Locating bat roosts through the coupling of diffusion-type models and static acoustic detectors

Lucy Henley

October 6, 2022

A thesis presented for the degree of

Doctor of Philosophy

Department of Mathematics

Cardiff University

Contents

Summary

Bats play a vital role in ecosystems around the world, but human activities have put their habitats at risk. To ensure that habitats are protected, ecologists must first locate their roosts, the structures in which they sleep. Unfortunately, locating roosts can be extremely difficult and time consuming; bats are small, nocturnal creatures that fly, and are therefore elusive and difficult to track. This thesis aims to use mathematics to develop a method to estimate the locations of bat roosts and reduce the survey effort required by ecologists.

Firstly, we use data from radiotracking surveys, in which the locations of individual bats are tracked over multiple nights, to inform a model of bat movement throughout the night. We show that movement is in two distinct phases: (i), dispersal away from the roost, (ii), followed by a gradual return back to the roost. We describe these two phases using partial differential equation models and stochastic agent based simulations.

Next, we develop a method to estimate the location of bat roosts using a combination of these movement models and acoustic bat surveys. We fit movement models to data from acoustic bat detectors placed around the landscape using Approximate Bayesian Computation.

Finally, we discuss the design of bat surveys and optimise the placement of bat detectors in order to minimise the expected error in roost estimates. We use an iterative process to place detectors one by one, using Bayesian Global Optimisation to determine the optimal location for the next detector at each iteration.

Acknowledgements

Firstly, I would like to thank my supervisors Dr. Thomas Woolley, Professor Owen Jones and Professor Fiona Mathews, without whom this thesis would not have been a success. I am endlessly thankful for their support, guidance and advice on all things maths, bats and writing.

I would also like to thank the bat research group at the University of Sussex, particularly Dr. Domnhall Finch and Kieran O'Malley, who provided me with bat survey data, and helped greatly with understanding bat behaviour.

Equally importantly, I would like to thank the incredible friends I've made in Cardiff, especially Emily, Layla, Tim, Josh and Sam. This PhD has been challenging throughout, but especially in the last two years due to the global pandemic and I am certain that without them, I would never have made it to the end.

And last but certainly not least, my hearfelt thanks go to Liam, my parents and my dog Mabel for helping to keep me sane throughout frequent crises.

Chapter One

Introduction

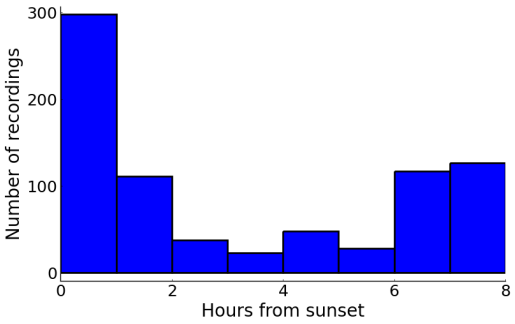
Bats are an important part of ecosystems around the world, playing a vital role in controlling insect populations, seed dispersal and pollination [1]. There are more than 1270 bat species worldwide, occupying a huge range of habitats on each continent except Antartica and bats are among the most ecologically diverse groups of mammals on the planet [2]. Insectivorous species, such as the 18 species found in the UK, feed primarily on airborne insects and contribute to suppressing insect populations including agricultural pests and species which can spread diseases such as mosquitoes. Due to their role as predators, they are sensitive to changes in the population of insects and can therefore act as ecological indicators of biodiversity and pollution [3]. However, bats are susceptible to human impacts due to their sensitivity to light, noise and climate change. Additionally, habitat fragmentation due to roads and building work can reduce foraging opportunities and lead to a significant risk of population decline [4].

In the maternity season during summer, hundreds of animals can group together in a single roost to have young, and as such it is imperative that these roosts are identified and protected. The disturbance of bat roosts has been identified as a significant cause of the population decline of bat species in Europe during the past century [5, 6]. As a result, bats are protected by law in Europe under the EUROBATS agreement [7] and under domestic law in the UK [8, 9]. Under these laws, it is an offence to disturb or harm a bat, or to damage, destroy or obstruct access to a place used as a roost. However, in

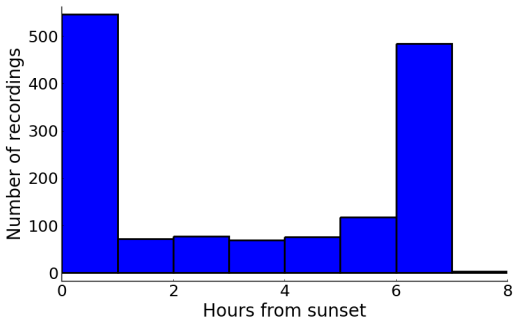
the UK, it is possible to apply for a mitigation license to allow activities which may cause damage to a roost, for example building development, to proceed legally. As part of the licence application, an ecological impact assessment is required to identify local bat roosts and propose strategies to mitigate the impact on bats due to damage to roosts. Critically, identifying roosts is the first step in ensuring that they can be protected from human activity and mitigating the damage to populations.

In this thesis, we will use mathematics in a number of ways to solve current problems regarding understanding bat movement [10], and subsequently use the models we derive to help identify the locations of bat roosts. We will specifically consider the movement of Greater Horseshoe Bats, a species that is classified as Near Threatened across Europe owing to significant declines in its distribution and abundance over the last 50 years [3]. Great Britain, and particularly south-west England are a stronghold for the species, and several detailed ecological studies have been conducted in this region [11, 12, 13]. Greater Horseshoe bats roost predominantly in caves, although in recent years colonies have been found to roost in abandoned buildings, particularly in rural barns and stables.

There are a variety of methods used for identifying or locating bat roosts, and in general all are highly labour intensive. Back-tracking surveys were in common use until around 20 years ago, and involved teams of bat workers with bat detectors searching in the evening, just after sunset, or in the early morning, just before sunrise [14]. There are two main reasons for focusing on times close to sunrise and sunset. Firstly, these are the times that bats are in closest proximity to the roost as they commute away from/towards the roost. Secondly, there are clear temporal patterns in bat activity levels throughout the night [15]. These patterns depend on the species, but in general, activity levels peak shortly after dusk and, in some species, shortly before dawn, and therefore it is easiest to locate flying bats at these times. A representation of these temporal patterns is shown in Figure 1.1 for two surveys of Greater Horseshoe bats. For each survey, there are two clear peaks in the number of recordings, one at sunrise and a second at sunset, however the initial sunset peak is more pronounced, particularly in the Braunton survey shown in Figure 1.1a. When searching for a roost using back-tracking techniques, workers would spread out in a line



(a) The number of recordings for an acoustic survey in Braunton on 11/07/2016.



(b) The number of recordings for an acoustic survey in Buckfastleigh on 26/06/2016.

Figure 1.1: The temporal pattern in activity levels for two Greater Horseshoe bat surveys in Devon, England, in summer 2016. The temporal patterns are demonstrated using the number of bat calls recorded by acoustic detectors in each hour from sunset to sunrise during each of the surveys.

close to a possible roost location and communicate by walkie talkie when a bat call was detected. The team would then work out in which direction the bat was moving (away from the roost in the evening, or towards the roost in the early morning), and attempt to follow the bat's path in order to hone in on the roost. This method would require multiple workers tracking bats over multiple nights, since it is impossible to follow a bat flying at high speed over long distances through often difficult terrain, and was therefore highly labour intensive.

Luckily, the improvement in quality and availability of tracking technology has led to improved survey methods. Radiotracking surveys are commonly used to identify the movement and habitat use of bats by using radio transmitters to locate bats, and are often used to locate roosts [16, 17, 18]. In order to track bats, they must first be caught using a humane method, such as a harp trap, which consists of a frame of wires designed to stop the bat's flight, with a large bag attached beneath the trap to collect falling bats [19]. Once a bat is caught, a small radio transmitter is attached to its back, often using surgical glue, and it is released. The transmitter must be less than 5% of the bat's weight in order to avoid disrupting flight patterns [20]. Images of a typical transmitter attached to a Greater Horseshoe bat are shown in Figure 1.2. The signal from a transmitter is then picked up by field workers using scanning radio-receivers. The precise position of the

Figure 1.2: A Greater Horseshoe bat with radio transmitter glued to its back. The radiotransmitters are very thin antennae, and are highlighted here in white. Photographs taken by Professor Fiona Mathews.

bat can be difficult to determine: generally, triangulation is used to estimate the bat's location, requiring at least 2 field workers recording a bearing at the same time. However, flight speed can exceed 20mph [21], and it can be difficult for multiple workers to record a bearing simultaneously. As such, estimates are subject to human error. Field workers follow the bat and attempt to maintain contact throughout the night, regularly scanning for the signal and taking recordings of location until either the signal is lost, or the bat returns to the roost. Due to the nature of the tracking, bat locations are not recorded at regular intervals, rather only when the signal is found.

Although an improvement on back-tracking, radio tracking surveys are still highly labour intensive as they require bats to be caught and tagged, and then teams of workers follow each bat over the entire night for each night of the survey. Transmitters have a limited range, and workers therefore must remain close to the bat, generally within 1km-2km or even closer if the bat is roosting within a building or underground structure, in order to pick up the signal, which can often be difficult in a rural environment with obstacles such as impassable waterways and hedgerows. Transmitters also have a limited battery life and are often detached and lost before the end of the survey, meaning that surveys can be cut short [22]. Additionally, there are generally numerous potential roost locations, and it is not always possible to conduct physical surveys of each site.

Since data collection is so expensive, it is useful to derive mathematical models that

use the data collected from radiotracking studies in order to gain more insight into bat behaviour. For example, these models can be useful in identifying roosts. An approach to locating birds nests from GPS tracking surveys that result in similar data to that of radiotracking surveys has been implemented in the R package NestR [23]. The package uses recursive movement patterns (periodic returns to places of ecological significance) [24], to identify the locations of ecologically relevant locations such as nests. Additionally, the technique of geographic profiling has been successfully used, along with radiotracking studies, to narrow down search areas for roosts of pipistrelle bats [25]. Geographic profiling is a method commonly used in criminology to determine the most probable area of an offender's residence [26]. It is used in cases of serial linked crimes, and assumes that crimes are most likely to occur close to the offender's home. In the case of locating a bat roost, a jeopardy surface (a surface representing the probability of a roost being located on each grid square) is produced using the locations of known foraging sites and the distance to each grid square. Whilst this results in a reduced search area and could reduce search effort, both locating foraging sites and conducting radiotracking surveys are still likely to be highly labour intensive.

Radio tracking studies can also be complemented with acoustic surveys that use acoustic detectors. Acoustic detectors are microphones which record the ultrasound frequencies produced by bat calls, and can produce timestamped lists of bat detections. Over the last decade, there has been a rapid increase in the use of acoustic surveys for bats, particularly in the ecological consultancy sector that informs development, due to technological improvements in both detector hardware and analytical software, and falling capital costs. Acoustic detectors are used throughout the world to study bat populations and quantify biodiversity [27, 28, 29] and acoustic surveys of buildings are routinely carried out to determine the presence of roosts and inform planning for development [30]. However, acoustic surveys have the important disadvantage that although they record the presence of bats in flight, they cannot record flight direction or be used to locate roosts in the same way that radiotracking studies can. Mathematical models can help us to interpret acoustic records and can allow us to gain more insight into the presence of possible roosts, as well

as the location of roosts, as we would expect many more records at detectors close to a roost than at detectors further away.

There are already models using data from static acoustic detector studies which aim to estimate the importance of certain locations to local bat populations. The software Circuitscape [31, 32] uses algorithms from electronic circuit theory and graph theory to predict connectivity in heterogenous landscapes. It represents landscapes as conductive surfaces, assigning low resistance values to grid squares containing landscape features that are most permeable to movement, and high resistances to grid squares containing movement barriers. A network is then constructed by converting grid squares to nodes. Adjacent notes are connected by edges representing resistors, with resistance values taken as the mean of the resistances of the cells they connect. Then, the connectivity of each node in the landscape is calculated using the resistance distance, a metric defined as the effected resistance between a pair of nodes when multiple possible pathways between the two nodes are considered. The resistance distance, κ_{uv}, between a pair of nodes u and v is calculated as

$$\kappa_{uv} = R_{uv} \sum_{x=1}^{n} \sum_{y=1}^{n} \frac{1}{R_{xy}} \tag{1.1}$$

where R_{xy} is the resistance of the resistor connecting nodes x and y and n is the total number of nodes in the network. Once these resistance distances are known for each pair of nodes, voltage sources are placed within the landscape (when considering the connectivity of a landscape for bats, the roost is the source). The current passing through each node is calculated using Ohm's law, which states that the current, I, which passes through a resister depends on both the voltage applied, V, and the resistance, R, such that $I = V/R$. In Circuitscape, the resistance is replaced with the resistance distance as given by equation (1.1), and then the current, or connectivity, between adjacent nodes u and v is given by

$$I_{uv} = \frac{V}{\kappa_u v}. \tag{1.2}$$

The current quantifies the importance of each location to the population in question, and

can therefore be used to identify important movement corridors between habitat patches. Circuitscape has been successfully used along with data from acoustic surveys to model the importance of landscapes surrounding known roosts, and quantify the importance of different locations [11]. However, this form of analysis requires the location of the roost to be known beforehand and is not able to predict the location of an unknown roost.

The software Ecobat offers a web-based interface to upload data and calculate an indicator of the bat activity and quantify the relative importance of a site [33]. The user inputs timestamped data from acoustic surveys showing each time a bat was recorded at detectors located at various sites. Then, Ecobat compares the activity levels of each species recorded at each detector over each night of the survey to local and national datasets. The user is then provided with a report showing the activity level quantified as a percentile of the comparison datasets. For example, a site assigned a percentile value of 70 has greater activity than 70% of comparison sites. Ecobat can help to quantify the probability of the presence of a roost based on the percentile of activity levels (sites with high percentiles are likely to be located close to a roost, whilst those with low percentiles are unlikely to be located close to a roost), however, it does not provide an estimate of the location of a possible roost.

Mathematical models are an invaluable tool in understanding ecology as they help us to understand the mechanisms that lead to certain patterns of behaviour [34]. There are many possible formalisms, such as stochastic or deterministic, continuous or discrete, depending on the population and behaviour [35]. Partial differential equation (PDE) models can provide a useful approximation to real life whilst simplifying the mathematics by excluding noise. As there are a finite number of animals in a population, adding stochastic dynamics can improve the model approximations.

In this thesis, we focus on diffusion-type models (we define diffusion-type models as those which contain a diffusion element) for bat movement. Diffusion-type models have been used for decades to describe animal movement, and can be formalised as stochastic processes or with PDEs [36, 34]. With improvements in electronic animal tracking devices in recent years, increasingly detailed data can now be gathered and used to inform complex

models of animal movement.

Stochastic processes are able to effectively describe movement of individual animals under the influence of group dynamics and landscape effects [36]. The Ornstein-Uhlenbeck (OU) process was first introduced in 1930 as an adaptation of Brownian motion which includes an overall drift towards a specific location [37]. The OU process is a stochastic process with two components, a random diffusion element and a deterministic convection or drift element. The first method for modelling animal locations using a stochastic process in continuous time in 1977 [38] was derived from the OU process, and provided a description of an animal's home range, defined as the smallest geographical area in which the animal spends a fixed proportion of time [39]. In this model of animal movement, the position of the animal in (x, y) coordinates is described by the equilibrium, long-term, distribution of the OU process given by

$$\boldsymbol{U}(t) \sim N(\boldsymbol{\mu}, \boldsymbol{\gamma}), \tag{1.3}$$

where $\boldsymbol{U}(t)$ and $\boldsymbol{\mu}$ are d-dimensional vectors and $\boldsymbol{\gamma}$ is a $d \times d$ covariance matrix. Since 1977, stochastic processes have been used to study the movement on various scales of a wide range of animal species ranging from the migration of large land mammals such as elk [40] to constructing flight models of bumblebees [41].

However, following the dynamics of each individual in a colony can prove to be computationally expensive, particularly when studying large populations under the influence of multiple external factors [36]. In contrast, PDE models, also frequently used to model a wide range of biological systems, reduce complexity by excluding individual dynamics, focussing only on the dynamics of the group as a whole [42]. In the context of animal movement, PDEs have been applied to a range of problems including home-range formation and territory use [43], insect dispersal [44] and flocking behaviour [45]. PDE diffusion models have been used to model the movement of bats in both homogeneous and heterogeneous environments [46, 47]. Whilst PDEs are highly useful in analysing the properties of ecological systems, their use in empirical and statistical ecology is much less common [48]. This is likely due to the difficulties in choosing PDE models and fitting them to data.

Fitting parameters to a PDE model often requires numerically solving the PDE for various parameter values [49], and as a result simpler or easier to solve stochastic process models are often favoured by non-mathematical ecologists.

Overall, both stochastic process and PDE models are useful for solving different problems: stochastic models can effectively describe individual dynamics, and can be simpler to solve by ecologists, whilst PDE models exclude individual variation and can provide useful approximations. In this thesis, we will use and compare the results from stochastic process models and PDE models.

In Chapter 3, we combine statistical methods and modelling to characterise bat motion, using discrete and stochastic models to help derive continuous, PDE models. We use time-location data from radiotracking studies that track bat motion from when they first leave their roost at sunset to when they return in the morning. By extracting the mean squared displacement (the ensemble average of squared displacement from the roost over time) from the data we see two distinct movement phases, an initial linear dispersal followed by a gradual return to the roost. We use this data to develop PDE models to describe motion for each movement phase. Diffusion models in both one and two dimensions are discussed to describe the dispersal of bats away from the roosts. Two models are used to describe movement for the remainder of the night whilst bats are foraging, a convection-diffusion model and a model describing diffusion on a shrinking domain. Convection-diffusion models are widely used in ecology to model population migration, however we will show here that a convection-diffusion model is not consistent with radiotracking data. Instead, a shrinking domain diffusion model [50, 51] provides a better description of bat movement whilst foraging.

Next, in Chapter 4, we discuss a method for estimating the location of bat roosts using data from static detector surveys along with the models derived in Chapter 3. Finally, in Chapter 5, we look at designing surveys such that we can optimise the location of detectors and improve the estimates for roost location.

Chapter Two

Mathematical Techniques

In this chapter, we will conduct a literature review of the mathematical techniques used in this thesis for model-fitting. We will introduce Bayesian statistics and Approximate Bayesian Computation and their uses for fitting parameters, and Bayesian Global Optimisation for optimising functions. Approximate Bayesian Computation will be used to fit models of bat movement to radio tracking data, and later to estimate the location of a roost given bat survey data. Bayesian Global Optimisation will be used to plan bat surveys, optimising the location of bat detectors in order to improve the estimates of roost locations.

1 Bayesian statistics

Bayesian statistics is a statistical paradigm that provides useful mathematical tools for updating knowledge about a parameter given related data [52]. A Bayesian model consists of three parts. Firstly, the prior probability distribution, which describes our initial knowledge of the parameter and is generally chosen before data collection. Secondly, the likelihood function, which describes the probability of possible parameter values given the data collected. Lastly, the posterior probability distribution which reflects our updated knowledge of the parameter, and is calculated by combining the prior and likelihood function. The posterior is used to make inferences about the parameters, and is dependent on both the informativeness of the prior and the observed data encoded in the likelihood

function.

The posterior probability distribution, $p(\theta \mid \boldsymbol{Y})$, of a parameter θ conditioned on observations $\boldsymbol{Y} = \{Y_1, Y_2, ..., Y_n\}$ is given by Bayes' Theorem [52],

$$p(\theta \mid \boldsymbol{Y}) = \frac{p(\boldsymbol{Y} \mid \theta)p(\theta)}{p(\boldsymbol{Y})}, \tag{2.1}$$

where $p(\theta \mid \boldsymbol{Y})$ is the posterior probability distribution, formally describing the probability that the parameter value is θ, given observations, $\boldsymbol{Y}$. The likelihood function, $p(\boldsymbol{Y} \mid \theta)$, describes the probability of observing $\boldsymbol{Y}$ if the parameter value is θ. The prior distribution, $p(\theta)$, describes the initial knowledge of possible parameter values, and $p(\boldsymbol{Y})$ defines the marginal likelihood, or the total probability of observing the evidence. Often, the observations, $\boldsymbol{Y}$, are fixed, and the probability of observing $\boldsymbol{Y}$ is therefore constant. In that case, the marginal likelihood, $p(\boldsymbol{Y})$, can be dropped from the equation, and equation (2.1) is expressed as

$$p(\theta \mid \boldsymbol{Y}) \propto p(\boldsymbol{Y} \mid \theta)p(\theta). \tag{2.2}$$

A simple example of Bayesian inference using probability density functions uses conjugate priors. For some likelihood functions, if a certain prior is chosen, the resulting posterior is a distribution in the same probability distribution family as the posterior (for example, both the prior and posterior distributions could be Beta distributions, each with different shape parameters). These priors are known as conjugate priors. We will discuss an example of Bayesian inference with a Beta distribution as the conjugate prior, and a Binomial distribution as the likelihood function.

The Binomial distribution is a discrete distribution describing the number of successes, Y, in a series of n independent trials,

$$Y \sim B(n, \theta) \tag{2.3}$$

$$p(Y|\theta) = \binom{n}{Y}\theta^Y(1-\theta)^{n-Y}, \tag{2.4}$$

where θ is a parameter describing the success probability of each trial and

$$\binom{n}{Y} = \frac{n!}{Y!(n-Y)!} \tag{2.5}$$

is the binomial coefficient. For a Binomial likelihood, the conjugate prior is a Beta distribution,

$$\theta \sim \text{Beta}(\alpha, \beta). \tag{2.6}$$

The probability density function of this Beta distribution is given as

$$p(\theta) = C\theta^{\alpha-1}(1-\theta)^{\beta-1}, \tag{2.7}$$

where C is a constant. The posterior for θ is then given by

$$\begin{aligned} p(\theta|Y) &\propto P(Y|\theta)p(\theta) \\ &\propto \theta^{Y}(1-\theta)^{n-Y}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &\propto \theta^{\alpha+Y-1}(1-\theta)^{\beta+n-Y-1}. \end{aligned} \tag{2.8}$$

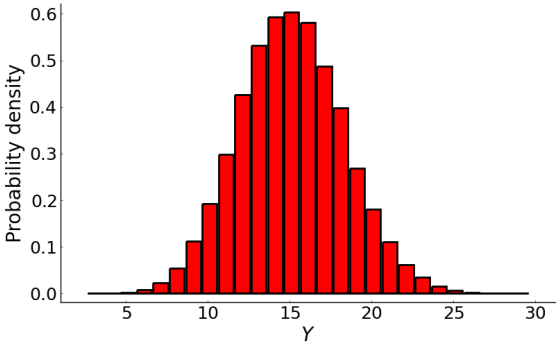
Comparing equation (2.7) with equation (2.8) shows that the posterior is a Beta distribution,

$$\theta \sim \text{Beta}(\alpha + Y, \beta + n - Y), \tag{2.9}$$

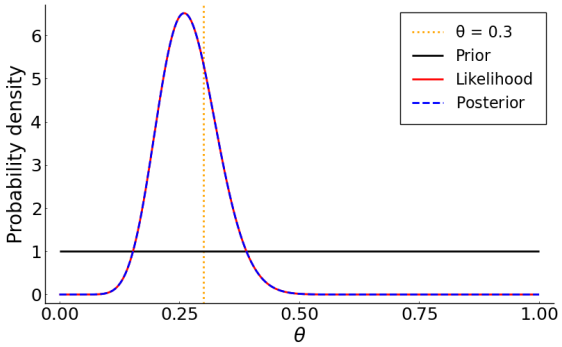
showing that a Beta distribution is a conjugate prior when the likelihood function is a Binomial distribution.

The posterior distribution of the parameter θ for the conjugate prior example is illustrated in Figure 2.1 for varying priors and observed data. Clearly, the posterior is affected by changes in both the prior and observed data. When an uninformative (in this case, uniform) prior is chosen, as in Figure 2.1b, the posterior is determined solely by the likelihood function, as the prior does not add any additional information. In contrast, an informative prior can affect the shape of the posterior. In Figure 2.1c, a weakly informative

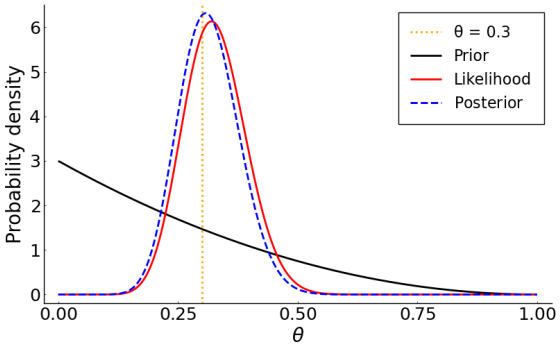
prior is used, and the posterior distribution is shifted slightly from the likelihood function. In Figure 2.1d, a highly informative prior was used, and the resulting posterior has a lower variance because more information about the parameter is available.



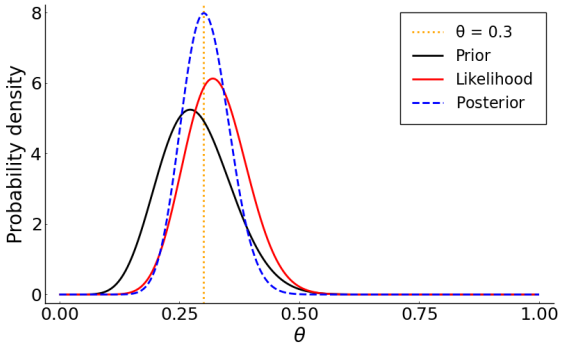
(a) A histogram showing the number of successes, Y, in $n = 50$ trials. To construct the histogram, 10^5 samples were drawn from the Binomial distribution given by equation (2.4).



(b) The prior and posterior distributions for parameters $\alpha = 1$ and $\beta = 1$ and observation $Y = 13$ successes. The prior distribution is $\theta \sim \text{Beta}(1, 1)$ and the posterior distribution is $\theta \sim \text{Beta}(14, 38)$.



(c) The prior and posterior distributions for parameters $\alpha = 1$ and $\beta = 3$ and observation $Y = 16$ successes. The prior distribution is $\theta \sim \text{Beta}(1, 3)$ and the posterior distribution is $\theta \sim \text{Beta}(17, 38)$.



(d) The prior and posterior distributions for parameters $\alpha = 10$ and $\beta = 25$ and observation $Y = 16$ successes. The prior distribution is $\theta \sim \text{Beta}(10, 25)$ and the posterior distribution is $\theta \sim \text{Beta}(26, 59$.

Figure 2.1: An example of Bayesian inference to estimate the probability distribution of the parameter θ from a Binomial distribution, $Y \sim B(n, \theta)$, with $n = 50$. The true parameter value is $\theta = 0.3$ and the posterior, given by equation (2.9), is compared for varying priors and observed data.

This method of Bayesian inference is extremely useful for problems with a calculable likelihood function. However, for many problems, including those discussed in this thesis, the likelihood function is intractable. In this case, a useful approach is to use Approximate Bayesian Computation.

1.1 Approximate Bayesian computation (ABC)

Approximate Bayesian Computation (ABC) is an approach to Bayesian inference using simulation and random sampling [53, 54, 55] and is widely used for problems where the analytical form of the likelihood function is intractable [56]. We will use ABC in Chapter 3 to fit mathematical movement models to bat survey data, and in Chapter 4 to estimate the location of bat roosts given acoustic survey data.

ABC replaces the calculation of the likelihood function, $p(\boldsymbol{Y} \mid \theta)$, with simulation of a model using a specific parameter value $\theta = \theta'$ to produce an artificial dataset $\boldsymbol{X}$. For each possible value of θ', there is a non-zero probability of being in the posterior, and the values of θ' that produce artificial datasets $\boldsymbol{X}$ closest to the observations, $\boldsymbol{Y}$, are most likely to be added. To compare the two datasets, we generate low-dimensional vectors of summary statistics, $\boldsymbol{S}(\boldsymbol{Y})$ and $\boldsymbol{S'}(\boldsymbol{X})$, which summarise observations, $\boldsymbol{Y}$, and simulated data, $\boldsymbol{X}$, respectively. In many cases, the observations may be high dimensional. For example if we wished to estimate the parameters, α and β, of a Beta distribution, as in the conjugate prior example in Section 1, we may choose a vector of $n = 200$ samples drawn from the distribution as our observations, $\boldsymbol{Y}$. Typically, the dimension of $\boldsymbol{S}$ will be less than 5. The choice of summary statistics is critical to the efficiency and accuracy of ABC, and we will discuss this later, in Section 1.1.1.

The distance between the simulated and observed datasets is defined by a distance metric, $\rho(\boldsymbol{X}, \boldsymbol{Y})$, often defined as the Euclidean distance between the summary statistics of the observed and simulated data,

$$\begin{aligned} \rho(\boldsymbol{X}, \boldsymbol{Y}) &= \|\boldsymbol{S}(\boldsymbol{Y}) - \boldsymbol{S'}(\boldsymbol{X})\|_2. \\ &= \sqrt{\sum_{i=1}^{d} (\boldsymbol{S_i}(\boldsymbol{Y}) - \boldsymbol{S_i'}(\boldsymbol{X}))^2}, \end{aligned} \tag{2.10}$$

$$\tag{2.11}$$

where d is the number of dimensions in $\boldsymbol{S}$. We restrict the posterior to observations for which the distance metric, $\rho(\boldsymbol{X}, \boldsymbol{Y})$, is smaller than some threshold value, ϵ, $\rho(\boldsymbol{X}, \boldsymbol{Y}) < \epsilon$,

such that the posterior is given by $p(\theta \mid \rho(\boldsymbol{X}, \boldsymbol{Y}) < \epsilon)$. For small ϵ, $\epsilon \to 0$, assuming that the model is correct, the simulated posterior distribution produced approximates the true posterior, $p(\theta \mid \boldsymbol{X} = \boldsymbol{Y})$ [55], and this simulated posterior provides a probability distribution for the parameter θ. Often, ϵ is chosen such that a certain percentage of the sample which have the lowest distance metric are chosen.

A pseudocode for the ABC method is as follows.

Algorithm 1 The ABC algorithm.

```
Set threshold parameter, ϵ, and sample size n
Set prior distribution for parameters, θ: p(θ)
Calculate summary statistics S(Y) from observed data Y
i ← 1
while i < n do
    Sample θ′ from p(θ)
    Simulate X_i dependent on θ = θ′
    Calculate summary statistics S′(X)_i from X_i
    ρ ←| S(Y) − S′(X) |
    if ρ < ϵ then
        θ_i ← θ′
        i ← i + 1
    end if
end while
```

The parameters θ_i are an approximate sample from the posterior distribution $p(\theta \mid \boldsymbol{Y})$, and the posterior mean, $\mathbb{E}(\theta \mid \boldsymbol{Y})$, is estimated using

$$\mathbb{E}(\theta \mid \boldsymbol{Y}) = \frac{1}{n} \sum_i \theta_i. \tag{2.12}$$

1.1.1 Choice of Summary Statistics

As discussed before, the choice of summary statistics is critical to the efficiency and accuracy of ABC estimates. The most simple choice of statistics is to simply use the raw data, such that $\boldsymbol{S}(\boldsymbol{Y}) = \boldsymbol{Y}$ and $\boldsymbol{S'}(\boldsymbol{X}) = \boldsymbol{X}$, however in general it is significantly more efficient and accurate to summarise the data in some way. In statistics, efficiency is a measure of the quality of an experimental design: an efficient experiment will require fewer observations than an inefficient experiment to achieve a given error performance. The

accuracy of the estimate depends on the informativeness of summary statistics: if the statistics chosen effectively summarise the shape of the data, the estimate will be accurate, whilst if the statistics are uninformative, the estimate will not be accurate [57].

