter estimation in the Section 4.4.2. Figure 7.1 shows how the standardised first order Sobol' indices for each of the outputs. It shows that apart from the minimum, which is almost entirely θ_1, θ_3 and u control the variance and the indices don't really change for later quantiles with the changes being mostly down to noise. It is interesting to see that very quickly θ_3 dominates the variance of the output.

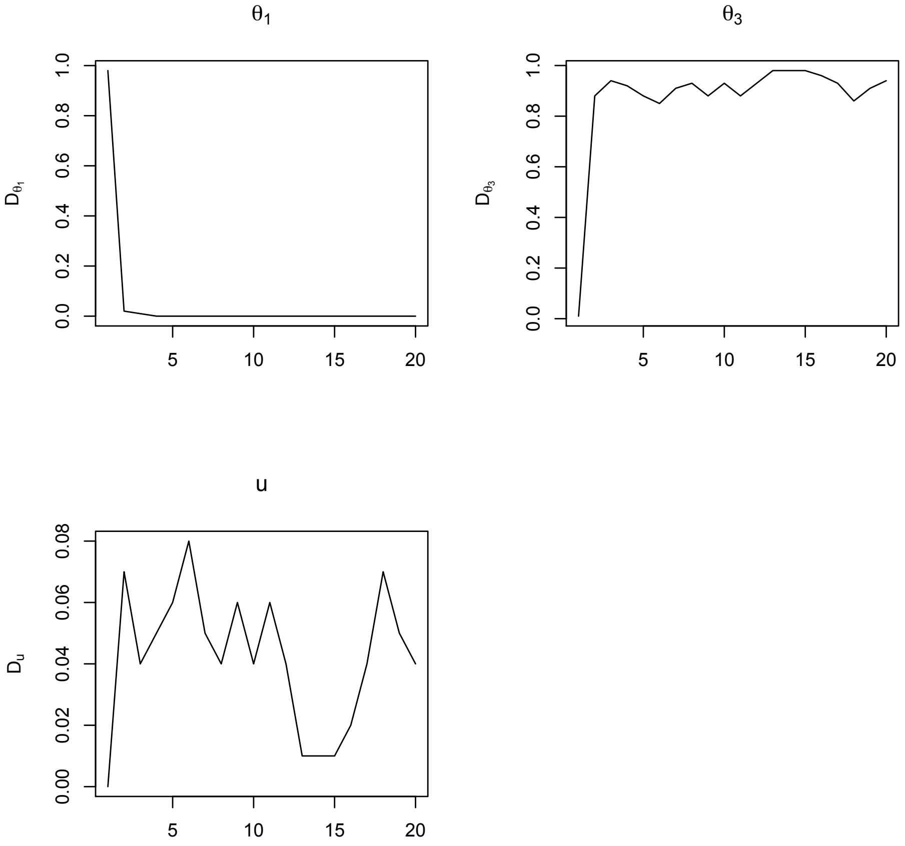

Figure 7.1: The first order indices for the θ_1 θ_3 and u with the original prior $\theta_3 \sim U(\cdot|0, 1/3)$ for each of the 20 outputs. The other inputs are zero.

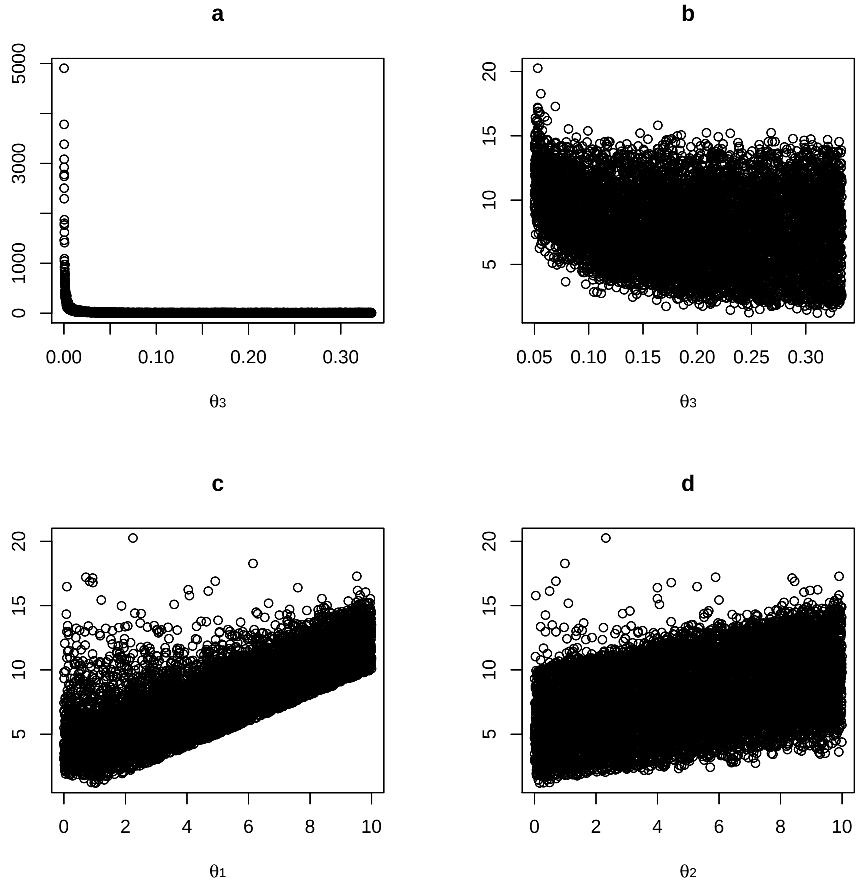

Figure 7.2: The observations of the 10/19 quantile for runs of the model for the whole prior of θ_3 in (a) and $\theta_3 \sim U(\cdot|0.05, 1/3)$ in (b), (c) and (d). In (b), (c) and d we show the results for θ_3, θ_1 and θ_2 respectively.

Figure 7.2 shows samples of the 10/19 quantile for different values of the parameters. It shows, in (a), that when $\theta_3 < 0.05$ it dominates the output. However if we change the prior distribution such that $\theta_3 \sim U(\cdot|0.05, 1/3)$ then we can see that θ_3 no longer dominates the output as seen in (b-d).

Figure 7.3 shows the standardised first order sensitivity indices for each of the inputs for each of the outputs with the new prior distribution. It is interesting that the beginning quantiles are most sensitive to θ_1, the minimum service time whereas the later quantiles are more sensitive to the arrival rate, θ_3, and random input that determines the arrival times, u. The other inputs, θ_2 and w, the random input that determines the service times effect the middle quantiles. This is expected as these are the quantiles when the queue length is roughly zero and the inter-departure times depend on the service times only.

Additionally this shows that learning a little bit about the value of θ_3 has not only reduced the variance but has meant that the standardised indices of the other parameters are much higher. This illustrates that the Sobol' indices are very sensitive to their prior distributions. Examining this is beyond the scope of this thesis but for a method of estimating the value of learning some data, which will reduce the variance of the input distributions, see (Strong and Oakley, 2013; Strong et al, 2014).

7.2.2 Ricker Model

We are going to examine the Ricker Model, as described in Wood (2010) and Section 4.2.1. We examined most of the summaries described in Wood (2010). These are:

- the coefficients, β_1 and β_2, of autoregression.
- the mean of the outputs and the number of outputs that are equal zero.
- the autocovariances up to lag 5.

We sampled the parameters and the random inputs $10,000$ times with priors suggested by Wilkinson (2014),

$$\begin{aligned} \log(r) &\sim U(\cdot|3,5), \\ \sigma &\sim U(\cdot|0,0.8) \quad \text{and} \\ \phi &\sim U(\cdot|4,20). \end{aligned}$$

Figure 7.4 shows the results of the global sensitivity analysis. The global sensitivity of the Ricker model shows that for most of the summaries $\log r$ often has an effect on the output as too does ϕ. For the mean (summary 3), the Sobol' index of ϕ is nearly 0.8. u, the random inputs associated with the observation model has very little effect on all of the summary statistics.

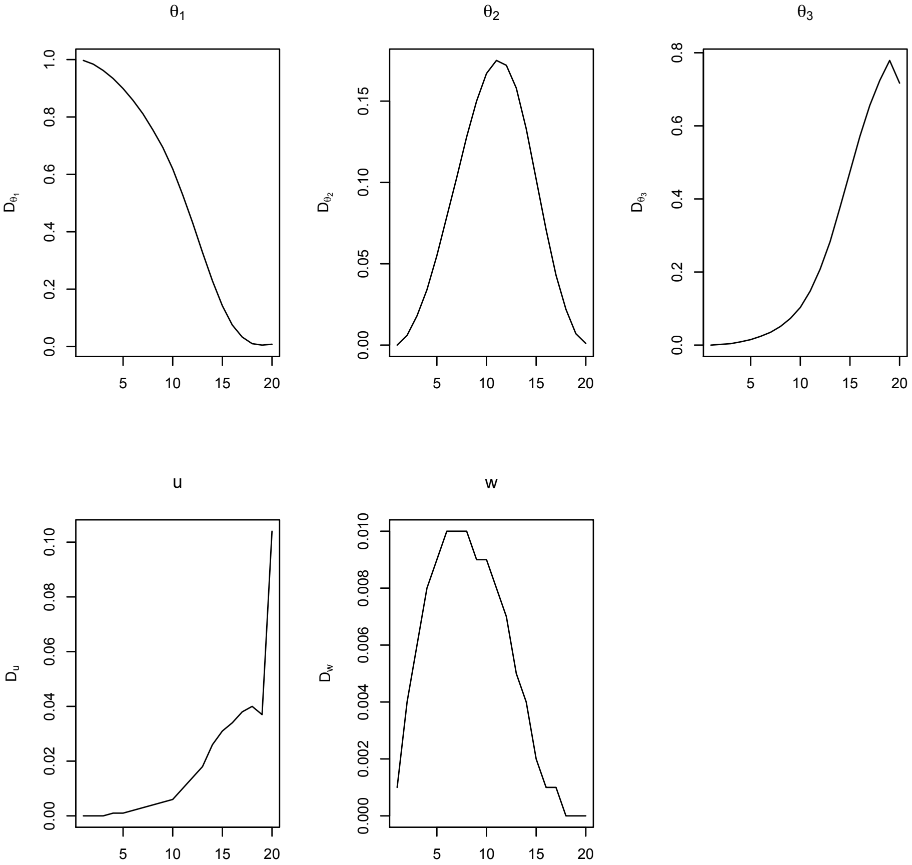

Figure 7.3: The first order indices for the θ_1, θ_2, θ_3, u and w when the prior $\theta_3 \sim U(\cdot|0.05, 1/3)$ for each of the 20 outputs.

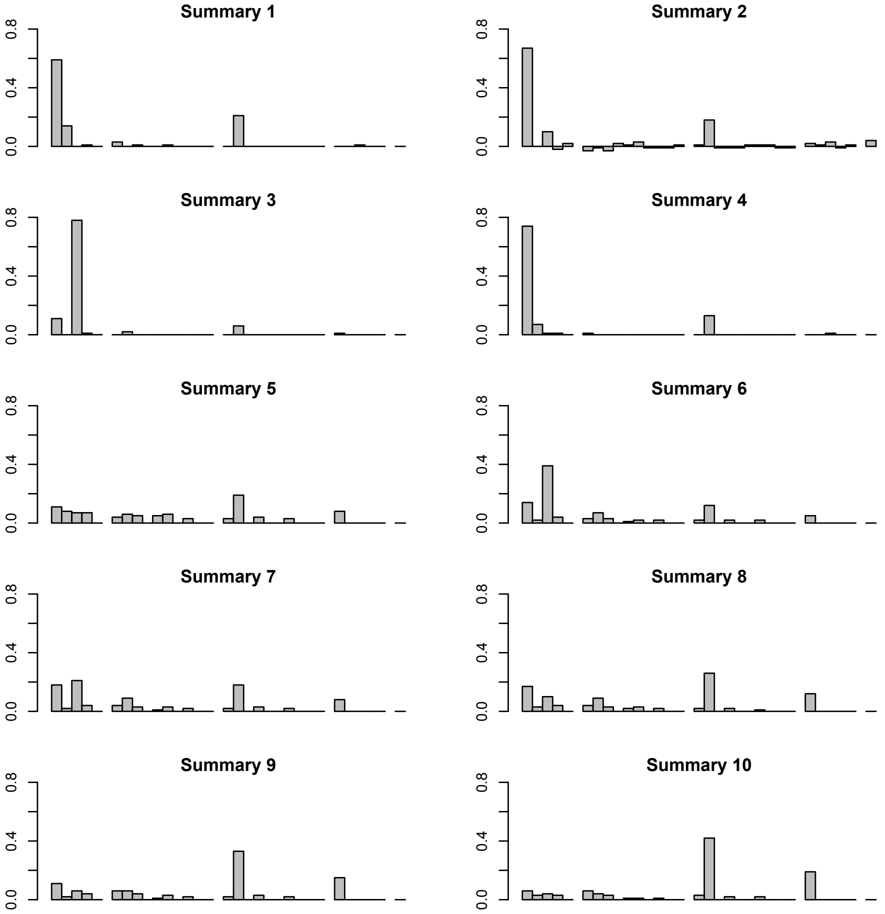

Figure 7.4: The sensitivity indices for the inputs of the Ricker model. The first 5 inputs are the first order indices, the next 10 the second order indices, the other sets being the third, fourth and fifth order indices respectively with a lexicographical ordering within each set of inputs with the inputs being in the order r, σ, ϕ, e and u.

The third order interaction between $\log r$, σ and e has a large effect on all of the summaries. The fourth order interaction between $\log r$, σ, ϕ and e also contributes quite a lot to the variance for the auto covariance summaries.

There is quite a lot of variance for all of the summary statistics caused by interacting parameters. The total sensitivity indices for the random inputs is roughly a quarter for the covariates of autoregression (summaries 1 and 2), roughly 0.1 for the mean and number of outputs equal to 0 (summaries 3 and 4) and about 0.5 for the autocovariances of lag 0 (summary 5) and increasing roughly linearly from 0.3 to 0.73 for lags 1 to 5 (summaries 6 to 10). With the exception of the covariates of autoregression, the total variance for the u was 0. This means that almost all of the stochastic variation is down to the random inputs in the background model, e.

7.3 Submodels and Arbitrary decisions

We have seen how a stochastic model can be written as a deterministic function of its parameters and its 'random inputs'. In this section we look at decomposing this function into a number of separate functions, linked together in the form of a Directed Acyclic Graph (DAG). Such a decomposition is not unique, of course, but often we can choose the decomposition so that the individual functions correspond to parts of the model that have meaning in the real process being modelled, linked together in an order that reflects the real-world sequence of events (or sometimes, in a computer-based model, the sequence of calculations within the model). We call the elements of a decomposition of this sort 'submodels'.

Definition 1. *A submodel is part of a model expressed as a deterministic function.*

Usually the inputs to a submodel will be uncertain and they may include parameters, random inputs, and the outputs from other submodels (its 'parents' in the DAG). While the DAG usually induces only a partial ordering on its nodes, in practice the ordering on the nodes representing submodels (as opposed to inputs) is often strict. This ordering will be important later in thinking about variance partitioning.

We also want to extend the type of flexibility we represent within a model. Sometime there are a number of different ways in which a part of a process can be modelled. For example, in the woodhoopoe model (Section 2.2), there is a part of the model where a subordinate leaves their group and searches for a vacant dominant position in another group. The subordinate either goes left or right and searches for up to 5 groups in that direction. If another researcher comes along and says that rather than go left or right they go the opposite way to where they went last time, this could be put into the model as an alternative. In general, suppose that we have a submodel that is uncertain, and we have different versions $S_i, i = 1, \ldots, d$ with probabilities w_i, which define a prior distribution over these models. We can think of $s \in \{S_1, \ldots, S_d\}$ as an additional input to the submodel, representing the uncertain modelling choice being made in part of the model. This method is similar to the trigger parameter method of Tarantola et al (2002) where the presence or absence of stochastic errors is added with probability 0.5.

The sensitivity indices for s can be estimated like the other parameters but for small d it may be more efficient to estimate the closed sensitivity index of inputs i by estimating

$$\underline{\tau}_i^2 = \sum_{l=1}^{d} p(s^{(l)}) \sum_{m=1}^{d} p(s^{(m)}) \int f(x, s^{(l)}) \left(f(x_i, x'_{-i}, s_i^{(l)}, s_{-i}^{(m)}) - f(x', s^{(m)}) \right) dx dx'$$

than sampling s and s' and estimating it using the method of Tarantola et al (2006b).

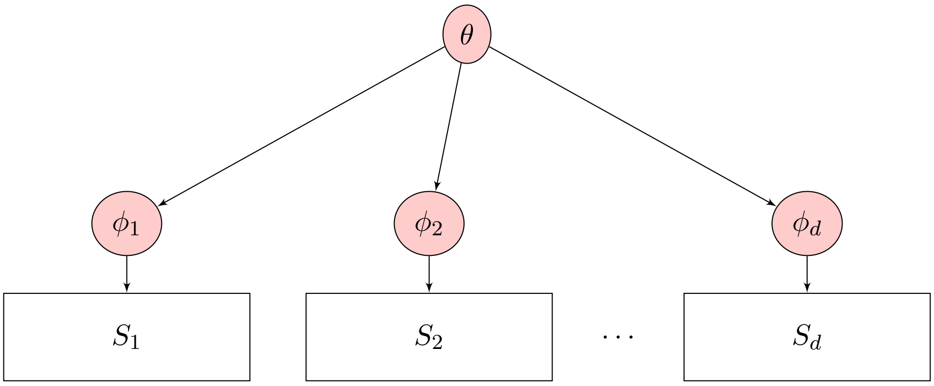

Figure 7.5: The directed acyclic graph of the submodel choice where all of the submodels are parameterised from the same parameters θ.

If we have a number of functions, S_i, that share the same parameters, θ, but are parameterised differently, ϕ_i, as shown in Figure 7.5, then we can distinguish between the uncertainty caused by s and θ. This is demonstrated on the bird synchrony model in Section 7.4.1.

Previously the test of alternative submodels has been used to test the importance of each submodel to the model as a whole (Tarantola et al, 2002; Iooss and Ribatet, 2009). However this differs from what we are describing here. Above we are assuming that the decision of which submodel to use in the model is uncertain and sensitivity index of s is the expected reduction in variance should we learn the true value of s. However in the other methods, the choice of submodel has already been made and they see what difference would have been made had we chosen another submodel. This is known as robustness analysis rather than sensitivity analysis. In robustness analysis the variance of the models is increased by adding alternative models and then the additional variance is deemed to be the importance of the submodel to the output (Grimm et al, 2014).

We are now going to propose a method of partitioning the variance between the submodels, in a way that depends on the (partial) ordering represented by the DAG linking them. More precisely, we will use Sobol decomposition to partition the variance between the inputs to the model—including parameters, random inputs and choices of submodel versions—and then use the ordering of the submodels to associate the terms in the Sobol decomposition with particular submodels. Note that this means that the Sobol terms will be grouped differently if we decompose the model into submodels in a different way. In order to describe this we will look at a toy model

$$\begin{aligned} \mu &\sim N(m, v) && (7.4) \\ Y &\sim N(\mu, \sigma^2) && (7.5) \end{aligned}$$

By coupling the random inputs as described in Chapter 4 (i.e. u_1 is used to evaluate the first submodel and w the second), we are able draw a directed acyclic graph (DAG) of this model as shown in Figure 7.6. In this model, the submodel, $m+\sqrt{v}\Phi^{-1}(u_1)$, is evaluated before $\mu+\sigma\Phi^{-1}(w)$ and μ is the state of the model after the first submodel. Note that this representation of the model gives a strict ordering between the two submodels, but not between the inputs. By treating the inputs to a submodel as coming before the inputs to a later submodel, we can refine the ordering in a natural way; in the example, we think of mu, v and u_1 as coming before σ^2 and w. This

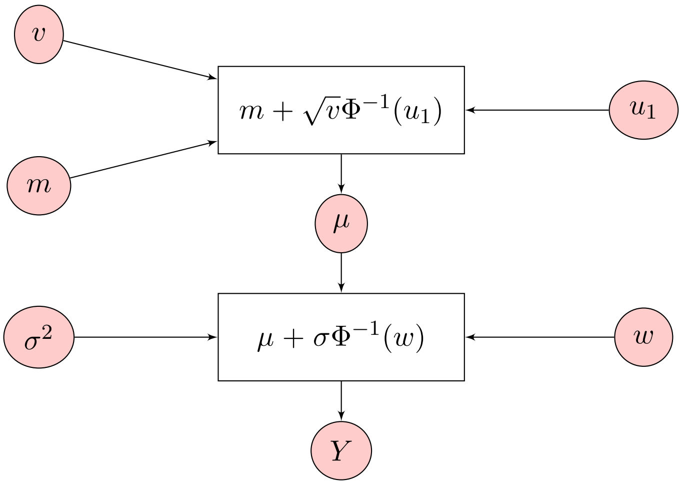

Figure 7.6: The directed acyclic graph of the model described in equations 7.4 and 7.5.

refined ordering is the basis of the way we will define the sensitivity index of a submodel.

In a general model, we can label the submodels A_i and their respective inputs x_i for $i = 1 \ldots r$ and Q_i as the state of the model after submodel A_i has been evaluated with Q_0 is the initial state of the model and Q_r is the output of the model. The state, Q_i, may include things that are unchanged by A_i and may need to be used in further submodels. A DAG of a general model is shown in Figure 7.7. In the toy example, A_1 is $m + \sqrt{v}\Phi^{-1}(u_1)$, with inputs m, v and u_1, A_2 is $\mu + \sigma\Phi^{-1}(w)$, with inputs σ^2 and w and Q_1 is μ and Q_r is Y.

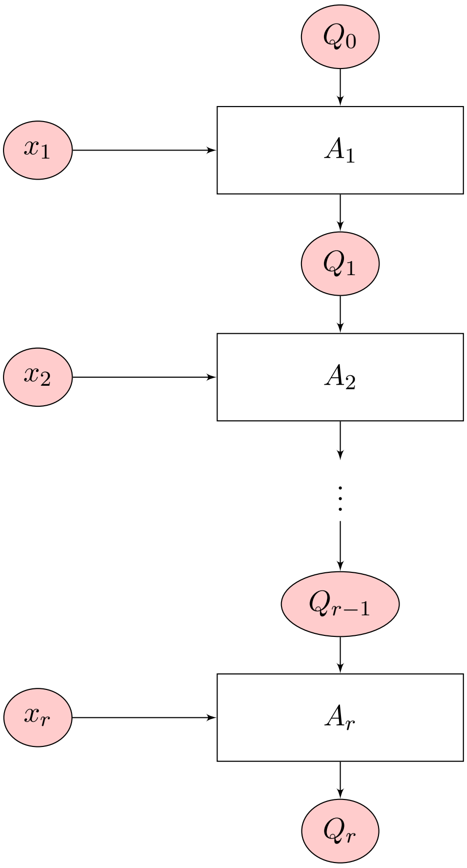

Figure 7.7: The directed acyclic graph of a general model.

Before we can define submodel sensitivity, we need to consider the case where there is *not* a strict ordering between the submodels. Suppose that in the toy example that we add a submodel

$$\sigma^2 = IG(a, b, u_2)$$

where $IG(a, b, u_2)$ is the inverse of the cumulative distribution of $Inv -$

$Gamma(a, b)$ evaluated at u_2. This submodel has to be evaluated before the submodel $\mu + \sigma\Phi^{-1}(w)$ but it does not matter whether it is evaluated before or after the submodel $m+\sqrt{v}\Phi^{-1}(u_1)$. In this formulation, A_1 is either $m + \sqrt{v}\Phi^{-1}(u_1)$ with Q_1 being μ and A_2 being $IG(a, b, u_2)$ or $IG(a, b, u_2)$ with Q_1 being σ^2 and A_2 being $m + \sqrt{v}\Phi^{-1}(u_1)$. Either way Q_2 is μ, σ^2. As the order that these two submodels are evaluated does not affect the state Q_2, we say that the submodels A_1 and A_2 are commutative submodels. More formally

Definition 2. *Two submodels, A_i and A_{i+1}, are commutative submodels if, conditional on the state, Q_{i-1}, being the same before either submodel is evaluated, the state after both submodels have been evaluated, Q_{i+1} will be the same regardless of the order in which the submodels are evaluated in.*

The DAG for the toy model, with u_2 being the random input for the submodel $IG(a, b, u_2)$, is shown in Figure 7.8. Now lets we add a further level of complexity by saying that

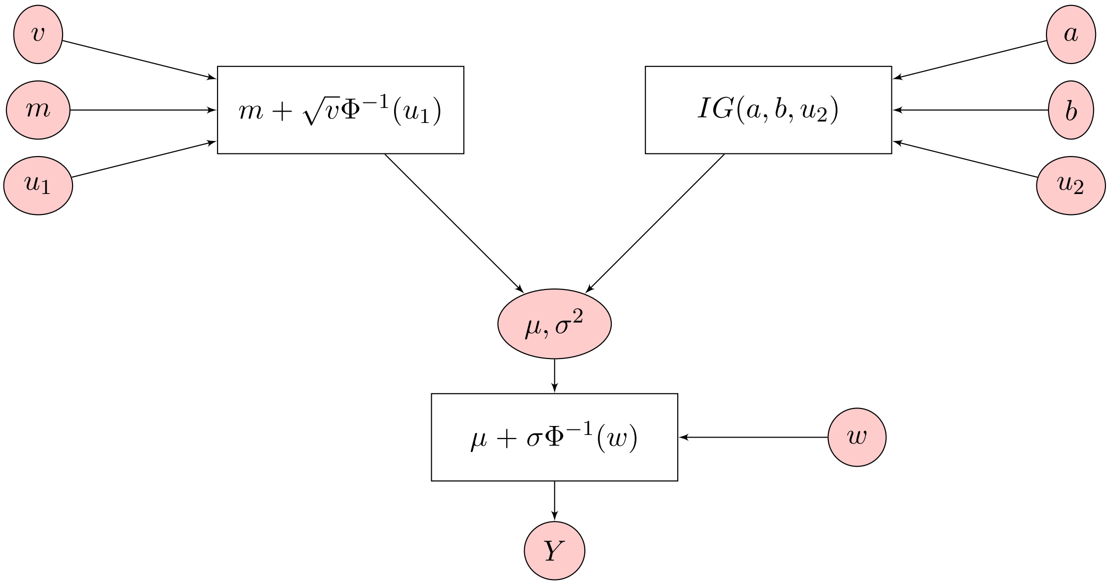

Figure 7.8: The directed acyclic graph of the toy model with the commutative submodels.

$$\begin{aligned} m &= f(x_1) \\ a &= g(x_2). \end{aligned}$$

In this model, the submodel $f(x_1)$ needs to be evaluated before $m+\sqrt{v}\Phi^{-1}(u_1)$ and $g(x_2)$ needs to be evaluated before $IG(a, b, u_2)$ however the order in which these two sets of submodels are evaluated does not matter. We say that these two sets of submodels are commutative submodel sets. More formally

Definition 3. *Sets of submodels $A = \{A_i, \ldots, A_j\}$ and $B = \{A_{j+1}, \ldots, A_k\}$ are commutative submodel sets if conditional on the state, Q_{i-1}, being the same before either set has been is evaluated, the state after both sets have been evaluated Q_{k+1} will be the same regardless of the order in which the sets are evaluated.*

If we look again at the final submodel, $\mu + \sigma\Phi^{-1}(w)$, which has inputs μ, σ^2 and w we can partition the variance of Y as

$$Var(Y) = V_w + \underline{\tau}^2_{\mu\sigma^2} + V_{w\mu} + V_{w\sigma^2} + V_{w\mu\sigma^2}.$$

If we were to learn μ and σ^2 then we would expect the variance of Y to reduce by $\underline{\tau}^2_{\mu\sigma^2}$. This means that $\underline{\tau}^2_{\mu\sigma^2}$ has been caused in other submodels and the rest is caused in this submodel. We now define a metric of how the model outputs are sensitive to the submodels.

Definition 4. *Let x_i be the inputs to submodel A_i and ϕ_i be all of the inputs to the previous submodels, $A_{1:i-1}$, that are not commutative with A_i, then the submodel sensitivity index of A_i is defined as*

$$S_{A_i} = \underline{\tau}^2_{x_i\phi_i} - \underline{\tau}^2_{\phi_i}.$$

This corresponds to the inputs in a particular order, induced by the DAG and extended to the inputs as described above. This essentially gives the expect reduction in variance by learning x_i and ϕ as opposed to just learning ϕ. It follows from the decomposition of the Sobol index above that $S_{A_i} \geq \underline{\tau}^2_{x_i}$.

Definition 5. *If A_i and A_j are commutative with inputs x_i and x_j respectively, then the submodel interaction index is*

$$S_{A_iA_j} = \underline{\tau}^2_{x_ix_j} - \underline{\tau}^2_{x_i} - \underline{\tau}^2_{x_j}.$$

For example in the toy model described by equations 7.4 and 7.5, the submodel sensitivity index of the submodel $m + \sqrt{v}\Phi^{-1}(u_1)$ is the closed sensitivity index of the inputs, $\underline{\tau}^2_{vmu_1}$ because $x_1 = \{v, m, u_1\}$ and $\phi_1 = \{\}$. The submodel sensitivity index of $\mu + \sigma\Phi^{-1}(w)$ is the closed sensitivity index of the inputs to the submodel, $x_2 = \{\sigma^2, w\}$, and inputs to the previous submodel, $\phi_2 = \{m, v, u_1\}$, less the closed sensitivity index of the inputs to the previous submodel. This is

$$V_{\sigma^2} + V_w + V_{\sigma^2 v} + \cdots + V_{wv} + \cdots + V_{\sigma^2 wv} + \cdots + V_{\sigma^2 wvmu_1}.$$

An extended, iterative version of the toy model is described by

$$\mu_i = Y_{i-1} + \sqrt{v}\Phi^-1(u_i) \tag{7.6}$$

and then

$$Y_i = \mu_i + \sigma\Phi^{-1}(w_i) \tag{7.7}$$

for $i = 1 \dots n$. The DAG of this model is shown in Figure 7.9. The submodels described in equation 7.6 and equation 7.7 are repeated. These submodels form an iteration of an iterative model. The beginning of the iteration the submodel described by equation 7.6 and the end of the iteration is the submodel described in equation 7.7. The iteration is of length 2. More formally

Definition 6. *If there exist integers $l > 0$ and j such that*

$$A_i = A_{(t-1)l+i}$$

for $i = j \dots l-1+j$ and $t = 2 \dots T$ for some $T > 1$, in the sense that if the inputs to A_i and $A_{(t-1)l+i}$ are the same then the outputs remain the same, then the model is an iterative model. The part of the model between A_j and A_{l-1+j} is known as an iteration, and is effectively executed T times. The submodel A_j is known as the start submodel and A_{l-1+j} is the end submodel and the iteration has length l.

If the general model described above is an iterative model, then we can re-label the submodels in an iteration as $A_{j,t}$ for $j = 1 \dots l$ in iteration t. t increases every time the start submodel of an iteration is evaluated (for $t = 1 \dots T$). This means we can add a plate to the DAG as shown in Figure 7.10 for a section of the model. We now say that an edge creating a cycle in the graph means that the model is an iterative model with the edge moving from the end submodel to the start submodel implying a plate. The state, Q_i, is also implied.