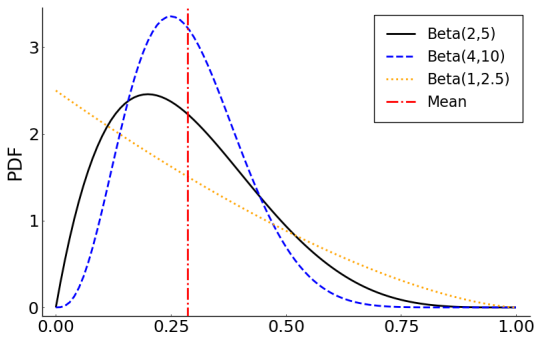
Commonly used statistics include a measure of location, such as the mean, a measure of dispersion, such as the standard deviation or variance, and measures of the shape of the distribution, such as skew or kurtosis [57]. If more than one variable is measured, a measure of statistical dependence, such as a correlation coefficient, can also be added. The quality of ABC approximations decays with the dimension of $\boldsymbol{X}$ [58], because as more statistics are added, there are more opportunities for random discrepancies between $\boldsymbol{S}(\boldsymbol{Y})$ and $\boldsymbol{S'}(\boldsymbol{X})$ and therefore a larger threshold parameter, ϵ, must be used to find enough samples to add to the posterior. As ϵ increases, the acceptance rate of poorly matched samples also increases, and therefore the quality of ABC posteriors decreases. Therefore, low-dimensional vectors of summary statistics are preferable. However, these statistics must be informative, effectively summarising the shape of the data. Examples of ABC using various summary statistics to estimate the parameter values of a Beta distribution are shown in Figure 2.2. The least accurate posterior is from the example using raw data as summary statistics in Figure 2.2b, and this is because the distance metric is the sum of the distance between each sample,

$$\rho = \sum_{i=1}^{n} (\boldsymbol{Y}_i - \boldsymbol{X}_i)^2, \tag{2.13}$$

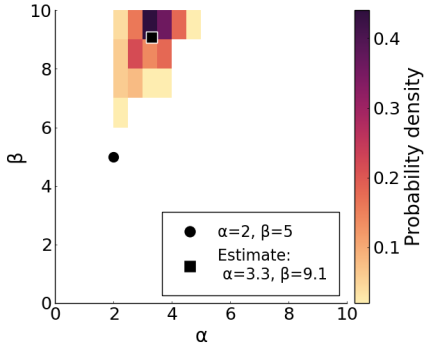
meaning that the vector of summary statistics is large, $n = 200$. As each sample in both $\boldsymbol{Y}$ and $\boldsymbol{X}$ is identically distributed in the Beta distribution, the probability of $\boldsymbol{Y}_i - \boldsymbol{X}_i < \epsilon$ for a given i is low. Using the mean as a summary statistic, as in Figure 2.2c, negates this issue, as we reduce the dimension of the vector of summary statistics to $n = 1$, comparing the centre of the distribution produced by $\boldsymbol{Y}$ and $\boldsymbol{X}$ rather than each independent sample. In this case, the posterior includes a family of distributions with similar means, each with parameters $\beta \approxeq 2.5\alpha$. Three different distributions with $\beta = 2.5\alpha$ are plotted in Figure 2.2a, showing that each of these has a mean of 0.29 despite each distribution having a very

different shape. To further narrow down the family of distributions in the posterior, we add the standard deviation to the summary statistics, shown in Figure 2.2d. This time, we use both a measure of central tendency (the mean) and a measure of spread (the standard deviation), and the resulting posterior is significantly more accurate than the previous examples, a narrow distribution centred close to the true parameter values. This shows the importance of informative summary statistics in producing accurate estimates. A commonly used measure of the usefulness of a statistic is sufficiency: a statistic, S, can be described as Bayes sufficient for parameter θ if $\theta \mid S$ has the same distribution as $\theta \mid Y$ for almost all possible Y, such that in an ideal ABC algorithm with sufficient S and a small threshold parameter, $\epsilon \to 0$, the ABC posterior equals the true posterior [59]. In practice, low dimensional sufficient statistics do not exist for most distributions, and instead we must look for low dimensional statistics as close to sufficient as possible.

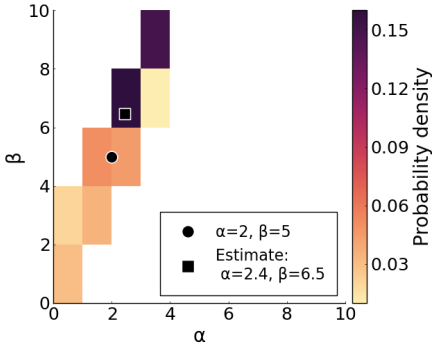
ABC will be used here to fit parameters to an agent-based movement model using the data extracted from radio tracking data. The tracking locations will be used as the observations, $\boldsymbol{Y}$, and we will use the mean squared displacement from the roost calculated from the radio tracking data as the summary statistic, $\boldsymbol{S}(\boldsymbol{Y})$. We will use agent-based simulations to generate simulated bat locations, $\boldsymbol{X}$, and calculate mean squared displacement for summary statistics, $\boldsymbol{S}'(\boldsymbol{X})$. The parameters in the movement model are diffusion and convection coefficients, D, and χ, respectively, and these lead to a vector of parameters, $\boldsymbol{\theta} = (D, \chi)$.



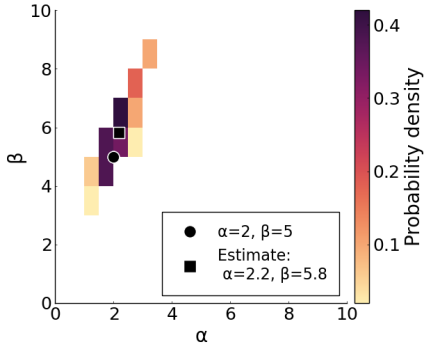
(a) The probability density functions for Beta distributions with varying parameters α and β. The mean of each distribution is the same, and is marked as a vertical line.



(b) The posterior distribution when the raw data are used as summary statistics, $\boldsymbol{S}(\boldsymbol{Y}) = \boldsymbol{Y}$ and $\boldsymbol{S'}(\boldsymbol{X}) = \boldsymbol{X}$.



(c) The posterior distribution when the mean of the data is used as the summary statistic, $\boldsymbol{S}(\boldsymbol{Y}) = \overline{\boldsymbol{Y}}$ and $\boldsymbol{S'}(\boldsymbol{X}) = \overline{\boldsymbol{X}}$.



(d) The posterior distribution when the mean and standard deviation are used as summary statistics, $\boldsymbol{S}(\boldsymbol{Y}) = \big(\overline{\boldsymbol{Y}}, \sigma(\boldsymbol{Y})\big)$ and $\boldsymbol{S'}(\boldsymbol{X}) = \big(\overline{\boldsymbol{X}}, \sigma(\boldsymbol{X})\big)$.

Figure 2.2: Examples of the results of ABC to estimate the parameters, α and β, of a Beta distribution. The observed data is a vector of $n = 200$ samples drawn from the distribution, $\boldsymbol{Y}_i \sim \text{Beta}(2, 5)$, and different summary statistics are used for each example. For each example, the number of samples was $N = 10000$, the Euclidean distance was used as the distance metric, and the threshold parameter, ϵ, was chosen such that the best 1% of samples were added to the posterior. The estimates are calculated as the mean of the posterior.

2 Bayesian Global Optimisation (BGO)

Bayesian Global Optimisation (BGO) is a class of machine-learning based optimisation methods able to optimise expensive black-box derivative-free functions. Expensive here refers to a limited computational budget, usually due to limited scheduling or finances, meaning that only a limited number of evaluations can be performed. A black-box function, $f(x)$, is defined as a function for which knowledge can only be obtained by sampling at some point, x. Derivative-free evaluation signifies that only the function itself, $f(x)$, and no derivatives, $\nabla_x f(x)$, are evaluated. BGO will be used in Chapter 5 to optimise the location of detectors for bat surveys in order to minimise the error in roost estimation.

BGO is typically used to solve global optimisation problems of the form

$$\max_{x \in A} f(x), \tag{2.14}$$

where the feasible set A and objective function $f(x)$ have the following properties [60, 61]:

1. The input, x, is in $\mathbb{R}^d$, where the number of dimensions d is not "too" large, here this will mean $d \leq 20$.

2. The feasible set, A, is typically a hyper-rectangle in d dimensions.

3. The objective function, f, is continuous.

4. The objective function, f, is expensive to evaluate, and the number of evaluations that can be performed is limited.

5. Only the objective function, f, is evaluated. First- or second-order derivatives are not evaluated.

The BGO method has 2 components: (i) a Gaussian process (GP) model for the objective function $f(x)$, and (ii) an acquisition function used to decide where to sample next. First, we evaluate the objective function, $f(x)$, at a set of points x chosen according to an initial space-filling design. These initial points are often chosen by sampling uniformly at

random [61]. The initial function values are then used to iteratively allocate the remainder of a budget of N function evaluations, chosen based on available computing power. At each iteration, i, the objective function is modelled using Gaussian Process Regression (explained below in Section 2.1) to produce a distribution for the objective function for general points, x [61]. Then, the acquisition function is used to choose the next sample point, x_{i+1}. Once the N function evaluations have been completed, a solution is returned, either the point evaluated with the largest objective function $f(x)$, or the point with the largest posterior mean if this point was not sampled.

2.1 Gaussian Process Regression (GPR)

Gaussian Process Regression (GPR) is a statistical approach for regression [62]. A Gaussian process (GP) is a stochastic process for which any finite sample has a multivariate Gaussian distribution [60].

A multivariate Gaussian distribution is defined as a random vector $\boldsymbol{X} \in \mathbb{R}^d$ such that $\boldsymbol{a}^T\boldsymbol{X}$ is a univariate Gaussian for all $\boldsymbol{a} \in \mathbb{R}^d$. The distribution is determined by its mean, $\boldsymbol{\mu}$, and covariance matrix, $\boldsymbol{\Sigma}$, we write $\boldsymbol{X} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The probability density, $\rho(\boldsymbol{x})$, of a multivariate Gaussian distribution is given by

$$\rho(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}\sqrt{\det\boldsymbol{\Sigma}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}, \tag{2.15}$$

where d is the dimension of $\boldsymbol{X}$.

In order to model a function, $f(x)$, using GPR, we approximate it using a multivariate Gaussian distribution. Provided the covariance matrix, $\boldsymbol{\Sigma}$ is non-singular, the probability density of a function value at point x_i, $f(x_i)$, is given by equation (2.15). GPR approximates the function, $f(x)$, using a sample set of function values $\boldsymbol{f}(\boldsymbol{x}_{1:k}) = [f(x_1), ..., f(x_k)]$ at points $\boldsymbol{x_{1:k}} = x_1, ..., x_k \in \mathbb{R}^n$ to calculate a conditional mean of a GP. The distribution of the GP is defined by the mean, $\boldsymbol{\mu}$, and covariance matrix, $\boldsymbol{\Sigma}$, at points $\boldsymbol{x}_{1:n}$, and is written as

$$\boldsymbol{X} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{\mu}(\boldsymbol{x}_{1:k}), \boldsymbol{\Sigma}(\boldsymbol{x}_{1:k})). \tag{2.16}$$

An example of a GPR for the function $f(x) = \sin(x)$, calculated using the Julia package GaussianProcesses.jl [63] is shown in Figure 2.3. The package GaussianProcesses.jl provides solvers for calculating the distribution of a GP, given by equation (2.16), for a variety of mean and kernel functions. The function values $\boldsymbol{f}(\boldsymbol{x}_{1:k})$ are observed at points $\boldsymbol{x}_{1:k}$ and a GPR is fit, conditioned on $\boldsymbol{f}(\boldsymbol{x}_{1:k})$. The conditional mean of the multivariate Gaussian distribution is used as the regression estimate. Note that the only information about the shape of the function that is required to calculate a GPR is the function values, $\boldsymbol{f}(\boldsymbol{x}_{1:k})$, observed at points $\boldsymbol{x}_{1:k}$. As such, GPR is particularly useful when we have no information about the underlying shape or periodicity of the function in question.

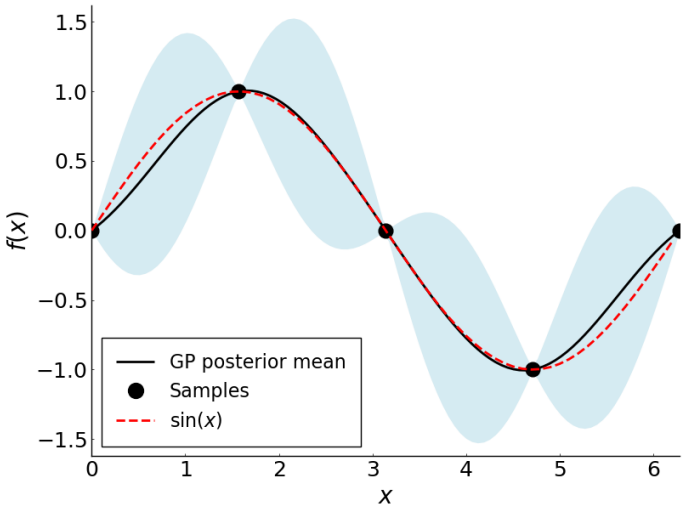


Figure 2.3: Gaussian process regression for the function $f(x) = \sin(x)$ with 5 samples at $\boldsymbol{x} = [0, \pi/2, \pi, 3\pi/2, 2\pi]$. The ribbon represents a 95% confidence interval around the mean from the GP posterior probability distribution.

First we assume a GP distribution for the function $f(x)$, specified using a mean function, $\mu_0(x)$, and covariance function, or kernel, $\boldsymbol{\Sigma}_0(x, x')$,

$$f \sim GP(\mu_0(x), \boldsymbol{\Sigma}_0(x, x')). \tag{2.17}$$

The mean function, $\mu_0(x)$, represents the mean of the distribution at point x, and therefore the estimate of $f(x)$, whilst the kernel, $\boldsymbol{\Sigma}_0(x, x')$, represents the covariance between pairs of points, x, x'. The choice of mean and covariance functions completely

determines the resulting distribution, and is discussed below in Section 2.1.1.

A mean vector, $\boldsymbol{\mu}_0(\boldsymbol{x}_{1:k})$, where the notation $\boldsymbol{x}_{1:k}$ denotes the sequence $x_1, ..., x_k$, is constructed by evaluating the mean function, $\mu_0(x)$, at each point x_i,

$$\boldsymbol{\mu}_0(\boldsymbol{x}_{1:k}) = \begin{bmatrix} \mu_0(x_1) \\ \vdots \\ \mu_0(x_k) \end{bmatrix}. \tag{2.18}$$

A covariance matrix, $\boldsymbol{\Sigma}_0(\boldsymbol{x}_{1:k}, \boldsymbol{x}_{1:k})$, is constructed by evaluating the kernel function, $\Sigma_0(x_i, x_j)$, at each pair of points x_i, x_j,

$$\boldsymbol{\Sigma}_0(\boldsymbol{x}_{1:k}, \boldsymbol{x}_{1:k}) = \begin{bmatrix} \Sigma_0(x_1, x_1) & \dots & \Sigma_0(x_1, x_k) \\ \vdots & \ddots & \vdots \\ \Sigma_0(x_k, x_1) & \dots & \Sigma_0(x_k, x_k) \end{bmatrix}. \tag{2.19}$$

A kernel function, $\Sigma_0(x_i, x_j)$, will be defined below in Section 2.1.1. The kernel is chosen such that pairs of points x_i, x_j that are close in input space have a large positive correlation, and those points far in input space have a small positive correlation. For any finite set of points $\boldsymbol{x}_{1:n}$, the resulting distribution on $[f(x_1), ..., f(x_k)]$ is

$$\boldsymbol{f}(\boldsymbol{x}_{1:k}) \sim \boldsymbol{\mathcal{N}}\left(\boldsymbol{\mu}_0(\boldsymbol{x}_{1:k}), \boldsymbol{\Sigma}_0(\boldsymbol{x}_{1:k}, \boldsymbol{x}_{1:k})\right) \tag{2.20}$$

$$\sim \boldsymbol{\mathcal{N}}\left(\begin{bmatrix} \mu_0(x_1) \\ \vdots \\ \mu_0(x_k) \end{bmatrix}, \begin{bmatrix} \Sigma_0(x_1, x_1) & \dots & \Sigma_0(x_1, x_k) \\ \vdots & \ddots & \vdots \\ \Sigma_0(x_k, x_1) & \dots & \Sigma_0(x_k, x_k) \end{bmatrix}\right). \tag{2.21}$$

Given a training set consisting of a set of function values $\boldsymbol{f}(\boldsymbol{x}_{1:n})$ for some n, we can estimate μ_0 and Σ_0, then calculate the distribution of $\boldsymbol{X}$ at a new set of n' test points, $\boldsymbol{x}_*$,

$$\boldsymbol{X}(\boldsymbol{x}_*) \mid [\boldsymbol{X}(\boldsymbol{x}_{1:k}) = \boldsymbol{f}(\boldsymbol{x}_{1:k}))]. \tag{2.22}$$

This conditional distribution is normal, with mean and variance depending on $\boldsymbol{f}(\boldsymbol{x}_{1:k})$. To

do so, let $k = n + n'$ and $\boldsymbol{x}_{n+1:k} = \boldsymbol{x}_*$ such that the GP over $[\boldsymbol{f}(\boldsymbol{x}_{1:n}), f(\boldsymbol{x}_*)]$ is given by equation (2.21). We can then calculate the conditional probability of $\boldsymbol{X}(\boldsymbol{x}_*)$ given these observations using Bayes' rule, equation (2.1) [64]. First, we write the joint distribution of the training set $\boldsymbol{f}(\boldsymbol{x}_{1:n})$ and the new set of points $f(\boldsymbol{x}_*)$ as

$$\begin{bmatrix} \boldsymbol{X}_{1:n} \\ \boldsymbol{X}(\boldsymbol{x}_*) \end{bmatrix} \sim \boldsymbol{\mathcal{N}} \left(\begin{bmatrix} \boldsymbol{\mu}_{1:n} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Sigma}_* \\ \boldsymbol{\Sigma}_*^T & \boldsymbol{\Sigma}_{**} \end{bmatrix} \right), \tag{2.23}$$

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_0(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n}), \tag{2.24}$$

$$\boldsymbol{\Sigma}_* = \boldsymbol{\Sigma}_0(\boldsymbol{x}_{1:n}, \boldsymbol{x}_*), \tag{2.25}$$

$$\boldsymbol{\Sigma}_{**} = \boldsymbol{\Sigma}_0(\boldsymbol{x}_*, \boldsymbol{x}_*), \tag{2.26}$$

where we have introduced the notation $\boldsymbol{X}_a = \boldsymbol{X}(\boldsymbol{x}_a)$ and $\boldsymbol{\mu}_a = \boldsymbol{\mu}_0(\boldsymbol{x}_a)$. The training set covariances are written as $\boldsymbol{\Sigma}$, training-test set covariances as $\boldsymbol{\Sigma}_*$ and test set covariances as $\boldsymbol{\Sigma}_{**}$. We already know the values for the training set $\boldsymbol{f}_{1:n}$, and are interested in the conditional distribution of $\boldsymbol{X}_*$ given $\boldsymbol{X}_{1:n}$,

$$\boldsymbol{X}_* \mid \boldsymbol{X}_{1:n} \sim \boldsymbol{\mathcal{N}} \left(\mu', \boldsymbol{\Sigma'} \right), \tag{2.27}$$

where μ' is the conditioned mean function and Σ' is the conditioned variance function. The formula for a conditional distribution from a joint Gaussian distribution is [65]

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \sim \boldsymbol{\mathcal{N}} \left(\begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix}, \begin{bmatrix} \boldsymbol{A} & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{B} \end{bmatrix} \right) \tag{2.28}$$

$$\Rightarrow \boldsymbol{x} \mid \boldsymbol{y} \sim \boldsymbol{\mathcal{N}} \left(\boldsymbol{a} + \boldsymbol{C}\boldsymbol{B}^{-1}(\boldsymbol{y} - \boldsymbol{b}), \boldsymbol{A} - \boldsymbol{C}\boldsymbol{B}^{-1}\boldsymbol{C}^T \right). \tag{2.29}$$

Finally, we use equation (2.29) to rewrite the conditional distribution of $\boldsymbol{X}(\boldsymbol{x}_*)$ given

$\boldsymbol{X}(\boldsymbol{x}_{1:n})$ from equation (2.26) as

$$\begin{aligned} \boldsymbol{X}_* \mid \boldsymbol{X}_{1:n} &\sim \boldsymbol{\mathcal{N}}\left(\boldsymbol{\mu}', \boldsymbol{\Sigma}'\right), \\ \boldsymbol{\mu}' &= \boldsymbol{\mu}_{1:n} + \boldsymbol{\Sigma}_* \boldsymbol{\Sigma}_{**}^{-1} (\boldsymbol{f}_{1:n} - \boldsymbol{\mu}_*), \\ \boldsymbol{\Sigma}' &= \boldsymbol{\Sigma} - \boldsymbol{\Sigma}_* \boldsymbol{\Sigma}_{**}^{-1} \boldsymbol{\Sigma}_*^T. \end{aligned} \tag{2.30}$$

The posterior mean, $\boldsymbol{\mu}'$, is a weighted average between the prior $\boldsymbol{\mu}_{1:n}$ and an estimate based on the data $\boldsymbol{f}_{1:n}$, with a weight dependent on the chosen kernel. The posterior variance function $\boldsymbol{\Sigma}'$ is the prior covariance $\boldsymbol{\Sigma}$ minus a term that corresponds to the reduction in variance from observing $\boldsymbol{f}_{1:n}$.

2.1.1 Choosing the mean function μ_0 and Kernel Σ_0

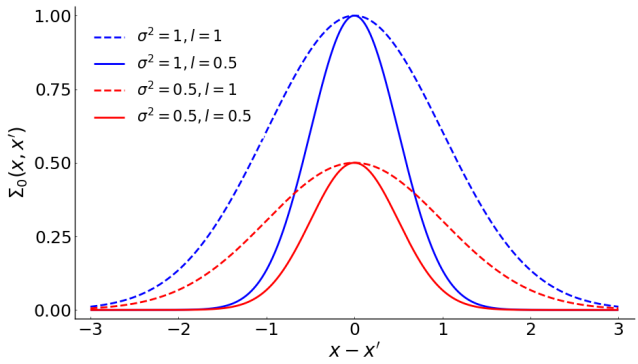


Figure 2.4: Examples of Squared Exponential kernels with different values of variance, σ^2, and lengthscale, l, calculated using equation (2.33).

The mean function should be chosen to represent the mean of the data. The most common option is to choose a constant value [62],

$$\mu_0(x) = \overline{\boldsymbol{f}(\boldsymbol{x}_{1:k})}, \tag{2.31}$$

$$= \sum_{i=1}^{k} \frac{f(x_i)}{k} \tag{2.32}$$

as it is usually simplest and we often do not have more information about the shape of the

target function, $f(x)$.

The covariance kernel should have the property that points closer in input space are more strongly correlated. The kernel is additionally required to be a positive semi-definite function such that the covariance matrix has non-negative eigenvalues and it is possible to calculate the inner product $\Sigma_* \Sigma_{**}^{-1} \Sigma_*^T$. One commonly used kernel is the Gaussian, or power exponential kernel, popular because it produces a smooth prior on functions sampled from the GP [62],

$$\Sigma_0(x, x') = \sigma^2 \exp\left(-\frac{||x - x'||^2}{2l^2}\right). \tag{2.33}$$

The hyperperameters σ^2 and l are parameters of the kernel describing the variance and length scale of the kernel, respectively. We can indicate the hyperparameters using a vector $\boldsymbol{\theta} = (\sigma^2, l)$. Examples of squared exponential functions with various hyperparameters are shown in Figure 2.4. The variance, σ^2, is a scale factor, which describes the height of the kernel. The lengthscale, l, describes the width of the kernel, and determines the correlation distance between two points, $|x - x'|$. A high value of l will produce a wide kernel with a large correlation between points, (x, x'), even when the distance between them, $|x - x'|$, is large.

2.1.2 Tuning hyperparameters

The variance σ^2 determines the average distance of the function away from its mean. Examples of GPRs using kernels of different variance values are shown in Figure 2.5. When the variance is too low, as in Figure 2.5a, the resulting GPR is flat and remains close to the mean function. When the variance is high, as in Figure 2.5b, the GPR is able to capture large variations, however the modelled function results in high uncertainty and can over-emphasise the effect of outliers.

The lengthscale, l, determines the scale of variation, or smoothness, of the function. In general, small lengthscales mean that the function value can have large variations over a small space, while large lengthscales characterise smooth functions that vary over larger spaces. The lengthscale determines the distance that we can reliably extrapolate from the

sample data. Examples of GPRs using kernels of different lengthscales are shown in Figure 2.6. When the lengthscale is too low, as in Figure 2.6b, the resulting GPR can change too quickly, resulting in oscillations in the function. When the lengthscale is too high, as in Figure 2.6a, the resulting GPR is too smooth and does not accurately capture variations in the function, and the variance in the distribution is reduced, suggesting that the function is represented precisely by the mean of the distribution when this is not necessarily the case.

To optimise hyperparameters, we can use the method of maximum likelihood estimate (MLE) [61]. Given observations $\boldsymbol{f} = \boldsymbol{f}_{1:n}$ and a set of hyperparameters, $\boldsymbol{\theta} = (\sigma^2, l)$, which define the covariance matrix, $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, using equations (2.33) and (2.19), we calculate the likelihood of obtaining these observations under the prior. In this case, the posterior distribution is a multivariate Gaussian distribution, with probability density given by equation (2.15). The likelihood is therefore defined as

$$P(\boldsymbol{f}|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{d/2}\sqrt{\det\boldsymbol{\Sigma_0}(\boldsymbol{\theta})^{-1}}} e^{-\frac{1}{2}\boldsymbol{f}^T\boldsymbol{\Sigma_0}(\boldsymbol{\theta})^{-1}f}. \tag{2.34}$$

To obtain the MLE, we set $\boldsymbol{\theta}$ to the value that maximises this likelihood,

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \frac{1}{(2\pi)^{d/2}\sqrt{\det\boldsymbol{\Sigma_0}(\boldsymbol{\theta})^{-1}}} e^{-\frac{1}{2}\boldsymbol{f}^T\boldsymbol{\Sigma_0}(\boldsymbol{\theta})^{-1}f}. \tag{2.35}$$

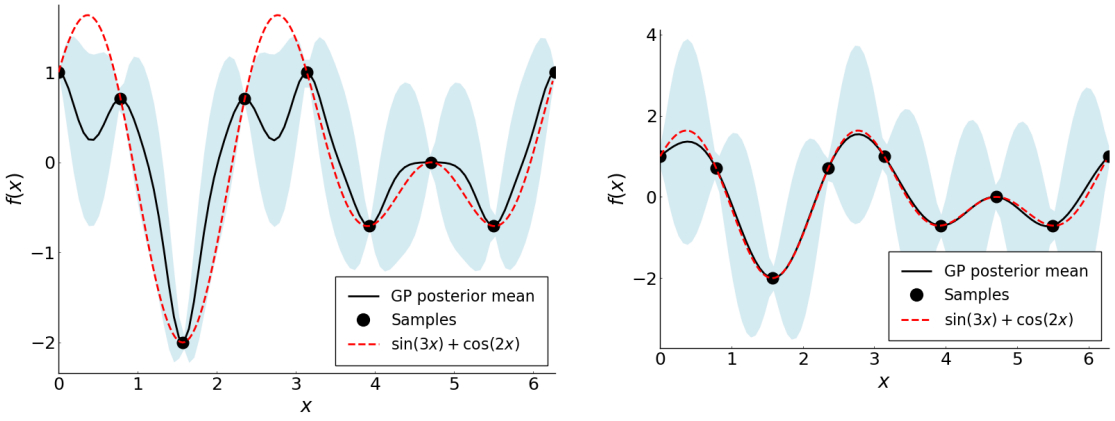
To find the maximum, we then set the derivative with respect to $\boldsymbol{\theta}$ to zero,

$$\frac{\partial}{\partial\boldsymbol{\theta}}\left[\frac{1}{(2\pi)^{d/2}\sqrt{\det\boldsymbol{\Sigma_0}(\boldsymbol{\theta})^{-1}}} e^{-\frac{1}{2}\boldsymbol{f}^T\boldsymbol{\Sigma_0}(\boldsymbol{\theta})^{-1}f}\right] = 0. \tag{2.36}$$

In practice, this calculation is non-trivial, especially with large numbers of points or in many dimensions. Instead of solving equation (2.36) analytically, a numerical optimisation method such as conjugate gradients [64] can be used to maximise equation (2.35) to tune hyperparameters.

An example of a GPR with well-tuned hyperparameters is shown in Figure 2.7b. The hyperparameters are optimised by calculating the likelihood of observations, $\boldsymbol{f}(\boldsymbol{x})$, for

varying hyperparameter values, shown in Figure 2.7a. The pair of parameters that give the maximum likelihood are used to calculate the GPR. The lengthscale is sufficiently small to capture variations in the data, whilst large enough to avoid adding extra oscillations in the function, and the variance is large enough to capture deviations from the mean of the function, whilst small enough to avoid adding excess uncertainty in the posterior distribution. Note that the GPR does not accurately represent the function for all values of x. This is because the function value at more points is needed to fully represent the function. Next, we discuss an acquisition function, the expected improvement, which is used in BGO to decide which point to sample the function at next.



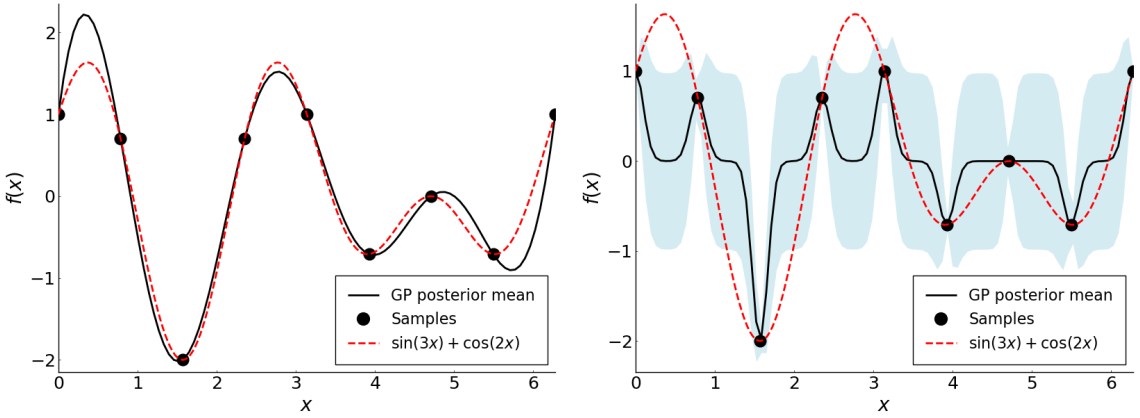
(a) Kernel variance $\sigma^2 = 0.2$.

(b) Kernel variance $\sigma^2 = 5.0$.

Figure 2.5: GPRs with varying values of kernel variance σ^2 for the function $f(x) = \sin(3x) + \cos(2x)$ with samples at $(0, \pi/4, \pi/2, ..., 2\pi)$. The lengthscale is fixed, $l = 0.6$. The ribbon represents a 95% confidence interval around the mean from the GP posterior probability distribution. The values of σ^2 and l are chosen for presentation clarity only, and are not likely to be optimal for this particular function.