The DAG of the iterative model described by equations 7.6 and 7.7 is shown in Figure 7.11. Looking at Figure 7.9 every time $Y_{i-1} + \sqrt{v}\Phi^{-1}(u_i)$, which we are going to call $A_{1,i}$, is evaluated, v remains the same. Similarly when $\mu_i + \sigma\Phi^{-1}(w_i)$, which we denote $A_{2,i}$, is evaluated σ^2 remains the same. We can collect all of the $A_{1,i}$s (called A_1) and $A_{2,i}$s (called A_2) and then the iterative submodel sensitivity indices of model of submodels A_1 and A_2 is the sum of the submodel sensitivities indices $A_{1,1:n}$ and $A_{2,1:n}$ respectively. More formally, if the model is an iterative model then we can re-label the submodels in an iteration as $A_{j,t}$ for $j = 1 \dots l$ in iteration t. t increases every time the start submodel of an iteration is evaluated (for $t = 1 \dots T$).

Definition 7. *For an iterative model with iteration start A_j and end A_{j+l} then the iterative submodel sensitivity index of A_i, for $i = j \dots j+l-1$ is*

$$V_{A_i} = \sum_{t=1}^{T} S_{A_{i,t}}.$$

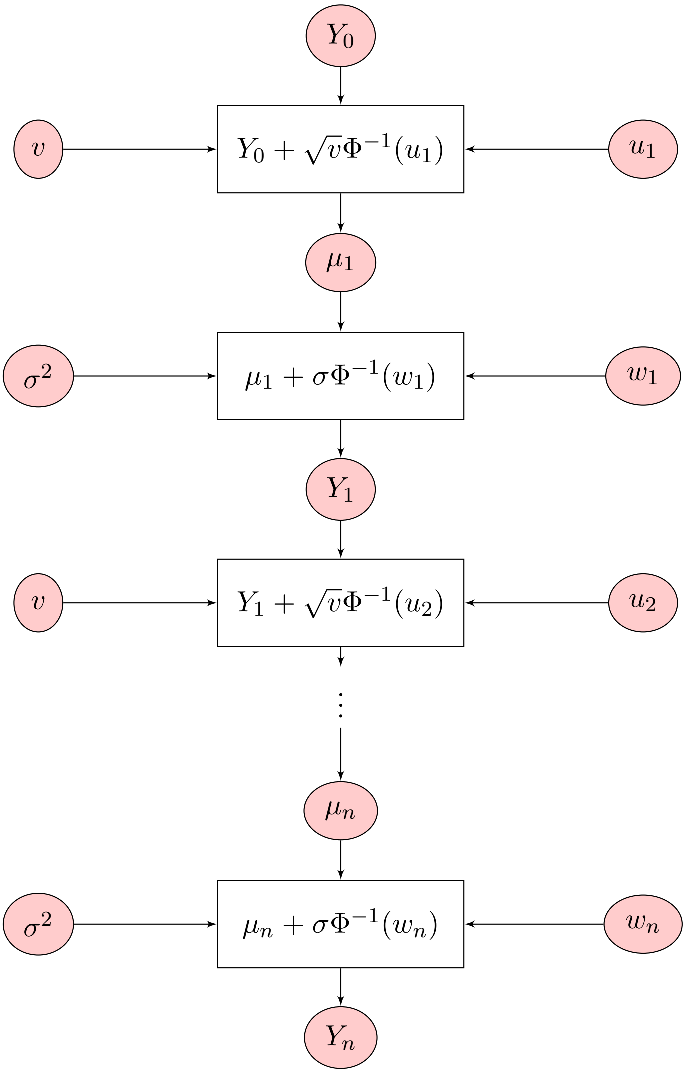
Y_0
v
$Y_0 + \sqrt{v}\Phi^{-1}(u_1)$
u_1
μ_1
σ^2
$\mu_1 + \sigma\Phi^{-1}(w_1)$
w_1
Y_1
v
$Y_1 + \sqrt{v}\Phi^{-1}(u_2)$
u_2
$\vdots$
μ_n
σ^2
$\mu_n + \sigma\Phi^{-1}(w_n)$
w_n
Y_n

Figure 7.9: The directed acyclic graph of the model described in equations 7.6 and 7.7.

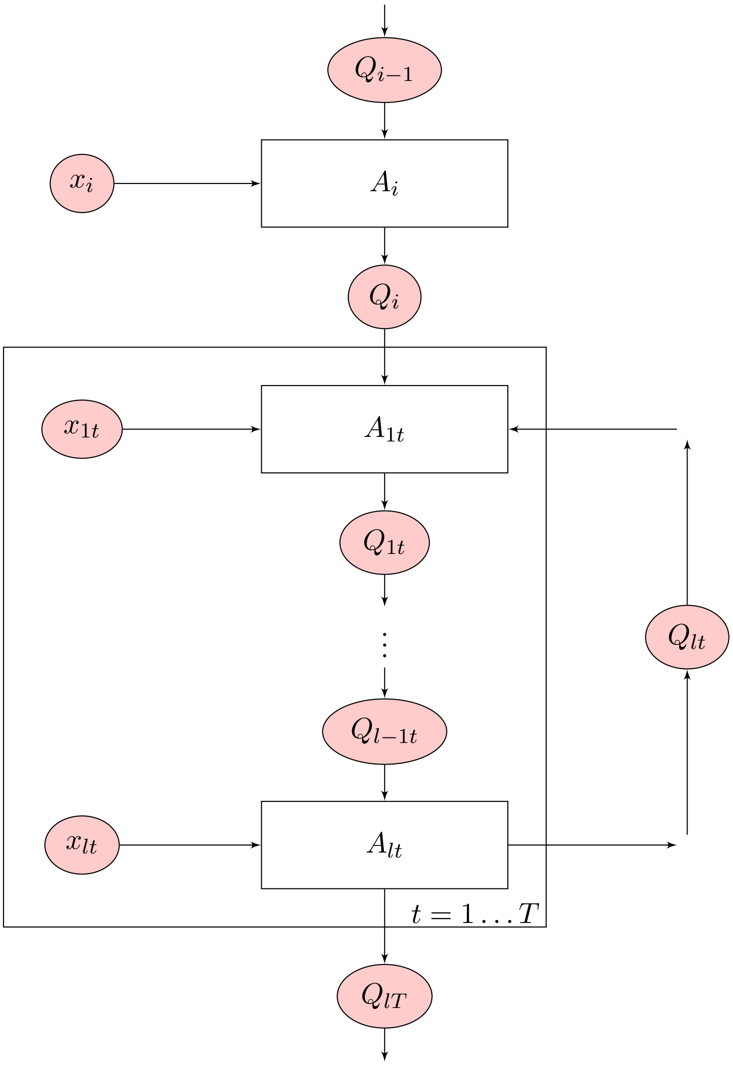

Figure 7.10: Part of the directed acyclic graph of an iterative model.

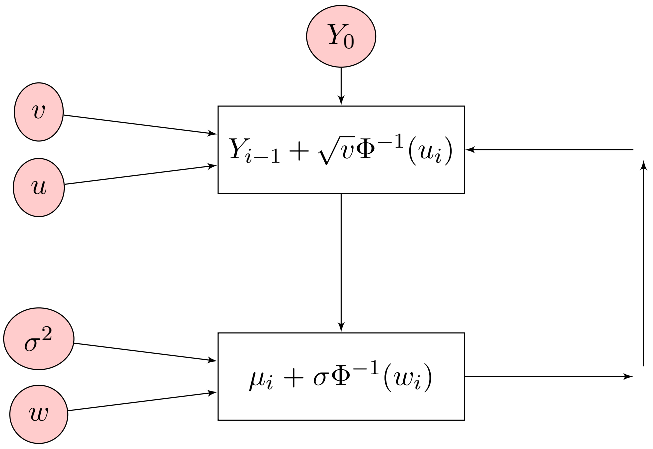
Y_0
v
u
$Y_{i-1} + \sqrt{v}\Phi^{-1}(u_i)$
σ^2
w
$\mu_i + \sigma\Phi^{-1}(w_i)$

Figure 7.11: The directed acyclic graph of the iterative model described in equations 7.7 and 7.6.

For a submodel A_k which is not part of an iteration then

$$V_{A_k} = S_{A_k}.$$

If submodel A_i is in an iteration with length greater than one, it is not possible to calculate the iterative submodel sensitivity index of A_i. We can however calculate upper and lower bounds.

Proposition 5. *If we have an iterative model with submodel, A_i with inputs θ, ϕ are the inputs to other submodels in the iteration and η are the inputs all the from submodels not involved in the iteration, A_j for $j < i$, then we can say*

$$\underline{\tau}^2_{\theta\eta} - \underline{\tau}^2_{\eta} \leq V_{A_i} \leq \underline{\tau}^2_{\theta\phi\eta} - \underline{\tau}^2_{\phi\eta}.$$

Proof. First separate θ into $\theta_{1:T}$ where θ_t is the inputs to the submodel at time t and let ϕ_t be all the parameters that have entered the model in other submodels between A_{it-1} and A_{it}. This means that the submodel sensitivity index for A_{it} is

$$S_{A_it} = \underline{\tau}^2_{\theta_{1:t}\phi_{1:t}\eta} - \underline{\tau}^2_{\theta_{1:t-1}\phi_{1:t}\eta}.$$

We can rewrite

$$\begin{aligned}\underline{\tau}^2_{\theta_{1:t}\phi_{1:t}\eta} \quad &= \quad \underline{\tau}^2_{\theta_{1:t-1}\phi_{1:t}\eta} + \underline{\tau}^2_{\theta_t\phi_{1:t}\eta} - \underline{\tau}^2_{\theta_t} - \underline{\tau}^2_{\phi_{1:t}\eta} \\ &\quad\quad + \underline{\tau}^2_{\theta_{1:t}\eta} - \underline{\tau}^2_{\theta_{1:t-1}\eta}\end{aligned}$$

and therefore

$$S_{A_it} = \underline{\tau}^2_{\theta_{1:t}\eta} - \underline{\tau}^2_{\theta_{1:t-1}\eta} + \underline{\tau}^2_{\theta_t\phi_{1:t}\eta} - \underline{\tau}^2_{\theta_t\eta} - \underline{\tau}^2_{\phi_{1:t}\eta}.$$

The total first order variance, over time T is

$$\begin{aligned}
V_{A_i} &= \sum_{t=1}^{T} S_{A_i t} \\
&= \sum_{t=1}^{T} \left(\underline{\tau}^2_{\theta_{1:t}\eta} - \underline{\tau}^2_{\theta_{1:t-1}\eta} + \underline{\tau}^2_{\theta_t \phi_{1:t}\eta} - \underline{\tau}^2_{\theta_t \eta} - \underline{\tau}^2_{\phi_{1:t}\eta} \right) \\
&= \underline{\tau}^2_{\theta_{1:T}\eta} - \underline{\tau}^2_{\eta} + \sum_{t=1}^{T} \left(\underline{\tau}^2_{\theta_t \phi_{1:t}\eta} - \underline{\tau}^2_{\theta_t \eta} - \underline{\tau}^2_{\phi_{1:t}\eta} \right) \\
&\geq \underline{\tau}^2_{\theta_{1:T}} - \underline{\tau}^2_{\eta}
\end{aligned}$$

as

$$\underline{\tau}^2_{\theta_t \phi_{1:t}\eta} - \underline{\tau}^2_{\theta_t \eta} - \underline{\tau}^2_{\phi_{1:t}\eta} \geq 0.$$

Also

$$\begin{aligned}
\underline{\tau}^2_{\theta\phi\eta} - \underline{\tau}^2_{\phi\eta} &\geq \underline{\tau}^2_{\theta_{1:T}\eta} + \sum_{t=1}^{T} \left(\underline{\tau}^2_{\theta_t \phi_{1:T}\eta} - \underline{\tau}^2_{\theta_t \eta} - \underline{\tau}^2_{\phi_{1:T}\eta} \right) \\
&\geq \underline{\tau}^2_{\theta_{1:T}\eta} + \sum_{t=1}^{T} \left(\underline{\tau}^2_{\theta_t \phi_{1:t}\eta} - \underline{\tau}^2_{\theta_t \eta} - \underline{\tau}^2_{\phi_{1:t}\eta} \right) \\
&= V_{A_i}.
\end{aligned}$$

□

The iterative submodel indices for the toy example are then

$$\underline{\tau}^2_{Y_0 vu} \leq V_{A_1} \leq \overline{\tau}^2_{Y_0 vu}$$

and

$$\underline{\tau}^2_{\sigma^2 w} \leq V_{A_1} \leq \overline{\tau}^2_{\sigma^2 w}.$$

Up until now we have only described sensitivity analysis however these methods could be used to perform robustness analysis on the models. We have already seen that by changing some of the submodels and examining the sensitivity of these changes on the model output we can perform robustness analysis (Grimm et al, 2014; Railsback and Grimm, 2012). Additionally by treating some of the assumptions as submodels, for example the choice of a square grid rather than a hexagonal one (Birch et al, 2007), we can then perform global sensitivity analysis on the model to see whether the model output is sensitive to the assumptions.

7.4 Demonstration on Individual based models

We are going to demonstrate these methods on the two individual based models described in Chapter 2.

7.4.1 Bird synchrony model

Jovani and Grimm (2008) built an individual based model that modelled the laying times of birds in order to try and find out which individual traits lead to the synchronicity of breeding amongst colonial birds. They say that in order for the birds to lay their eggs it is important that there is calm and that neighbouring birds assess each others' stress level and when it is calm enough they lay their eggs. The model is described in Section 2.1.

Coupling the model

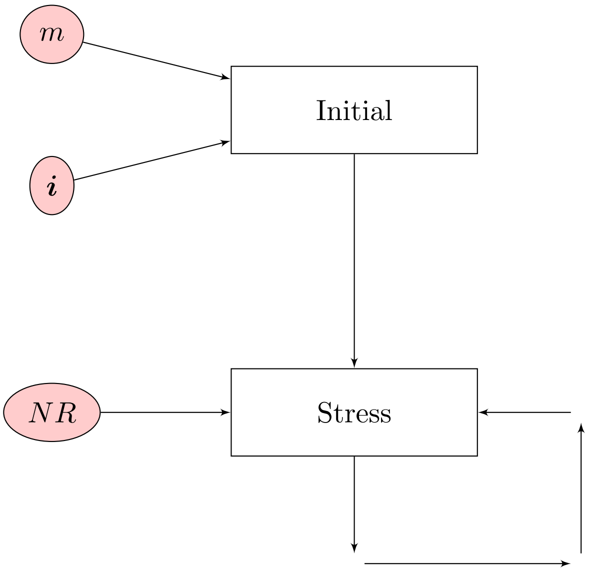

Figure 7.12: The original breeding synchrony model.

The original model has two submodels and three inputs. The Initial submodel, that sets the initial stress levels of the birds, has the maximum stress level, m+10, and the random inputs involved in generating these are $\boldsymbol{i}$. The Stress submodel works out the stress levels for the birds and has one input, the neighbourhood relevance parameter NR. The original model is iterative with an iteration of length 1 and the DAG is shown in Figure 7.12.

Uncertain Inputs

The input distributions for the three inputs are

$$\begin{aligned} NR &\sim U(\cdot|0,1), \\ m &\sim U(\cdot|0,50) \text{ and} \\ \boldsymbol{i} &\sim U(\cdot|0,1) \end{aligned}$$

where $\boldsymbol{i}$ is 15^2 samples from a uniform distribution between 0 and 1. The initial stress levels are then

$$10 + m\boldsymbol{i}.$$

Outputs

We are going to examine six outputs. The first is the mean laying time; this is not a measure of synchrony but we are going to see how the inputs effect this. We will examine the standard deviation of the laying times; this is a measure of the synchrony of the laying times with lower the value of the standard deviation the more synchronised the laying times will be. Another measure of synchrony which we will examine is the interquartile range. If the interquartile range is small then the laying times will be more synchronised.

We will measure the skewness estimated by

$$\left(\frac{(n-1)m_3}{nm_2}\right)^{\frac{3}{2}}$$

where m_i is the sample of the ith moment (Joanes and Gill, 1998; Meyer et al, 2013). If we found that $m_2 = 0$ (i.e. if all of the laying times are the same) then we set the skewness to 0. The skewness is not a measure of synchrony but has been used when describing the laying times of colonial seabirds (Gochfeld, 1980).

In addition to these inputs we will examine the spatial autocorrelation. In order to do this we will use Moran's I function (Moran, 1950). Moran's I function is a measure of global spatial correlation and is defined as

$$\frac{n\sum_i\sum_j w_{ij}(x_i-\bar{x})(x_j-\bar{x})}{\sum_i\sum_j w_{ij}\sum_i(x_i-\bar{x})^2}$$

where x_i is the ith bird's laying time of which there are n, $\bar{x}$ is the mean laying time and w_{ij} is the reciprocal of the distance between bird i and j. If $i = j$ then $w_{ij} = 0$. A Moran's I value close to 1 indicates clustering whereas a value close to -1 indicates dispersion. If we found that all of the laying times were the same we set Moran's I function to be 1. We have two measures of w_{ij}; Moran$_1$ is where the w_{ij} is the Manhattan distance (Black, 2006) between i and j and Moran$_2$ is where w_{ij} is the Chebyshev distance,or chessboard distance (Cantrell, 2000) between i and j.

Sensitivity analysis

Table 7.4 shows the standardised first order and the total effect indices. As the model is quick to simulate from, we used the method described by Liu and Owen (2006) Section 6.3.3 in order to calculate these indices with the number of points simulated as 5000. The iterative submodel sensitivity

		NR	m	$\boldsymbol{i}$
FO	Mean	0	0.99	0
	SD	0.70	0.07	0.01
	IQR	0.69	0.05	0.01
	Skew	0.02	0.03	0.06
	Moran$_1$	0.12	0.32	0
	Moran$_2$	0.11	0.32	0.01
TI	Mean	0	0.99	0
	SD	0.91	0.28	0.02
	IQR	0.92	0.28	0.06
	Skew	0.72	0.88	0.89
	Moran$_1$	0.58	0.83	0.46
	Moran$_2$	0.56	0.82	0.46

Table 7.4: The first order (FO) and total sensitivity indices (TI) for the outputs of the original model. Moran$_1$ is the distance with four neighbours and Moran$_2$ is the distance with eight neighbours.

	Initial	Stress
Mean	0.99	0.00
SD	0.08	0.91
IQR	0.07	0.92
Skew	0.27	0.72
Moran$_1$	0.42	0.58
Moran$_2$	0.43	0.56

Table 7.5: The submodel sensitivity indices for the breeding synchrony model. Moran$_1$ is the distance with four neighbours and Moran$_2$ is the distance with eight neighbours.

sensitivity indices for the Initial and Stress submodels are shown in Table 7.5. We found that the Stress submodel was driving the measures of synchrony and mean was driven entirely by the Initial submodel. This suggest that the stress hypothesis is a good reason why the birds synchronise their breeding behaviour. The other outputs seem to be driven by a mix of inputs especially as over half of the skewness and 0.4 of the Moran I variation is down to the interaction of all three of the inputs.

As mentioned earlier, the submodel sensitivity indices are dependent on the order that you learn the inputs. If we were to learn NR first the corresponding version of the submodel sensitivity indices are shown in Table 7.6. We found that for the Mean, the results are the same for both orderings and differ slightly for the standard deviation and the interquartile range;

however, we did find that the ordering did make a large difference for the skewness and spatial outputs.

	Initial	Stress
Mean	0.99	0.00
SD	0.29	0.70
IQR	0.30	0.69
Skew	0.97	0.02
Moran_1	0.88	0.12
Moran_2	0.88	0.11

Table 7.6: The submodel sensitivity indices if you were to learn NR before m and $\boldsymbol{i}$. Moran_1 is the distance with four neighbours and Moran_2 is the distance with eight neighbours.

Robustness analysis

We now perform robustness analysis on the model. In doing so we need to test the model by changing the model details and submodels. In order to do this we added all of the alternatives submodels described in Section 2.1.2 and tested their effect on the outputs. The model was coupled according to Figure 7.13.

The bird layout in the model is a square grid of size $Size \times Size$ where $Size$ is an integer uniformly between and including 15 and 30. If $Arr = 1$, with probability 0.5, the birds will arrive according to the Stochastic Arrival submodel and will be present at the start of the model, i.e. as in the original model, if $Arr = 0$ with probability 0. The ith bird arrives

$$\min\{n : 1 - (1-\lambda)^n \geq p_i, n \in \mathbb{N}_0\}$$

where $\lambda \sim \text{Gamma}(\cdot|2, 0.25)$ and p_i is the random inputs sampled from a uniform distribution between 0 and 1.

The stress level will be reduced stochastically if $Red = 1$, with probability 0.5, and deterministically if $Red = 0$, with probability 0.5. If the stress level is reduced stochastically, it will be reduced, after the neighbourhood reduction, by

$$1 + \sigma \boldsymbol{s}$$

where $\sigma \sim U(\cdot|0, 1)$ and each element of $\boldsymbol{s}$ is sampled from a standard Gaussian distribution.

The shape of the grid cell seems rather arbitrary so we are going to look at seeing if this has an effect on the model. The birds will have either four or eight neighbours with equal probability and this is denoted in Figure 7.13

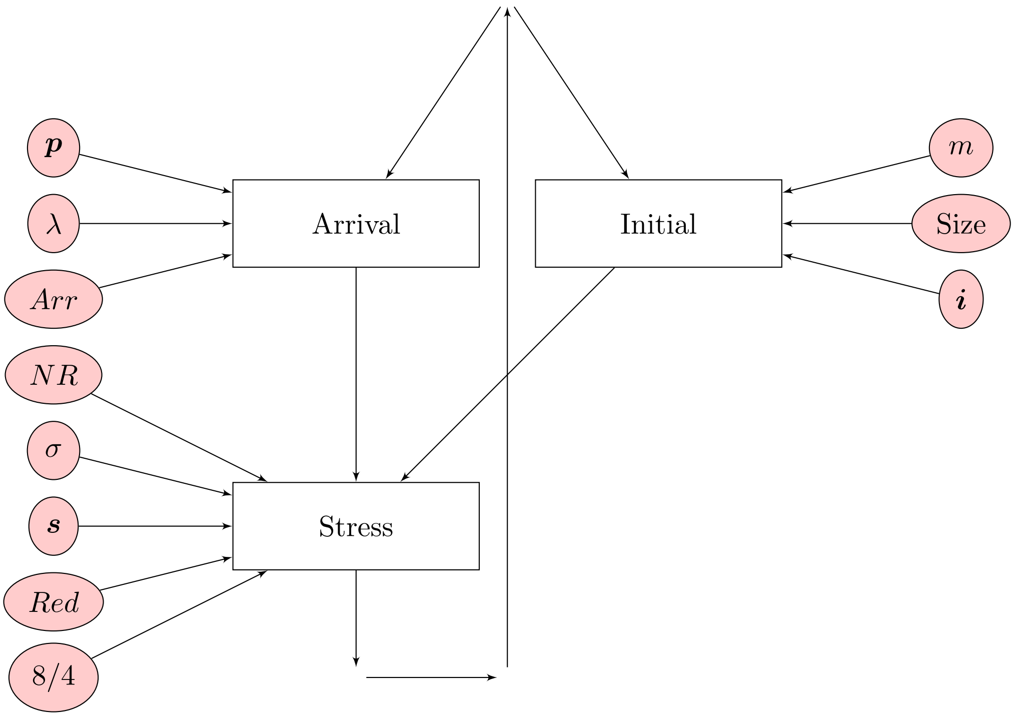

Figure 7.13: The breeding synchrony model with additional submodels.

by 8/4. The inputs relate to the submodels Initial, Arrival and Stress as described in Figure 7.13. The submodels, Arrival and Initial are commutative submodels. We ran the model using the method of Liu and Owen (2006) (Section 6.3.3) with 2000 points as this analysis requires a lot more model runs. The standardised first order and total indices for the model described in Figure 7.13 are shown in Table 7.7.

The total sensitivity indices for the random inputs for the standard deviation, the mean and the interquartile range are zero. This suggests that the the stochasticity of the model has a very small effect on these outputs. For the other outputs the initial random inputs have an effect when interacting with other inputs as too do $\boldsymbol{p}$ and $\boldsymbol{s}$ for the skew of the model. That said, whether the stochasticity is on or not does have an effect on the outputs of the model particularly the stochastic arrival of the birds.

Whether the neighbourhood consists of the 4 neighbours or to 8 neighbours doesn't really make a difference to the outputs that measure the synchronicity of the whole population and the mean, which suggests that the choice of the neighbours does not make a difference for these outputs. However the total indices for the measures of spatial autocorrelation and skewness of the laying times are not negligible and thus this detail of the model

		NR	m	λ	σ	$\boldsymbol{i}$	$\boldsymbol{p}$	$\boldsymbol{s}$	Red	Arr	8/4	Size
FO	Mean	0.10	0.59	0.06	0	0	0	0	0	0.03	0	0
	SD	0.27	0.05	0.09	0.01	0	0	0	0.03	0.24	0.01	0
	IQR	0.43	0.03	0.03	0.01	0	0	0	0.02	0.10	0.01	0
	Skew	0.05	0.08	-0.01	0	0.01	0	0	0.01	0.20	0	0.02
	Moran$_1$	0.02	0.01	0.01	0	-0.01	0	0	0.09	0.18	0	0.02
	Moran$_2$	0.01	0	0.01	0	-0.01	0	0	0.09	0.18	0	0.02
TI	Mean	0.22	0.65	0.13	0	0	0	0	0	0.23	0	0
	SD	0.46	0.16	0.19	0.02	0	0	0	0.04	0.40	0.01	0
	IQR	0.64	0.19	0.10	0.02	0	0	0	0.03	0.21	0.01	0
	Skew	0.50	0.39	0.25	0.04	0.25	0.04	0.08	0.09	0.42	0.10	0.17
	Moran$_1$	0.24	0.58	0.03	0	0.05	0	0	0.51	0.62	0.09	0.11
	Moran$_2$	0.24	0.63	0.02	0	0.06	0	0	0.56	0.65	0.10	0.12

Table 7.7: The first order (FO) and total sensitivity indices (TI) for the outputs of the complete model. Moran$_1$ is the distance with four neighbours and Moran$_2$ is the distance with eight neighbours.

does have an effect on these outputs.

Similarly, the size only affects the results of the Moran I's function and the skewness with only its high orders none negligible. The size of the grid has no effect on the outputs of synchronicity and the mean of the laying times.

In the robustness analysis we have shown that when we added new submodels and changed the details of the model, the outputs relating to the breeding synchrony were still mostly effected by NR. NR also contributes to the mean in the alternative model which was not present in the original model. Having said this the mean is still mostly affected by m. This leads us to believe that this robustness analysis supports the conclusions of Jovani and Grimm (2008). However the model's structure and details are sensitive to the spatial autocorrelation and the skew of breeding times which leads us to believe that the model is not robust to other outputs and therefore further work would be needed on the model in order to answer other questions about breeding times of colonial birds.

Submodel sensitivity

Using the natural ordering, the model is iterative and therefore we can only give bounds for the iterative submodel sensitivity indices (ISSIs) for the Arrival and Stress submodels which are shown in Table 7.8. We can see that for the standard deviation and the interquartile range, the two outputs that most measure the synchronicity of the breeding, that the majority of the variance is caused by the Stress submodel which suggests that the breeding synchrony in the model is down to this submodel.

Output	Initial		Arrival		Stress	
Mean	0.59	0.65	0.15	0.30	0.10	0.22
SD	0.05	0.16	0.41	0.50	0.33	0.52
IQR	0.03	0.19	0.18	0.25	0.48	0.69
Skew	0.09	0.40	0.21	0.54	0.10	0.54
Moran$_1$	-0.01	0.56	0.20	0.64	0.12	0.66
Moran$_2$	-0.02	0.60	0.20	0.66	0.11	0.68

Table 7.8: The ISSIs for each submodel. For the Initial submodel, this can be calculated exactly; for the other two submodels we can only get the bounds. Moran$_1$ is the distance with four neighbours and Moran$_2$ is the distance with eight neighbours

As in the original model, the mean was mostly caused by the Initial submodel. Having said this, the Stress submodel does now cause some uncertainty on the mean.

First bird's stress level

We set all of the alternative submodels on, with the $Size = 15$ and the neighbourhood being the eight neighbours, and examined the stress level of the first bird to arrive after each iteration for the first 40 iterations. We estimated the ISSIs for the three submodels using the method of Liu and Owen (2006) with a sample size of 2000 and these results are shown in Figure 7.14.

The plot shows the minimum ISSIs for the Arrival, Stress and Initial submodels as well as the ISSIs interaction for the Initial and Arrival submodels and the variance that cannot be assigned to any of the submodels. For the stress level after the first iteration the unknown variance is zero because the model, for this output, is not iterative. The variation in the early iterations is mostly attributed to the Arrival submodel with the variation in the middle iterations attributed to a combination of the Initial and Stress submodels. In the final iterations we are unable to distinguish in which submodel the variance is attributed to. Of the variance that we can distinguish, most was attributed to the interaction of the Initial and Arrival submodels.

Test with re-parameterisation

In this section we demonstrate sensitivity analysis on a submodel when the submodels have the same input distribution, like in Figure 7.5. We will test what effect the birds' arrival have on the model output. We will assume, possibly ecologically incorrectly, that the birds have an initial stress level that is determined at the beginning of the simulation and is reduced before the birds arrive with $NR = 0$. When stochastic arrival is on then the model runs as above with initial values $\boldsymbol{i}$ but when stochastic arrival is off the initial values will be

$$i_j + \sum_{k=1}^{a_j} \sigma\epsilon_{jk} + a_j$$

where a_j is the arrival time of bird j. This is the initial value plus the stochastic reductions before arrival plus the arrival time. If stochastic reductions are off then $\epsilon_{jk} = 0 \quad \forall j, k$. The idea is that when the bird j arrives at time a_j, when stochastic arrival is on, it will have the same stress level if the stochastic arrival was off and $NR = 0$ at time a_j. We are unable to distinguish between the variance caused by the input and the variance caused by the submodel decision. As when we examined the individual stress level, we set $Size = 15$ and only examined the eight neighbours and using the method of Liu and Owen (2006), with a sample size of 2000, we estimated the standardised Sobol' indices which are shown in Table 7.9. It shows that whether birds arrive at different times in the model has an effect on the model outputs.

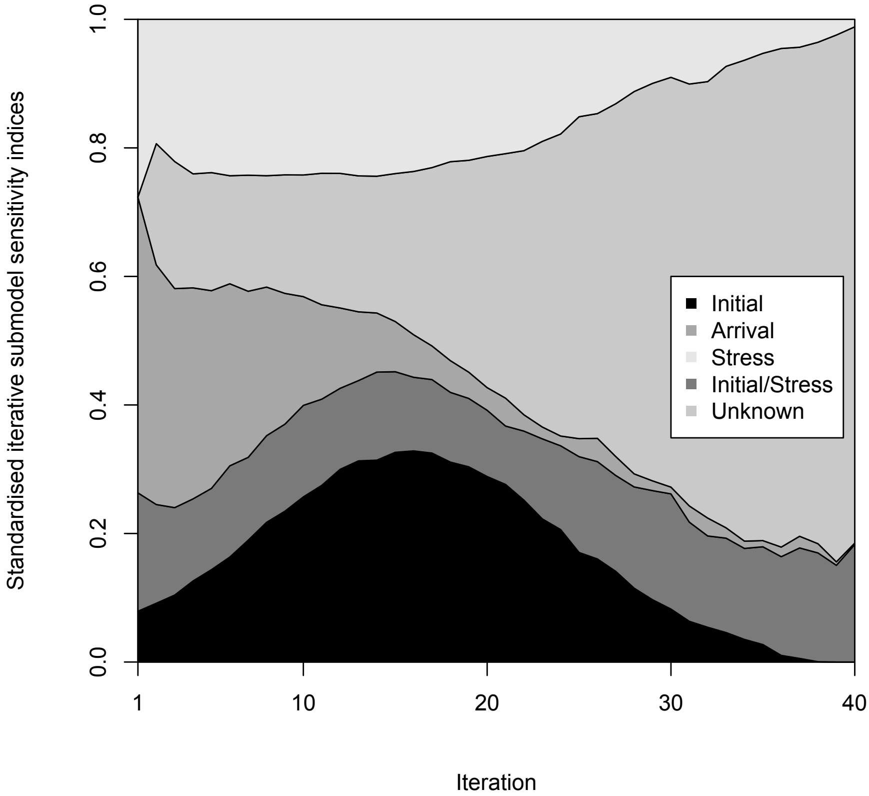

Figure 7.14: The standardised iterative submodel sensitivity indices for the stress level of the first bird to arrive.