2.2 Acquisition function: expected improvement

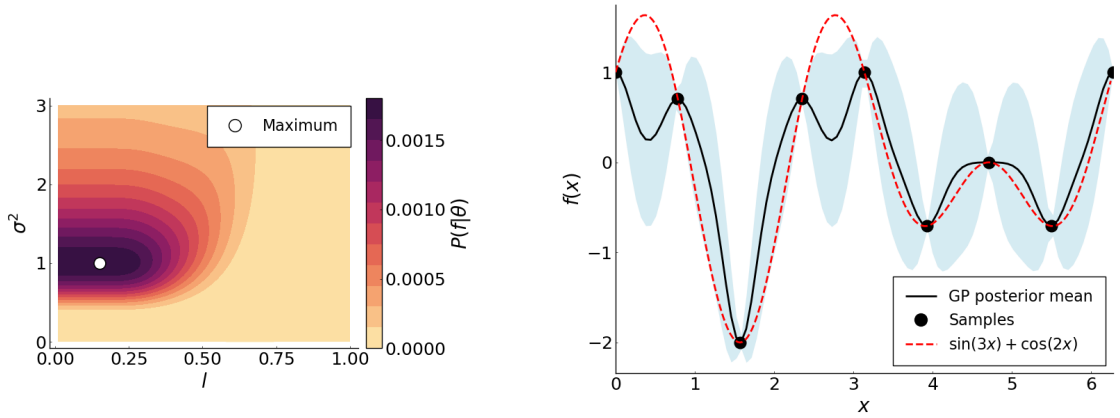
The acquisition function is used at each iteration, i, of a BGO process to decide at which point, $\boldsymbol{x}_i$, to sample the function, $f(\boldsymbol{x})$, next. The most commonly used acquisition function is the expected improvement, as it is simple to use and performs well in most situations [66]. At each iteration the expected improvement acquisition function aims to choose the point, x, which gives the maximum improvement in function value over the current



(a) Lengthscale $l = 1.5$.

(b) Lengthscale $l = 0.1$.

Figure 2.6: GPRs with varying values of lengthscale l for the function $f(x) = \sin(3x) + \cos(2x)$ with samples at $(0, \pi/4, \pi/2, ..., 2\pi)$. The kernel variance is fixed, $\sigma^2 = 0.5$. The ribbon represents a 95% confidence interval around the mean from the GP posterior probability distribution. Note that in Figure 2.6a, the confidence ribbon is not visible because the lengthscale is too high, and the GPR suggests that the function value at each point is known exactly. The values of σ^2 and l are chosen for presentation clarity only, and are not likely to be optimal for this particular function.



(a) The likelihood, $p(\boldsymbol{f}|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = [l, \sigma^2]$ is a vector of hyperparameters, of observations $\boldsymbol{f}(\boldsymbol{x})$, calculated using equation (2.34). The value of $\boldsymbol{\theta}$ with the maximum likelihood is marked as a white circle.

(b) GPR for the function $f(x) = \sin(3x) + \cos(2x)$ with samples at $(0, \pi/4, \pi/2, ..., 2\pi)$. The hyperparameters are those which give the maximum likelihood of observations $\boldsymbol{f}(\boldsymbol{x})$, shown in Figure 2.7a. The lengthscale is $l = 0.2$ and kernel variance is 1.0. The ribbon represents a 95% confidence interval around the mean from the GP posterior probability distribution.

sample optimum. Choosing where to evaluate is based on a trade off between exploration of unknown areas and exploitation of areas with the highest prior mean. In the context of optimisation, evaluating at points with high expected quality is desirable as optima are likely to be located close to these points. Evaluating at points with high uncertainty is also valuable as it increases knowledge in locations far from previously measured points, and this reduces the chances of getting stuck at local minima and missing global optima. The "exploration vs. exploitation tradeoff" is also common in other optimisation domains [67].

At iteration n, we have n points $\boldsymbol{x}_{1:n}$ at which the function is sampled at, and observed values $\boldsymbol{f}(\boldsymbol{x}_{1:n})$. The current optimal point is the previously evaluated point with the largest observed value, and $f_n^* = \max_{m \leq n} f(x_m)$ is the function value at this point. If we have one extra evaluation to complete, we choose a point x and observe $f(x)$. After the next evaluation, the value of the best point observed will be

$$f_{n+1}^* = \begin{cases} f(x), & \text{if } f(x) \geq f_n^*, \\ f_n^*, & \text{if } f(x) < f_n^*, \end{cases} \tag{2.37}$$

The improvement in optimum value after this iteration is $\max([f(x) - f_n^*], 0)$. We want to choose the point x to maximise the improvement, however the function value $f(x)$ is unknown until after the function is evaluated. Instead, we attempt to choose the point, x, that maximises the expected improvement.

We model our experiment on probability space, $(\Omega, F, \mathbb{P})$, where Ω is the set of outcomes, F is the set of possible events, and $\mathbb{P}$ is the probability measure on the sample space (Σ, F). Then, we can write the expected improvement as an integral over probability space,

$$\begin{aligned} EI_n(x) &:= E_n[\max([f(x) - f_n^*], 0)] \\ &:= \int_\Omega \max([f(x) - f_n^*], 0) d\mathbb{P}. \end{aligned} \tag{2.38}$$

where $E_n[.]$ is the expectation taken under the posterior distribution by the gaussian process regression over the sampled function values $\boldsymbol{f}(\boldsymbol{x}_{1:n})$. The domain, Ω We assume

that the function value, $f(x)$, is normally distributed according to the posterior distribution at point x, $f(x) \sim \mathcal{N}(\mu_n(x), \sigma_n(x)^2)$. We can reparameterise our estimate of $f(x)$ using the standard normal distribution, $f(x) = \mu_n(x) + \sigma_n(x)\epsilon$, where $\epsilon \sim \mathcal{N}(0,1)$ and $-\infty < \epsilon < \infty$. The probability density function (pdf) and cumulative distribution function (cdf) of ϵ are given by

$$\phi(\epsilon) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\epsilon^2}{2}} \tag{2.39}$$

$$\Phi(\epsilon) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\epsilon} e^{\frac{-\epsilon^2}{2}} d\epsilon, \tag{2.40}$$

respectively. Then, the expected improvement from equation (2.38) can be written in terms of ϵ,

$$EI_n(x) = \int_{-\infty}^{\infty} \max([\mu_n(x) + \sigma_n(x)\epsilon] - f_n^*], 0)\phi(\epsilon)\ d\epsilon, \tag{2.41}$$

where we have used the change of variables theorem [68]. The change of variables theorem states that if a random variable, X, is continuous, the expectation value can be written as an integral with respect to the probability density function of X,

$$E[X] = \int_{\Gamma} X d\mathbb{P} = \int_{-\infty}^{\infty} X f(X) dX. \tag{2.42}$$

Due to the maximum function, the integrand is non-zero only where $f_n^* > \mu_n(x) + \sigma_n(x)\epsilon$, and we can therefore rewrite equation (2.42) as

$$\int_{-\infty}^{\frac{\mu_n(x) - f_n^*}{\sigma_n(x)}} (\mu_n(x) + \sigma_n(x)\epsilon - f_n^*)\phi(\epsilon)\ d\epsilon. \tag{2.43}$$

Using the definitions of the pdf and cdf from equations (2.39) and (2.40), along with

integration by parts, we can calculate the expectation value,

$$\begin{aligned} EI_n(x) =& \int_{-\infty}^{\frac{\mu_n(x)-f_n^*}{\sigma_n(x)}} (\mu_n(x) + \sigma_n(x)\epsilon - f_n^*)\phi(\epsilon)\ d\epsilon, \\ =&(f_n^* - \mu_n(x))\Phi\left(\frac{\mu_n(x) - f_n^*}{\sigma_n(x)}\right) - \sigma_n(x)\int_{-\infty}^{\frac{\mu_n(x)-f_n^*}{\sigma_n(x)}} \epsilon\ \phi(\epsilon)\ d\epsilon, \\ =&(f_n^* - \mu_n(x))\Phi\left(\frac{\mu_n(x) - f_n^*}{\sigma_n(x)}\right) + \frac{\sigma_n(x)}{\sqrt{2\pi}}\left[e^{\frac{-\epsilon^2}{2}}\right]_{-\infty}^{\frac{\mu_n(x)-f_n^*}{\sigma_n(x)}}, \\ =&(f_n^* - \mu_n(x))\Phi\left(\frac{\mu_n(x) - f_n^*}{\sigma_n(x)}\right) + \sigma_n(x)\phi\left(\frac{f_n^* - \mu_n(x)}{\sigma_n(x)}\right). \end{aligned} \tag{2.44}$$

Tidying up the notation, we arrive at the expression

$$EI_n(x) = \Delta_n(x)\Phi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right) + \sigma_n\phi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right), \tag{2.45}$$

where $\mu_n(x)$ and $\sigma_n(x)$ are the posterior mean and standard deviation of the posterior distribution at point x, respectively, and $\Delta_n(x) = \mu_n(x) - f_n^*$. The optimum choice for the next point is then the point with the largest expected improvement,

$$\boldsymbol{x}_{n+1} = \underset{n}{\operatorname{argmax}}\, EI_n(x). \tag{2.46}$$

There are a variety of approaches to solving equation (2.46) to calculate the next optimum point, $\boldsymbol{x}_{n+1}$. Unlike the objective function, $f(x)$, $EI_n(x)$ is inexpensive and easy to evaluate from sampled function values $\boldsymbol{f}(\boldsymbol{x}_{1:n})$, and allows for evaluation of first- and second-order derivatives. As such, we can use a first-order or second-order optimisation method to solve equation (2.46), such as gradient ascent or Newton's root finding method, a second-order iterative optimisation method [69]. When using Newton's root finding method, at each iteration, a parabola is fit to the function $f(x)$ at the trial value x_i, with the same slope and curvature as the function at that point. At the next iteration, the trial value x_{i+1} is updated to the maximum of the parabola.

2.3 The Bayesian global optimisation algorithm

Now that we have discussed the two components of the BGO method in detail (GPR and the acquisition function), we will provide a summary of the process. Pseudocode for the BGO method is shown below:

Set the budget of function evaluations, N

Place a Gaussian process prior on the objective function, $f(x)$

Evaluate $f(x)$ at $n_0 < N$ points chosen according to an initial space-filling design

Set $n = n_0$

while $n < N$ **do**

Update the posterior distribution of f, given by $\boldsymbol{X}_* \mid \boldsymbol{X}_{1:n}$, using the current vector of function values, $\boldsymbol{f}(\boldsymbol{x}_{1:n})$ and equation (2.30)

Use the mean and standard deviation of the current posterior distribution at point x to calculate the acquisition function in equation (2.45)

Determine x_n, the maximum of the acquisition function over x

Evaluate the function, $f(x_n)$, at the next point x_n

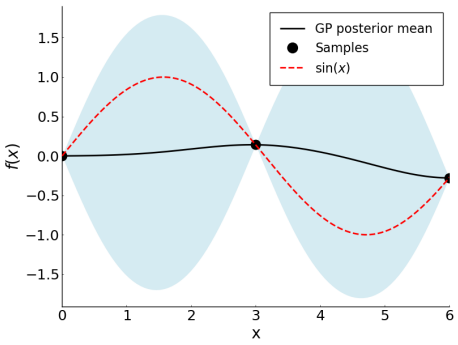
Increment n, $n \to n + 1$

Return a solution, either the point evaluated with the largest objective function $f(x)$, or the point with the largest posterior mean if this point was not sampled

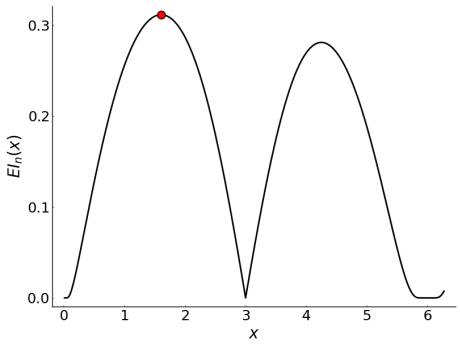
end while

An example of one iteration of a BGO process to find the maximum of the function $\sin(x)$ is shown in Figure 2.8. First a Gaussian process regression is calculated using the Julia package GaussianProcesses.jl [63] from an initial 3 points, shown in (a). Then, the expected improvement acquisition function is calculated over a grid with spacing $\Delta x = 0.01$ using equation (2.45), shown in (b). An additional sample is taken, using the point x that gives the maximum expected improvement value (indicated with a red marker in (b)), and the Gaussian process regression is calculated using the 3 initial samples as well as the new sample, shown in (c). As we are maximising the function, the optimal point is the point with the highest calculated function value, $f_n^* = \max(f(x))$. When we compare

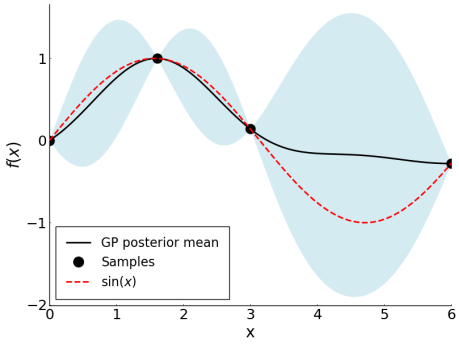
the calculated maximum with the true maximum in Figure 2.8c we see that after just one iteration, the optimal point found using BGO is very close to the true maximum of the function, showing that the optimisation process was successful.



(a) Gaussian process regression for the function $\sin(x)$ with 3 samples at points $x = 0.0, 3.0, 6.0$. The ribbon represents a 95% confidence interval around the mean from the GP posterior probability distribution.



(b) The expected improvement for the Gaussian process regression shown in (a). The maximum is indicated with a red marker.



(c) Gaussian process regression for the function $\sin(x)$ with 4 samples. The ribbon represents a 95% confidence interval around the mean from the GP posterior probability distribution.

Figure 2.8: An example of the process for one iteration in a Bayesian Optimisation process for the function $f(x) = \sin(x)$.

2.4 Optimising a noisy function

BGO will be used here to optimise detector placement, based on the stochastic movements of a finite number of bats in a colony. Due to this stochasticity, we need to modify the BGO algorithm to account for noisy dynamics. When the noise is additive, we can

approximate the function as

$$g(x) = f(x) + \eta(x), \tag{2.47}$$

where $\eta(x)$ represents noise and $f(x)$ approximates the true function without noise. We assume the noise is normal, distributed as $\eta(x) \sim \mathcal{N}(0, \sigma^2_{noise})$, and uncorrelated between variables. We then transform a noise-free kernel $\mathbf{\Sigma}_0$ into a noisy kernel $\mathbf{\Sigma}_{noise}$,

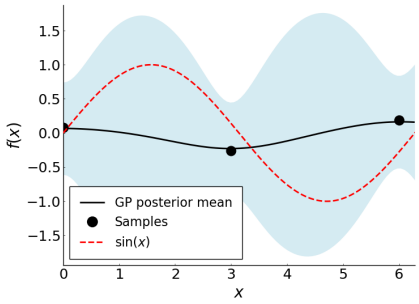
$$\mathbf{\Sigma}_{noise} = \mathbf{\Sigma}_0 + \sigma^2_{noise}\mathbf{I} \tag{2.48}$$

$$= \begin{bmatrix} \Sigma_0(x_1, x_1) & \dots & \Sigma_0(x_1, x_k) \\ \vdots & \ddots & \vdots \\ \Sigma_0(x_k, x_1) & \dots & \Sigma_0(x_k, x_k) \end{bmatrix} + \sigma^2_{noise}\mathbf{I}, \tag{2.49}$$

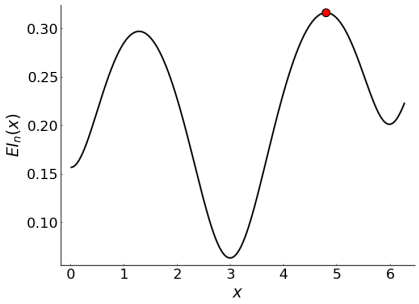
where $\mathbf{I}$ is the identity matrix. When the function is noisy, we also need to modify the definition of the maximum function value, f^*_n, to use the posterior mean instead of the point with the highest calculated function value, $f^*_n = \max(\boldsymbol{\mu}(x))$. This is because the addition of noise can generate fake maxima that are not found at the same point as the true maxima of the noiseless function.

An example of one iteration of a BGO process for a noisy function using GaussianProcesses.jl [63] is shown in Figure 2.9. First a Gaussian process regression is calculated from an initial 3 points, shown in (a). Then, the expected improvement acquisition function is calculated over a grid with spacing $\Delta x = 0.01$ using equation (2.45), shown in (b). An additional sample is taken, using the point x that gives the maximum expected improvement value (indicated with a red marker in (b)), and the Gaussian process regression is calculated using the 3 initial samples as well as the new sample, shown in (c). As the function is noisy, the optimal point is the point with the maximum posterior mean, $f^*_n = \max(\boldsymbol{\mu}(x))$. In this case, the maximum point is not as close to the true maximum as for the case with no noise, in Figure 2.8. However, when we repeat the process for a further iteration, we arrive at a maximum point which is much closer to the true maximum, showing that BGO can be effective at optimising noisy functions, although it is likely that more iterations are needed to optimise functions with noise. Additionally, we note that noise increases

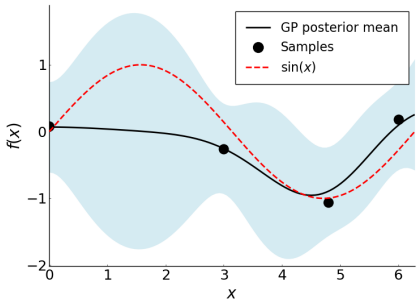
uncertainty and therefore the confidence intervals are wider than in the noise-free example in Figure 2.8.



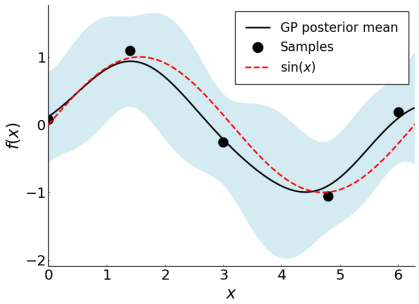
(a) Gaussian process regression for the function $f(x) = \sin(x) + 0.25\eta(x)$, where $\eta(x) \sim \mathcal{N}(0, 1)$, with 3 samples at $x = 0, 3, 6$.



(b) The expected improvement for the Gaussian process regression shown in (a). The maximum is indicated with a red marker.



(c) Gaussian process regression for the noisy function $f(x) = \sin(x) + 0.25\eta(x)$, where $\eta(x) \sim \mathcal{N}(0, 1)$, with 4 samples.



(d) Gaussian process regression for the noisy function $f(x) = \sin(x) + 0.25\eta(x)$, where $\eta(x) \sim \mathcal{N}(0, 1)$, with 5 samples.

Figure 2.9: An example of the process over 4 iterations in a Bayesian Optimisation process for the noisy function $f(x) = \sin(x) + 0.25\eta(x)$, where $\eta(x) \sim \mathcal{N}(0, 1)$. The ribbons represent 95% confidence intervals around the mean from the GP posterior probability distribution.

3 Overview of mathematical techniques

Methods using Bayesian statistics are useful for optimising functions and estimating parameters in a variety of situations. In Chapter 3, we will use ABC to estimate parameters to fit movement models to data from radio tracking surveys. In Chapter 4, we will use the movement models developed in Chapter 3 to design a method to estimate roost locations given data from bat surveys. We will compare a method using a random search and

deterministic movement models for movement to a method using ABC and stochastic movement models. We will use BGO in Chapter 5 along with the method developed in Chapter 4 to determine the optimal positions to place acoustic detectors in order to produce the most useful data for locating a roost. Next, in Chapter 3, we will look at mathematical models of bat movement.

Chapter Three

Deterministic models for bat movement

In this chapter our aim is to use data gained from a radio tracking survey to produce a mathematical formulation of bat movement which qualitatively and quantitatively matches the data. The data suggests that Greater Horseshoe bat movement can be split into 2 phases during the night. We aim to provide a mechanistic description of these 2 phases, which allow us to understand and describe the motion. Later, in Chapter 4, we will use this model of motion to help estimate the most likely locations of Greater Horseshoe bat roost.

1 Radiotracking survey

A radiotracking study was conducted at three Greater Horseshoe bat roosts in Devon, UK, to study the usage of the landscape surrounding the roosts [12]. 12 bats were fitted with radio tags and studied over 24 nights between 19/05/2009 and 12/06/2009. Due to a limited number of workers and limited battery life on the tags, each bat was not tracked every night. Three day roosts were used by bats in the study, with some bats using different roosts on different days. The roost locations were known before the study was conducted, and workers attempted to identify the roost used by each bat on each night of the survey by scanning for the radio signal at each of the known roosts close to sunset. The roost used by each bat was not identified on every night.

For this analysis, only the data from nights when a bat's roost was known was used, since the dispersal from the roost is important. After removing data from nights with unknown roosts, 53 bat-nights remain and were used for this analysis. Here, one bat-night refers to an event in which one bat was tracked over one night. Two bat-nights could refer to two nights in which one bat was tracked, or one night in which two bats were tracked. A total of 322 bat locations were used for this analysis.

A map of the three roosts and 322 recorded locations is shown in Figure 3.1, overlaid on a map of the surrounding landscape. We note that the ocean is located to the South and West of the roost locations, and there is little land in these directions.

The trajectory of a bat, referred to here as bat 1, over the 6 nights it was tracked, is shown in Figure 3.2. Bat 1 used 2 of the 3 roosts identified during the course of the study and visited different areas whilst foraging, taking a different route on each night. The number of locations recorded varies each night because the signal was lost during the night on some nights. The recorded locations of all bats over the study period are displayed in a polar histogram in Figure 3.3. This histogram shows that most records are located close to the roost, and are distributed over all angles. We note that there is an anisotropy in recordings, with fewer recordings between 0°and 135°. However, when we refer to the landscape shown in Figure 3.1, we see that to the South and West of the survey area is

the ocean. The South-West corresponds to angles between 0°and 135°, and therefore there is very little suitable landscape in these directions, especially further than 1000m from the roost locations.

A histogram of the recorded distance from the roost of all bats over the study period is shown in Figure 3.4a. The histogram shows that the majority of recorded locations are close to the roost, with the 90th percentile at $r = 1.51$km.

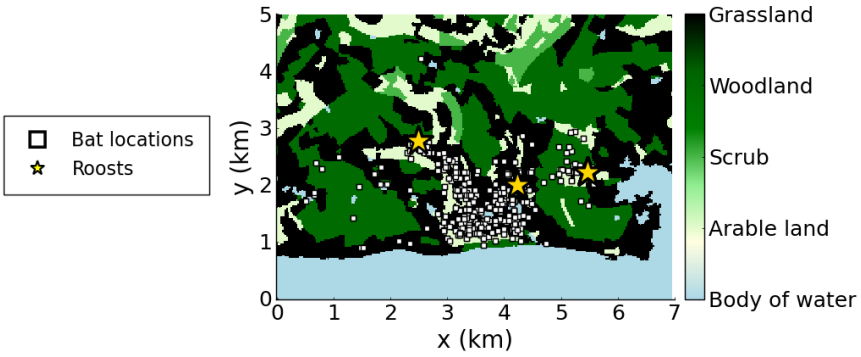


Figure 3.1: The locations of the roost and all recorded bat tracking locations overlaid on a map of the landscape surrounding the survey area. The label 'Body of water' refers to all bodies of water, including ocean, rivers and lakes, and the large body of water to the South and West of the survey area is the ocean.

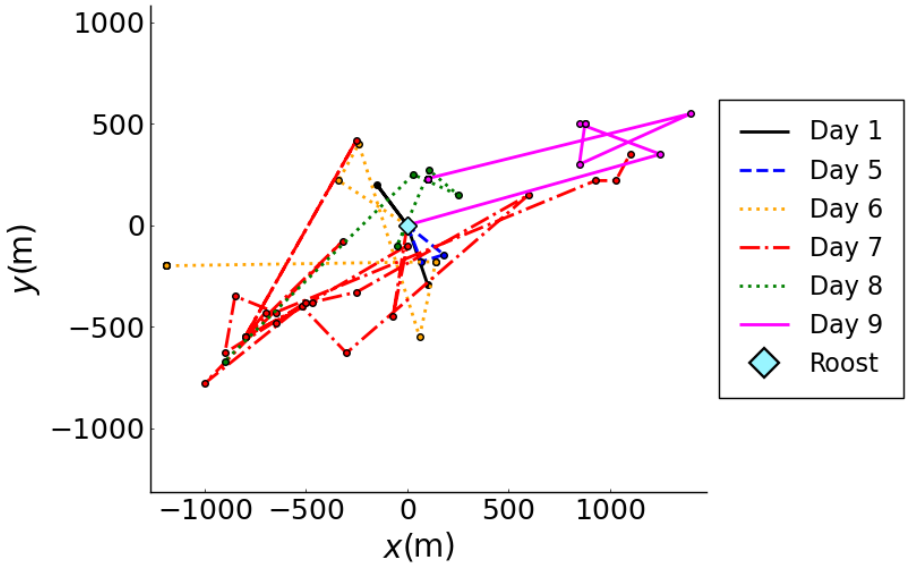


Figure 3.2: The displacement from the roost of bat 1 over 6 nights of the survey. The roost used by bat 1 during the course of the study is shown as a diamond.

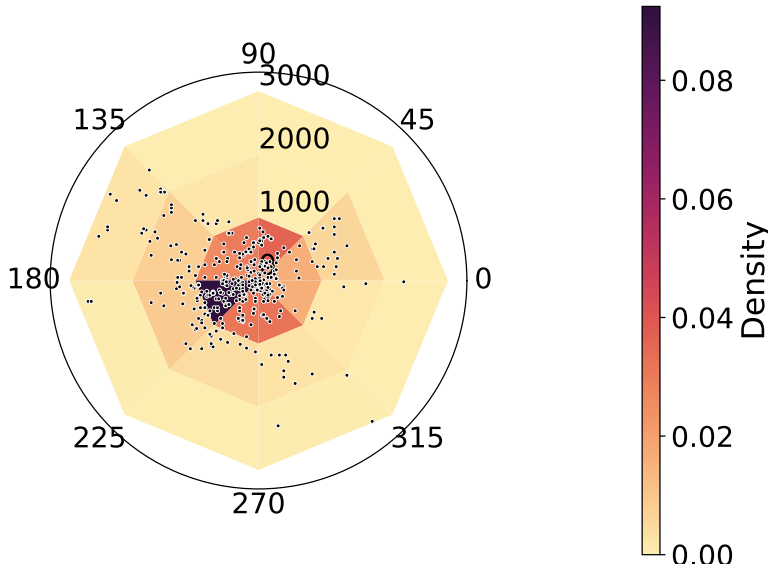
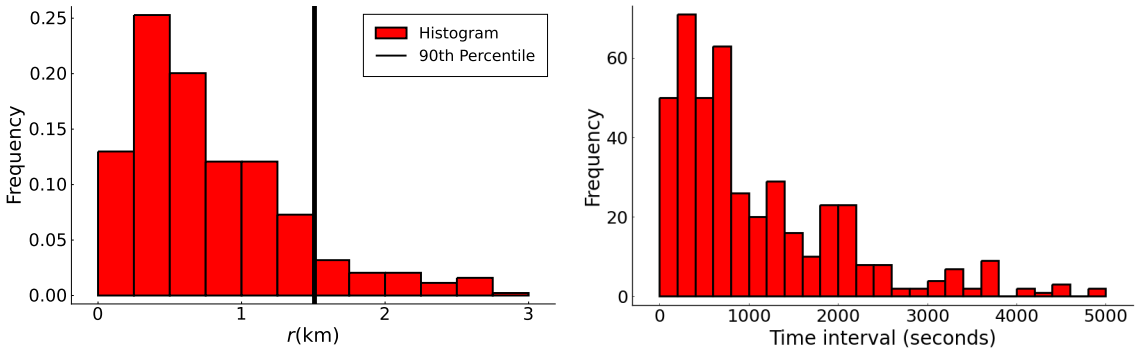


Figure 3.3: Polar histogram of bat locations over the study period. The density for each segment is calculated as the number of locations recorded in the segment divided by the area of the segment. A scatter plot displaying each recording is overlaid on the histogram.



(a) A histogram of the distance from the roost over the study period. The 90th percentile, at $r = 1.51$km, is shown as a vertical line.

(b) A histogram of the time intervals between consecutive recordings. Outliers with time intervals over the 75th percentile at 5000 seconds have been removed.

Figure 3.4: Histograms of the distance from the roost and the time intervals between recordings for the data recorded during the radiotracking survey.

A histogram of time intervals between consecutive recordings is shown in Figure 3.4b, demonstrating the irregularity of recording intervals. We will use the mean-squared distance (MSD) from the roost as a function of time to summarise the data and this will be used as a measure to fit the movement models to the data. In order to calculate the MSD at a given time, we require regularly spaced recordings. The locations were linearly interpolated

between recordings at intervals of $\Delta t = 200$ seconds, as the distribution in Figure 3.4b peaks between 100 and 200 seconds. The MSD from the roost was calculated from the interpolated positions using

$$\left\langle r^2(t) \right\rangle = \frac{1}{N} \sum_{i=1}^{N} ||\boldsymbol{x_i}(t) - \boldsymbol{x_i}(0)||^2, \tag{3.1}$$

where $\boldsymbol{x_i}(t)$ is the location (x, y) of bat-night i at time t, and N is the total number of observations, x_i at time t. The MSD is shown in Figure 3.5. The standard error is given by

$$SE = \sigma_{\langle r^2(t) \rangle} = \sqrt{\frac{\Sigma(r_i^2(t) - \langle r^2(t) \rangle)^2}{N(N-1)}}, \tag{3.2}$$

where $\langle r^2(t) \rangle$ is the MSD, given by equation (3.1). The standard error is useful in calculating confidence intervals around the mean: for a large sample, $1.96SE$ is a 95% confidence interval. This 95% confidence interval is shown as an orange ribbon around the MSD in Figure 3.5. The roost that each bat returned to at the end of the night is not known in the majority of cases, as the signal is lost throughout the night. As a result, we have assumed for the purpose of calculating the MSD that bats have returned to the same roost. As shown in Figure 3.5, the MSD reaches 0 by the end of the night, suggesting that the recordings from the end of the night are all from bats that returned to the same roost at the end of the night. The data indicates two movement phases: phase 1 is an initial rapid dispersal from the roosts, and phase 2 is a gradual return whilst bats are foraging.

During phase 1, for $0 \leq t < 1.5$ hours, the MSD seems to increase linearly as bats are dispersing. The standard error, and therefore the 95% confidence interval, grows during this phase as the bats spread out throughout the survey area, and are therefore found at different distances from the roost. During phase 2, for $1.5 \leq t < 8$ hours, the MSD decreases at an increasing rate as bats move back towards the roost, shrinking to zero at $t \approx 8$ hours. The variation shrinks to zero during this phase as bats start to converge on the roost.