Output	Arrival	Other inputs	Interaction
Mean	0.32	0.57	0.11
SD	0.23	0.58	0.19
IQR	0.09	0.75	0.16
Skew	0.23	0.37	0.40
Moran$_1$	0.09	0.55	0.36
Moran$_2$	0.07	0.55	0.38

Table 7.9: The sensitivity indices of the arrival and all the other parameters.

7.4.2 Woodhoopoe model

We are going to perform sensitivity and robustness analysis on the simplified woodhoopoe model described in Section 2.2.

Coupling the model

We coupled the model as demonstrated in the DAG in Figure 7.15. The submodels Leave, Predation and Left/Right for a set that is a commutative submodel set with the Subordinate Order submodel. The age of woodhoopoe j at the beginning of the model is

$$\min\left\{n : \frac{n}{24} \geq u_{ij}, n \in \mathbb{N}^{+}\right\}.$$

A subordinate j leaves its territory at time t if $u_m^{j\;t} < \theta_2$. If subordinate j leave their territory, they are killed by a predator if $u_p^{j\;t} < \theta_3$. If subordinate j leaves their territory at time t and survives the predation part then they go to the left territory if $u_s^{j\;t} < 0.5$ and right otherwise. The subordinates move in an order with the subordinate with the lowest $\boldsymbol{u_o^{\cdot\;t}}$ moving first and the second lowest moving second and so on. When there are n newborns in territory k then the jth newborn (for $j = 1 \ldots n$) is a male if $u_b^{j\;k\;t}$. Finally woodhoopoe j dies of natural mortality if $u_d^{j\;t} < \theta_1$. All of the random inputs are sampled uniformly between 0 and 1.

Uncertain Inputs

The input distributions for the three parameters are the same as described in Section 4.4.3,

$$\begin{aligned}\theta_1 &\sim U(\cdot|0, 0.05),\\ \theta_2 &\sim Beta(\cdot|1, 1) \quad \text{and}\\ \theta_3 &\sim U(\cdot|0, 0.5).\end{aligned}$$

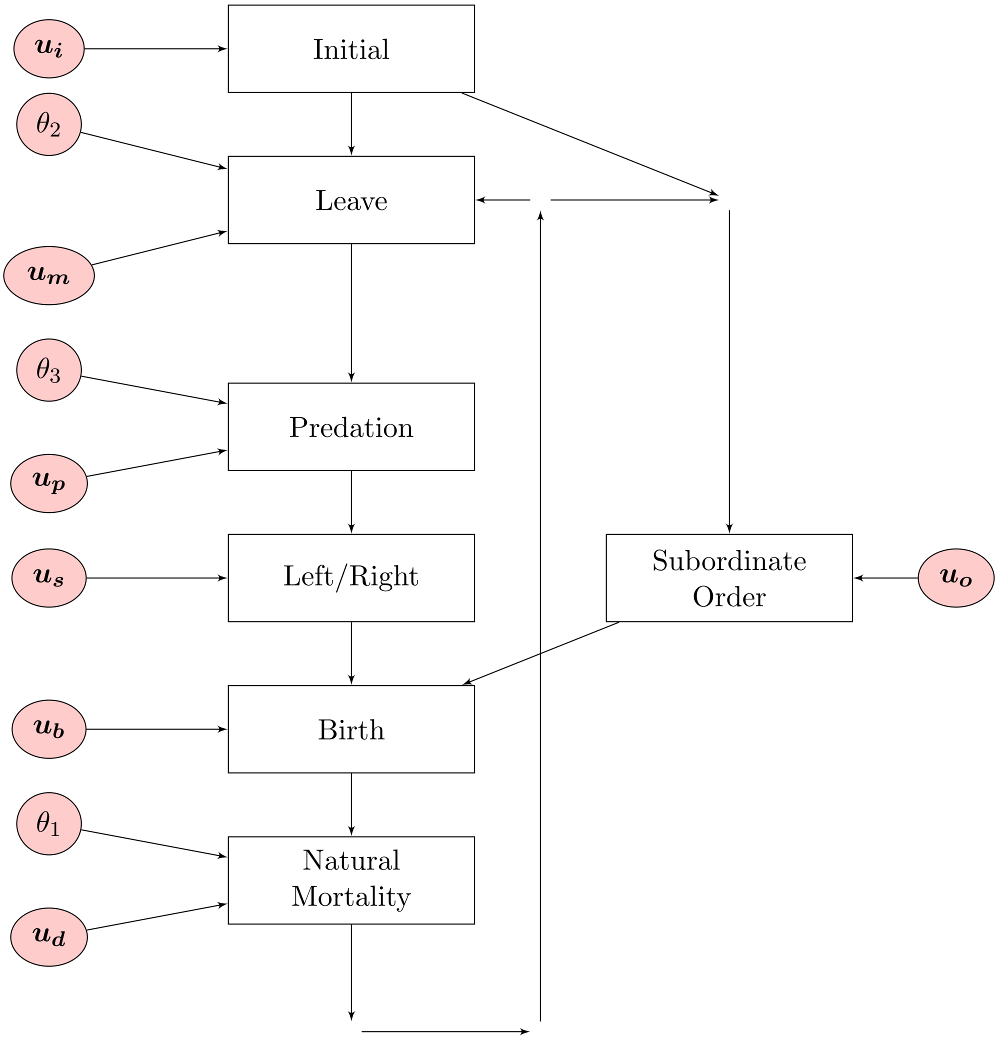

Figure 7.15: The woodhoopoe model.

Outputs

We are going to examine the same summary statistics as described in Section 4.4.3. These are the 0.25 (a), 0.5 (b) and 0.75 (c) quantiles of the population size; the minimum (d), mean (e) and maximum (f) of the number of groups with vacant dominant positions and the 0.25 (g), 0.75 (h) quantiles and mode (i) of the group sizes evaluated once a year (in the 11th month) for the 20 years of the simulation. If the model reaches its absorbing state all of the outputs are still defined. However, whilst performing this analysis, we did not run an input set that resulted in the model reaching its absorbing state.

Variance based sensitivity analysis

We estimated the sensitivity indices using the method described by Tarantola et al (2006b) (Section 6.3.3) with 2000 samples from each input. We found that the majority of the variance is caused by the parameters; this is shown in Table 7.10. It also shows that for all of the outputs except the ones that count the number of vacant dominant spots that the total sensitivity of the random inputs is nearly zero which means that the variance down to the aleatory parts is quite small.

We estimated the submodel sensitivity indices and the results are shown in Table 7.11. We found that most of the variance was attributed to the submodels with the parameters in them, namely the Leave, Predation and Natural Mortality submodels. The other submodels had very little impact on the model output.

Robustness analysis

We now perform robustness analysis by experimenting with various submodels. There are a number of alternative decisions that could have been made when formulating the model. We are going to experiment with different versions of the left/right submodel. Each of the following submodels will have an equal probability of being used in the model:

1. The subordinate will move as described above.

2. In this submodel a subordinate will attempt to move to any other territory at random. More formally, subordinate j at time t and at the ath attempt during this iteration will move to the kth territory that it hasn't visited in this iteration (with territory 1 being the first and territory 24 being the 24th) such that

$$k = \min \left\{ n : \frac{n}{25-a} \geq b_a, n \in \mathbb{N}^+ \right\}$$

		a	b	c	d	e	f	g	h	i
FO	θ_1	0.42	0.36	0.32	0.48	0.66	0.62	0.40	0.28	0.21
	θ_2	0.19	0.21	0.23	0.02	0.02	0.02	0.17	0.21	0.29
	θ_3	0.24	0.25	0.25	0.06	0.06	0.07	0.21	0.23	0.28
	$\boldsymbol{u_i}$	0	0	0	0	0	0	0	0	0
	$\boldsymbol{u_o}$	0	0	0	0	0	0	0	0	0
	$\boldsymbol{u_m}$	0	0	0	0	0	0	0	0	0
	$\boldsymbol{u_p}$	0	0	0	0	0	0	0	0	0
	$\boldsymbol{u_s}$	0	0	0	0	0	0	0	0	0
	$\boldsymbol{u_b}$	0	0	0	0	0	0	0	0	0
	$\boldsymbol{u_d}$	0	0	0	0.01	0.01	0.01	0	0	0
TI	θ_1	0.56	0.53	0.51	0.92	0.91	0.91	0.60	0.53	0.42
	θ_2	0.30	0.34	0.37	0.19	0.11	0.11	0.33	0.41	0.44
	θ_3	0.31	0.34	0.35	0.36	0.24	0.25	0.32	0.37	0.41
	$\boldsymbol{u_i}$	0	0	0	0	0	0	0	0	0
	$\boldsymbol{u_o}$	0	0	0	0.04	0.03	0.02	0	0.01	0
	$\boldsymbol{u_m}$	0	0	0	0.09	0.06	0.04	0.01	0.03	0.01
	$\boldsymbol{u_p}$	0	0	0	0.10	0.07	0.05	0.01	0.03	0.01
	$\boldsymbol{u_s}$	0	0	0	0.08	0.06	0.04	0.01	0.02	0
	$\boldsymbol{u_b}$	0	0	0	0.11	0.07	0.06	0.01	0.02	0
	$\boldsymbol{u_d}$	0.01	0.01	0.01	0.23	0.14	0.14	0.02	0.04	0.01

Table 7.10: The first order and total sensitivity indices for the outputs of the woodhoopoe model. The letters a-i correspond to the summary of the output described in 7.4.2.

where

$$b_a = (25 - a)\left(b_{a-1} - \frac{k_{a-1}}{25 - a}\right)$$

and the subordinate visited the k_{a-1}th territory searched at attempt $a - 1$ for $a \geq 2$ and $b_1 = u_s^{j\ t}$.

3. Subordinate will move left or right at their first search, with equal probability; every time they leave from this moment they will move the opposite way to their previous move. Subordinate j will move left on its αth time that they leave their home territory if α is odd and $u_s^{j1} < 0.5$ or if α is even and $u_s^{j1} > 0.5$ otherwise it will move right.

In addition to this we will look at adjusting the number of territories a subordinate searches in before they return back to their original territory which is 5 in the original model. We will say that the number is discretely uniform from 1 to 6 as du Plessis (1992) observed that the maximum distance a woodhoopoe travelled to settle in a new territory was 6 territories away.

	a	b	c	d	e	f	g	h	i
Leave	0.19	0.21	0.23	0.02	0.02	0.02	0.17	0.21	0.29
	0.30	0.34	0.37	0.19	0.11	0.11	0.33	0.41	0.44
Predation	0.24	0.25	0.25	0.06	0.06	0.07	0.21	0.23	0.28
	0.31	0.34	0.35	0.37	0.25	0.27	0.32	0.37	0.41
Left/Right	0	0	0	0	0	0	0	0	0
	0	0	0	0.08	0.06	0.04	0.01	0.02	0
Subordinate Order	0	0	0	0	0	0	0	0	0
	0	0	0	0.04	0.03	0.02	0	0.01	0
Birth	0	0	0	0	0	0	0	0	0
	0	0	0	0.11	0.07	0.06	0.01	0.02	0
Natural	0.42	0.36	0.32	0.55	0.67	0.65	0.40	0.28	0.21
Mortality	0.56	0.53	0.51	0.92	0.91	0.91	0.60	0.53	0.42

Table 7.11: Upper and lower bounds for the iterative submodel sensitivity indices for the outputs of the woodhoopoe model. The letters a-i correspond to the summary of the output described in 7.4.2.

Additionally, the order in which the subordinates attempt to leave their territory could be important. We examined three alternative submodels with equal probability of being used:

1. The subordinates move in the same order as described above.

2. The subordinates move in the order they entered the model. This essentially means that the older subordinates move first and therefore have a greater chance of becoming dominants in other territories.

3. Another alternative, is to run the subordinates in a random order but this order remains the same for the rest of the simulation. The woodhoopoe alive in the model with the lowest $\boldsymbol{u}_{\boldsymbol{o}}^{\cdot 1}$ will always attempt to move first and the woodhoopoe with the second lowest will always move second and so on.

We found that the choice of submodel and the order of which the birds moved did not have an effect on the results. In fact the total and first order indices were 0 to 4 decimal places for all of the outputs for the submodel selection parameters.

We did find that the number of attempts that a woodhoopoe that left its territory has to find a new territory with a vacant dominant slot did affect the results. It had a small effect on the population and the group sizes (about 0.05 for each of the outputs) and had a large effect on the number of vacant dominant positions, the first order index being about 0.25 for the three summaries of the vacant positions. The other parameters (described in the original model) contributed roughly 0.5 meaning that quite a lot of the variance was down to the interactions between the number of searches made by a subordinate and the other inputs.

In the simplified model, Railsback and Grimm (2012) do not discuss the implication of this seemingly arbitrary parameter. The decision to fix this value at 5 causes quite a large difference in the model output and thus making any conclusions from the model less reliable, especially about the number of vacant dominant positions.

7.5 Discussion

In this chapter we developed a method of performing variance-based sensitivity analysis on stochastic models by treating the random inputs as an additional parameter, refining a method described in Iooss and Ribatet (2009). Rather than in Iooss and Ribatet (2009), where the random inputs are one "macroparameter", we have coupled the random inputs into groups of inputs (Jacques et al, 2006). By coupling the random inputs, we were able to partition the variance not only between aleatory and epistemic parts, but between the different parts of the aleatory uncertainty, essentially allowing us to identify which random inputs are important and which are not.

We demonstrated this method on two stochastic models, a queuing model and the Ricker model. In the queuing model we were able to say that the random inputs that determined the arrival times of customers are more influential that the random inputs of the service time. In the Ricker model we showed that a complex interaction of the random inputs and parameters had the highest sensitivity index, something that would be impossible to find using the "macroparameter method" or meta-model methods (Iooss and Ribatet, 2009; Marrel et al, 2012).

If we were uncertain about a what form a submodel should take, we demonstrated a method of treating this decision as an unknown input. If multiple submodels have the same inputs or same prior predictive input distributions but are parameterised differently, then it may be useful to distinguish between variance accounted by the choice in submodel and the variance accounted by the inputs to the submodel.

In the method described in this chapter, all potential submodels need to be fully specified and written into the model which can be a time consuming task (Grimm and Railsback, 2005; Railsback and Grimm, 2012). Furthermore, prior distributions need to be specified for the submodels which may not be such an easy task and as we have shown empirically, the sensitivity indices are sensitive to the input distributions.

The method of submodel choice described in this chapter was developed in order to perform robustness analysis (Weisberg, 2006) by testing the effects of the details of complex models on the output of the model and seeing what happens if we change some of the submodels. We tested two IBMs, adding uncertainty to the model by changing some of the submodels and details of the model. We then performed sensitivity analysis on the output of the new model. This sensitivity analysis partitions the variance of the new model, rather than measuring the additional variance caused by the changes between the old model and the new.