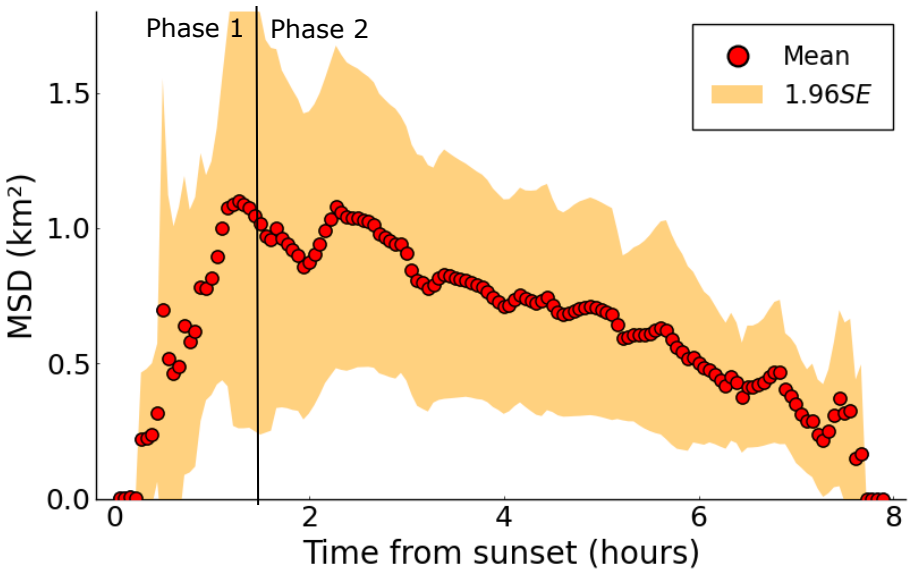


Figure 3.5: The mean-squared distance (MSD) for all radiotracked bats. The red dots are the averaged values over 53 trajectories and a 95% confidence interval, calculated as $1.96SE$, is shown as an orange ribbon.

2 Modelling methods

As discussed in Section 1, the movement is in two distinct phases, a dispersal followed by a return to the roost. In the next section various models for each phase will be compared. Although the data is discrete and stochastic in nature, we will assume that the underlying probability distribution of the ensemble dynamics is continuous in both space and time. Therefore, we will be able to use partial differential equations to analytically describe the evolution of the probability distribution.

2.1 Phase 1: dispersal

Diffusion models are widely used to model animal movement, specifically dispersal, for a number of species [34]. Diffusion describes the movement of matter due to a random walk process. In this case, we wish to describe the dispersal during phase 1 of movement as bats fly away from the roost to the surrounding areas. It is commonly accepted that bats tend to remain within an area around the roost known as the Core Sustenance Zone, and will forage within this area for the majority of the night [70]. As a result, a diffusion model on a bounded domain is considered here. First we will consider a one-dimensional diffusion model before extending to a two-dimensional model in polar coordinates.

2.1.1 A diffusion model in one dimension

First we will consider diffusion on a one-dimensional domain, $x \in [0, R] \subset \mathbb{R}$, where $R < \infty$. The probability density, $\phi(x, t)$, of finding a bat at position x at time t is given by

$$\frac{\partial \phi(x,t)}{\partial t} = D \frac{\partial^2 \phi(x,t)}{\partial x^2}, \tag{3.3}$$

where the diffusion coefficient, D quantifies the rate of spread. The diffusion coefficient can be a function of both time and position, but here it is a positive constant.

The boundary conditions,

$$\begin{aligned} \frac{\partial \phi(x=0,t)}{\partial x} &= 0, \\ \frac{\partial \phi(x=R,t)}{\partial x} &= 0, \end{aligned} \tag{3.4}$$

specify zero-flux across the boundary, such that bats cannot enter or leave the domain. The initial condition,

$$\phi(x, t=0) = \delta_0(x), \tag{3.5}$$

where $\delta_0(x)$ is the Dirac delta function, specifies that all bats begin the night at the roost before moving away to begin foraging. Theoretically, the boundary and initial conditions are not consistent. However, any computational solution requires a discretisation of the domain, and therefore the initial condition will be represented by a stepwise approximation of the delta function,

$$\phi(x, t=0) = \begin{cases} \frac{1}{\epsilon} & 0 < x < \epsilon \\ 0 & x > \epsilon \end{cases}, \tag{3.6}$$

for some $\epsilon > 0$. Given this initial condition, the computational solution will converge smoothly to a stable solution. The roost is placed at $x = 0$ to mimic the behaviour of diffusion in polar coordinates as we will be extending to polar coordinates later.

2.1.2 Mean squared displacement in the one dimensional diffusion model

The relationship between the expected mean squared displacement (MSD) and time t can be calculated using moments of the probability density ϕ. Although we can solve equation (3.3) directly, we will instead calculate the MSD through calculating the moments of the equation, as this is more useful when considering trajectory data. The moment generating function,

$$\langle x^n \rangle = \int_0^R x^n \phi(x,t) dx, \tag{3.7}$$

gives the nth moment of any probability density function, $\phi(x,t)$ [71]. The MSD is the 2nd moment, and is given by

$$\left\langle x^2 \right\rangle = \int_0^R x^2 \phi(x,t) dx. \tag{3.8}$$

Taking the time derivative of both sides of equation (3.8)and substituting $\frac{\partial \phi(x,t)}{\partial t}$ from the diffusion equation (3.3),

$$\begin{aligned}
\frac{d}{dt}\left\langle x^2 \right\rangle &= \frac{d}{dt}\int_0^R x^2 \phi(x,t) dx, \\
&= \int_0^R x^2 \frac{\partial \phi}{\partial t} dx, \\
&= D \underbrace{\left[x^2 \frac{\partial \phi}{\partial x} \right]_0^R}_{=0} - D \int_0^R 2x \frac{\partial \phi}{\partial x} dx, \\
&= -2D \left[x\phi \right]_0^R + 2D \int_0^R \phi dx, \\
&= -2DR\phi(R,t) + 2D,
\end{aligned} \tag{3.9}$$

and therefore, integrating with respect to time gives

$$\left\langle x^2 \right\rangle = 2D \left(t - \int_0^t R\phi(R,\tau) d\tau \right). \tag{3.10}$$

For t small, $\phi(R,t) \approx 0$, since the probability of reaching the boundary over a short period of time is small because the initial condition is approximately a delta function, as given by equation (3.6). Therefore, over a short timescale, that is for

$$\sqrt{2Dt} \ll R, \tag{3.11}$$

the expected MSD for diffusion in one dimension is directly proportional to time,

$$\left\langle x^2 \right\rangle \approx 2Dt. \tag{3.12}$$

An expression for the mean squared displacement over a long timescale can also be derived using equation (3.8). Over long timescales, we expect the probability density to be uniformly spread across the domain,

$$\phi(x,t) \approx \frac{1}{R}. \tag{3.13}$$

Substituting this into equation (3.10) gives

$$\left\langle x^2 \right\rangle = \frac{R^2}{3} \tag{3.14}$$

and therefore the mean squared displacement is constant over long timescales. The MSD given by equation (3.14) may not immediately appear to be consistent with the expression in equation (3.10). However, we note that the probability density at R, $\phi(R,t)$ varies on a short timescale, for $\sqrt{2Dt} \ll R$, and is only constant at larger timescales. Therefore, we cannot simply substitute the long timescale value of $\phi(R,t)$ into equation (3.10).

The expected MSD is plotted in Figure 3.6, using a numerical solution to the diffusion equation, equation (3.3), solved using a Runge-Kutta Order 4 method implemented in `DifferentialEquations.jl` [72]. The plot shows an initial linear section which tends to a constant value, consistent with equations (3.12) and (3.14).

2.1.3 A discretised diffusion model

The diffusion equation on a bounded domain can be solved using a discretised ODE description [73]. Here, we use a discretised ODE description of the domain to numerically solve the diffusion PDE for phase 1 of movement. The discretised formulation discussed here will also provide a framework for the models used later for phase 2 of movement.

The domain can be discretised into n boxes, each of length $h = R/n$. Let x_i be the position of the outer edge of box i, and assume that the probability density, $\phi_i(x,t)$, is constant on each box and evolves over time according to the diffusion process. A diagram of the motion is shown in Figure 3.7. A finite difference approximation is used to describe

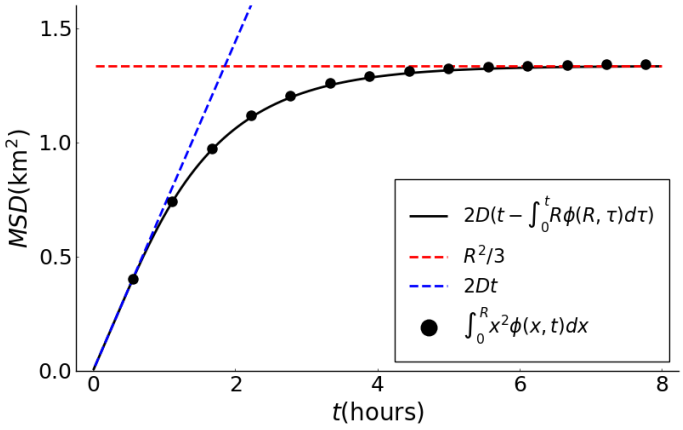


Figure 3.6: The expected MSD for a 1D diffusion model from equation (3.10) using a numerical solution to the diffusion equation with $D = 100\text{m}^2\text{s}^{-1}$ and $R = 2000\text{m}$. The expectation value calculated using equation (3.10) is shown along with the short and long timescale expressions in equation (3.12) and equation (3.14) for comparison.

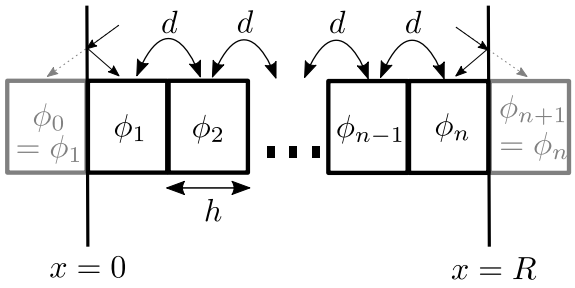


Figure 3.7: A diagram to illustrate the movement of probability density between boxes in the discretised diffusion model. Diffusion between boxes is represented by d and the probability density in each box i is denoted by ϕ_i.

the movement of probability density between boxes. The central difference approximation to the second order derivative at box i is given by

$$\left.\frac{\partial^2\phi}{\partial x^2}\right|_{x=x_i} = \lim_{h\to 0}\frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{h^2}, \tag{3.15}$$

where $\phi_i = \phi(x_i)$ and $\phi_{i\pm1} = \phi(x_i \pm h)$.

Substituting the discretisation given by equation (3.15) into the diffusion equation (3.3) gives the equation for box 1 at $x = 0$,

$$\frac{d\phi_1}{dt} = \frac{D}{h^2}(\phi_0 - 2\phi_1 + \phi_2). \tag{3.16}$$

However, due to the reflective boundary condition in equation (3.4), any bats in a trajectory that would pass through the boundary are reflected back in the direction of the roost. Bats that would pass from box 1 to an imaginary box 0 are instead reflected back, and therefore the value of ϕ in the imaginary box 0 is the same as in box 1, $\phi_0 = \phi_1$, and

$$\frac{d\phi_1}{dt} = \frac{D}{h^2}(\phi_2 - \phi_1). \tag{3.17}$$

Similarly, from equation (3.15), for box n at $x = R$,

$$\frac{d\phi_n}{dt} = \frac{D}{h^2}(\phi_{n-1} - 2\phi_n + \phi_{n+1}). \tag{3.18}$$

Due to the reflective boundary condition between box n and box $n+1$, $\phi_{n+1} = \phi_n$, and

$$\frac{d\phi_n}{dt} = \frac{D}{h^2}(\phi_{n-1} - \phi_n). \tag{3.19}$$

Collecting equations (3.15)-(3.19), the set of equations describing the full system is

$$\frac{d\phi_i}{dt} = \begin{cases} d(\phi_i - \phi_{i+1}), & \text{for } i = 1, \\ d(\phi_{i-1} - 2\phi_i + \phi_{i+1}), & \text{for } 2 \leq i \leq n-1, \\ d(\phi_{i-1} - \phi_i), & \text{for } i = n, \end{cases} \tag{3.20}$$

where the discretised diffusion coefficient is given by $d = D/h^2$. The initial condition corresponding to equation (3.5) is approximated by assuming that at time $t = 0$, probability density is concentrated in the first box,

$$\phi_i(0) = \begin{cases} \frac{1}{h}, & \text{for } i = 1, \\ 0, & \text{for } 2 \leq i \leq N. \end{cases} \tag{3.21}$$

Later, in Figure 3.8, we will show that the non-smooth initial condiiton quickly converges to a smooth probability density profile. A diagram illustrating the spread of probability

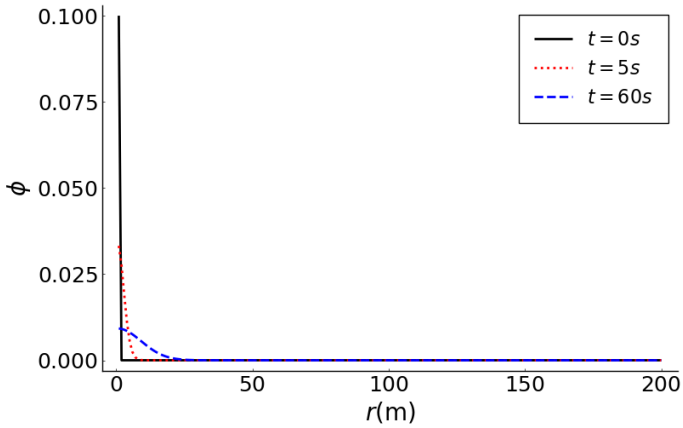


Figure 3.8: The value of $\phi(r)$ at position r for a 1D discretised diffusion simulation with parameters $N = 100$, $R = 2000$m and $D = 100\text{m}^2\text{s}^{-1}$ and initial condition given by equation (3.21) for $t \leq 1$ minute.

density due to the diffusion process is shown in Figure 3.7. This generates a system of n ODEs describing motion over the domain at each timestep and which can be solved using a numerical ODE solver (here, we use a Runge-Kutta Order 4 method implemented in `DifferentialEquations.jl` [72]). The equations for $i = 1$ and $i = N$ correspond to reflective, zero-flux boundary conditions. The result of a simulation with $N = 100$ boxes in a domain of length $R = 2000$m and diffusion coefficient $D = 100\text{m}^2\text{s}^{-1}$ is shown in figures **??** and **??** The probability density over the initial 60 seconds, shown in Figure 3.8, illustrates that, whilst a discrete, non-smooth initial condition is used in equation (3.21), within 60 seconds, the solution converges to a smooth probability density.

The diffusion process spreads the probability density out from the left side of the domain, where ϕ is initially high and tends to homogenise the probability density across the domain over time. After 5 hours, the probability density ϕ is evenly spread throughout the domain and the probability distribution is eventually uniform.

2.1.4 Diffusion in polar coordinates

Next we will consider a diffusion model in polar dimensions to describe dispersal during phase 1 of movement. Bats are not restricted to a single dimension in the real world, and are instead able to fly in any direction in 3D. We will not include the third dimension here,

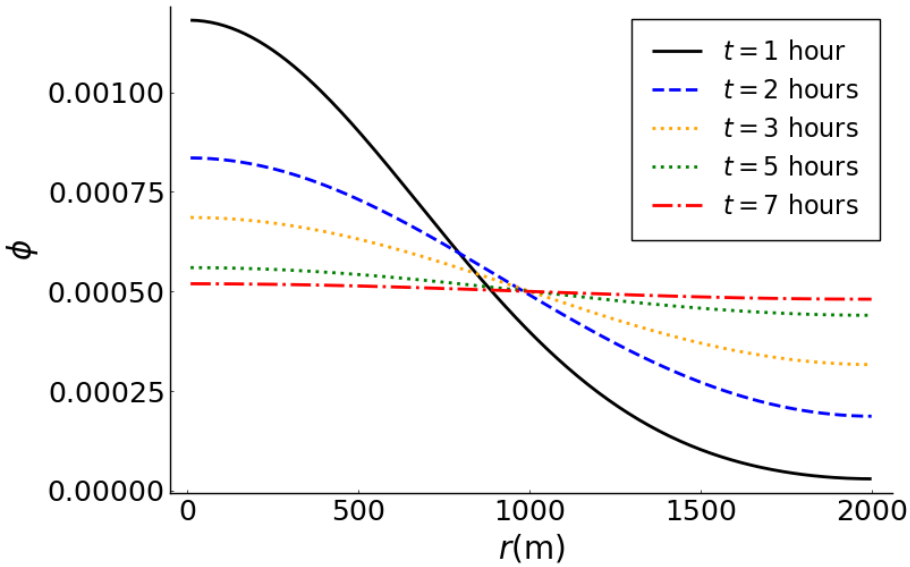


Figure 3.9: The value of $\phi(r)$ at position r for a 1D discretised diffusion simulation with parameters $N = 100$, $R = 2000$m and $D = 100\text{m}^2\text{s}^{-1}$ and initial condition given by equation (3.21) after $t = 1$ hour, 2 hours and 5 hours.

as height is not measured in the radio tracking survey, and it is not needed to describe landscape use. If the roost is at (x_0, y_0) and bats leave the roost at time $t = 0$, the 2D diffusion equation describes the probability density $\phi(x, y, t)$ of finding a bat at position (x, y) at time t,

$$\frac{\partial \phi(x, y, t)}{\partial t} = D\nabla^2 \phi(x, y, t), \tag{3.22}$$

where ∇^2 is the Laplacian, D is the diffusion coefficient, a positive constant that quantifies the rate of spread. The Core Sustenance Zone is denoted by $\Omega \subset \mathbb{R}^2$ and modelled as a disk of radius R centred around the roost. Therefore we will consider the diffusion equation in polar coordinates. Since the domain and initial condition are angularly symmetric, ϕ is only dependent on r and not on the angle, and we drop the angular dependence from the Laplacian. The diffusion equation in radial polar coordinates is then given by

$$\frac{\partial \phi(r, t)}{\partial t} = \frac{D}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \phi(r, t)}{\partial r}\right), \tag{3.23}$$

where r is the distance from the roost, given by $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$. The initial condition,

$$\phi(r, t = 0) = \delta_0(r), \tag{3.24}$$

specifies that all bats begin the night at the roost at position $r = 0$ before moving away at time t to begin foraging. The boundary condition,

$$\frac{\partial \phi(r = R, t)}{\partial r} = 0, \tag{3.25}$$

specifies zero-flux across the boundary such that bats cannot enter or leave the boundary.

2.1.5 Mean squared displacement in the diffusion model in polar coordinates

The relationship between the expected mean squared displacement (MSD) and time t can be calculated using the moment-generating function [71]. The derivation shown here is original, however the method is commonly used for calculating moments using both PDEs and stochastic processes [74, 75, 76]. The 2nd moment defines the MSD, and is defined similarly to equation (3.7),

$$\left\langle r^2 \right\rangle = \int_\Omega r^2 \phi(r, t) d\omega, \tag{3.26}$$

where $\omega = (r, \theta) \subset \Omega$. Taking the time derivative of both sides and substituting $\frac{\partial \phi(r,t)}{\partial t}$ from equation (3.23),

$$\begin{aligned}
\frac{d}{dt}\left\langle r^2\right\rangle &= \frac{d}{dt}\int_\Omega r^2\phi(r,t)d\omega,\\
&= \int_0^{2\pi}\int_0^R r^3\frac{\partial\phi}{\partial t}drd\theta,\\
&= \int_0^{2\pi}\int_0^R r^3\frac{D}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\phi}{\partial r}\right)drd\theta,\\
&= \int_0^{2\pi}\left(\underbrace{\left[Dr^2\left(r\frac{\partial\phi}{\partial r}\right)\right]_0^R}_{=0} - \int_0^R 2rD\left(r\frac{\partial\phi}{\partial r}\right)dr\right)d\theta,\\
&= \int_0^{2\pi}\int_0^R -2r^2D\frac{\partial\phi}{\partial r}drd\theta,\\
&= \int_0^{2\pi}\left(\left[-2r^2D\phi\right]_0^R + \int_0^R 4rD\phi dr\right)d\theta,\\
&= -4\pi R^2D\phi(R,t) + 4D\int_\Omega \phi d\omega, \qquad (3.27)
\end{aligned}$$

and therefore,

$$\frac{d}{dt}\left\langle r^2\right\rangle = 4D(1-\pi R^2\phi(R,t)). \qquad (3.28)$$

Integrating with respect to time gives

$$\left\langle r^2\right\rangle = 4D\left(t - \pi R^2\int_0^t \phi(R,\tau)d\tau\right). \qquad (3.29)$$

For t small, $\phi(R,t)\approx 0$, since the probability of a reaching the boundary over a short period of time is small due to the initial condition. Therefore, over a short timescale, while

$$t \ll \pi R^2, \qquad (3.30)$$

the expected MSD for diffusion is directly proportional to time,

$$\left\langle r^2\right\rangle \approx 4Dt. \qquad (3.31)$$

As noted in Section 1 the MSD in Figure 3.5 is indeed linear during phase 1, and therefore

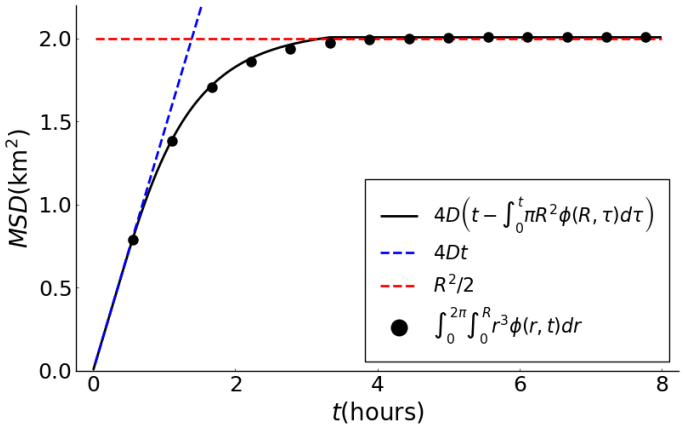


Figure 3.10: The expected MSD for the polar diffusion model defined by equation (3.29) using a numerical solution to the diffusion equation with $D = 100\text{m}^2\text{s}^{-1}$ and $R = 2000\text{m}$. The expectation value calculated using equation (3.26) is shown along with the short and long timescale expressions in equation (3.31) and equation (3.33) for comparison.

consistent with a diffusion model. An expression for mean-squared displacement over a long timescale can also be derived using equation (3.26). Over long timescales, we expect the probability density to be uniformly spread across the domain,

$$\phi(r, t) = \frac{1}{\pi R^2}. \tag{3.32}$$

Substituting this into equation (3.26) gives

$$\begin{aligned} \left\langle r^2 \right\rangle &= \frac{1}{\pi R^2} \int_0^{2\pi} \int_0^R r^3 dr d\theta \\ &= \frac{1}{2} R^2, \end{aligned} \tag{3.33}$$

and therefore the mean squared displacement is constant over long timescales.

The expected MSD is plotted in Figure 3.10, using a numerical solution to the diffusion equation, equation (3.23), solved using `DifferentialEquations.jl` [72]. The plot shows an initial linear section then flattening out to a constant value, consistent with equations (3.31) and (3.33).

2.2 A discretised ODE solution to the two dimensional diffusion model

The polar diffusion equation can be discretised using central difference approximations in the same way as for one dimensional cartesian PDEs [77, 78, 79], and this discretisation will be used to solve the equation in a circular domain. For a circular domain in polar coordinates, $\Omega = [0, R] \times [0, 2\pi]$, the domain can be discretised into N annuli, each of width $h = R/N$. Due to the assumption that the domain is symmetric, we will not consider angular movement around the annuli, and we can reduce the problem to only one dimension. The probability density in each annulus is taken as constant and we let ϕ_i be the value for the ith annulus. The distance from the origin to the outer edge of annulus i is given by r_i. Central difference approximations to equation (3.23) are used to describe the movement of probability density between annuli. First we will expand the differential in equation (3.23),

$$\frac{\partial \phi(r,t)}{\partial t} = D\left(\frac{\partial^2 \phi(r,t)}{\partial r^2} + \frac{1}{r}\frac{\partial \phi(r,t)}{\partial r}\right). \tag{3.34}$$

The first and second order derivatives at r_i are approximated by the central difference approximations,

$$\left.\frac{\partial \phi}{\partial r}\right|_{r=r_i} \approx \frac{\phi(r_i + h) - \phi(r_i - h)}{2h}, \tag{3.35}$$

$$\left.\frac{\partial^2 \phi}{\partial r^2}\right|_{r=r_i} \approx \frac{\phi(r_i - h) - 2\phi(r_i) + \phi(r_i + h)}{h^2}. \tag{3.36}$$

The central difference approximation to equation (3.34) is then

$$\left.\frac{\partial \phi}{\partial t}\right|_{r=r_i} \approx D\left(\frac{\phi(r_i - h) - 2\phi(r_i) + \phi(r_i + h)}{h^2} + \frac{1}{r_i}\frac{\phi(r_i + h) - \phi(r_i - h)}{2h}\right). \tag{3.37}$$

The distance r_i at annulus i is given by

$$r_i = ih, \tag{3.38}$$

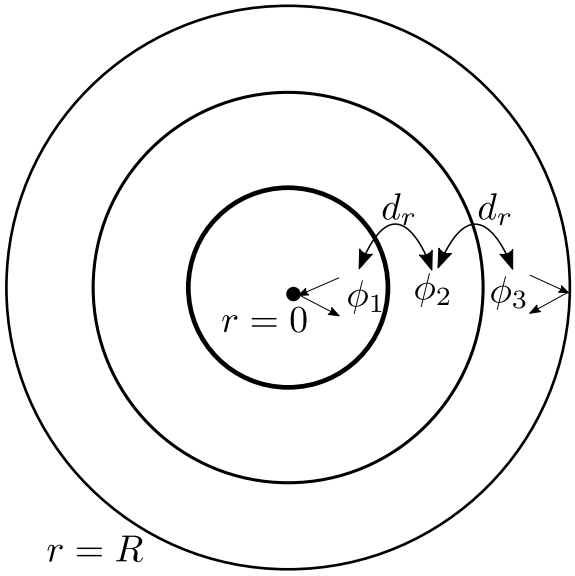


Figure 3.11: A diagram of the discretised diffusion process. Here we illustrate the domain discretised into 3 sections, but the diagram extends analogously to any number of compartments. The probability density shifts between annuli according to the diffusion operator, $d_r = D\left(\frac{\partial^2\phi(r,t)}{\partial r^2} + \frac{1}{r}\frac{\partial\phi(r,t)}{\partial r}\right)$. Reflective boundary conditions ensure that probability density never leaves the domain.

and equation (3.37) can therefore be written as

$$\frac{\partial\phi(r_i)}{\partial t} \approx \frac{D}{h^2}\left(\phi(r_i - h) - 2\phi(r_i) + \phi(r_i + h)\right) + \frac{D}{2ih^2}\left(\phi(r_i + h) - \phi(r_i - h)\right). \tag{3.39}$$

Thus our discrete approximation satisfies the system of equations

$$\frac{d\phi_i}{dt} = \frac{D}{h^2}(\phi_{i-1} - 2\phi_i + \phi_{i+1}) + \frac{D}{2ih^2}(\phi_{i+1} - \phi_{i-1}). \tag{3.40}$$

The equation for annulus 1 at $r = 0$ is

$$\frac{d\phi_1}{dt} = \frac{D}{h^2}(\phi_0 - 2\phi_1 + \phi_2) + \frac{D}{2h^2}(\phi_2 - \phi_0). \tag{3.41}$$

However, due to the reflective boundary condition in equation (3.25), any bats in a trajectory that would pass through the boundary are reflected back in the direction of the roost. Bats that would pass from annulus 1 to an imaginary annulus 0 are instead reflected back, and therefore the value of ϕ in the imaginary annulus 0 is the same as in

annulus 1, and $\phi_0 = \phi_1$.

$$\frac{d\phi_1}{dt} = \frac{D}{h^2}(\phi_2 - \phi_1) + \frac{D}{2h^2}(\phi_2 - \phi_1). \tag{3.42}$$

Similarly, from equation (3.40), for annulus N at $r = R$,

$$\frac{d\phi_N}{dt} = \frac{D}{h^2}(\phi_{N-1} - 2\phi_N + \phi_{N+1}) + \frac{D}{2h^2N}(\phi_{N+1} - \phi_{N-1}). \tag{3.43}$$

Due to the reflective boundary condition between annulus N and annulus $N+1$, $\phi_{N+1} = \phi_N$, and

$$\frac{d\phi_N}{dt} = \frac{D}{h^2}(\phi_{N-1} - \phi_N) + \frac{D}{2Nh^2}(\phi_N - \phi_{N-1}). \tag{3.44}$$

Collecting equations (3.40)-(3.44), the set of equations describing the full system is

$$\frac{d\phi_i}{dt} = \begin{cases} \frac{D}{h^2}(\phi_{i+1} - \phi_i) + \frac{D}{2h^2}(\phi_{i+1} - \phi_i), & \text{for } i = 1, \\ \frac{D}{h^2}(\phi_{i-1} - 2\phi_i + \phi_{i+1}) + \frac{D}{2ih^2}(\phi_{i+1} - \phi_i), & \text{for } 2 \leq i \leq N-1, \\ \frac{D}{h^2}(\phi_{i-1} - \phi_i) + \frac{D}{2ih^2}(\phi_i - \phi_{i-1}), & \text{for } i = N. \end{cases} \tag{3.45}$$

The initial condition corresponding to equation (3.5) is approximated by assuming that at time $t = 0$, probability density is concentrated in the first annulus at $t = 0$,

$$\phi_i(0) = \begin{cases} \frac{1}{2\pi h}, & \text{for } i = 1, \\ 0, & \text{for } 2 \leq i \leq N. \end{cases} \tag{3.46}$$

2.3 Comparison of one and two dimensional diffusion models

Diffusion simulations in 1D and 2D were simulated, each with $N = 1000$ particles, diffusion coefficient $D = 100\text{m}^2\text{s}^{-1}$ and domain size $R = 2000\text{m}$, using `DifferentialEquations.jl` [72] to validate the model. The mean squared displacement for each is shown in Figure 3.12. The mean squared displacements for both discretised simulations are initially linear and consistent with the analytical results calculated in Section 2.1.2 and Section 2.1.5.

The curve then flattens and tends towards a constant value as expected.

There are two differences in the shape of the curves between the 1D and 2D simulations, the gradient of the initial dispersal and the final value. In order to decide whether bats are moving in one or two dimensions, we need to look at the values of either D or R. From a histogram of the distance from roost recorded in the radio tracking survey, Figure 3.4a, we can calculate a domain size R. When we remove the outliers by taking the 90th percentile, we arrive at a domain size of $R = 1.51$km. Whilst the 90th percentile is an arbitrary measure, there is a clear drop in the frequency of recordings around the 90th percentile shown in Figure 3.4a. The frequency drops from 8% for $1.25 < R \leq 1.5$km to just 2.5% for $1.25 < R \leq 1.5$km, less than a third of the number of recordings in the previous bin.

For one-dimensional bounded diffusion, this value corresponds to a long timescale expected MSD of

$$MSD = \frac{R^2}{3} = \frac{1.51^2}{3} = 0.76\text{km}^2. \tag{3.47}$$

For polar bounded diffusion, it corresponds to a long timescale expected MSD of

$$MSD = \frac{R^2}{2} = \frac{1.51^2}{2} = 1.14\text{km}^2. \tag{3.48}$$

From the MSD obtained during the radio tracking study, shown in Figure 3.5, we see that the peak in MSD is 1.15km^2. It is important to note here that the uncertainty calculated from the standard error in Figure 3.5 is large, and we therefore should consider that the uncertainty in the domain size calculated from the MSD is also large. With more data, we would be able to reduce the uncertainty in MSD over time, however, given the data available, 1.15km^2 is our best estimate of the peak MSD. Comparing the peak value of 1.15km^2 to the values obtained for one-dimensional and polar diffusion, we can see that the expected MSD for polar diffusion is very similar. We therefore conclude that, given the data available, bat movement during dispersal from the roost is better approximated by polar diffusion on a bounded domain.

In the following section, we will derive a model to describe the decrease in MSD during

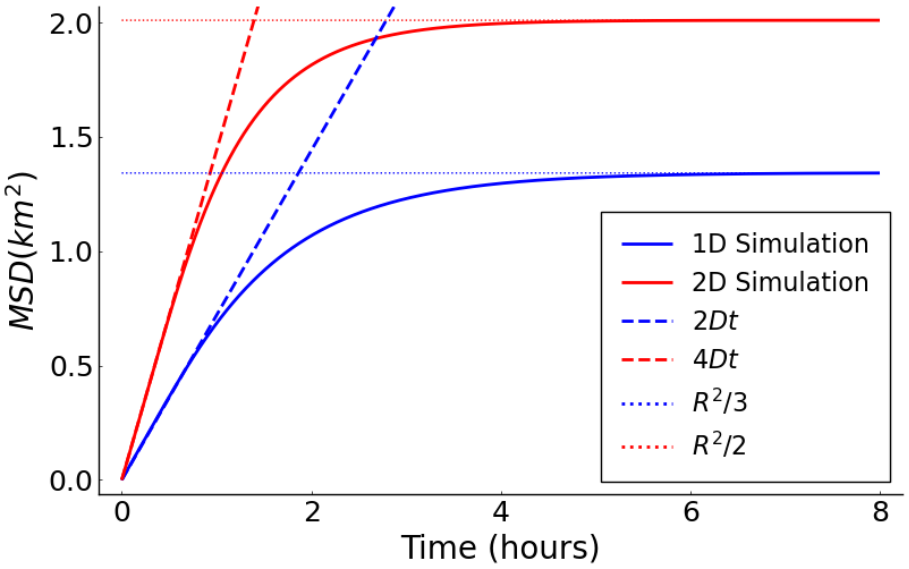


Figure 3.12: A comparison of the mean squared displacement for discretised diffusion simulations in bounded one and two dimensional domains, both with $N = 1000$ boxes, $D = 100\text{ms}^{-2}$ amd $R = 2000\text{m}$, compared to the analytical results from Section 2.1.2 and Section 2.1.5. The 1D simulation solves the system of equations given by equation (3.20) and the 2D simulation solves the system of equations given by equation (3.45).

phase 2 of movement. We will expand on the two-dimensional diffusion model described here, adding a mechanism after phase 1 which acts to push bats back towards the roost during the remainder of the night.

3 Phase 2: return to roost

The diffusion model described in Section 2.1 explains the initial dispersal in phase 1, however it cannot explain the decrease in MSD for phase 2. For the second phase of movement, three different models are considered, a convection-diffusion model, an agent based model and a diffusion model on a shrinking domain. Convection-diffusion models describe movement under the influence of two processes, diffusion and convection. The diffusion component corresponds to dispersal as discussed in Section 2.1, whilst the convection component describes a drift in a particular direction, or towards a specific location. These models are commonly used to model animal movement in response to external factors, for example a drift towards patches of high resources or away from predators. However we will show here that a diffusion model in a shrinking domain provides a more accurate description of bat movement whilst foraging.

3.1 A convection-diffusion model in two dimensions

During phase 2 of movement, the MSD is decreasing as bats return towards a point, the roost location (x_0, y_0) at $r = 0$. We will first consider a convection-diffusion model to describe this drift. The convection-diffusion equation is given by [80]

$$\frac{\partial \phi}{\partial t} = D\nabla^2\phi + \nabla \cdot (\boldsymbol{v}\phi), \tag{3.49}$$

where ∇^2 is the Laplacian operator, ∇ is the gradient, D is the diffusion coefficient and $\boldsymbol{v}$ is the vector flow velocity. We choose a constant radial velocity so that $\boldsymbol{v} = \chi\hat{\boldsymbol{r}}$, where χ is the convection coefficient and defines the rate of convection, and equation (3.49) becomes

$$\frac{\partial \phi}{\partial t} = D\nabla^2\phi + \chi\nabla \cdot \phi, \tag{3.50}$$

We assume that the domain is an angularly symmetric disk in 2D polar coordinates, and that the vector flow is also purely radial, such that there is no angular component. Then,

equation (3.50) becomes

$$\frac{\partial \phi(r,t)}{\partial t} = \frac{D}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \phi(r,t)}{\partial r}\right) + \boldsymbol{\chi}\left(\frac{\partial \phi(r,t)}{\partial r} + \frac{\phi}{r}\right). \tag{3.51}$$

The boundary condition,

$$\frac{\partial \phi(r=R,t)}{\partial r} = 0, \tag{3.52}$$

specifies zero-flux across the boundary such that bats cannot enter or leave the domain. As bats are heading towards the roost at $r = 0$, the convection component of equation (3.51) describes a drift towards $r = 0$. As we assume in this model that bats undergo diffusive movement whilst dispersing from the roost for a time T before their behaviour changes, $\boldsymbol{\chi}$ is time dependent,

$$\boldsymbol{\chi}(t) = \begin{cases} \mathbf{0}, & \text{for } t < T, \\ \boldsymbol{\chi}_0, & \text{for } t \geq T, \end{cases} \tag{3.53}$$

where $\boldsymbol{\chi}_0$ is a negative constant.

3.2 A discretised convection-diffusion model

The convection-diffusion model will be solved using the method of lines, as with the diffusion model in Section 2.1. The domain Ω of length R is discretised into N annuli, each of size $h = R/N$. The probability density in each annulus i is denoted by ϕ_i and evolves over time according to the diffusion and convection processes, as illustrated in Figure 3.13. The diffusion component is unchanged from Section 2.1, however this time there is an additional convection component pushing bats away from the domain boundary and back towards the roost at $r = 0$. The convection process shifts probability density towards the

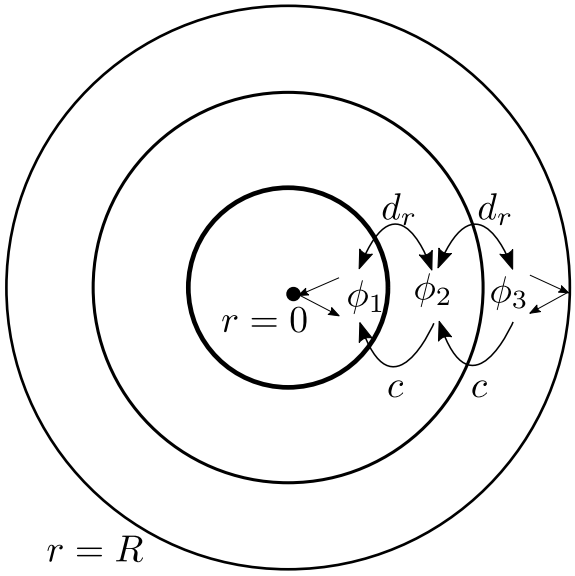


Figure 3.13: A diagram to illustrate the movement of probability density between annuli in the discretised convection-diffusion model. Here we illustrate the domain discretised into 3 sections, but the diagram extends analogously to any number of compartments. As there is no angular component to this movement, the problem is reduced to one radial dimension. Diffusion between annuli is represented by d_r and the drift due to convection is represented by c. The probability density in each annulus i is denoted by ϕ_i.

left, towards annulus $i = 1$, and the discretised equations are, for $t > T$,

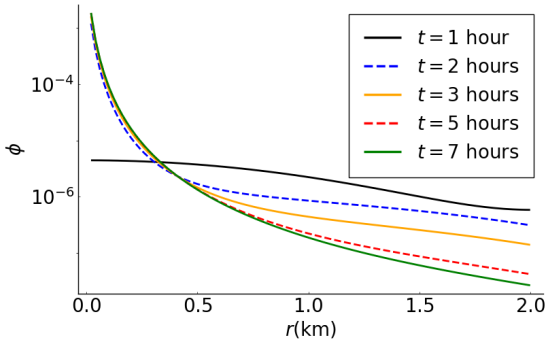
$$bv\frac{d\phi_i}{dt} = \begin{cases} d(\phi_{i+1} - \phi_i) + \frac{d}{2}(\phi_{i+1} - \phi_i) - c\phi_{i+1}, & \text{for } i = 1, \\ d(\phi_{i-1} - 2\phi_i + \phi_{i+1}) + \frac{d}{2i}(\phi_{i+1} - \phi_{i-1}) - \frac{c}{i}(\phi_{i+1} - \phi_i), & \text{for } 2 \leq i \leq N-1, \\ d(\phi_{i-1} - \phi_i) + \frac{d}{2i}(\phi_i - \phi_{i-1}) - \frac{c}{i}\phi_i, & \text{for } i = N. \end{cases} \tag{3.54}$$

where the discretised diffusion coefficient is $d = D/h^2$ as before and $c = \chi_0/h$ is the discretised convection coefficient.

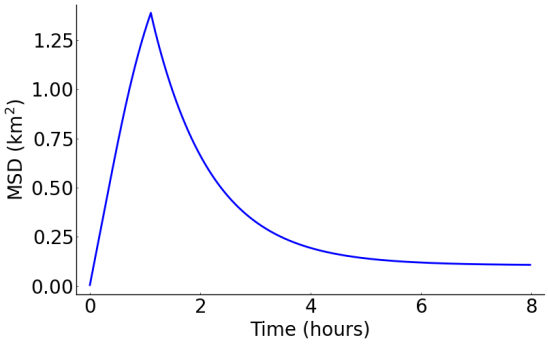
The model was simulated using `DifferentialEquations.jl` [72], using a time dependent convection coefficient as in equation (3.53) with $T = 1.5$ hours. The simulation was run with a domain of length $R = 2000$m, split into $N = 100$ annuli. The diffusion coefficient was $D = 100\text{m}^2\text{s}^{-1}$, and the convection coefficient was $\chi_0 = -15\text{ms}^{-1}$, both chosen for illustrative purposes. The results of this simulation are shown in Figure 3.14. The probability density ϕ is shown in Figure 3.14a. The convection process pushes ϕ

uniformly in the direction of the drift, towards the left side of the domain, whereas the diffusion tends to spread ϕ across the domain. The system eventually reaches a steady state when diffusion and convection are balanced.

The MSD for the same convection-diffusion simulation is shown in Figure 3.14b. The convection-diffusion simulation shows the initial rapid dispersal expected from the diffusion model, however the shape of the curve for phase 2 is clearly inconsistent with the radio tracking data shown in Figure 3.5. The convection-diffusion simulation yields a convex curve whereas the MSD in Figure 3.5 is concave, and the decrease in MSD slows with time as the system tends towards a steady state. The MSD never reaches 0 because the diffusion term acts to spread bats out whilst the convection term is pushing them back towards the roost, and these eventually balance, without the colony returning to the roost.



(a) The probability density ϕ after $t = 1$, 2, 3, 5 and 7 hours.



(b) The mean squared displacement for $0 \leq t \leq 8$ hours.

Figure 3.14: The results of a convection-diffusion simulation with parameters $R = 2000$m, $N = 100$, $D = 100\text{m}^2\text{s}^{-1}$, $\chi_0 = -15\text{ms}^{-1}$ and $T = 1.1$ hours.

In Figure 3.14b, the curve is concave because the MSD decreases slowly at first, when bats are most spread out and furthest from the roost, and the rate of decrease increases with time. One possible model that may provide a concave curve in the MSD is to use a convection-diffusion model in which the convection coefficient is spatially dependent as well as time dependent. In this case, we can engineer the spatial dependence such that when bats are far from the roost, they drift slowly, and the drift speeds up as they get closer to the roost. This corresponds to the temporal patterns shown in Figure 1.1: during the middle of the night, activity levels are low, and close to sunrise, activity levels increase

as bats return to the roost. In this case, bats drift back to the roost at a rate dependent on their distance from the roost,

$$\chi(r,t) = \begin{cases} 0, & \text{for } t < T, \\ r^{\beta}\chi_0, & \text{for } t \geq T, \end{cases} \tag{3.55}$$

where β is a constant. The results of simulations with $-2 \leq \beta \leq 2$ are shown in Figure 3.15. The plots show that for each value of the exponent β, the curves are convex rather than concave as diffusion eventually balances convection and bats stop moving once they reach the roost. When β is negative, increasing χ pushes bats closer to the roost, however the value must be artificially inflated in order to overcome diffusion, and does not solve the convexity problem.

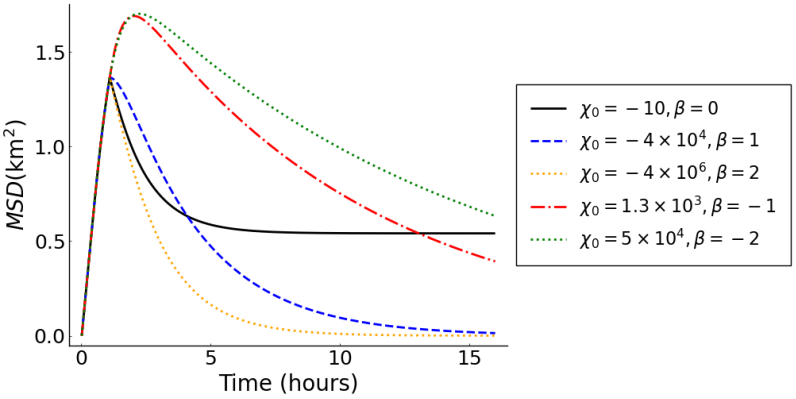


Figure 3.15: The mean squared displacement for spatially dependent convection-diffusion models with convection coefficient χ of the form given in equation (3.55). The simulation parameters are $R = 2000$m, $N = 100$, $D = 100\text{m}^2\text{s}^{-1}$ and $T = 1.5$ hours.

From Figure 3.15 we can see that a convection-diffusion model of this form cannot produce a concave curve because under this model diffusion will always cause bats to spread out, and convection and diffusion processes lead to a steady state without the colony ever reaching the roost. Next we will consider a model that solves this problem by selecting only the bats furthest from the roost to drift back towards the roost. We will first derive a stochastic model describing bat movement and then use it to inform a PDE model.

3.3 A stochastic diffusion model

Next, we will look at using a stochastic process to model bat movement. As the PDE convection-diffusion framework was insufficient to reproduce the observed behaviour, it is important to explore the role of noise. As bat colonies are of a finite size, introducing stochasticity brings the models discussed here closer to real life. We will define stochastic differential equations (SDEs), and discuss the stochastic Euler scheme for approximating solutions. Then, we will look at applying the Euler scheme to solve the stochastic diffusion equation, first in 1D before extending to 2D. First, we define an SDE as an equation of the form

$$dX_t = a(X_t, t)dt + b(X_t, t)dW(t), \tag{3.56}$$

where $X_t \in \mathbb{R}^d$ is the position of a particle at time t and $a(X_t, t)$ and $b^2(X_t, t)$ are variables which represent the infinitesimal mean and variance of X_t, respectively. The terms dX_t and $dW(t)$ represent the infinitesimal changes in X_t and $W(t)$ at time t, respectively. $W(t)$ describes Brownian motion, a random function of time which represents the random noise that 'drives' the stochastic process. Brownian motion has the following three properties:

1. the position $W(t)$ is a continuous function of time t;
2. the increment $W(t_2) - W(t_1)$ is Gaussian with mean $\mu = 0$ and variance $\sigma^2 = t_2 - t_1$, such that $W(t_2) - W(t_1) \sim \mathcal{N}(0, t_2 - t_1)$;
3. $W(t)$ is a Markov process such that at any time t, conditional on knowing $W(t)$ (the present value), the future state of W ($W(s)$ with $s > t$) is independent of the past state of W ($W(s)$ with $s < t$).

Suppose we are given $a(X_t, t)$ and $b(X_t, t)$ and wish to estimate the expectation,

$$f := E\left[V\left(X_{[0,T]}\middle| X_0 = x\right)\right], \tag{3.57}$$

$$:= E_{x,0}\left[V\left(X_{[0,T]}\right)\right], \tag{3.58}$$

for some variable V. If we are able to simulate from the SDE (3.56) then we can use a Monte Carlo estimate. A Monte Carlo simulation involves generating a large number, N, of sample paths for the SDE over a specified time scale, $0 < t < T$. We define the sample path n at time t as $X^{(n)} = \{X^{(n)}(t) : t \geq 0\}$, where each $X^{(n)}$ is independent and satisfies equation (3.56). We write the observable of path n as $V_n = V\left(X^{(n)}\right)$, and then we can write the direct Monte Carlo estimate of f,

$$\hat{f} = \frac{1}{N}\sum_{n=1}^{N} V_n. \tag{3.59}$$

Applying the central limit theorem to the estimate, for large N, the sample mean, $\hat{f}$, is approximately normal, with mean f and variance

$$\mathrm{var}\left(\hat{f}\right) = \sigma_{\hat{f}}^2 = \frac{1}{N}\mathrm{var}\left(V\left(X_{[0,T]}\middle| X_0 = x\right)\right) := \frac{1}{N}\sigma_V^2. \tag{3.60}$$

Formally, we think of our estimate $\hat{f}$ having an error of order $\sqrt{\sigma_v^2/N}$, such that we can write $\hat{f}$ as

$$\hat{f} = f + \mathcal{O}\left(\sqrt{\frac{\sigma_v^2}{N}}\right). \tag{3.61}$$

A stochastic drift-diffusion process is defined as a stochastic Markov process with continuous time, t, and for which sample paths, $X(t)$, are continuous functions of t [81]. As with Brownian motion, the increment $X(t_2) - X(t_1)$ is Gaussian, however unlike Brownian motion the mean and variance can take different values, such that we can approximate the increment $X(t_2) - X(t_1)$, where $t_2 = t_1 + \delta$, as $X(t_2) - X(t_1) \sim \mathcal{N}(\mu(X(t), t), 2D(t_2 - t_1))$, where D is the diffusion coefficient. We can write the stochastic drift-diffusion equation as an SDE of the form given by equation (3.56),

$$dX(t) = \chi\left(X(t), t\right) dt + \sqrt{2D} dW(t). \tag{3.62}$$

The first term defines a deterministic convection component of the motion, determined by the mean drift, $\chi(X_t, t)$, whilst the second term defines the random component of the

motion driven by the Brownian motion described by $dW(t)$.

When there is no drift component, the mean, $\chi(X(t), t)$, is zero, and equation (3.62) is reduced to the stochastic diffusion equation,

$$dX(t) = \sqrt{2D}dW(t). \tag{3.63}$$

The sample paths for most stochastic processes cannot be simulated exactly. Instead, they are approximated by discretising in time. Next, we will define the stochastic Euler scheme for discretising sample paths.

3.3.1 The stochastic Euler scheme

The stochastic Euler scheme is a time stepping method to approximate a sample path $X(t)$ using a discretisation in time with some timestep Δt. We first define discretised time as $t_k = k\Delta t$ and $X_k^{(n)}$ to be an approximation to $X^{(n)}(t_k)$, the position of particle n at time t_k. We need the process to have approximately the correct mean and variance over a timestep of size Δt. Using the Markov Chain property of the stochastic process, we can then write the position of particle n at time t_{k+1} in terms of its position at time t_k, $X_k^{(n)}$,

$$X_{k+1}^{(n)} = X_k^{(n)} + a\left(X_k^{(n)}, t_k\right)\Delta t + b\left(X_k^{(n)}, t_k\right)\Delta W_k, \tag{3.64}$$

where ΔW_k represents the Gaussian increment defined by Brownian motion,

$$\begin{aligned} \Delta W_k &= W_{k+1} - W_k, \\ &\sim \mathcal{N}(0, \Delta t). \end{aligned} \tag{3.65}$$

Since the mean of ΔW_k is zero, the expectation value of the second, stochastic, term in equation (3.64) is also zero, $E\left[b\left(X_k^{(n)}, t_k\right)\Delta W_k\right] = 0$. As the second term on the RHS of equation (3.64) is deterministic, the expectation value of this term is $a\left(X_k^{(n)}, t_k\right)\Delta t$. The

expectation of the change in position between time t_k and t_{k+1} is then given by

$$\begin{aligned} E\left[X_{k+1} - X_k\right] &= E\left[a(X_k, t_k)\Delta t\right] + E\left[b\left(X_k^{(n)}, t_k\right)\Delta W_k\right], \\ &= a(X_k, t_k)\Delta t. \end{aligned} \tag{3.66}$$

We can then use the Euler stepping equation, equation (3.64), to write the location of particle n, $X_K^{(n)}$, at time $t = \Delta tK$ in terms of the initial location $X_0^{(n)}$ and a sum of increments at timesteps k,

$$X_K^{(n)} = X_0^{(n)} + \sum_{k=0}^{K-1}\left[a\left(X_k^{(n)}, t\right)\Delta t + b\left(X_k^{(n)}\right)\Delta W_k\right]. \tag{3.67}$$

For the diffusion equation defined in equation (3.63), the variables, a and b, are $a\left(X_k^{(n)}, t\right) = 0$ and $b\left(X_k^{(n)}\right) = \sqrt{2D}$. Substituting these into the Euler stepping scheme as defined in equation (3.67) approximates the solution, X_K, to equation (3.63) at timestep K,

$$\begin{aligned} X_K &= X_0 + \sum_{k=1}^{K}\sqrt{2D\Delta t}\Delta W_k, \\ &= X_0 + \sqrt{2D\Delta t}\sum_{k=1}^{K}\Delta W_k, \end{aligned} \tag{3.68}$$

where X_0 is the initial condition, $X(t = 0)$. Since the random numbers, ΔW_k, are distributed according to the standard normal distribution, $\Delta W_k \sim \mathcal{N}(0, 1)$, the mean position of particles at any timestep k is X_0. The variance is $2DK\Delta t$, and therefore the solution can be written as

$$X_K = \mathcal{N}(X_0, 2DK\Delta t). \tag{3.69}$$

The probability density of a normal distribution with mean μ and variance σ^2 is given by

$$\mathcal{N}(\mu, \sigma^2) \sim \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right), \tag{3.70}$$

Substituting the mean and variance from equation (3.69) gives the probability density of $X(t)$,

$$\phi(x,t) = \frac{1}{\sqrt{4\pi Dt}} \exp\left(\frac{-(x-X_0)^2}{4Dt}\right). \tag{3.71}$$

3.3.2 Extending to two dimensions

Now that we have defined the SDE in one dimension, we can extend to two dimensions. First, we write the position as a vector,

$$\boldsymbol{X}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}, \tag{3.72}$$

where x and y are variables that define the position in the x and y dimensions, respectively. We also write the Brownian motion operator as a vector,

$$\boldsymbol{W}(t) = \begin{pmatrix} W_x(t) \\ W_y(t) \end{pmatrix}, \tag{3.73}$$

where W_x and W_y are independent standard Brownian motions corresponding to the first and second dimensions, respectively. We can now write a general SDE in terms of these vector quantities,

$$d\boldsymbol{X}(t) = \boldsymbol{a}(\boldsymbol{X}_t, t)dt + \boldsymbol{b}(\boldsymbol{X}_t, t)d\boldsymbol{W}(t). \tag{3.74}$$

The functions $\boldsymbol{a}$ and $\boldsymbol{b}$ are now defined in two dimensions, $\boldsymbol{a}(\boldsymbol{X}_t, t) \in \mathbb{R}^2$ and $\boldsymbol{b}(\boldsymbol{X}_t, t) \in \mathbb{R}^{2\times 2}$, where $\boldsymbol{b}$ is a positive semidefinite matrix. The special case,

$$\boldsymbol{a}(\boldsymbol{X}_t, t) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \boldsymbol{b}(\boldsymbol{X}_t, t) = \begin{pmatrix} \sqrt{2D} & 0 \\ 0 & \sqrt{2D} \end{pmatrix}, \tag{3.75}$$

gives diffusion in 2D and leads to the SDE

$$d\boldsymbol{X}(t) = \sqrt{2D}d\boldsymbol{W}(t). \tag{3.76}$$

The solution to equation (3.76) is

$$\boldsymbol{X}(t) \sim \boldsymbol{\mathcal{N}}(\boldsymbol{X}(0), 2Dt). \tag{3.77}$$

Substituting the mean and variance from equation (3.69) gives the probability density of $\boldsymbol{X}(t)$,

$$\phi(\boldsymbol{X}, t) = \frac{1}{\sqrt{4\pi Dt}} \exp\left(\frac{-(\boldsymbol{X} - \boldsymbol{X}_0)^2}{4Dt}\right). \tag{3.78}$$

An estimate of the MSD based on a finite number of stochastic diffusion simulations is shown, along with the theoretical expected MSD from equation (3.31), $\langle r^2 \rangle \approx 4Dt$, in Figure 3.16. The MSD calculated from the stochastic simulations hovers around the theoretical mean of $4Dt$, and the error comes from the distribution about the mean.

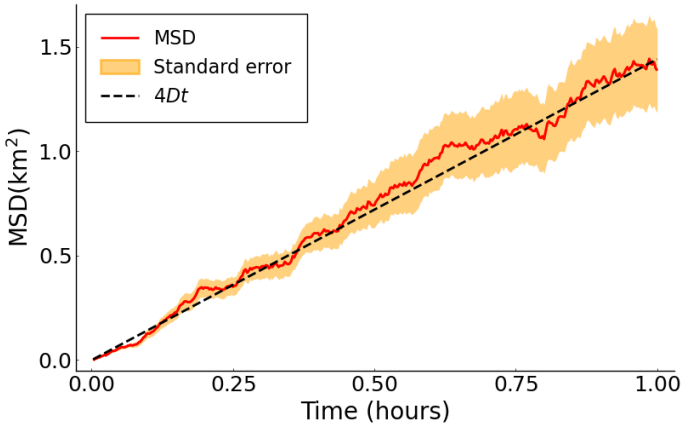


Figure 3.16: The mean squared displacement for stochastic and PDE diffusion models in two dimensions with diffusion coefficient $D = 100\text{m}^2\text{s}^{-1}$. The stochastic simulation uses $N = 100$ bats and a timestep of $\Delta t = 50$ seconds. The standard error is shown as a yellow ribbon.

3.4 A stochastic model for phase 2, bats returning to the roost

As discussed in Section 3.1, the convection-diffusion model does not provide a good model for phase 2 because convection and diffusion balance, and bats never return to the roost. In this section we will derive an agent-based model that solves this problem by selecting only the bats furthest from the roost to drift back. All bats will undergo diffusion

throughout the night, and during phase 2 the bat furthest from the roost will also undergo movement towards the roost. Later, we will use this agent-based model to inform a PDE model for phase 2 of movement.

3.4.1 A stochastic convection-diffusion model

Next, we will define the stochastic convection-diffusion model, returning to the drift-diffusion equation given by equation (3.62),

$$dX(t) = \chi\left(X(t), t\right) dt + \sqrt{2D} dW(t). \tag{3.79}$$

with $\chi\left(X(t), t\right) \neq 0$. As in Section 3.3.2, converting to two dimensions is simple. We define $\boldsymbol{X}(t)$ and $\boldsymbol{W}(t)$ as vectors, as in equations (3.72) and (3.73), and similarly define the drift coeffecient, $\chi\left(X(t), t\right)$ as a vector,

$$\boldsymbol{\chi}\left(X(t), t\right) = \begin{pmatrix} \chi_1\left(X(t), t\right) \\ \chi_2\left(X(t), t\right) \end{pmatrix}. \tag{3.80}$$

The vector drift-diffusion equation can then be written as

$$d\boldsymbol{X}(t) = \boldsymbol{\chi}\left(\boldsymbol{X}(t), t\right) dt + \sqrt{2D} d\boldsymbol{W}(t). \tag{3.81}$$

Using the Euler stepping equation, equation (3.64), the discretised solution at timestep K is then

$$\boldsymbol{X}_K^{(n)} = \boldsymbol{X}_0^{(n)} + \sum_{k=0}^{K-1} \left[\boldsymbol{\chi}\left(\boldsymbol{X}_k^{(n)}, t\right) \Delta t + \sqrt{2D} \Delta \boldsymbol{W}_k\right]. \tag{3.82}$$

In phase 2 of movement bats are returning to the roost, and so we define the drift, $\boldsymbol{\chi}(\boldsymbol{X}(t), t)$, in radial lines towards the roost at $\boldsymbol{X}_0 = (0, 0)$. As in Section 3.1, the expression for the coefficient $\boldsymbol{\chi}\left(\boldsymbol{X}(t), t\right)$ is time-dependent, and also dependent on the distance from the

roost,

$$\boldsymbol{\chi}\left(\boldsymbol{X}(t),t\right)=\begin{cases}0, & \text{for } t<T,\\ -\chi_0\|\boldsymbol{X}(t)\|_2^{\beta}\boldsymbol{X}(t), & \text{for } t\geq T,\end{cases} \tag{3.83}$$

where χ_0 is a positive constant.

3.4.2 A 'leapfrog' model

We now use this convection model to develop an agent-based model that selects only the bats furthest from the roost at each timestep to drift back towards the roost in order to solve the problem of the closest bats returning to the roost first. The convection coefficient is given by equation (3.83) for only the furthest bat at each timestep and is set to zero for all other bats. At each timestep, the convection coefficient is calculated as follows. First, find the index of bat at the furthest distance from roost,

$$j_{max}=\underset{j\in n}{\operatorname{argmax}}\left(\sqrt{x_j^2+y_j^2}\right). \tag{3.84}$$

The convection coefficient is 0 for $t<T$, during phase 1, and for $t>T$ it is non-zero only for the furthest bat,

$$\boldsymbol{\chi}^{(j)}\left(\boldsymbol{X}_k^{(j)},t\right)=\begin{cases}0, & \text{for } t<T \text{ or } j\neq j_{max},\\ \begin{pmatrix}\chi_1(\boldsymbol{X}^{(j)}(t),t)\\ \chi_2(\boldsymbol{X}^{(j)}(t),t)\end{pmatrix} & \text{for } t\geq T \text{ and } j=j_{max}.\end{cases} \tag{3.85}$$

The discretisation for bat j at timestep K is then written as

$$\boldsymbol{X}_K^{(j)}=\boldsymbol{X}_0^{(j)}+\sum_{k=1}^{K}\left[\boldsymbol{\chi}\left(\boldsymbol{X}_k^{(j)},K\Delta t\right)\Delta t+\sqrt{2D}\Delta\boldsymbol{W}_k\right]. \tag{3.86}$$

3.4.3 Validating the leapfrog model with radiotracking data

Next, we will fit the parameters for the stochastic leapfrog model to the data collected in Section 1. First, we look at the diffusion coefficient, D. As there is another mechanism

added in phase 2 in addition to diffusion, the MSD is not linear in phase 2. Therefore, we cannot calculate the diffusion coefficient, D, for phase 2 from the gradient as for phase 1, so we assume that it is time-dependent,

$$D(t) = \begin{cases} D_1, & \text{for } t < T, \\ D_2, & \text{for } t \geq T, \end{cases} \tag{3.87}$$

where D_1 and D_2 are the diffusion coefficients for phase 1 and phase 2 respectively, and are both positive constants.

Next, we determine the exponent, β. The estimated MSD for simulations with $N = 100$ bats with various β are shown in Figure 3.17. The curves for $\beta = 0$ and $\beta = -1$ are convex, however for $\beta = -2$ the curve is concave initially, and a similar shape to the radio tracking data.

It is important to note that the behaviour is dependent on the number of bats, as well as the exponent and convection coefficient. As only one bat, the furthest from the roost, drifts back towards the roost at each timestep, changing the value of N changes the proportion of bats which drift. The estimated MSD for simulations with various N and χ are shown in Figure 3.18. The behaviour clearly changes with varying N, and simulations with equal χ/N produce similar behaviour. A simple way to negate this dependence would be to instead select a certain percentage of bats in the colony to drift back at each step, however here we will consider only simulations with a single bat drifting. We choose $N = 53$, since this corresponds to the number of bat-nights used in the analysis of the radio tracking survey in Section 1. Whilst the size of the colony is likely to be larger than $N = 53$, the colony size was not counted at any point, and we therefore have no indication of a more suitable value of N to choose.