The robustness analysis is specific to an output of the model. If a model is used to make a specific decision then this decision should be robust to the model details and structure but other outputs need not. However, it is often the ultimate goal of these types of models to build a virtual laboratory. If a model is robust enough to be used as a virtual laboratory, adding new plausible submodels should not have a large effect on any of the outputs that the model could be used to test. What effect a new submodel requires for a model to pass robustness analysis is subject to the user. For example, in the bird synchrony model, we concluded that the measures of synchrony were robust in this model despite the addition of the arrival having an effect on the model output.

We defined the submodel sensitivity index and the iterative submodel sensitivity index as measures of the sensitivity to the output a submodel gives. This gives a way of identifying which components of the model are "live", where the variance is added to the model, and which parts are "dead" and do not contribute to the variation (Tarantola et al, 2002). This can enable the model developer to see which submodels are important during the model development phase.

The submodel sensitivity indices are sensitive to the ordering of the inputs. In this chapter we used them with the order defined by the model structure, however, it could be plausible that you could, from field experiments, learn a number of the true values of the uncertain inputs or even intermediate state of the model. The corresponding version of submodel sensitivity indices could then be used to decide what would then be the next best inputs to learn.

For commutative submodels and submodel sets we cannot assign the higher-order Sobol' indices to any submodel but only collectively. The commutative submodels interact in the next submodel but as we treat the input to that submodel, Q as an individual input and it is possible to learn this without running the submodel it does not make sense to say that this variance is caused in the next submodel. Although all of the IBMs that we examine here have commutative subomodels or commutative submodel sets this is not always the case for example the Cortés-Avizanda et al (2014) and Hovel and Regan (2008) models.

When the model is iterative it can mean that we are unable to estimate the iterative submodel sensitivity indices exactly but it is possible to find bounds on the value of them. The submodels can be seen as a collection of repeated submodels. For example the mortality submodel sensitivity index in the woodhoopoe model is the sum of the mortality submodel sensitivity indices for every time.

Sometimes a parameter is involved in more than one submodel. If this is the case you can either group the submodels or further bounds would be needed to be calculated as it is not possible to define which submodel the Sobol' index of this parameter belongs to. Additional work could be done to separate this by using methods of variance based sensitivity of correlated inputs (Kucherenko et al, 2012).

Chapter 8

Conclusions and Future work

In this thesis we have looked at methods of quantifying uncertainty in ecological simulation models. As the probabilistic behaviour in this class of models is implicit, parameter estimation and sensitivity analysis are difficult to do. We have developed novel methods of improving parameter estimation when the likelihood is intractable and developed a method of performing sensitivity and robustness analysis on stochastic simulation models. We demonstrated these methods on a selection of “toy” and realistic models. We have also investigated a size-structured ecosystem model consisting of numerically solved differential equations, carrying out fully Bayesian inference using a combination of optimisation and MCMC.

8.1 Thesis summary

In Chapter 1 we introduced the reasons behind building a simulation model rather than a statistical model. It is often the case that modellers are wondering why something happens and building a simulation model better answers that question. Additionally it is often the case that the aim is to build a virtual laboratory so that experiments that would have in reality possibly irreversible or expensive consequences can be performed. We also introduced a specific type of simulation model, individual-based models (IBMs). IBMs model the individuals in the system explicitly in order to see what individual behaviours result in the emergent behaviour of the whole system.

Two IBMs that were used later in the thesis are described using the ODD protocol (Grimm et al, 2006, 2010) in Chapter 2. The woodhoopoe model was used to demonstrate the method parameter estimation that we developed in Chapter 3. We perform sensitivity and robustness analysis on these two models in Chapter 7.

We gave an overview of methods of parameter estimation in Chapter 3. We began by introducing MCMC methods and describing ways of speeding them up. We showed a novel doubly parallel tempering algorithm that is a mix between the parallel MCMC algorithm, described by Cui et al (2011), and the parallel tempering algorithm (Swendsen and Wang, 1986). This algorithm can be used when the target distribution is multi-modal and the likelihood is moderately expensive to calculate.

Most simulation models have an intractable likelihood meaning that the likelihood either cannot be written down or is expensive to calculate which means that standard MCMC methods, as described in Section 3.2, cannot be used. We described an overview of a method to overcome this, namely ABC, in Section 3.3. We described the idea of ABC and some algorithms that improve the efficiency of ABC. One such algorithm is ABC-MCMC which is a hybrid between ABC and MCMC (Marjoram et al, 2003). This algorithm is not favoured among ABC users as it suffers from poor mixing when the Markov chain is in the tails of its distribution (Sisson et al, 2007).

In Chapter 4 we developed a method of improving ABC-MCMC by considering the random inputs as additional parameters and coupled them so that small movements in parameter space resulted in small movements in the model output. We also introduced a Gibbs step that enables the parameters to move without a further run of the model. We showed empirically that this improved the mixing of the algorithm with the greatest improvements being in the tails of the distribution.

We fitted a multi-species size-based model of the North Sea (Blanchard et al, 2014) to landings data using a Bayesian framework in Chapter 5. This involved using a mixture of space filling and optimisation algorithms before using the doubly parallel tempering algorithm which we described in Section 3.2.3. This work addresses an important issue in size-based modelling: parameter estimation or calibration.

We give an overview of sensitivity analysis in Chapter 6, mainly focusing on variance-based sensitivity indices and methods of estimating them.

The majority of models that we wanted to examine and perform sensitivity analysis on are stochastic and few methods of dealing with stochastic models exist (Iooss and Lemaître, 2015). We used the idea of coupling the random inputs, as described in Chapter 4, to treat the random inputs, the stochastic parts of the model, as additional inputs and hence to quantify sensitivity of model outputs to stochasticity.

We also treated submodel uncertainty, uncertainty about which submodel to use in the model, as an additional parameter. This enabled us to perform robustness analysis by testing alternative submodels whether the different versions of submodels represent genuine scientific uncertainty or essentially arbitrary modelling or coding choices. We also defined the iterative submodel sensitivity analysis, which is the amount of variance that is created in each submodel and shows which parts of the model are important to model output which could be useful when developing these models, which in practice are often iterative in form.

8.2 Discussion and further work

8.2.1 CG-ABC

We found that CG-ABC had a much greater improvement for the queuing model than for the woodhoopoe model. Additionally we found that for the queuing model the total sensitivity index of the random inputs, $\widetilde{\overline{\tau}^2_u}$, was much larger than that in the woodhoopoe model. We postulate that this is not a coincidence and the absolute value of $\widetilde{\overline{\tau}^2_u}$, as opposed to the standardised value, compared to the tolerance level, ϵ, should give an idea of how much CG-ABC improves over ABC-MCMC especially in the region of non-negligible posterior mass.

By treating the random inputs as parameters we are creating an MCMC algorithm that is analogous to other MCMC problems where the likelihood is unknown but can be, at the cost of one run of the model, calculated exactly. This means that we can use more sophisticated MCMC methods to optimise the algorithm such as building an emulator (Kennedy and O'Hagan, 2001), Multiple-try MCMC (J. Liu and Wong, 2000) and Differential Evolution Adaptive Metropolis (DREAM) (ter Braak and Vrugt, 2008; ter Braak, 2006) just to name a few.

We showed empirically that the CG-ABC algorithm improves the performance of the ABC-MCMC algorithm in 3 dimensions. We believe that it would be be good in higher dimensional problems compared to other ABC methods.

8.2.2 Size-based models

We fitted the multi-species size-based model to landings observations by formulating a likelihood relating the model output to reality. Another common way of performing parameter estimation for this class of models is to fit the model to indicators, such as the size of the slope in the spectrum, and check that the indicators have plausible values similar to ABC (Pope et al, 2006; Thorpe et al, 2015; Hall et al, 2006). Some indicators, or summary statistics as they are known in the ABC literature, are chosen and then the uncertain parameters are fitted to replicate these indicators. This leads to another question in this field: which indicators should we fit to the model to in order to best learn the uncertain parameters? We will discuss this in the Section 8.2.3.

It is also important to take account of the uncertainty of the parameters fitted using these methods as this will lead to uncertainties in the model output (Harwood and Stokes, 2003). Formal measures of uncertainty will enable multi-species forecasts to be reported to decision makers in a manner that is comparable to single-species decision tables. This would help further develop the use of formal risk assessment in ecosystem approaches to fisheries management, which has been fairly limited to date but is a burgeoning area of research (Plagányi et al, 2014).

One of the additional problems with performing parameter estimation is the computational expense of these models. A potential method that we did not use in the work in this thesis but could be used on other size-based models is the use of an emulator, a statistical stochastic model in order to represent the more complex ecological model (Kennedy and O'Hagan, 2001). This statistical model increases the uncertainty of the model output but this uncertainty is also taken account of when estimating the model output. Building an emulator "gets the most of the model from a finite number of runs". Currently size-based models are quick to run compared to some of the models that the emulation methods have been developed for (for example Vernon et al, 2010) which should enable the emulator to become more accurate and less uncertain.

Local sensitivity analysis is common for this class of models but it is important to examine sensitivities caused by some interactions of parameters (Saltelli et al, 2004). Although considered important in the field and performed on other marine ecosystem models (Morris et al, 2014), global sensitivity has not been done with size-based models. One of the reasons that this has not been done is the computational cost of the model and the high number of parameters as many methods of global sensitivity analysis are very expensive.

Some ongoing work uses emulators to fit a sized-based model to catch data as well as some indicators using a Bayesian framework (Spence, unpublished). Additionally some work is being done that tests what effect adding a temperature dependent growth submodel has on to the multi-species North Sea model using methods developed in Chapter 7 (Spence, unpublished).

8.2.3 Sensitivity and Robustness analysis

It is possible to use variance-based sensitivity analysis to estimate the (partial) Expected Value of Perfect Information (Strong and Oakley, 2013; Strong et al, 2014). This means that by treating the size-based model as a decision problem it is possible to determine which data sets or indicators need to be collected in order to reduce the uncertainty of the decision to be made.

When we perform variance based sensitivity analysis on stochastic models we showed that it doesn't matter how the random inputs are coupled, so long as the prior predictive distribution is the same, the Sobol' indices still remain the same. However this does not seem to be the case for local methods such as derivative based sensitivity (Sobol' and Kucherenko, 2009) and the Morris method (Morris, 1991). Having said this we believe that coupling the random inputs could greatly improve the estimation of local sensitivity as the estimates of the derivatives will be smoother, as with the method of common random numbers (Owen, 2013b).

The methods we developed and demonstrated here give a way of testing different versions of submodels and arbitrary decisions. The order in which submodels are run within a ABM is sometimes not obvious and in reality the submodels run simultaneously, possibly using information from the other submodels as they are been run. For example in the Hovel and Regan (2008) model, three entities move one at a time when in reality they move simultaneously interacting with each other along the way. It is not possible to do this using a computer so the submodels have to be run one at a time for one time step and thus the order in which the submodels are run should not have a large effect on the output.

If the order does have an effect on the results then the submodels could be run with smaller time steps. This would mean that the submodels would be able to interact with each other more often thus acting more and more like they are running simultaniously. However as the time step decreases the amount of computation required increases and although specialist software does exist that enable ABMs to be run optimally (Coakley et al, 2012), there does become a critical point when it becomes too computationally expensive to run the model. Alternatives could be to change the order of the submodels every time step or to use the Stromer-Verlet or leapfrog integrator, where the order the submodels are run reverse every time step, to run the models (Verlet, 1967).

It is possible to incorporate uncertainty by adding individual heterogeneity to the model. This was done in the full woodhoopoe model (Neuert et al, 1995). The number of groups searched by an individual subordinate, the parameter that we found had a large effect on the number of vacant dominant positions in the simplified model, was stochastically simulated uniformly between 1 and 6 for each search flight rather than being fixed as in the simplified model. This takes advantage of the heterogeneity of individuals in the model and the system which is one of the advantages of building IBMs. It would be interesting to see what effect individual heterogeneity has on some uncertainty of outputs in IBMs.

In this thesis we have explored a number of statistical themes of importance for complex simulation models. We have extended the repertoire of parameter estimation methods and assessed their properties for a number models. We defined a method of formally quantifying uncertainties for stochastic models and a method of assigning this uncertainty to different submodels within a larger simulation model. We developed a method of quantifying the uncertainty caused by modeller decisions whether it is the choice of submodels or the arbitrary modelling choices. Taken together, these techniques serve to improve the understanding and handling of uncertainty in practical simulation modelling.

Bibliography

Akaike H (1974) A new look at the statistical model identification. IEEE Transactions on Automatic Control 19:716–723

Andersen KH, Beyer JE (2006) Asymptotic Size Determines Species Abundance in Marine Size Spectrum. American Society of Naturalists 168(1):54–61

Andersen KH, Beyer JE (2013) Size structure, not metabolic scaling rules, determines fisheries reference points. Fish and Fisheries DOI:10.1111/faf.12042

Andersen KH, Pedersen M (2009) Damped trophic cascades driven by fishing in model marine ecosystems. Proceedings of the Royal Society of London B: Biological Sciences 227:795–802

Andersen KH, Ursin E (1977) A multispecies extension to the Beverton and Holt theory of fishing, with accounts of phosphorus circulation and primary production. Meddelelser fra Danmarks Fiskeri- og Havundersøgelser 7(319-345)

Andersen KH, Zhang L (2011) The size spectrum approach to physiology structured community modeling. Tech. rep., Technical University of Denmark

Andrieu C, Roberts GO (2009) The pseudo-marginal approach for efficient Monte Carlo computations. Annals of Statistics 41:697–725

Andrieu C, Thoms J (2008) A tutorial on adaptive MCMC. Statistics and Computing 18:343–373

Andrieu C, Doucet A, Holenstein R (2010) Particle Markov Chain Monte Carlo methods. Journal of Royal Statistical Society B 72:269–342

Andrieu C, Doucet A, Lee A (2012) Discussion of Constructing summary statistics for approximate Bayesian Computation: semi-automatic Approximate Bayesian computation. Journal of Royal Statistics Society 74:451–452

Baragatti M, Grimaud A, Pommeret D (2013) Likelihood-free parallel tempering. Statistics and Computing 23(4):535–549

Beaumont MA (2003) Estimation of Population Growth of Decline in Genetically Monitored Populations. Genetics 164:1139–1160

Beaumont MA (2010) Approximate Bayesian Computation in Evolution and Ecology. Annual Review of Ecology, Evolution, and Systematics 41:379–406

Beaumont MA, Zhang W, Balding DJ (2002) Approximate Bayesian Computation in population genetics. Genetics 162:2025–2035

Beaumont MA, Cornuet JM, Marin JM, Robert CP (2009) Adaptive Approximate Bayesian Computation. Biometrika 96:983–990

Beck MB (1987) Water quality modeling: a review of the analysis of uncertainty. Water Resources Research 23:1393–1442

Benoît E, Rochet MJ (2004) A continuous model of biomass size spectra governed by predation and the effect of fishing on them. Journal of Theoretical Biology 226:9–21

Berk R, Bickel P, Campbell K, Keller-McNutly S, Kelly E, Sacks J (2002) Workshop on statistical approaches for the evaluation of complex computer models. Statistical Science 17:173–192

Bigg GR, Cunningham CW, Ottersen G, Pogson GH, Wadley MR, Williamson P (2008) Ice-age survival of atlantic cod: agreement between palaeoecology models and genetics. Proceedings of the Royal Society of London B: Biological Sciences 275(1631):163–173, URL http://rspb.royalsocietypublishing.org/content/275/1631/163.abstract

Birch CPD, Oom SP, Beecham JA (2007) Rectangular and hexagonal grids used for observation, experiment and simulation in ecology. Ecological Modelling 206(3-4):347–359

Black PE (2006) Manhattan distance. In: Pieterse V, Black PE (eds) Dictionary of Algorithms and Data Structures, URL http://www.nist.gov/dads/HTML/manhattanDistance.html