Next, we fit the remaining parameters to the radio tracking data in Figure 3.5: diffusion coefficients for the two phases, D_1 and D_2, and the convection coefficient, χ_0. The diffusion coefficient for phase 1, D_1, was estimated using `LsqFit.jl`, a package for least squares fitting in Julia [82]. A straight line was fitted to the initial linear segment, for time

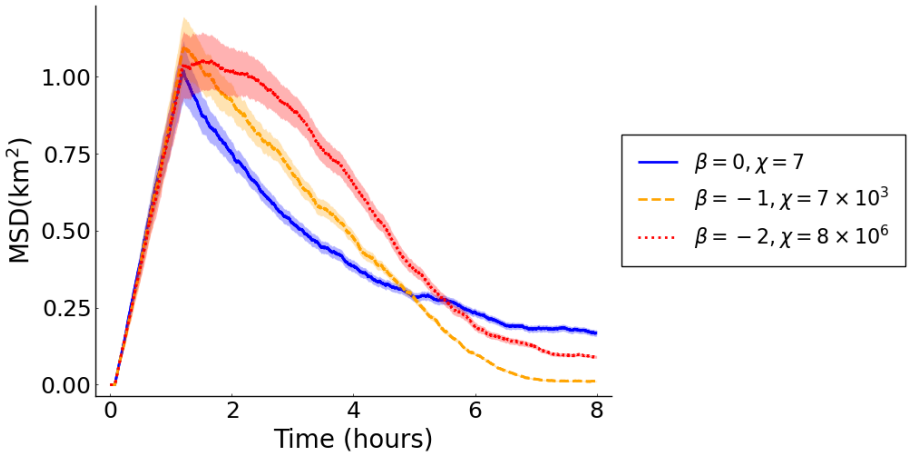


Figure 3.17: The mean-squared displacement for various exponents β. The parameters for these simulations were $D_1 = 65\text{m}^2\text{s}^{-1}$, $D_2 = 46.5\text{m}^2\text{s}^{-1}$, $N = 100$ and $T = 4350$ seconds. As these simulations are stochastic, the MSD for each parameter was taken as the mean over 10 simulations to reduce noise. The standard errors are shown as ribbons.

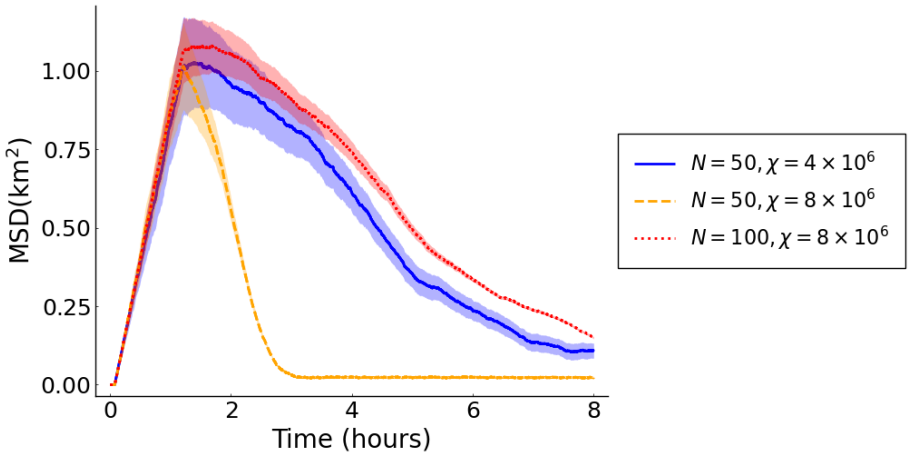


Figure 3.18: The mean-squared displacement for various numbers of bats, N, and convection coefficients, χ_0. The parameters for these simulations were $\chi = 8 \times 10^8$, $D_1 = 65\text{m}^2\text{s}^{-1}$, $D_2 = 46.5\text{m}^2\text{s}^{-1}$ and $T = 4350$ seconds. As these simulations are stochastic, the MSD for each parameter was taken as the mean over 10 simulations to reduce noise. The standard errors are shown as ribbons.

$0 \leq t < 1.5$ hours, and the gradient was used along with equation (3.29) to determine the diffusion coefficient as $D_1 = 63.4\text{m}^2\text{s}^{-1}$. For phase 2, the exponent $\beta = -2$ was used, and the parameters D_2 and χ_0 were fitted using the ABC algorithm set out in algorithm 1. The number of bats, N, was not fitted, because as noted before, simulations with equal χ_0/N produce similar behaviour. Adding an additional parameter to the fitting greatly

increases the computational cost, and in this case is unlikely to add significant benefits.

ABC was chosen here because the behaviour is stochastic, and therefore there is noise in the MSD calculated at each timestep. As a result, we cannot use a simple parameter sweep. Of course, it is possible to use a parameter sweep, averaging over multiple simulations at each set of parameter values, however this is significantly more computationally expensive than using ABC.

In this case, observations, $\boldsymbol{Y}$, are the locations recorded during the radio tracking survey, and artificial datasets, $\boldsymbol{X}$, are samples from stochastic simulations with parameters $\boldsymbol{\theta} = (D_2, \chi_0)$. The summary statistics, $\boldsymbol{S}(\boldsymbol{Y})$ and $\boldsymbol{S}'(\boldsymbol{X})$, are the estimated MSD at intervals of $dt = 200$ seconds for the observations and simulations, respectively. We use the Euclidean distance between summary statistics as the distance metric,

$$\rho(\boldsymbol{X}, \boldsymbol{Y}) = \|\boldsymbol{S}(\boldsymbol{X}) - \boldsymbol{S}'(\boldsymbol{Y})\|_2. \tag{3.88}$$

The ABC algorithm was run for a sample size of $n = 50000$, and ϵ was chosen such that the best 2% of parameter values were added to the posterior. The posterior distribution is shown in Figure 3.20. The posterior distribution shows that the parameters D_2 and χ are clearly correlated. The estimate for each parameter is calculated by taking the mean of the posterior, $D_2 = 31.5\text{m}^2\text{s}^{-1}$ and $\chi_0 = 5.38 \times 10^6\text{ms}^{-1}$. Note that, whilst this value of χ_0 may seem unrealistically large, the distance travelled at each timestep is dependent also on the distance from the roost, as given by equation (3.83). The drift speed, $\chi_0\|\boldsymbol{X}(t)\|_2^\beta$, is shown in Figure 3.19 for varying distances from the roost, $0.4 < \|\boldsymbol{X}(t)\|_2 \leq 2\text{km}$. We will show later, in Figure 3.22, that the maxmimum distance from the roost over time for this simulation varies between 0.4km and 2km. The drift speed of 34ms^{-1} found at 0.4km from the roost is therefore the maximum for this simulation. The average flight speed recorded for Greater Horseshoe bats during foraging is around 9ms^{-1}, although individuals generally fly faster when commuting between the roost and foraging spots than during foraging [18]. The drift speed calculated here is within the same order of magnitude, and we conclude that it is therefore biologically feasible.

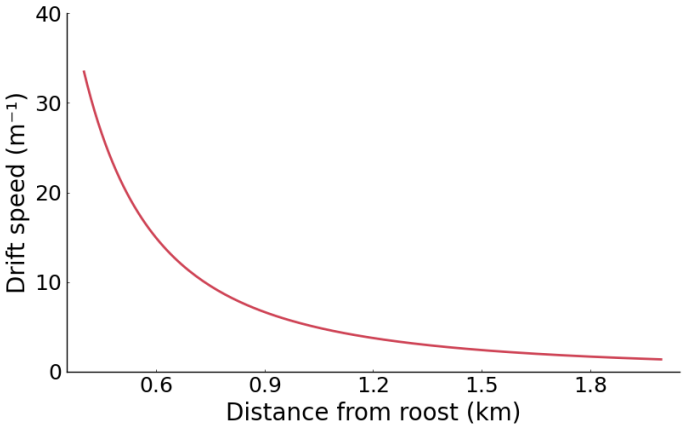


Figure 3.19: The drift speed, given by $\chi_0 \|\boldsymbol{X}(t)\|_2^{\beta}$, for varying distances from the roost, $0.4 < \|\boldsymbol{X}(t)\|_2 \leq 2$km.

The results of a simulation with the parameters calculated with ABC is shown in Figure 3.21 to compare with the radio tracking data. The curve provides a good fit to the radio tracking data, suggesting that this model provides a good description of bat behaviour when foraging. Ideally, it would be beneficial to test the significance of this fit by applying the model and parameters calculated here to radiotracking data from a different roost. However, collecting such data is extremely labour intensive and therefore beyond the scope of this thesis.

The maximum distance from the roost over time for the same simulation is plotted in Figure 3.22, showing that during phase 2 of movement the maximum distance from the roost is decreasing. As diffusion is unbounded for this simulation, some bats travelled much further from the roost than the rest of the colony during phase 1, and the initial steep decrease in phase 2 is due to these outliers drifting back quickly.

We have successfully derived a stochastic 'leapfrog' model to describe the movement during phase 2. Next we will use the leapfrog model to inform a partial differential equation model for phase 2, using diffusion on a shrinking domain, in which the maximum possible distance from the roost decreases over time. This is analogous to the leapfrog model as it also assumes that the bats furthest from the roost start to move back first. However, to convert to a PDE model, instead of moving only the furthest bat towards the roost at each time step, we will shrink the domain, causing all bats at the edge of the domain to

move back.

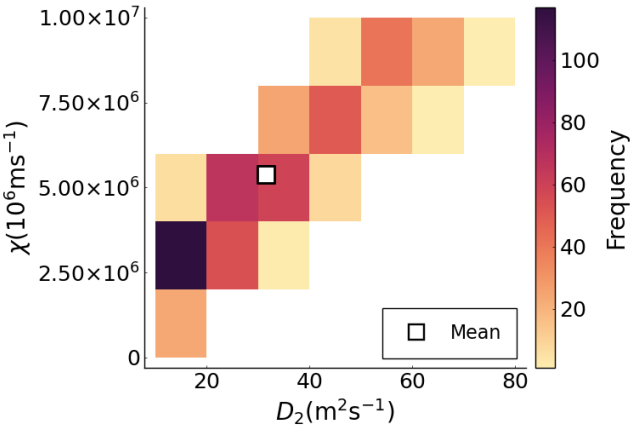


Figure 3.20: A 2D histogram of the posterior joint distribution for χ and D_2.

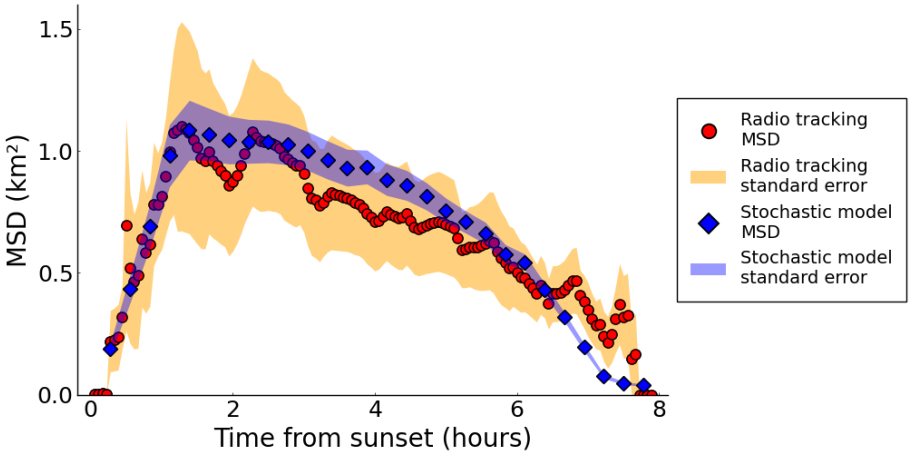


Figure 3.21: MSD for the stochastic model compared to radio tracking data. For the stochastic model, $N = 53$ bats were simulated with the parameters $D_1 = 63.4\text{m}^2\text{s}^{-1}$, $D_2 = 31.5\text{m}^2\text{s}^{-1}$ and $\chi = 5.38 \times 10^6\text{ms}^{-1}$. As the model is stochastic, the MSD shown is the mean value at each timestep over 1000 simulations to reduce noise.

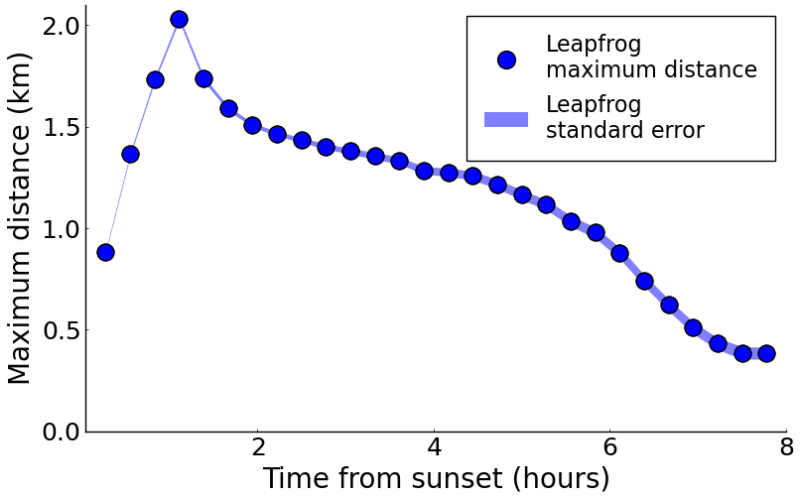


Figure 3.22: The maximum distance from the roost over time for the simulation shown in Figure 3.21.

3.5 Diffusion on a shrinking domain

Next we will consider diffusion on a shrinking domain, assuming that bats disperse at the beginning of the night, but tend to move back towards the roost during phase 2 of movement, narrowing the area in which they forage as the night goes on.

We will see that the shrinking of the domain causes a change in the kinetics of diffusion, and thus the problem is not a simple change in the scale of the domain. First, we recall our assumption that the domain is angularly symmetric, such that the system is completely homogeneous in the angular coordinate. Then, position and movement are described fully by one, radial, coordinate, r.

Let $\boldsymbol{\Omega}_t$ define the shrinking domain and $\phi(r,t)$ define the probability density at position r and time t. The conservation equation for probability density in the domain equates the rate of change of probability density with the flux in and out of a volume,

$$\frac{d}{dt}\int_{\boldsymbol{\Omega}_t}\phi(\boldsymbol{r},t)d\omega = \int_{\boldsymbol{\Omega}_t} -\nabla\cdot\boldsymbol{j}d\omega, \tag{3.89}$$

where $\boldsymbol{j}$ is the flux of material into and out of an elemental volume, ω, and $d\omega = rdrd\theta$ [83]. We apply the Reynolds' transport theorem [84] to the left-hand side,

$$\frac{d}{dt}\int_{\boldsymbol{\Omega}_t}\phi(r,t)d\omega = \int_{\boldsymbol{\Omega}_t}\frac{\partial\phi}{\partial t} + \nabla\cdot(\boldsymbol{v}\phi)d\omega, \tag{3.90}$$

where $\boldsymbol{v}(\boldsymbol{r},t)$ is the velocity field of the flow generated by the shrinkage of the domain,

$$\boldsymbol{v}(\boldsymbol{r},t) = \frac{d\boldsymbol{r}}{dt}. \tag{3.91}$$

Finally, we apply Fick's Law of Diffusion [85],

$$\boldsymbol{j} = D\nabla\phi, \tag{3.92}$$

to the right-hand side of equation (3.89), and we obtain the equation for diffusion on a

shrinking domain,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\boldsymbol{v}\phi) = D\nabla^2 \phi. \tag{3.93}$$

We note that the shrinking of the domain adds an additional, advective term to the original diffusion equation, $\nabla \cdot (\boldsymbol{v}\phi)$. This term can be written as $\nabla \cdot (\boldsymbol{v}\phi) = \phi \nabla \cdot \boldsymbol{v} + \boldsymbol{v} \cdot \nabla \phi$. The term $\phi \nabla \cdot \boldsymbol{v}$ is interpreted as the transport of material around the domain, whilst $\boldsymbol{v} \cdot \nabla \phi$ is the dilution which arises due to volume changes as the domain shrinks [51].

Next we will discuss the form of shrinkage. We wish for our PDE model to approximate the behaviour of the stochastic leapfrog model from Section 3.4.2. The leapfrog model selects only the bats furthest bats from the roost to drift back towards the roost, whilst all other bats continue diffusing normally. We consider a domain shrinking peripherally, in which shrinking is restricted to an annulus of width δ at the outer edge of the domain. Let the radius of the domain at time t be $R(t)$. Then, we say that shrinkage is zero everywhere except in a region of width δ at the edge of the domain,

$$\frac{\partial \boldsymbol{v}}{\partial r} = \begin{cases} 0, & 0 \leq r < R(t) - \delta, \\ S_{edge}(t), & R(t) - \delta \leq r \leq R(t), \end{cases} \tag{3.94}$$

where $S_{edge}(t) < 0$ defines the strain caused by shrinking in the element at the edge of the boundary. A diagram illustrating equation (3.94) is shown in Figure 3.23. Integrating, we

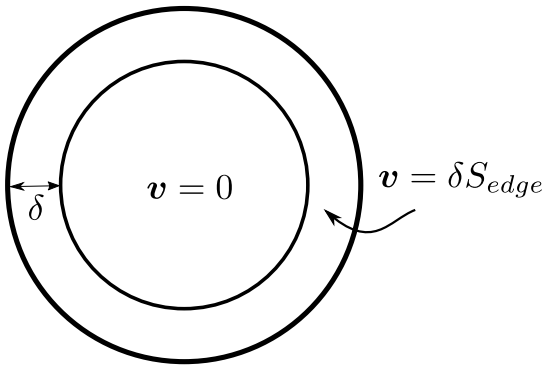


Figure 3.23: A diagram showing a domain shrinking peripherally, such that shrinking is restricted to an annulus of width δ at the edge of the domain. The velocity field of the flow and strain generated by the shrinkage of the domain are denoted $\boldsymbol{v}$ and S_{edge}, respectively.

obtain the flow,

$$\boldsymbol{v} = \begin{cases} 0, & 0 \leq r < R(t) - \delta, \\ \delta S_{edge}(t), & R(t) - \delta \leq r \leq R(t). \end{cases} \tag{3.95}$$

Returning to equation (3.93), we see that, since the flow as defined in equation (3.95) is spatially dependent, the advective term, $\nabla \cdot (\boldsymbol{v}\phi)$, is also now spatially dependent. This spatial dependence makes equation (3.93) difficult to solve analytically, and so we will instead choose to simulate and make some approximations. In the next section, we will use the discretised diffusion simulations discussed in Section 2.2 to develop a diffusion simulation on a discretised shrinking domain.

3.5.1 A discretised diffusion simulation on a shrinking domain

The diffusion process on a shrinking domain can be simulated using a modification of the discrete diffusion model described in Section 2.2. We will use a circular domain of time-dependent radius $R(t)$ discretised into $N(t)$ annuli of width h.

There are two separate processes working in this simulation, illustrated in Figure 3.24, diffusion and shrinking of the domain. Diffusion moves probability density between annuli. Domain shrinking reduces the size of the domain by removing the annuli at the edge of the domain and adding the probability density to the furthest remaining annulus. These two processes do not necessarily happen on the same timescale; the shrinking rate may be faster, or slower than the diffusion rate. We denote the timestep used for the discretisation of the diffusion PDE as dt. The time at timestep i is written as $t_i = idt$ and the end time of the simulation is written as T. The radius of the domain at timestep i is denoted R_i and the number of annuli at timestep i is given by $N_i = R_i/h$. The radius of the outer edge of annulus j is written r_j.

Next, we consider conservation of probability density. The total probability in the domain must always remain constant,

$$2\pi \int_0^R \phi r dr = 1. \tag{3.96}$$

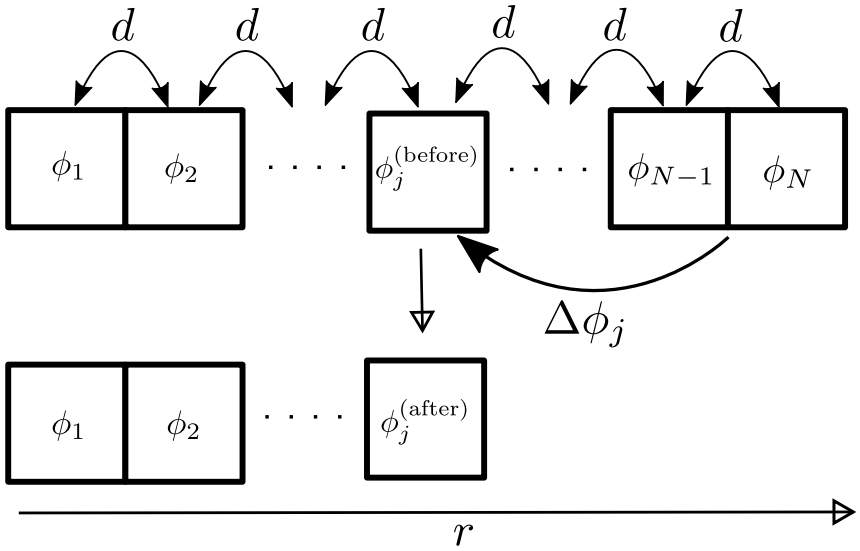


Figure 3.24: A diagram illustrating a discretised diffusion process on a shrinking domain. Box j represents annulus j, the diffusion process is denoted by d and the probability in annulus j is denoted by ϕ_j. Here we show the process for one shrinking event, in which the annuli $j+1$ to N are removed and their probability densities are added to annulus j. We write the probability density in annulus j before and after the shrinking event as $\phi_j^{(\text{before})}$ and $\phi_j^{(\text{after})}$, respectively, where $\phi_j^{(\text{after})}$ is given by equation (3.104). We define the change in probability density in annulus j as $\Delta\phi_j = \phi_j^{(\text{after})} - \phi_j^{(\text{before})}$.

In other words, all bats are located within the domain at all times, as the boundaries are reflective and bats cannot move into, or out of, the domain. Since we are working on a discretised set of annuli, and the area of annulus j is given by $\pi(r_j^2 - r_{j-1}^2)$, the continuous form of the conservation equation (3.96) is analogous to the discretised form,

$$\sum_{j=1}^{N} \phi_j \pi (r_j^2 - r_{j-1}^2) \approx \frac{1}{2\pi}. \tag{3.97}$$

where $r_j = jh$, and we drop the indexing for N_i. Each time an annulus is removed, we must ensure that the conservation equation is always satisfied. Specifically, we consider the continuity equation for the two annuli at the outer edge of the domain, $N-1$ and N. We define a shrinking event as a single event in which any number of annuli at the outer edge of the domain are removed, and their probability density moved to the last remaining annulus. If we consider a shrinking event in which only the outer annulus is removed, the

event only affects the probability density in the last two annuli, and the probability density in all other boxes remains constant.

We define the probability density in annulus j before a shrinking event as $\phi_j^{(\text{before})}$ and after the shrinking event as $\phi_j^{(\text{after})}$. Then, we note that, since annulus N is removed during shrinking, $\phi_N^{(\text{after})} = 0$, and we write the continuity equation for the two outer annuli as

$$\phi_N^{(\text{before})}\left(r_N^2 - r_{N-1}^2\right) + \phi_{N-1}^{(\text{before})}\left(r_{N-1}^2 - r_{N-2}^2\right) = \phi_{N-1}^{(\text{after})}\left(r_{N-1}^2 - r_{N-2}^2\right). \tag{3.98}$$

Rearranging for $\phi_{N-1}^{(\text{after})}$, we get

$$\phi_{N-1}^{(\text{after})} = \phi_{N-1}^{(\text{before})} + \phi_N^{(\text{before})}\left(\frac{r_N^2 - r_{N-1}^2}{r_{N-1}^2 - r_{N-2}^2}\right). \tag{3.99}$$

Expanding the spatial term,

$$\begin{aligned}\frac{r_N^2 - r_{N-1}^2}{r_{N-1}^2 - r_{N-1}^2} &= \frac{r_N + r_{N-1}}{r_{N-1} + r_{N-2}} \underbrace{\frac{r_N - r_{N-1}}{r_{N-1} - r_{N-2}}}_{=\frac{h}{h}=1}, \\ &= \frac{r_N + r_{N-1}}{r_{N-1} + r_{N-2}},\end{aligned} \tag{3.100}$$

we arrive at

$$\begin{aligned}\phi_{N-1}^{(\text{after})} &= \phi_{N-1}^{(\text{before})} + \phi_N^{(\text{before})}\left(\frac{r_N + r_{N-1}}{r_{N-1} + r_{N-2}}\right), \\ &= \phi_{N-1}^{(\text{before})} + \phi_N^{(\text{before})} c_N,\end{aligned} \tag{3.101}$$

where we have defined c_N as

$$c_N = \left(\frac{r_N + r_{N+1}}{r_{N-1} + r_{N-2}}\right). \tag{3.102}$$

We note that for large r_N (and small h), $c_N \approx 1$. As noted before, the timescale for shrinking may not be the same as the timescale for diffusion, dt. When the domain shrinks and annuli are removed at a rate faster than dt, more than one annulus will be removed in

a single timestep. If, instead of just removing one annulus in a given timestep, we remove the outer 2 annuli, $N-1$ and N, and add their probabilities to annulus $N-2$, we can write down an expression for $\phi_{N-2}^{(\text{after})}$,

$$\phi_{N-2}^{(\text{after})} = \phi_{N-2}^{(\text{before})} + \phi_{N-1}^{(\text{before})} c_{N-1} + \phi_{N}^{(\text{before})} c_N c_{N-1}. \tag{3.103}$$

Generalising this, we arrive at an expression for $\phi_j^{(\text{after})}$ for any $j < N$ when we remove annuli $j+1$ to N,

$$\phi_j^{(\text{after})} = \phi_j^{(\text{before})} + \sum_{k=j+1}^{N} \phi_k^{(\text{before})} \prod_{l=j+1}^{k} c_l. \tag{3.104}$$

Now that we have an expression for the change in probability density in annulus j, we write an algorithm for the discretised diffusion model on a shrinking domain:

Define $h = 1\text{m}$, $dt = 10$ seconds, and $T = 8$ hours

Set N_i for all i

Set $i \leftarrow 0$

while $t_i < T$ **do**

Solve the system of ODEs describing the discretisation for diffusion on a stationary domain, without convection given by equation (3.45), for the current N_i annuli over timestep $t_i \leq t < t_{i+1}$ using a ODE solver (here we use an RK45 solver implemented in Julia [72]).

if $N_{i+1} < N_i$ **then** Remove annuli N_i to N_{i+1} and add the probability from the removed annuli to the last remaining annulus, calculating ϕ using equation (3.104).

end if

Set $i \leftarrow i + 1$

end while

Next, we will discuss the form of the shrinking rate. Since the observed MSD calculated for phase 2 in Section 1 can be approximated by a negative parabola, we wish to choose a

shrinking rate to give an MSD of the form

$$\left\langle r^2 \right\rangle \propto \beta - \alpha t^2, \tag{3.105}$$

where α and β are parameters of the model. We will assume that during phase 1, the domain is of a fixed size. The domain begins to shrink at time $t = t_s$, where t_s is a parameter of the model, and shrinks at a rate given by the parameter α.

Next, we will infer a form for the time-dependent radius, $R(t)$, of the shrinking domain. Firstly, we consider the maximum distance from the roost in the leapfrog simulation, Figure 3.22. The maximum distance clearly decreases over time whilst the bats return to the roost, however, the use of a stochastic simulation means that any outliers contribute to the maximum distance. In particular, we see a sharp decrease in the maximum distance when the leapfrog mechanism begins at the start of phase 2. Additionally, we note that the maximum distance does not reach 0km at the end of the night because diffusion continues to cause bats to spread out. As a result of these outliers, fitting a radius to the results of the leapfrog model is difficult. Instead, we consider a very simple approximation. Later, in Section 4, we will compare the maximum distance for the leapfrog model and the radius of the shrinking domain derived here, and we will see that they are almost identical for the middle part of the night, $2 < t < 6$ hours.

First, we make the assumption that the diffusion rate is larger than the rate at which the domain changes size. In this case, the solution should approximate the homogeneous steady state on a bounded domain. Note that this assumption is not necessarily correct: we simply use it to infer the form of the radius. Later, we will show that the radius derived using this assumption produces the desired behaviour, whether the diffusion rate is larger than the shrink rate or not.

As shown in Section 2.1, the solution to the diffusion equation on a bounded domain tends to uniformity. On a circular domain, as $t \to \infty$,

$$\phi \to \frac{1}{\pi R^2}. \tag{3.106}$$

The expected MSD for a domain of radius R can be calculated using its probability distribution,

$$\begin{aligned}\left\langle r^2 \right\rangle &= \int_\Omega r^2 \phi(r,t) d\Omega, \\ &= \int_0^{2\pi} \int_0^R \frac{r^3}{\pi R^2} dr d\theta, \\ &= \frac{R^2}{2},\end{aligned} \tag{3.107}$$

and we note that, if probability is approximately constant, then the MSD is approximately proportional to the radius squared. From this, we can choose the radius so that the expected MSD matches the observed MSD from the radiotracking data. We choose the form of the domain radius, $R(t)$, to give a concave MSD. We set the radius as

$$R(t) = \sqrt{2(R_0^2 - \alpha(t - t_s)^2)}, \tag{3.108}$$

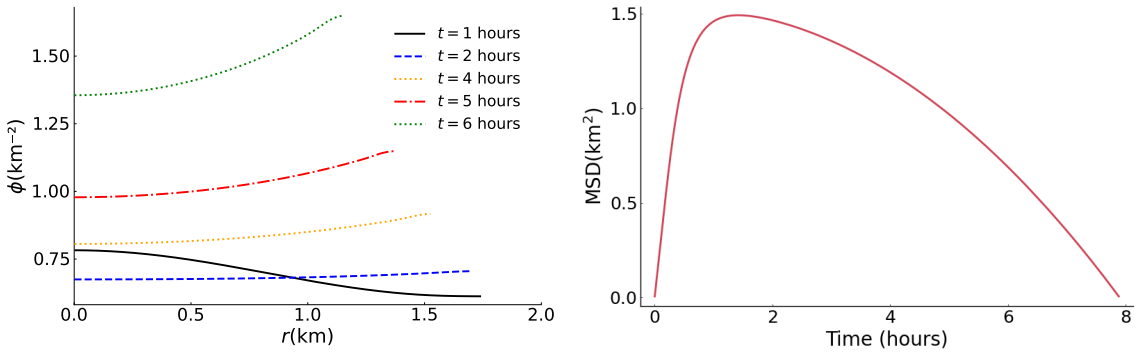
which gives an expected MSD of

$$\left\langle r^2 \right\rangle (t) = {R_0}^2 - \alpha(t - t_s)^2. \tag{3.109}$$

The result of a simulation of diffusion on a domain shrinking with time-dependent radius $R(t)$ given by equation (3.108) is shown in Figure 3.25. The probability density, shown in Figure 3.25a, spreads across the domain due to the diffusion process, however the shrinking causes the probability to increase at the right edge of the domain. We note that the probability density is not constant, and in fact varies by around 18% across the domain at $t = 6$ hours. However, this variation occurs over a large distance: at $t = 6$ hours, the radius of the domain is $R = 1.2$km, and therefore the gradient remains small. Additionally, Figure 3.25b illustrates that the mean squared displacement is a concave curve, consistent with the shape of the radiotracking data.

For comparison, a similar simulation was run, with a much smaller diffusion coefficient of $D = 63.4\text{m}^2\text{s}^{-1}$, (we will see in Section 4 that this is the value calculated for phase 1

of movement), and the results are shown in Figure 3.26. In this case, we note that the probability density, shown in Figure 3.26a is not constant, and by $t = 6$ hours, ϕ varies by up to 100%. Even so, the MSD, shown in Figure 3.26b, is a concave curve, qualitatively consistent with the shape of the radiotracking data. As a result, we conclude that, even when our assumption that the diffusion rate is larger than the shrinkage rate does not hold true, and the probability density is not uniform, the simulation produces the desired behaviour. We can therefore choose the parameters, R_0 and t_s to shrink the domain such that the expected MSD matches the MSD calculated from the radiotracking data.

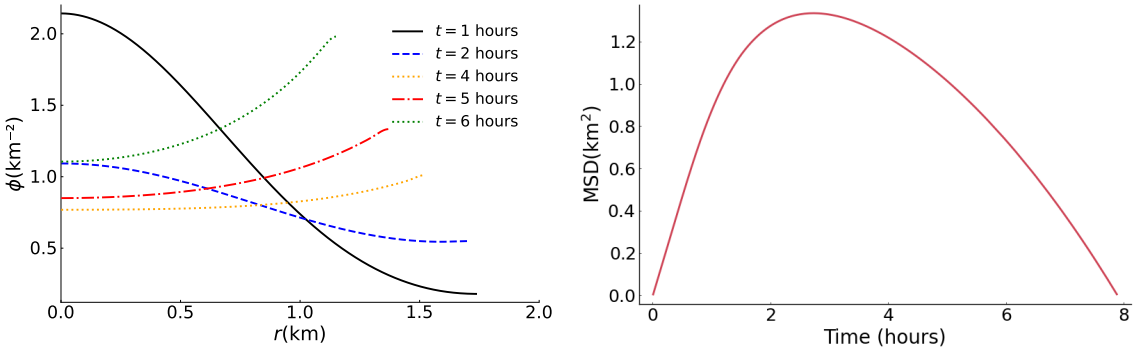


(a) Probability density, ϕ.

(b) Mean squared displacement.

Figure 3.25: Results of a diffusion simulation on a shrinking domain at $t = 1$, 2, 4, 5 and 6 hours. The initial condition is a delta function at $r = 0$ and the time-dependent radius of the domain is $R(t) = \sqrt{2(R_0^2 - \alpha(t - t_s)^2)}$, with $R_0 = 1800$m, $t_s = 1000$ seconds and $\alpha = 3.24\text{m}^2\text{s}^{-2}$. The simulation uses $N = 1800$ annuli and a diffusion coefficient of $D = 200\text{m}^2\text{s}^{-1}$.

Next, we will fit the diffusion and shrinking domain models to phase 1 and phase 2 of the radiotracking data, and assess the goodness of fit for this model. We will also compare the results of the shrinking domain and leapfrog models and discuss some ecological mechanisms that explain the behaviour.



(a) Probability density, ϕ.

(b) Mean squared displacement.

Figure 3.26: Results of a diffusion simulation on a shrinking domain at $t = 1$, 2, 4, 5 and 6 hours. The initial condition is a delta function at $r = 0$ and the time-dependent radius of the domain is $R(t) = \sqrt{2(R_0^2 - \alpha(t - t_s)^2)}$, with $R_0 = 1800$m, $t_s = 1000$ seconds and $\alpha = 3.24\text{m}^2\text{s}^{-2}$. The simulation uses $N = 1800$ annuli and a diffusion coefficient of $D = 63.4\text{m}^2\text{s}^{-1}$.

4 Model fitting using radio tracking data

Here we will fit the diffusion model to phase 1 of the radio tracking data, and the shrinking domain diffusion model to phase 2 of the radio tracking data using a grid search. The diffusion coefficient for the initial, linear dispersal was calculated using `LsqFit.jl`, a package for least squares fitting in Julia [82]. A straight line was fit to the initial linear segment, for time $0 \leq t < 1.25$ hours, and the gradient was used along with equation (3.26) to determine the diffusion coefficient as $D = 63.4\text{m}^2\text{s}^{-1}$. In Section 5, we will discuss the relationship between the diffusion coefficient calculated here and the measured flight speed of Greater Horseshoe bats. For the return phase, the shrinking rate was chosen to give a negative parabola for the MSD, as in equation (3.108). The parameters to fit are α, the rate at which the domain shrinks, t_s, the time at which the domain begins to shrink and R_0, the initial size of the domain. Since bats return to the roost at sunrise, the MSD is 0km^2 at the end of the night. Therefore, the shrink rate, α, is calculated from the parameters R_0 and t_s to ensure the domain size shrinks to 0km at sunrise,

$$\alpha = \frac{R_0^2}{(T - t_s)^2}, \tag{3.110}$$

where $T = 8$ hours is the time to sunrise.

To fit the parameters, we use a grid search. We define our distance metric as the Euclidean distance between simulated and observed data,

$$\rho(\boldsymbol{X}, \boldsymbol{Y}) = \|\boldsymbol{X} - \boldsymbol{Y}\|_2, \tag{3.111}$$

where $\boldsymbol{X}$ corresponds to the MSD values from the radio tracking data and $\boldsymbol{Y}$ corresponds to MSD values calculated in the shrinking domain simulation. A contour plot showing the distance metric, $\rho(\boldsymbol{X}, \boldsymbol{Y})$, for a range of parameters, R_0 and t_s, is shown in Figure 3.27. The plot shows that there is a clear relationship between the two parameter values; as R_0 increases, t_s must decrease to give a similar distance metric. The optimal set of parameters is determined as the point with the minimum distance metric and is marked at $R_0 = 1460\text{m}$

and $t_s = 100$s. These parameters together give a shrinking rate of $\alpha = 0.00266\text{m}^2\text{s}^{-1}$.

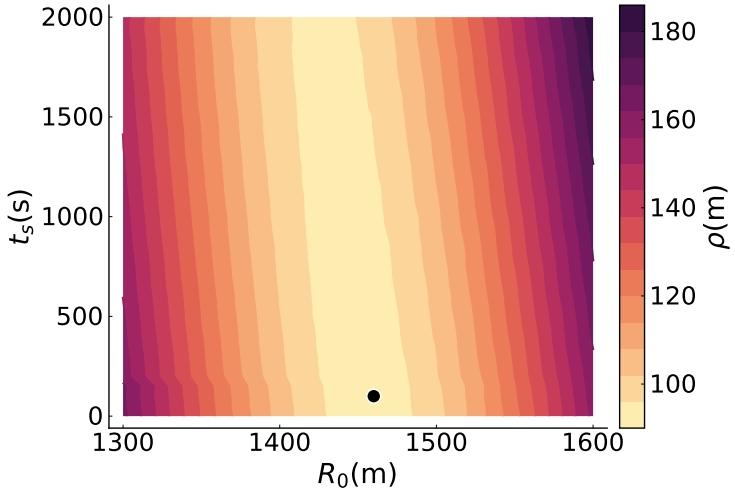


Figure 3.27: A contour plot showing the distance metric, ρ, for a range of parameter values R_0 and t_s. The optimum set of parameters is marked with a black circle at $R_0 = 1460$m, $t_s = 100$s.

The MSD simulation using fitted estimates of the true parameters is shown in Figure 3.28. The curve shows a good fit with the radio tracking data, within the standard error for the majority of the night. Additionally, the distance from the roost for all recordings from the radio tracking data was calculated, and 89% of recordings were within $R_0 = 1460$m, suggesting that this is a good estimate for maximum foraging radius for the majority of bats in this study.

4.1 Comparison of shrinking domain and leapfrogging models

The shrinking domain and leapfrogging models use different mechanisms to push bats back towards the roost, however both act to reduce the size of the domain over the course of the night. In each case, the mechanism driving movement within the domain is diffusion, and the PDE description is equivalent to a stochastic agent-based simulation with infinite numbers of agents.

A comparison of the radius of the shrinking domain and maximum distance from the roost for the leapfrogging models is shown in Figure 3.29. At the start of phase 2, the leapfrogging maximum distance is higher than the shrinking domain maximum distance,

whilst towards sunset the shrinking domain maximum distance is higher. However, during the middle part of the night, $2 < t < 6$ hours, the maximum distance for the leapfrogging simulation and radius of the shrinking domain are almost identical, and we therefore conclude that the two mechanisms are approximately equivalent for the majority of bats for this part of the night.

The differences in maximum radius during the start of phase 2 and end of the night are due to outliers and the differences in the boundary at the edge of the domain. The leapfrogging model does not include a hard boundary, and bats are free to travel to any distance away from the roost. Therefore, a small number of individuals travel significantly further from the roost. In contrast, the shrinking domain model has a reflective boundary such that no bats are able to leave the domain. This is why the maximum distance from the roost is much higher for the leapfrogging model than for the shrinking domain model.

The difference in maximum distance at the end of the night is again due to the differences in boundary conditions. The shrinking domain model forces bats to converge on the roost at a given rate. In contrast, the convection and diffusion components in the leapfrogging model eventually balance out, leading to a steady state before the MSD reaches 0, as with the simple convection-diffusion models described in Section 3.1.

Next, we will discuss some biological explanations for the behaviour described by the models developed in this chapter.

5 Discussion

The initial rapid dispersal from the roost in phase 1 can be explained by competition for resources. If the colony spreads out across the domain, each individual has a larger area to themselves, and therefore more resources available to them.

We have shown that a diffusion model provides a good description of motion in phase 1 of movement. However, it is also possible that there is also some convection component to motion in phase one. Here, we will examine the expected MSD for a convection-diffusion model and place bounds on the size of the convection and diffusion coefficients. We focus

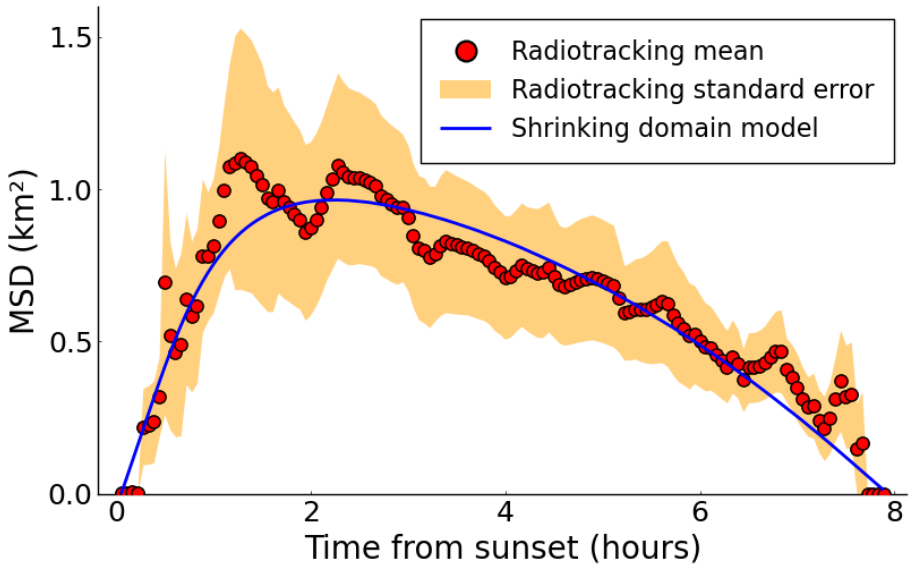


Figure 3.28: Comparison of the MSD for the radiotracking survey and a deterministic diffusion model on a shrinking domain of radius $R(t) = R_0 - \sqrt{\alpha(t - t_s)^2}$ with parameters $R_0 = 1460\text{m}$, $t_s = 100\text{m}$ and $\alpha = 0.00266\text{m}^2\text{s}^{-1}$.

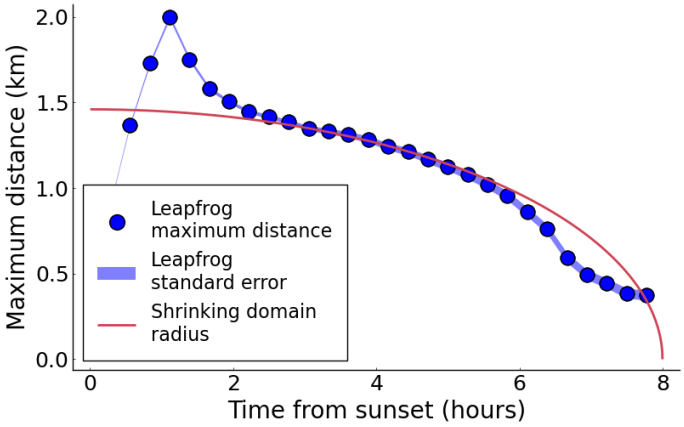


Figure 3.29: Comparison of the radius of the shrinking domain simulation shown in Figure 3.28 and the maximum distance for the leapfrogging simulation shown in Figure 3.21.

only on phase 1 of movement, and therefore set the timescale to $0 < t < 1.25$ hours.

We use the convection-diffusion model defined in equation (3.51),

$$\frac{\partial \phi(r,t)}{\partial t} = \frac{D}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \phi(r,t)}{\partial r}\right) + \boldsymbol{\chi}\left(\frac{\partial \phi(r,t)}{\partial r} + \frac{\phi}{r}\right). \tag{3.112}$$

where in this case the convection coefficient, χ, is a constant. The initial condition is

$$\phi(r, t = 0) = \delta_0(r). \tag{3.113}$$

We can again calculate the expected MSD for this model using the method of moments, as with the simple diffusion model, however, the function produced is not easy to interpret. Instead, we will simulate using the method of lines, using the discretisation described in equation (3.54). The MSD for simulations with $D = 63.4\text{m}^2\text{s}^{-1}$ and a range of convection coefficients is shown in Figure 3.30. The convection adds a quadratic component to the MSD because the flow is a constant velocity, and the squared displacement is therefore a quadratic.

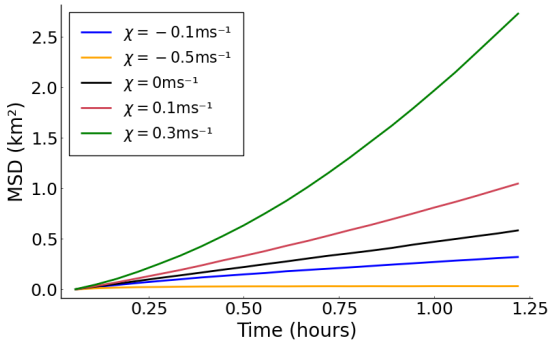


Figure 3.30: The MSD for convection-diffusion simulations, each with $D = 63.4\text{m}^2\text{s}^{-1}$, and with varying χ.

To set bounds on the size of χ and D, we will compare the MSD from phase 1 of the radio tracking data in Figure 3.5 to the MSD produced by convection-diffusion simulations with a range of diffusion and convection coefficients. We define a distance metric as the Euclidean distance between the MSD from the radio tracking data, $\boldsymbol{Y}$, and simulation MSD, $\boldsymbol{X}$,

$$\rho(\boldsymbol{X}, \boldsymbol{Y}) = \|\boldsymbol{S}(\boldsymbol{X}) - \boldsymbol{S}'(\boldsymbol{Y})\|_2. \tag{3.114}$$

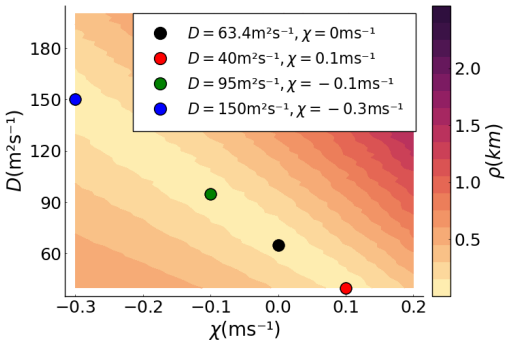
A contour plot of the distance metric for a range of convection and diffusion coefficients is shown in Figure 3.31a. There is a clear relationship between the two coefficients, and simulations with high convection coefficients require a low diffusion coefficient to produce

low distance metrics. The MSD for four different combinations of χ and D is shown in Figure 3.31b, along with the MSD from the radio tracking data for comparison. For each of the four combinations of χ and D shown here, the simulation MSD remains within the standard error for the majority of the time, suggesting that each could provide a good model for bat movement during phase 1.

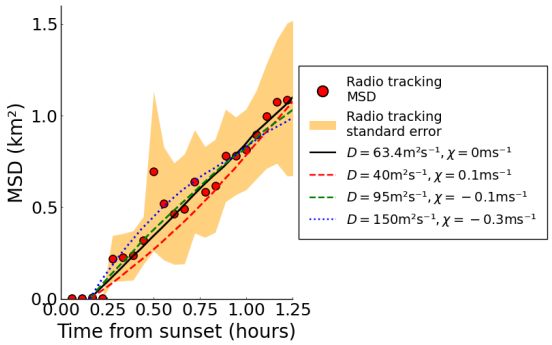
We will use the Peclet number to define the bounds on the relative size of χ and D. The Peclet number is given by

$$Pe = \frac{L\chi}{D} \tag{3.115}$$

where L is a characteristic lengthscale. We define $L = 3000$m, as this is the radius of the domain. We use the distance metric defined by equation (3.114) to determine bounds on χ and D, selecting pairs of coefficients which give a distance metric of $\rho < 0.5$km in Figure 3.31a. Calculating the Peclet number for these simulations gives a range of $-15 < Pe < 15$. Over this range, the magnitude of the convection coefficient is significantly smaller than the diffusion coefficient, and the effect of convection is small.



(a) The distance metric given by equation (3.114) for various convection and diffusion coefficients.



(b) The MSD for various convection and diffusion coefficients, shown along with the MSD from radio tracking data for comparison.

Figure 3.31: The results of convection-diffusion simulations with parameters $R = 2000$m, $N = 100$, $T = 1.25$ hours and varying convection and diffusion coefficients.

To investigate whether the diffusion coefficient calculated here for phase 1 of movement, $D = 63.4\text{m}^2\text{s}^{-1}$, is accurate, we will consider the flight speed of Greater Horseshoe bats. During foraging, Greater Horseshoe bats have been recorded flying at speeds of 8.1ms^{-1} [86]. compare the diffusion coefficient to this flight speed, we consider the mean speed

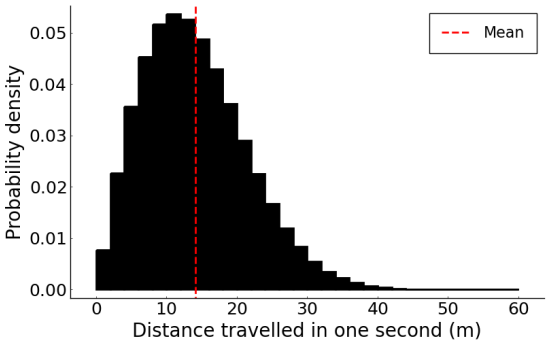
from the diffusion SDE given by equation (3.77). A histogram of the distance travelled in timestep with $dt = 1$ second is shown in Figure 3.32a. The mean is shown at 14.1m, suggesting that the flight speed for a diffusion coefficient of $D = 63.4\text{m}^2\text{s}^{-1}$ is 14.1ms^{-1}.

However, the speed of a diffusive particle is dependent on the timescale. A sample path generated from the diffusion SDE is shown in Figure 3.32b. The displacent for a timescale of $dt = 6$ seconds is not equal to the total distance travelled because the particle changes direction at each timestep. As a result, the speed decreases as the timestep is increased. The speed at varying timesteps is shown in Figure 3.32c, along with the Greater Horseshoe flight speed of 8.1ms^{-1}. At $dt = 3$ seconds, the diffusion speed is equal to the flight speed. We therefore conclude that the diffusion coefficient is of the correct magnitude, however, without knowledge of the timescale used to determine the flight speed, we cannot compare the speed from the diffusion model more precisely.

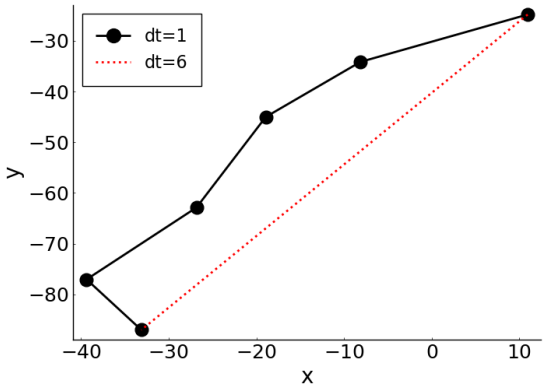
The model for phase 2 is not as easy to interpret as the model for phase 1, and there are a number of possible explanations.

The movement of bats within a domain with a shrinking boundary corresponds to those furthest from the roost always moving towards the roost, whilst those closer to the roost move diffusively. The probability density for these simulations in figures 3.25a and 3.26 show a clustering of probability density against the moving boundary, suggesting that bats behaving in this way would gather together. This behaviour could be explained by a desire for bats to move towards the locations of other foraging bats, as described by the leapfrogging model. In this case, bats would continue foraging until they notice that other bats in the colony are starting to return to the roost, at which point they would start to move back towards the roost, following the group. In other words, bats are happy to forage when they can hear calls all around them, but if all the calls they can hear come from one direction, towards the roost, they start to move in that direction.

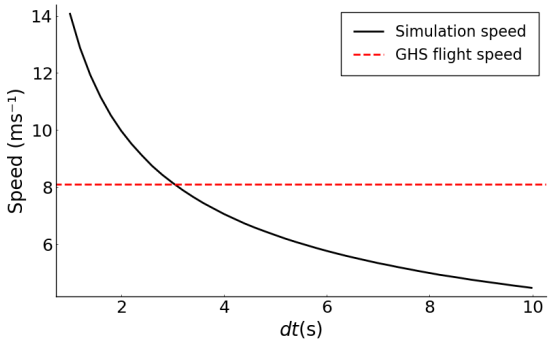
It is likely that the density of bats far away from the roost is often low enough that they do not hear calls from other bats constantly. However, foraging bats are not stationary, instead they fly around searching for prey [87]. In this case, we suggest that the mechanism sending bats towards the roost could instead be the absence of calls close to them: if a



(a) A histogram of the distance travelled in one timestep with $dt = 1$ second.



(b) A random walk trajectory at two timescales, $dt = 1$ second and $dt = 6$ seconds.



(c) The mean speed at various timescales. The mean flight speed of Greater Horseshoe bats, 8.1ms^{-1} is shown for comparison.

Figure 3.32: The mean speed for diffusive particles with diffusion coefficient $D = 63.4\text{m}^2\text{s}^{-1}$ at varying timescales, dt.

bat spends a significant period of time in silence, without hearing calls from other bats, it decides to head closer to the roost, towards where it will find other members of the colony.

In fact, a common foraging strategy amongst some bat species is to eavesdrop on the hunting calls of other bats to easily and quickly locate hunting grounds [88, 89]. This strategy is most common in landscapes dominated by cropland, where prey is difficult to find for a single bat due to patchy and ephemeral or unpredictable insect distribution and is uncommon in woodland where insect distribution is more reliable. Eavesdropping allows bats to locate areas with insects by following bats that have already found these hunting grounds. A satellite image of the area covered by bats in the survey is displayed in Figure 3.33, showing that the area is primarily farmland, with very little woodland,

and thus it is likely that bats foraging in this landscape may employ this eavesdropping technique. If bats are using eavesdropping whilst foraging, we would expect to see bats gathering together over the course of the night. It is not possible to test this hypothesis using standard radio tracking techniques, as we cannot track every bat in a colony at the same time: we might record one tagged bat in a specific location, but there is no way to test if there are others around.

The quadratic form of the radius of the shrinking domain, given by equation (3.108), suggests that bats move very slowly towards the roost at the start of phase 2, and the rate of return increases as time goes on. We consider two possible explanations for this. Firstly, a bat located far from the roost is likely to have less knowledge of its surroundings, and therefore may be more inclined to employ the eavesdropping strategy as it is more reliant on the behaviour of other bats in the colony. As the bat travels closer to the roost, navigation becomes easier because the landscape is more familiar, and the bat is therefore able to return directly to the roost rather than relying on others. Alternatively, the time of night may provide behavioural cues. Early in the night, the bat knows that there are several hours before the sun rises and it can therefore continue to forage safely and eavesdrop on other bats to locate abundant hunting grounds. Later on in the night, the sense of urgency to return to the roost becomes more pronounced, as the bat does not want to be stranded far from the roost, without shelter, when the sun rises, and therefore travels directly back to the roost.

Additionally, bats may have a mind-map of where they want to forage (for example, routes learnt from their mothers), and decide to turn back towards the roost once they have reached the edge of their foraging zone. It would be possible to test this hypothesis by tracking one bat over multiple days to see if they repeat their foraging routes, however this is beyond the scope of the work discussed here. An alternative explanation is the possibility that bats do not want to be further from the roost than the rest of the colony once they have eaten, and therefore start to travel back after foraging.

Figure 3.33: A satellite image of the area covered by bats in the survey.

Chapter Four

Estimating roost locations

As discussed in Chapter 1, locating bat roosts is generally very difficult and time consuming. In this chapter, we will discuss a method to estimate roost locations using data from static detector surveys to narrow down the search area and decrease the time required to locate roosts. A diffusion model is used to model bat movement during the initial, dispersive phase of movement at the start of the night, and is fit to data from acoustic detectors using a random search to produce an estimate for the roost location. Later, we will compare the results of roost estimates calculated using a deterministic model and random search to estimates calculated using a stochastic model and ABC.

1 Static detector surveys

Many bats use echolocation to navigate and catch prey [90], and acoustic surveys using bat detectors have proven to be a useful and cost-effective method to study populations [91]. Acoustic bat detectors are microphones calibrated to record the high-frequency sound of bat calls and allow for long-term, autonomous surveys. Surveys are conducted by placing detectors within a search area which record the calls of bats passing through their range. The coordinates of each detector are noted and a series of recordings is produced. The species for each recording can be identified in most cases using the mean frequency and shape of the waveform of the call [92, 93], and a list of times at which a bat of the species in question has been recorded at each detector is produced. For some surveys, not all

detectors are operational for each day of the survey due to limited battery life, malfunctions, or damage to the equipment, and so we use the total number of calls at each detector to calculate the mean number of calls recorded at each detector over each night of the survey in which the detector is operational. We will use only the data from the first 90 minutes after sunset, as this corresponds to the diffusive phase 1 of movement identified in Chapter 3. As shown in Figure 1.1, phase 1 also corresponds to the time of peak activity levels for Greater Horseshoe bats. Whilst there is often a second peak in activity levels close to sunrise, this is not always the case for all species - for example, two species of pippistrelle (the two most common bat species in the UK), have been shown to have only a single peak in activity levels, just after sunset [15]. As a result, we do not consider the activity levels at the end of the night, as this would likely not be generalisable to other species.

We will denote the mean number of calls at detector i as v_i and the location of detectors as a $2 \times n$ matrix, $\mathbf{X}$, where $\mathbf{X}_i \in \mathbb{R}^2$ is the location of the i-th detector. In general, if we are searching for a roost, we are unlikely to know the number of bats in the colony and therefore we normalise the number of calls, v_i, to the proportion of calls at detector i,

$$Y_i = \frac{v_i}{\sum_i v_i}. \tag{4.1}$$

We will look at the results of 6 acoustic surveys of Greater Horseshoe bats conducted at 4 different maternity roosts (used between May-September by females and their young) in Devon throughout summer 2016 [11]. Three surveys were conducted at Buckfastleigh (June, July and August), and one survey was conducted at each of Braunton (July), Gunnislake (August) and High Marks Barn (August). Each of these roosts are located in Devon, UK, close to the roosts studied in the radiotracking survey in Section 1, however, no data from acoustic surveys at these roosts were available.

Manual counts were taken at each roost throughout summer 2016, and the number of bats found in each roost is shown in Table 4.1. These counts were performed by workers entering the roost during the day, whilst bats were sleeping inside and counting each individual bat. All four of these roosts are large, used by hundreds of bats during the

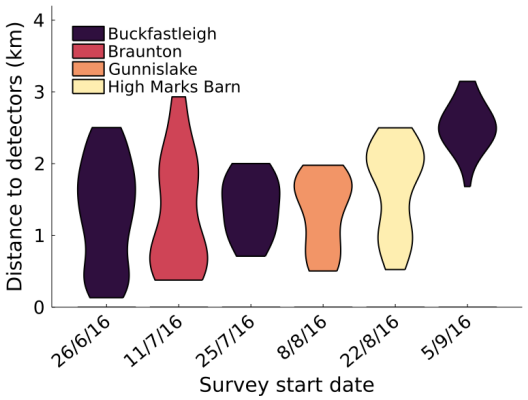
Date	Roost	Bat count	Number of detectors	Mean passes per detector
26/6/16	Buckfastleigh	1798	21	5.54
11/7/16	Braunton	512	19	3.63
25/7/16	Buckfastleigh	1798	25	3.91
8/8/16	Gunnislake	252	21	1.32
22/8/16	High Marks Barn	746	32	2.80
5/9/16	Buckfastleigh	1798	21	2.09

Table 4.1: Details of each of the six surveys at Buckfastleigh, Braunton, Gunnislake and High Marks Barn.

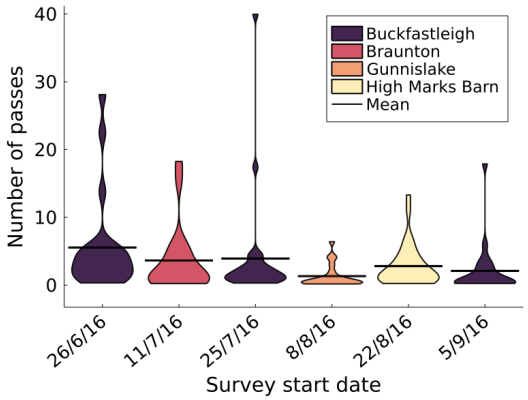
maternity season, however the Buckfastleigh roost is by far the largest, and is thought to be one of the largest Greater Horseshoe roosts in Europe, used by approximately 1800 bats during the maternity season in 2016.

All of these surveys were originally conducted to study how bats use the landscape surrounding the roost, and the impact of different terrain types on foraging behaviour. For example, it is generally recognised that Greater Horseshoe bats are highly dependent on linear features, such as hedgerows. They are used during foraging as corridors to navigate throughout the wider landscape [86, 13]. The roost sites were known beforehand, and detectors were placed as evenly as possible around the roost location, given constraints caused by access issues, whilst trying to cover a variety of different terrain types [11]. Details of the detectors used for each of the surveys are shown in Table 4.1. The surveys used between 19 and 32 detectors. The distribution of distances between each detector and the roost for each survey is shown in Figure 4.1a. The minimum distance between detectors and the roost is seen in the first Buckfastleigh survey, at a separation of 130m, whilst detectors are placed significantly further away in the last Buckfastleigh survey, with the closest placed 1.6km away. The number of nights over which detectors were active for each of the six surveys is shown in Figure 4.1c. All detectors were active for between 1 and 8 nights before either the survey ended or the detector malfunctioned. The mean number of active nights for each survey ranges between 3.8 and 5.5. The distribution of

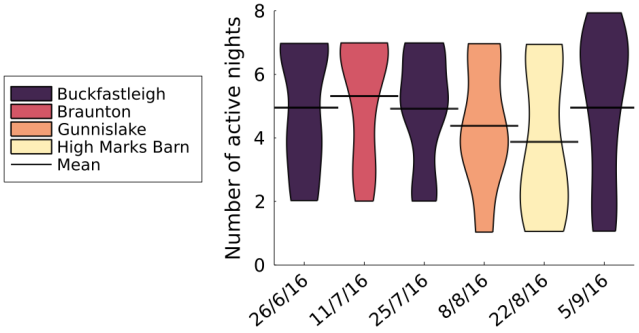
the number of bat passes recorded per night at each detector recorded during each survey is shown in Figure 4.1b. The number of passes recorded will depend greatly on the size of the colony using the roost as well as on the distance between the detectors and the roost and the landscape features surrounding each of the detectors. The first Buckfastleigh survey recorded the most passes per detector, a mean of 5.54 passes per detector, and the Gunnislake survey the least, at 1.32 passes per detector, unsurprising given the relative size of these roosts.