Blanchard JL, Andersen KH, Scott F, Hintzen NT, Piet G, Jennings S (2014) Evaluating targets and trade-offs among fisheries and conservation objectives using multispecies size spectrum model. Journal of Applied Ecology 51(3):612–662

Blum MGB (2010) Approximate Bayesian Computation: a Nonparametric Perspective. Journal of the American Statistical Association 105(491):1178–1187

Blum MGB, François O (2010) Non-linear regression models for Approximate Bayesian Computation. Statistics and Computing 20:63–73

ter Braak CJF (2006) A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces. Statistics and Computing 16:239–249

ter Braak CJF, Vrugt JA (2008) Differential Evolution Markov Chain with snooker updater and fewer chains. Statistics and Computing 18:435–446

Campolongo F, Cariboni J, Saltelli A (2007) An effective screening design for sensitivity analysis of large models. Environmental Modelling & Software 22:1509–1518

Cantrell CD (2000) Modern Mathematical Methods for Physicists and Engineers. Cambridge University Press

Cappé O, Guillin A, Marin JM, Robert CP (2004) Population Monte Carlo. Journal of Computational and Graphical Statistics 13:907–929

Cariboni J, Gatelli D, Liska R, Saltelli A (2007) The role of sensitivity analysis in ecological modelling. Ecological Modelling 203:167–182

Chen Q, Mynett AE (2003) Effects of cell size and configuration in cellular automata based prey-predator modelling. Simulation Modelling Practice and Theory 11:609–625

Chen Q, Han R, Ye F, Weifeng L (2011) Spatio-temporal ecological models. Ecological Informatics 6:37–43

Clark JS, Carpenter SR, Barber M, Collins SL, Dobson AP, Foley J, Lodge DM, Pascual M, Pielke Jr RA, Pizer W, Pringle CM, Reid W, Rose K, Sala O, Schlesinger WH, Wall D, Wear DN (2001) Ecological forecasts: an emerging imperative. Science 293:657–660

Coakley S, Gheorghe M, Holcombe M, Chin S, Worth D, Greenough C (2012) Exploitation of High Performance Computing in the FLAME Agent-Based Simulation Framework. In: Proceedings of the 14th International Conference on High Performance Computing and Communications

Cortés-Avizanda A, Jovani R, Donázar JA, Grimm V (2014) Bird sky networks: How do avian scavengers use social information to find carrion? Ecology 95(7):1799–1808

Csilléry K, Blum MGB, Gaggiotti OE, François O (2010) Approximate Bayesian Computation (ABC) in practice. Trends in Ecology and Evolution 25(7):410–418

Cui T, Fox C, Nicholls GK, O'Sullivan MJ (2011) Using Parallel MCMC Sampling to Calibrate a Computer Model of a Geothermal Reservoir. Tech. Rep. 686, University of Auckland Faculty of Engineering

Cukier RI, Levine H, Shuler KE (1978) Nonlinear sensitivity analysis of multiparameter model systems. Journal of Computational Physics 29(1):1–42

Cunningham PC (2007) A sensitivity analysis of an individual-based trout model. Master's thesis, Humbolt State University

Damuth J (1981) Population density and body size in mammals. Nature 290(23):699–700

Del Moral P (2004) Feynman-Kac Formulae Genealogical and Interacting Particle Systems with Applications. Springer: Probability and its Applications, New York

Del Moral P, Doucet A, Jasra A (2012) An adaptive sequential monte carlo method for approximate Bayesian computation. Statistics and Computing 22(5):1009–1020

Devroye L (1986) Non-Uniform Random Variate Generation. Springer-Verlag

Diggle PJ, Gratton RJ (1984) Monte Carlo Methods of Inference for Implicit Statistical Models. Journal of Royal Statistical Society B 46(2):193–227

Doucet A, Johansen AM (2011) A tutorial on particle filtering and smoothing: fifteen years on

Drechsler M (1998) Sensitivity analysis of complex models. Biological Conversation 86:401–412

Duane S, Kennedy AD, Pendleton BJ, Roweth D (1987) Hybrid Monte Carlo. Physics Letters B(195):216–222

Efron B, Stein C (1981) The Jackknife Estimate of Variance. The Annals of Statistics 9:586–596

Fearnhead P (2004) Filtering recursions for calculating likelihoods for queues based on inter-departure time data. Statistics and Computing 14(3):261–266

Fearnhead P, Prangle D (2012) Constructing summary statistics for approximate Bayesian Computation: semi-automatic Approximate Bayesian computation. Journal of Royal Statistical Society B 74:419–474

Fenchel T (1974) Intrinsic rate of natural increase: the relationship with body size. Oecologia 14:317–326

Franz M, Kramer-Schadt S, Kilian W, Wissel C, Groeneveld J (2010) Understanding the effects of rainfall on elephant-vegetation interactions around waterholes. Ecological Modelling 221:2909–2917

Fruth J, Roustant O, Kuhnt S (2014) Total Interaction Index: A variance-based sensitivity index for second-order interaction screening. Journal of Statistical Planning and Inference 147:212–223

Geman S, Geman D (1984) Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6(6):721–741

Girolami M, Calderhead B (2011) Riemann manifold Langevin and Hamiltonian Monte Carlo methods. Journal of Royal Statistical Society B 73:1–37

Glen G, Isaacs K (2012) Estimating Sobol sensitivity indices using correlations. Environmental Modelling & Software 37:157–166

Gochfeld M (1980) Reproductive Synchrony in Colonial Seabirds Current Perspectives in Research. In: Burger J, Olla BL, Winn HE (eds) Behaviour of Marine Animals, vol 4: Marine Birds, Plenium Press

Gordon NJ, Salmond D, Smith AFM (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEEE Transactions on Radar Signal Process 140:107–113

Grimm V (1999) Ten years of individual-based modelling in ecology: what have we learned about the future? Ecological Modelling 115:129–148

Grimm V, Railsback SF (2005) Individual-based Modelling and Ecology. Princeton Series in Theoretical and Computational Biology

Grimm V, Railsback SF (2012) Pattern-Oriented Modelling: a ‘muilti-scope’ for predictive systems ecology. Philosophical Transactions of the Royal Society B 367:298–310

Grimm V, Uchmański J (1996) Individual-based modelling in ecology: What makes the difference? Trends in Ecology and Evolution 11:437–441

Grimm V, Frank K, Jeltsch F, Brand R, Uchmański J, Wissel C (1996) Pattern-Oriented Modeling in population ecology. Science of total enviroment 183:151–166

Grimm V, Dorndorf N, Frey-Roos F, Wissel C, Wyszomirski T, Arnold W (2003) Modelling the role of social behaviour in the persistance of the alpine marmoto marmota marmota. Oikos 102:124–136

Grimm V, Berger U, Bastiansen F, Eliassen S, Ginot V, Giske J, Goss-Custard J, Grand T, Heinz SK, Huse G, Huth A, Jepson JU, Jørgensen C, Mooij WM, Müller B, Pe'er G, Piou C, Railsback SF, Robbins AM, Robbins MM, Rossmanith E, Rüger N, Strand E, Souissi S, Stillman RA, Vabø R, Visser U, DeAngelis DL (2006) A standard protocol for describing individual-based and agent-based models. Ecological Modelling 198:115–126

Grimm V, Berger U, DeAngelis DL, Polhil JG, Giske J, Railsback SF (2010) The ODD protocol: A review and first update. Ecological Modelling 221:2760–2768

Grimm V, Augusiak J, Focks A, Frank BM, Gabsi F, Johnston ASA, Liu C, Martin BT, Meli M, Radchuk V, Thorbek P, Railsback SF (2014) Towards better modelling and decision support: Documenting model development, testing and analysis using TRACE. Ecological Modelling 280:129–139

Hall SJ, Collie J, Duplisea DE, Jennings S, Bravington M, Link J (2006) A length-based multispecies model for evaluating community responses to fishing. Canadian Journal of Fisheries and Aquatic Sciences 63:1344–1359

Hartig F, Calabrese JM, Reineking B, Wiegand T, Huth A (2011) Statistical inference for stochastic simulation models - theory and application. Ecology Letters 14:816–827

Hartvig M, Andersen KH, Beyer JE (2011) Food web framework for size-structure populations. Journal of Theoretical Biology 272:113–122

Harwood J, Stokes K (2003) Coping with uncertainty in ecological advice: lessons from fisheries. Trends in Ecology and Evolution 18(12):617–622

Hastings WK (1970) Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika 51(1):97–109

Heath B, Hill R, Ciarallo F (2009) A survey of Agent-Based Modelling practices. Journal of Artificial Societies and Social Simulation 12(4):9

Heggland K, Frigessi A (2002) Estimating functions in indirect inference. Journal of Royal Statistical Society B 66:447–462

Hintze JL, Nelson RD (1998) Violin plots: a box plot-density trace synergism. The American Statistician 52(2):181–184

Holden PB, Edwards NR, Oliver KIC, Lenton TM, Wilkinson RD (2010) A probabilistic calibration of climate sensitivity and terrestrial carbon change in GENIE-1. Climate Dynamics 35(5):785–806

Holland EP, Aegerter JN, Dytham C (2009) Comparing resource representations and choosing scale in heterogeneous landscapes. Landscape Ecology 24:213–227

Homma T, Saltelli A (1996) Importance measures in global sensitivity analysis of nonlinear models. Reliability Engineering and System Safety 52:1–17

Horan S, Iman R (2008) A comparison of maximum/bounding and Bayesian/Monte Carlo for fault tree analysis. Sandia National Laboratories report

Hovel KA, Regan HM (2008) Using an individual-based model to examine the roles of habitat fragmentation and behaviour on predator-prey relationships in seagrass landscapes. Landscape Ecology 23:75–89

Iooss B, Lemaître P (2015) A review on global sensitivity analysis methods. In: Meloni C, Dellino G (eds) Uncertainty management in Simulation-Optimization of Complex Systems: Algorithms and Applications, Springer

Iooss B, Ribatet M (2009) Global sensitivity analysis of computer models with functional inputs. Reliability Engineering and System Safety 94(7):1194–1204

Iooss B, Dorpe FV, Devictor N (2006) Response surfaces and sensitivity analysis for an environmental model of dose calculations. Reliability Engineering and System Safety 91:1241–1251

J Liu FL, Wong WH (2000) The Multiple-Try Method and Local Optimization in Metropolis Sampling. Journal of the American Statistical Association 95:121–134

Jacques J, Lavergne C, Devictor N (2006) Sensitivity analysis in presence of model uncertainty and correlated inputs. Reliability Engineering and System Safety 91(10–11):1126 – 1134, DOI http://dx.doi.org/10.1016/j.ress.2005.11.047, URL http://www.sciencedirect.com/science/article/pii/S0951832005002231, the Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004) {SAMO} 2004 The Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004)

Jennings S, Blanchard JL (2004) Fish abundance with no fishing: predictions based on macroecology theory. Journal of Animal Ecology 73:632–642

Joanes DN, Gill CA (1998) Comparing measures of sample skewness and kurtosis. The Statistician 47:183–189

Johnson LR, Briggs CJ (2011) Parameter inference for an individual based model of chytridiomycosis in frogs. Journal of Theoretical Biology 277:90–98

Jovani R, Grimm V (2008) Breeding sychrony in colonial birds: From local stress to global harmony. Proceedings of the Royal Society of London B: Biological Sciences 275:1557–1563

Joyce P, Marjoram P (2008) Approximately Sufficient Statistics and Bayesian Computation. Statistical Applications in Genetics and Molecular Biology 7:26

Jul-Larsen E, Kolding J, OveråR, Nielsen JR, van Zwieten PAM (2003) Management, Co-Management or No Management? Tech. rep., Food and Agricilture Organization of the United Nation

Kaiser H (1979) The dynamics of populations as result of the properties of individual animals. Fortschritte der Zoologie 25:109–136

Kalman RE (1960) A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering 82(1):35

Katz RW (2002) Techniques for estimating uncertainty in climate change scenarios and impact studies. Climate Research 20:167–185

Kennedy MC, O'Hagan A (2001) Bayesian calibration of computer models. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63(3):425–464

Kitchell JF, Stewart DJ (1977) Applications of Bioenergetics Model to Yellow Perch (Perca flavescens) and Walleye (Stizostedion vitreum vitreum). Journal of the Fisheries Research Board of Canada 34:1922–1935

Kleijnen JPC (2007) DASE: Design and Anaysis of Simulation Experiments. Springer Science + Business Media

Kloprogge P, van der Sluijs JP, Petersen AC (2011) A method for the analysis of assumptions in model-based environmental assessments. Environmental Modelling & Software 36:289–301

Krzykacz B (1990) Samos: a computer program for the derivation of empirical sensitivity measures of results from large computer models. Technical report GRS-A-1700F

Kucherenko S, Feil R, Shah N, Mauntz W (2011) The identification of model effective dimensions using global sensitivity analysis. Reliability Engineering and System Safety 96:440–449

Kucherenko S, Tarantola S, Annoni P (2012) Estimation of global sensitivity indices for models with dependent variables. Computer Physics Communications 183:937–946

Law R, Plank MJ, James M, Blanchard JL (2009) Size-spectra dynamics from stochastic predation and growth of individuals. Ecology 90:802–811

Lee A, Łatuszyński K (2014) Variance bounding and geometric ergodicity of Markov chain Monte Carlo kernels for approximate Bayesian computation. Biometrika 101(3):655–671

Lee A, Andrieu C, Doucet A (2012) Discussion of Constructing summary statistics for approximate Bayesian Computation: semi-automatic Approximate Bayesian computation. Journal of Royal Statistical Society B 74:449–450

Lee JE, McVinish R, Mengersen K (2011) Population Monte Carlo algorithm in high dimensions. Methodology and Computing in Applied Probability 13(2):369–389

Lee PM (2004) Bayesian Statistics an introduction, 3rd edn. Oxford University Press

Li H, Wu J (2006) Uncertainty analysis in ecological studies. In: Wu J, Jones KB, Li H, Loucks OL (eds) Scaling and Uncertainty Analysis in Ecology: Methods and Applicationa, 43-64, Springer, pp 43–64

Liu R, Owen AB (2006) Estimating mean dimensionality of analysis of variance decompositions. Journal of the American Statistical Association 101:712–721

Macal CM, North MJ (2010) Tutorial on Agent-based modeling simulation. Journal of Simulation 4:151–162

Maclean M (2010) Individual-Based Simulation Models in Ecology. Master's thesis, The University of Sheffield

Maes J, Limburg KE, van de Putte A, Ollevier F (2005) A spatially explicit, individual-based model to assess the role of estuarine nurseries in the early life history of North sea herring, Clupea harengus. Fisheries Oceanography 14(1):17–31

Mara TA (2010) Extension of the RBD-FAST method to the computation of global sensitivity indices. Reliability Engineering and System Safety 95(4):354–360

Marin JM, Pudlo P, Robert CP, Ryder RJ (2012) Approximate Bayesian Computational methods. Statistics and Computing 22(6):1167–1180

Marjoram P, Molitor J, Plagnol V, Tavaré S (2003) Markov chain Monte Carlo without likelihoods. PNAS 100:15,324–15,328

Marrel A, Iooss B, Veiga SD, Ribatet M (2012) Global sensitivity analysis of stochastic computer models with joint metamodels. Statistics and Computing 22(3):833–847

Martínez I, Wiegand T, Camarero JJ, Batllori E, Gutiérrez E (2011) Disentangling the formation of contrasting tree-line physiognomies combining model selection and Bayesian parameterization for simulation models. The American Naturalist 117:136–152

McKay MD (1996) Variance-based methods for assessing uncertainty importance in nureg-1150 analysis. Tech. Rep. LA-UR-96-2695, Los Alamos Laboratories

McKay MD, Beckman RJ, Conover WJ (1979) A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. Technometrics 21(2):239–245

Metropolis NC, Ulam SM (1949) The Monte Carlo Method. Journal of the American Statistical Association 44(247):335–341

Metropolis NC, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of State Calculations by Fast Computing Machines. Journal of Chemical Physics 21:1087–1092

Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F, Chang CC, Lin CC (2013) e1071. R package

Moran PAP (1950) Notes on Continuous Stochastic Phenomena. Biometrika 37(1/2):17–23

Morris DJ, Spires DC, Cameron AI, Heath MR (2014) Global sensitivity analysis of an end-to-end marine ecosystem model of the North Sea: Factors affecting the biomass of fish and benthos. Ecological Modelling 273:251–263

Morris MD (1991) Factorial Sampling Plans for Preliminary Computational Experiments. Technometrics 33(2):161–174

Neal P (2012) Efficient likelihood-free Bayesian Computation for household epidemics. Statistics and Computing 22:1239–1256

Neal P, Huang CLT (2014) Forward simulation MCMC with applications to stochastic epidemic models. Scandinavian Journal of Statistics doi: 10.1111/sjos.12111

Neal P, Roberts GO (2006) Optimal scaling for partially updating MCMC algorithms. The Annals of Applied Probability 16(2):475–515

Nelder JA, Mead R (1965) A simplex algorithm for function minimization. Computer Journal 7:308–313

van Nes EH, Lammens EHRR, Scheffer M (2002) PISCATOR, an individual-based model to analyze the dynamics of lake fish communities. Ecological Modelling 152:261–278

Neuert C, du Plessis M, Grimm V, Wessel C (1995) Welche ökologischen faktoren bestimmen die gruppengröbe bei phoeniculus purpureus (gemeiner baumhopf) in südafrika? ein individuenbasiertes modell. Verhandlungen der Gesellschaft fur Okologie 24:145–149

Nielsen A, Berg CW (2014) Estimation of time-varying selectivity in stock assessments using state-space models. Fisheries Research 158:96–101

Oakley JE, O'Hagan A (2002) Bayesian inference for the uncertainty distribution of computer model outputs. Biometrika 89(4):769–784

Oakley JE, O'Hagan A (2004) Probabilistic Sensitivity Analysis of complex models: a Bayesian approach. Journal of the Royal Statistical Society: Series B 66(3):751–769

O'Hagan A (2006) Bayesian analysis of computer code: A tutorial. Reliability Engineering and System Safety 91:1290–1300

O'Hagan A, Buck CE, Daneshkhah A, Eiser JR, Garthwaite PH, Jenkinson DJ, Oakley JE, Rakow T (2006) Uncertain judgements: eliciting experts' probabilities. John Wiley and Sons

Owen AB (2013a) Better Estimation of Small Sobol' Sensitivity Indicies. ACM Transactions Modeling Computer Simulations 23:11:1–11:17

Owen AB (2013b) Monte Carlo theory, methods and examples. http://statweb.stanford.edu/ owen/mc/

Owen AB (2013c) Variance Components and Generalized Sobol' indicies. Uncertainty quantification 1:19–41

Pauly D, Christensen V, Walters C (2000) Ecopath, Ecosim, and Ecospace as tools for evaluating ecosystem impact of fisheries. ICES Journal of Marine Science 57:697–706

Petchey OL, Belgrano A (2010) Body-size distributions and size-spectra: universal indicators of ecological status? Biology Letters 6:434–437

Pimm SL, Rice JC (1987) The dynamics of multispecies, multi-life-stage models of aquatic food webs. Theoretical Population Biology 32:303–325

Pinnegar J (2014) Atlantis. MSCC/MASTS modelling workshop

Piou C, Berger U, Grimm V (2009) Proposing an information critation for individual-based models developed in a pattern-orientated modelling framework. Ecological Modelling 220:1957–1967

Plagányi ÉE, Punt AE, Hillary R, Morello EB, Thébaud O, Hutton T, Pillans RD, Thorson JT, Fulton EA, Smith ADM, Smith F, Bayliss P, Haywood M, Lyne V, Rothlisberg PC (2014) Multispecies fisheries management and conservation: tactical applications using models of intermediate complexity. Fish and Fisheries 15(1):1–22, DOI 10.1111/j.1467-2979.2012.00488.x, URL http://dx.doi.org/10.1111/j.1467-2979.2012.00488.x

du Plessis M (1992) Obligate cavity-roosting as a Constraint on Dispersal of Green (Red-Billed) Woodhoopoes: Consequences for Philopatry and the Likelihood of Inbreeding. Oecologia 90:205–211

Polovina JJ (1984) Model of a coral reef ecosystem. I: the ECOPATH model and its application to French Frigate Scholars. Coral Reefs 3:1–11

Pope JG, Rice J, Daan N, Jennings S, Gislason H (2006) Modelling an exploited marine fish community with 15 parameters - results from a simple size-based model. Journal of Marine Science 63:1029–1044

Pritchard JK, Seielstad MT, Perez-Lezaun A, Feldman MW (1999) Population growth of human y chromosomes: a study of y chromosome microsatellites. Molecular Biology and Evolution 16:1791–1798

Railsback SF, Grimm V (2012) Agent-based and individual-based modeling a practical introduction. Princeton University Press

Railsback SF, Cunningham PC, Lamberson RH (2006) A stratergy for parameter sensitivity and uncertainty analysis of Individual-based models. Ecological Modelling 111:207–222

Reckhow KH (1994) Water quality simulation modeling and uncertainty analysis for risk assessment and decision making. Ecological Modelling 72:1–20

Reuman DC, Mulder C, Raffaelli D, Cohen JE (2008) Three allometric relations of population density to body mass: theoretical integration and empirical tests in 149 food webs. Ecology Letters 11:1216–1228

Reynolds C (1997) Individual-based models. http://www.red3d.com/cwr/ibm.html.

Rice J, Gislason H (1996) Patterns of change in the size spectra of numbers and diversity of the North Sea fish assemblage, as reflected in surveys and models. Journal of Marine Science 53:1214–1225

Ricker WE (1954) Stock and recruitment. Journal of the Fisheries Research Board of Canada 11(5):559–623

Roberts GO, Rosenthal JS (2001) Optimal Scaling for Various Metropolis-Hastings Algorithms. Statistical Science 16(4):351–367

Roberts GO, Tweedie RL (1996) Exponential Convergence of Langevin Distributions and Their Discrete Approximations. Bernoulli 2(4):341–363

Rose K (1989) Sensitivity analysis in ecological simulation models. In: Singh, M. (Ed.), Systems and Control Encyclopedia, Pergamon Press, pp 4230–4234

Rossberg AG, Houle JE, Hyder K (2013) Stock-recruitment relations controlled by feeding interactions alone. Canadian Journal of Fisheries and Aquatic Sciences 70(10):1447–1455

Rubin DB (1984) Bayesianly justifiable and relevant frequency calculations for applied statisticians. Annals of Statistics 12:1151–1172

Saltelli A (2002) Making best use of model evaluations to compute sensitivity indices. Computer Physics Communications 145:280–297

Saltelli A, Tarantola S, Chan KPS (1999) A quantitative model independent method for global sensitivity of coupled reaction symptoms to certainties. Technometrics 41:39–56

Saltelli A, Chan KPS, Scott EM (2000) Sensitivity Analysis. Wiley

Saltelli A, Tarantola S, Campolongo F, Ratto M (2004) Sensitivity Analysis in Practise: A Guide to Assessing Scientific Models. Wiley

Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, Salsana M, Tarantola S (2008) Global sensitivity analysis - The primer. Wiley

Santner TJ, Williams BJ, Notz WI (2003) The design and analysis of computer experiments. Springer

Satterwaite FE (1959) Random Balance Experimentation. Technometrics 1(2):111–137

Savage VM, Gillooly JF, Brown JH, West GB, Charnov EL (2004) Effects of body size and temperature on population growth. The American Naturalist 163:429–441

Scott F, Blanchard JL, Andersen KH (2014) mizer: an R package for multispecies, trait-based and community size spectrum ecological modelling. Methods in Ecology and Evolution 5(10):1121–1125, DOI 10.1111/2041-210X.12256, URL http://dx.doi.org/10.1111/2041-210X.12256

Sheldon RW, Kerr SR (1972) The population of monsters in Loch Ness. Limnology and Oceanography 17:796–798

Sheldon RW, Parsons T (1966) A Continuous Size Spectrum for Particulate Matter in the Sea. Journal of the Fisheries Research Board of Canada 24(5):909–915

Silverman BW (1984) Discussion of Monte Carlo Methods of Inference for Implicit Statistical Models. Journal of Royal Statistical Society B 46(2):212–213

Silverman BW (1986) Density estimation for statistics and data analysis. Chapman and Hall

Sisson SA, Fan Y, Tanaka MM (2007) Sequential Monte Carlo without likelihoods. PNAS 104:1760–1765

Sobol' IM (1967) Distribution of points in a cube and approximate evaluation of integrals. USSR Computational Mathematics and Mathematical Physics 7(4):86–112

Sobol' IM (1993) Sensitivity estimates for nonlinear mathematical models. Mathematical Modeling and Computational Experiment 1:407–414

Sobol' IM (2001) Sensitivity estimates for nonlinear mathematical models and their Monte Carlo estimates. Mathematics and Computers in Simulation 55:271–280

Sobol' IM, Kucherenko S (2009) Derivative based global sensitivity measures and their link with global sensitivity indices. Mathematics and Computers in Simulation 79(3009-3017)

Sobol' IM, Kucherenko S (2010) A new derivative based importance criterion for groups of variables and its link with the global sensitivity indices. Computer Physics Communications 181:1212–1217

Sottoriva A, Tavaré S (2010) Integrating Approximate Bayesian Computation with Complex Agent-Based Models for Cancer research. In COMPSTAT 2010 Proceedings in Computational Statistics pp 55–66

Sousa VC, Fritz M, Beaumont MA, Chikhi L (2009) Approximate Bayesian Computation without summary statistics: the case of admixture. Genetics 181:1507–1519

Spiegelhalter DJ, Best NG, Carlin BP (1998) Bayesian deviance, the effective number of parameters and the comparison of arbitrary complex models. Tech. rep.

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A (2002) Bayesian measures of model complexity and fit. Journal of the Royal Statistical Society: Series B Series B Statistical Methodology 64:583–639

Storlie CB, Helton JC (2008) Multiple predictor smoothing methods for sensitivity analysis: Description of techniques. Reliability Engineering and System Safety 93:28–54

Stott PA, Kettleborough JA (2002) Origins and estimates of uncertainty in predictions of twenty first century temperature rise. Nature 416:723–726

Strong M (2011) Managing structural uncertainty in health economic decision models. PhD thesis, University of Sheffield

Strong M, Oakley JE (2013) An Efficient Method for Computing Single-Parameter Partial Expected Value of Perfect Information. Medical Decision Making 33(6):755–766

Strong M, Oakley JE, Chilcott J (2011) Managing structual uncertainty in health economic decision models: a discrepancy approach. Journal of the Royal Statistical Society: Series B 61(1):25–45

Strong M, Oakley JE, Brennan A (2014) Estimating multi-parameter partial Expected Value of Perfect Information from a probabilistic sensitivity analysis: a non-parametric regression approach. Medical Decision Making 34(3):311–326

Swendsen RH, Wang JS (1986) Replica Monte Carlo Simulation of Spin-Glasses. Physical Review Letters 57(21):2607–2609

Tarantola S, Giglioli N, Jesinghaus N, Saltelli A (2002) Can global sensitivity analysis steer the implementation of models for environmental assesments and decision making? Stochastic Environmental Research and Risk Assesment 16:63–76

Tarantola S, Gatelli D, Mara TA (2006a) Random balanced designs for estimation of first-order global sensitivity indices. Reliability Engineering and System Safety 91:717–727

Tarantola S, Nardon M, Salsana M, Gatelli D (2006b) A new estimator for sensitivity analysis of model output: an application to the e-business composite indicator. Reliability Engineering and System Safety 91:1135–1141

Tavaré S, Balding DJ, Griffiths RC, Donnelly P (1997) Inferring Coalescence Times from DNA Sequence Data. Genetics 145(2):505–518

Thorpe RB, Le Quesne WJF, Luxford F, Collie JS, Jennings S (2015) Evaluation and management implications of uncertainty in a multi-species size-structured model of population and community responses to fishing. Methods in Ecology and Evolution 6(1):49–58, DOI 10.1111/2041-210X.12292, URL http://dx.doi.org/10.1111/2041-210X.12292

Tian T, Xu S, Gao J, Burrage K (2007) Simulated maximum likelihood method for estimating kinetic rates in gene expression. Bioinformatics 23:84–91

Tissot JY, Prieur C (2012) Bias correction for estimation of sensitivity indices based on random balance designs. Reliability Engineering and System Safety 107:205–213

Verlet L (1967) Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. Physical Review 159:98–103

Vernon I, Goldstein M, Bower RG (2010) Galaxy Formation: a Bayesian Uncertainty Analysis. Bayesian Analysis 5(4):619–670

Vihola M (2012) Robust adaptive Metropolis algorithm with coerced acceptance rate. Statistics and Computing 22:997–1008

Vrugt JA, ter Braak CJF, Diks CGH, Robinson BA, Hyman JM, Higdon D (2009) Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptive Randomized Subspace Sampling. Journal of Nonlinear Sciences & Numerical Simulation 10(3):271–288

Wainwright HM, Finsterle S, Jung Y, Zhou Q, Birkholzer JT (2014) Making sense of global sensitivity analyses. Computers & Geosciences 65:84–94

Walters C, Christensen V, Pauly D (1997) Structuring dynamic models of exploited ecosystems from tropic mass-balance assessments. Reviews in Fish Biology and Fisheries 7(2):138–172

Ware DM (1978) Bioenergetics of Pelagic Fish: Theoretical Change in Swimming Speed and Ration with Body Size. Journal of the Fisheries Research Board of Canada 35(2):220–228

Wegmann D, Leuenberger C, Excoffier L (2009) Efficient Approximate Bayesian Computation Coupled With Markov Chain Monte Carlo Without Likelihood. Genetics Society of America 182:1207–1218

Weisberg M (2006) Robustness analysis. Philosophy of Science 73:730–742

Werner EE, Gilliam JF (1984) The ontogenetic niche and species interactions in size-structured populations. Annual Review of Ecology, Evolution, and Systematics 15:393–425

White EP, Ernest SKM, Kerkhoff AJ, Enquist BJ (2007) Relationships between body size and abundance in ecology. Trends in Ecology and Evolution 22(6):323–330

Wiegand T, Jeltsch F, Hanski I, Grimm V (2003) Using pattern-oriented modeling for revealing hidden information: A key for recording ecological theory and application. Oikos 100:209–222

Wiegand T, Revilla E, Knauer F (2004) Dealing with uncertainty in spatially explicit population models. Biodiversity and Conversation 13:53–78

Wilkinson DJ (2010a) Parameter inference for stochastic kinetic models of bacterial gene regulation: a Bayesian approach to systems biology. In: Bernardo JM, Bayarri MJ, Berger JO, Dawid AP, Heckerman D, Smith AFM, West M (eds) Bayesian Statistics 9, Oxford University Press

Wilkinson RD (2010b) Bayesian calibration of expensive computer models. In: Biegler L, Biros G, Ghattas O, Heinkenschloss M, Keyes D, Mallick B, Tenorio L, van Bloemen Waanders B, Willcox K (eds) Large scale inverse problems and quantification of Uncertainty, John Wiley and Sons, pp 195–216

Wilkinson RD (2013) Approximate Bayesian Computation (ABC) gives exact results under the assumption of model error. Statistical Applications in Genetics and Molecular Biology 12(2):129–141

Wilkinson RD (2014) Accelerating ABC methods using Gaussian processes. arXiv:14011436

Wood SN (2010) Statistical inference for noisy nonlinear ecological dynamic systems. Nature 466:1102–1104