(a) The distribution of distances from each detector to the roost for each of the six surveys, illustrated as violin plots.



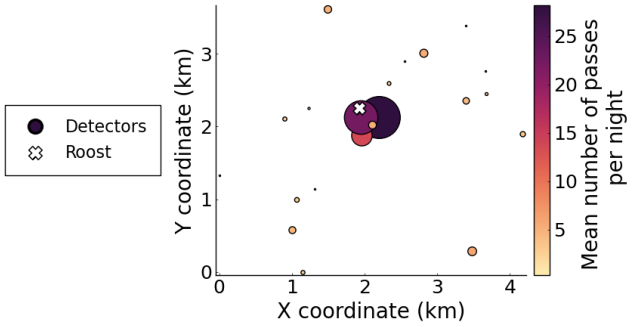
(b) Distributions of the number of passes recorded per night at each detector for each of the six surveys.



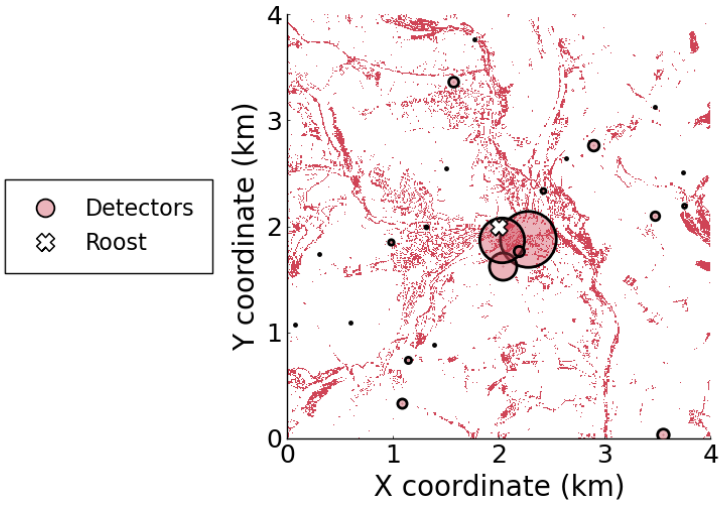
(c) Distributions of the number of nights that detectors were active for over each of the six surveys.

Figure 4.1: Details of each of the six surveys at Buckfastleigh, Braunton, Gunnislake and High Marks Barn. The width of each violin plot is scaled by the percentage of records at each value on the y axis.

The results of the first survey at Buckfastleigh, between 26/06/2016 and 03/07/2016, hereafter referred to as survey 1, are shown in Figure 4.2. In general, the detectors closest



(a) Markers are coloured according to the mean number of calls recorded each night in the first 90 minutes after sunset.



(b) Detector locations are superimposed on top of a map of hedgerows in the survey area.

Figure 4.2: The results of an acoustic survey of Greater Horseshoe bats at a large roost containing approximately 1800 bats in Buckfastleigh, Devon, referred to as Survey 1. The survey was conducted between 26/06/2016 and 03/07/2016. Circle markers are centred at the detector location and each marker i is sized according to the proportion of calls recorded at detector i, $\text{radius}_i = \boldsymbol{Y}_i$km, in the first 90 minutes after sunset. The location of the roost is shown as a white cross.

to the location of the roost recorded more calls than those farther away, as we would expect if bats are dispersing according to a diffusion model as described in Chapter 3. Next we will use the diffusion model to estimate the number of expected calls at each detector given a simulated roost location.

2 Adapting the diffusion model

As shown in Chapter 3, bat movement for the first 90 minutes after sunset can be modelled as diffusive. We will use a deterministic PDE diffusion model to generate simulated data and estimate the expected number of calls at each detector using the roost location, $\boldsymbol{z}$, as a parameter to be estimated using a random search. In this case, a deterministic model may be preferable to a stochastic model as simulating stochastic data is significantly more computationally expensive. Estimating the roost using stochastic simulations with 10^4 samples and ABC takes over 3 hours with a 2.7GHz processor and 8GB RAM when using 4 cores in parallel. In comparison, a random search with a deterministic model and 10^5 samples takes just 2 seconds on a single core using the same machine. We will later show roost estimation results for one of the surveys, survey 6, using a stochastic diffusion model to compare with the deterministic model.

We assume that bats leave the roost at sunset and fly away from the roost in search of food. Foraging bats are modelled as diffusive particles for the first 90 minutes after sunset. The 2D diffusion equation describes the probability density $\phi(u, v, t)$ of finding a specific bat at position (u, v) at time t,

$$\frac{\partial \phi(u, v, t)}{\partial t} = D\nabla^2 \phi(u, v, t), \tag{4.2}$$

where D is the diffusion coefficient, calculated using the relationship between mean squared distance travelled from the roost and time from sunset. A typical value calculated in Chapter 3 using radio tracking surveys is $D = 63.4\text{m}^2\text{s}^{-1}$. The initial condition,

$$\phi(u, v, 0) = \delta^2(z_u, z_v), \tag{4.3}$$

where $\delta^2(z_u, z_v) = \delta(z_u)\delta(z_v)$ is the two-dimensional Dirac delta function, specifies that all bats start the night at the roost, $\mathbf{z} = (z_u, z_v)$. The model assumes infinite space with boundary condition

$$\nabla\phi(u, v, t) \to 0, \quad \text{as} \ \mid u \mid, \mid v \mid \to \infty, \tag{4.4}$$

such that bats are free to travel any distance away from the roost.

We use the probability density function calculated from the solution to the SDE in Section 3.3.2, equation (3.78),

$$\phi(u,v,t) = \frac{1}{4\pi Dt} e^{-\frac{(u-z_u)^2+(v-z_v)^2}{4Dt}}, \tag{4.5}$$

$$\phi(x,y,t) = \frac{1}{4\pi Dt} e^{-\frac{x^2+y^2}{4Dt}}, \tag{4.6}$$

where we have used the coordinate transformation

$$x = u - z_u, \tag{4.7}$$

$$y = v - z_v. \tag{4.8}$$

Whilst diffusion here is unbounded, it can be shown that the probability of a bat travelling further than the maximum distance travelled from the roost in the radiotracking survey in Chapter 1, $R_f = 3000$m, is typically small. Specifically, at time $T = 90$ minutes, with a diffusion coefficient of $D = 63.4\text{m}^2\text{s}^{-1}$, the value calculated in Chapter 3, $\phi(R_f, T) \approx 10^{-6}$, and therefore the probability density stays negligible outside the radius for all times of interest. Thus, since R_f is large, the distance between the finite, but large, radius and the infinite domain case is small.

Integrating $\phi(x,y,t)$ over Ω, a circle of radius r, where r is the range of detectors, around detector i at position (x_i, y_i), and multiplying by the number of bats in the roost n gives the expected density of bats in range of detector i at time t. A diagram of the region, Ω, around a detector is shown in Figure 4.3. As no elementary indefinite integral exists for the function e^{-p^2}, the integral is approximated using a quadratic Taylor expansion around

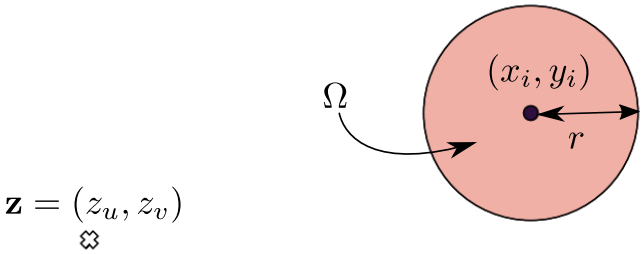


Figure 4.3: A diagram of the area of radius r, denoted as Ω, around detector i at location (x_i, y_i).

(x_i, y_i),

$$\begin{aligned} N_i(t) &= \int_\Omega \phi(x, y, t) n d\omega \\ &= \int_\Omega \frac{1}{4\pi Dt} e^{-\frac{x^2+y^2}{4Dt}} n d\omega \\ &\approx \frac{1}{4Dt} e^{-\frac{x_i^2+y_i^2}{4Dt}} \left[r^2 + \frac{r^4}{4} \left(\frac{x_i^2 + y_i^2}{8D^2t^2} - \frac{1}{2Dt} \right) \right] n. \end{aligned} \tag{4.9}$$

The total number of calls expected at each detector for $0 < t < T$ is calculated by integrating $N_i(t)$ over time,

$$N_i^*(T) = n \int_{t=0}^{t=T} N_i(t) dt. \tag{4.10}$$

Again, no elementary indefinite integral exists for the function $e^{1/t}/t$, and so the integral is approximated using a Simpson's rule numerical integration [94, 95]. From this $N^*(T)$, we can calculate the expected proportion of calls at detector i, γ_i, given roost location $\mathbf{z}$,

$$\gamma(\mathbf{z}) = \frac{N_i^*(T)}{\sum_i N_i^*(T)}. \tag{4.11}$$

Note that the number of bats, n, cancels when we use the proportion of calls in equation

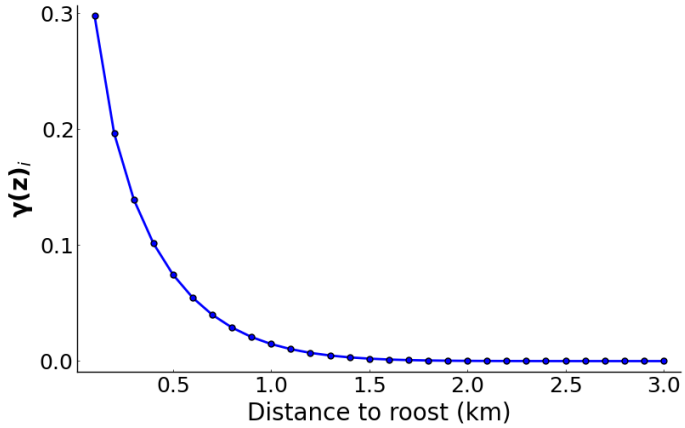


Figure 4.4: The expected proportion of calls, $\gamma(\mathbf{z})_i$, calculated using equation (4.11) at each of 30 detectors placed at intervals of 100m distance from the roost for a deterministic diffusion simulation with parameters $D = 63.4\text{m}^2\text{s}^{-1}$ and $T = 90$ minutes.

(4.11), and we therefore do not need to know the size of the colony using the roost. The value of $\gamma(\mathbf{z})_i$ for 30 detectors at intervals of 100m distance from a roost is shown in Figure 4.4, illustrating that the proportion of calls expected decays rapidly as we move away from the roost.

Next we will define a distance metric to compare the expected proportion of observations at each detector to data recorded from surveys and use this to estimate roost locations from survey data.

3 Estimating the roost location

We use a random search to estimate the roost location. First, we define a distance metric, $\rho(\gamma(\mathbf{z}), \mathbf{Y})$, using the squared Euclidean distance between the expected proportion of observations at each detector, $\gamma(\mathbf{z})$, and survey data, $\mathbf{Y}$,

$$\rho(\gamma(\mathbf{z}), \mathbf{Y}) = \frac{1}{n} \sum_{i}^{n} (\gamma(\mathbf{z})_i - \mathbf{Y}_i)^2, \tag{4.12}$$

where n is the number of detectors. The squared distance is chosen here to emphasise the importance of detectors which record high numbers of passes. Detectors which record particularly high numbers of bat passes are likely to be located close to the roost, whilst

those which record very few passes are likely to be located far from the roost. The squared distance as opposed to, for example, the Manhattan distance, places more importance on the detectors which record high number of passes, and therefore those which are likely to be close to the roost.

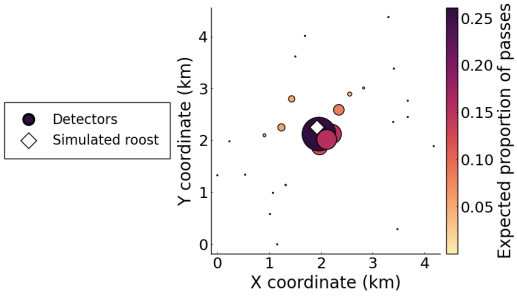
Roost locations, $\gamma(\mathbf{z})$, which result in a low distance metric, $\rho(\gamma(\mathbf{z}), \mathbf{Y}) < \epsilon$, where ϵ is some threshold value, are added to a set of plausible points, whilst those with a high distance metric, $\rho(\gamma(\mathbf{z}), \mathbf{Y}) > \epsilon$ are discarded.

The expected proportion of calls for various simulated roost locations, $\mathbf{z}$, is illustrated in Figure 4.5, and compared with the results of Survey 1. In Figure 4.5a, the expected proportion of calls is estimated for a simulated roost, $\mathbf{z}$, placed at the same location as the true roost, whilst in Figure 4.5b, the simulated roost is placed 1.8km away from the true roost. As shown in Figure 4.4, we expect detectors close to the roost to record a significantly higher proportion of calls when compared to those further away. We can clearly see in the true proportion of calls in Figure 4.5c that the detectors closest to the roost record the majority of calls, and those further away record very few.

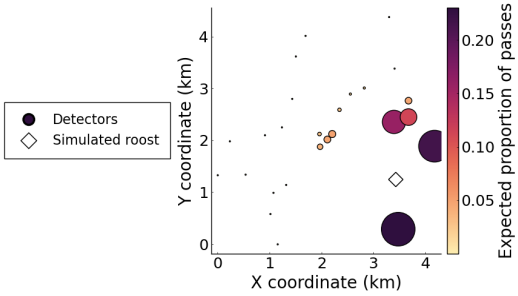
By inspection, we see that the expected proportion of calls is qualitatively more similar to the true proportion in Figure 4.5a than in Figure 4.5b, and comparing the distance metrics for each, calculated using equation (4.12) ($\rho(\gamma(\mathbf{z}), \mathbf{Y}) = 0.00185$ and $\rho(\gamma(\mathbf{z}), \mathbf{Y}) = 0.00897$, respectively) shows us that in this case the algorithm will select the true roost location over the false roost in Figure 4.5b.

A surface plot of the distance metric, $\rho(\gamma(\mathbf{z}), \mathbf{Y})$, is shown over a range of simulated roost locations in Figure 4.5d. Simulations with roost locations, $\mathbf{z}$, with low distance metrics, $\rho(\gamma(\mathbf{z}), \mathbf{Y}) < \epsilon$, will be added to the set of plausible points. The area with the lowest distance metric is located around the true roost, and very close to the three detectors that recorded the highest proportion of passes.

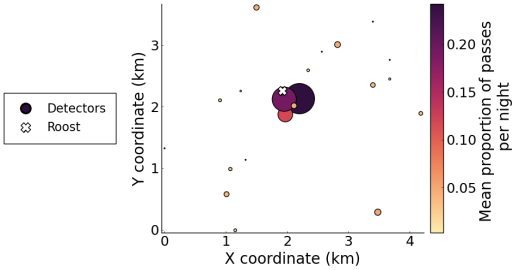
We use a random search to estimate the roost location. Whilst an optimisation algorithm could be used to find a single point which gives the minimum distance metric, it is useful for ecologists to have a defined area to search. For some surveys (for example, see the results for Survey 5 in Figure 4.10), there may be multiple local minima that a



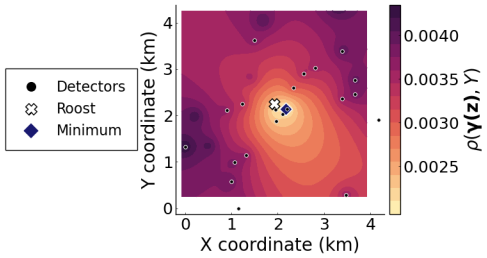
(a) An estimate of the expected proportion of calls at each detector, $\gamma(\mathbf{z})$, calculated using equation (4.11) for a simulated roost location $\mathbf{z} = (1.93, 2.26)$ km. The distance metric is calculated as $\rho(\gamma(\mathbf{z}), \mathbf{Y}) = 0.00185$.



(b) An estimate of the expected proportion of calls at each detector, $\gamma(\mathbf{z})$, calculated using equation (4.11) for a simulated roost location, $\mathbf{z} = (3.43, 1.26)$ km. The distance metric is calculated as $\rho(\gamma(\mathbf{z}), \mathbf{Y}) = 0.00897$.



(c) The true proportion of calls recorded at each detector, $\mathbf{Y}$.



(d) A contour plot of the distance metric, $\rho(\gamma(\mathbf{z}), \mathbf{Y})$, calculated for a grid of simulated roost locations, $\mathbf{z}$.

Figure 4.5: The true proportion of calls, $\mathbf{Y}$, and estimates of the expected number of calls, $\gamma(\mathbf{z})$, for varying roost locations at each detector from Survey 1.

single point estimate would not capture. To implement a random search, we start with an initial search area, $\boldsymbol{f}(\mathbf{z})$, for the roost location $\mathbf{z}$, defined as our initial knowledge of where the roost might be. Observations, $\mathbf{Y}$, are calculated from detector data using equation (4.1). We draw 10^5 samples for $\mathbf{z}$, uniformly distributed throughout our initial search area, simulate $\gamma(\mathbf{z})$ using equation (4.11) and calculate a distance metric, $\rho(\gamma(\mathbf{z}), \mathbf{Y})$, using equation (4.12) for each sample.

We define our threshold parameter, ϵ, such that the 2% of samples with the lowest distance metric are added to the set of plausible points. This is an arbitrary choice, chosen to give a usable search area, and is easily changeable if an ecologist wishes to have a larger or smaller search area. Finally, the roost estimate is calculated by taking the mean of the

plausible points.

The estimate is defined as

$$\zeta(\mathbf{X}, \mathbf{Y}) = \mathbb{E}(\boldsymbol{z} \mid \boldsymbol{X}), \tag{4.13}$$

$$= \frac{1}{\eta} \sum_j \boldsymbol{z}_j, \tag{4.14}$$

where η is the number of accepted samples in the set of plausible points and z_j is the jth accepted roost location in the set of plausible points.

Finally, we define the final search area as the area of a concave hull of the plausible points using an algorithm implemented in the Julia package ConcaveHull.jl [96, 97]. Note that, whilst the calculation for a concave hull is similar to that of a convex hull, they are not equivalent. A convex hull is calculated by first locating an extreme point, such as the point with the lowest value in y, and then finding subsequent points. At each iteration, the next point is the point (from the set of all remaining points) which gives the largest right hand turn [98]. In contrast, the concave hull algorithm considers only the k-nearest neighbours of the current point as candidates to become the next point in the polygon at each iteration. Increasing the value of k increases the smoothness of the resulting polygon, and the special case where k is equal to the total number of points leads to a standard convex hull. To calculate the concave hull of the search area for each of the surveys shown here, we use $k = 20$, equivalent to 1% of the number of accepted samples in the set of plausible points.

Next, we will test the roost finding algorithm using the six surveys shown in Figure 4.1.

3.1 Testing the roost finding algorithm

The roost finding algorithm is implemented for Survey 1 using a square initial search area, $\binom{0}{0} \leq \boldsymbol{z} \leq \binom{5}{5}$km, resulting in an initial search area of 25km^2. The final search area is 0.33km^2, around 1.3% of the initial search area. The roost estimate is taken as the mean of plausible points, $\zeta = (2, 2)$km and the error in estimate, defined as the Euclidean distance between the true roost and roost estimate, is 0.21km. A weighted average of the

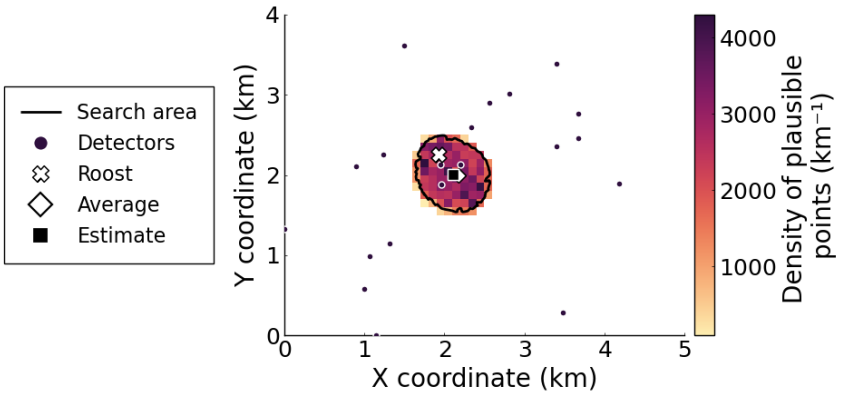


Figure 4.6: The density of plausible points, shown as a heatmap overlaid on the scatter plot of detectors, for Survey 1, conducted at Buckfastleigh, Devon between 26/06/2016 and 03/07/2016. A square initial search area was used, $\binom{0}{0} \leq \boldsymbol{z} \leq \binom{5}{5}$km. Plausible points are defined as the 2% of samples with the lowest distance metric, $\rho(\gamma(\mathbf{z}), \mathbf{Y})$.

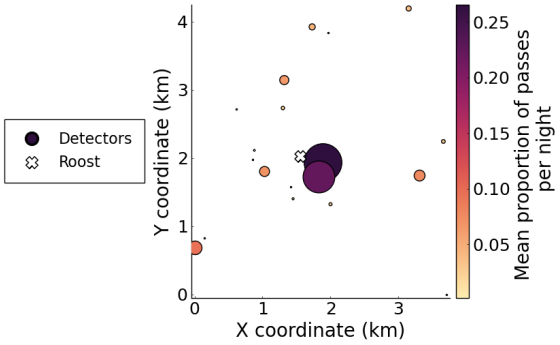
detector locations is shown for comparison,

$$\text{Average} = \sum_i \boldsymbol{Y}_i \boldsymbol{X}_i. \tag{4.15}$$

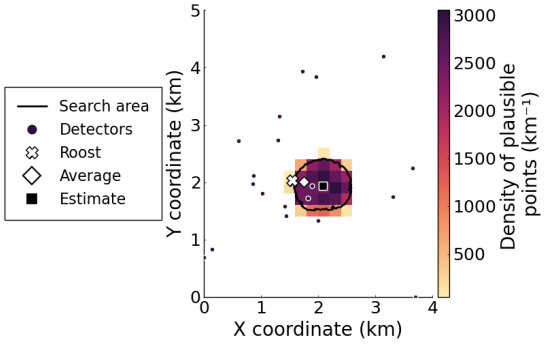
Taking a weighted average may seem to provide a useful estimate of roost location in some cases, and indeed in the majority of the surveys discussed here the average is located very close to the roost estimates calculated using the diffusion model. However, the average will only be useful for surveys in which detectors are distributed approximately evenly around the true roost location; if the roost is located outside of the detector array, it will never produce a useful estimate. For this survey, the method is highly successful, producing an accurate estimate and reducing the search area to a small fraction of its original size.

However, the method will not necessarily be as successful for other surveys, depending on the data recorded. The colony that uses this roost is much larger than an average colony, and as shown in Figure 4.1a, there are detectors placed very close to the roost.

The results for Surveys 2-6 are shown in figures 4.7-4.11. Clearly, some of these surveys are more successful than others. Survey 2, conducted at the second roost site in Braunton between 11/07/2016 and 18/07/2016 is shown in Figure 4.7. In this case, the colony using



(a) The true proportion of calls recorded at each detector, $\mathbf{Y}$.



(b) The density of plausible points, shown as a heatmap overlaid on the scatter plot of detectors. The error in roost estimate is 0.33km and the search area is 0.66 km^2.

Figure 4.7: The true proportion of calls, $\mathbf{Y}$, and the density of plausible points for Survey 2, conducted at Braunton, Devon between 11/07/2016 and 18/07/2016.

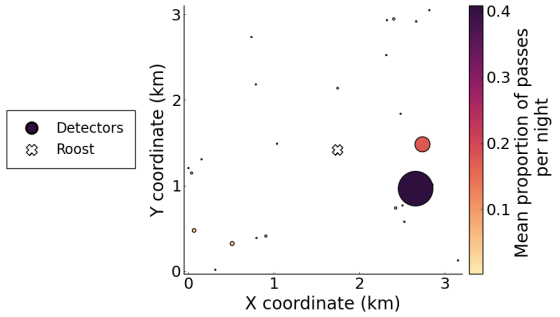
the roost is much smaller and therefore the number of passes, as shown in Figure 4.1b, recorded is less than half of that recorded in Survey 1. Additionally, as shown in Figure 4.1a, the closest detectors to the roost are further than those for Survey 1. Even so, the method is highly effective in locating the roost, with an error of just 0.33km and a search area of 0.66 km^2.

In contrast, Survey 3 (shown in Figure 4.8), conducted at Buckfastleigh between 25/07/2016 and 03/08/2016, is less successful. Two detectors recorded high numbers of passes in this survey, the highest at $(x, y) = (2.655, 0.967)$km and the second at $(x, y) = (2.655, 0.967)$km, 1.01km and 0.996km from the roost, respectively, and this explains why the model selected samples close to this region as plausible points. The reason for the high number of passes recorded at these detectors is not immediately obvious

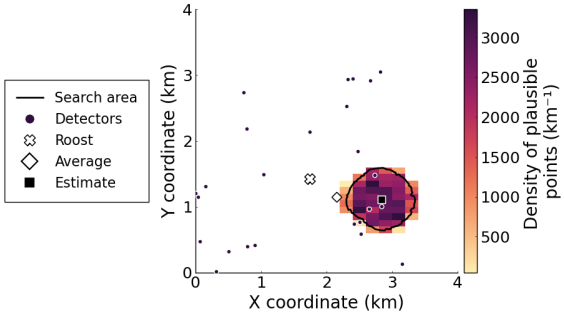
from the data, however when we look at the locations of hedgerows in the survey area, shown in Figure 4.8c, we see that the detector recording the highest number of passes at $(x, y) = (2.655, 0.967)$km was placed along one of the main hedgerow paths leading away from the centre of the survey area. As noted in Section 1, Greater Horseshoe bats are highly dependent on hedgerows for navigation and foraging, and it is likely that this particular hedgerow is used as a corridor between important parts of the landscape. In Section 4 we will suggest some possible adjustments to the model to account for the heterogeneity of the landscape.

The results for Survey 4 are shown in Figure 4.9. As shown in Figure 4.1, the colony using the roost at Gunnislake is much smaller than those using the other roosts, and the mean number of passes recorded at each detector was 1.32, much lower than for each of the other surveys. The detector that recorded the most passes is located at $(x, y) = (2.34, 2.45)$km, and the distribution of plausible points is clustered close to this point, but shifted slightly to the north, further from the roost. This can be explained by looking at the distance metric, shown in Figure 4.9b. The optimal point with the minimum distance metric is located closer to the roost than the detector, as there are 3 detectors south of the roost that recorded high numbers of passes. However, the contour around the optimum point is not centred around the optimum point, and instead is shifted further away from the roost, and the set of plausible points can take samples from anywhere in this contour.

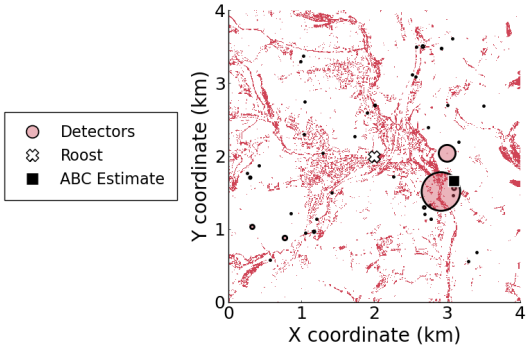
The results for Survey 5 are shown in Figure 4.10. In this case, detectors were not distributed evenly around the domain as the area to the South East of the roost is private land and it was not possible to obtain landowner permission to place detectors here. Many of the detectors close to the roost recorded high numbers of calls, and there are two detectors which recorded significantly more passes than the others. There are two peaks in the distribution of plausible points: one very close to the roost and the other clustered around the detector that recorded the most passes. The distance metric plotted in Figure 4.10b shows that the point with the minimum distance metric is located at the point of the detector with the most passes, and around the second peak in the distribution there is



(a) The true proportion of calls recorded at each detector, $\mathbf{Y}$.

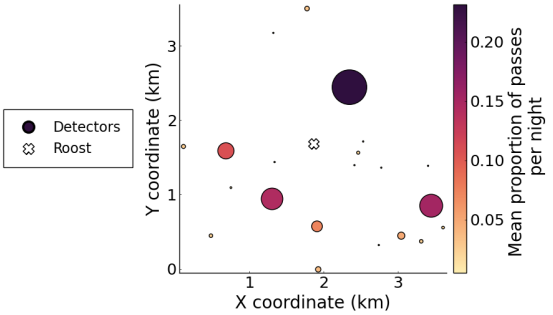


(b) The density of plausible points, shown as a heatmap overlaid on the scatter plot of detectors. The error in roost estimate is 0.79km and the search area is 0.65km^2.

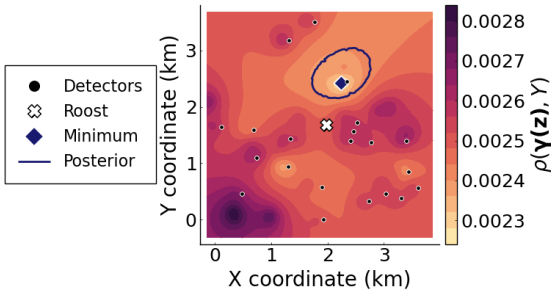


(c) The locations of hedgerows in the survey area. Hedges are coloured red, and markers representing detectors are overlaid.

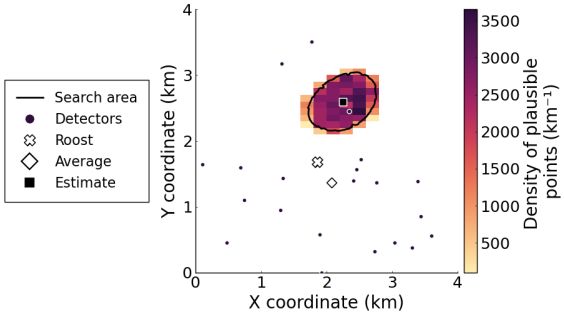
Figure 4.8: The true proportion of calls, $\mathbf{Y}$, the density of plausible points and location of hedgerows for Survey 3, conducted at Buckfastleigh, Devon between 25/07/2016 and 03/08/2016.



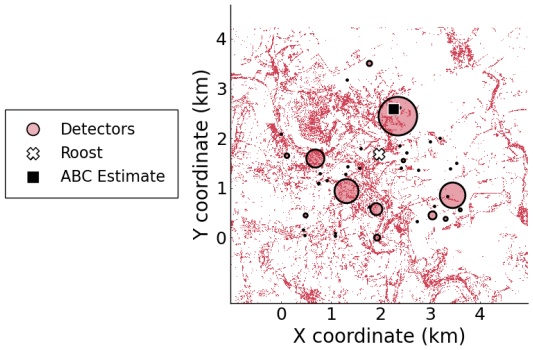
(a) The true proportion of calls recorded at each detector, $\mathbf{Y}$.



(b) A contour plot of the distance metric, $\rho(\gamma(\mathbf{z}), \mathbf{Y})$, calculated for a grid of simulated roost locations, $\mathbf{z}$.



(c) The density of plausible points, shown as a heatmap overlaid on the scatter plot of detectors. The error in roost estimate is 0.64km and the search area is 0.76km^2.



(d) The locations of hedgerows in the survey area. Hedges are coloured red, and markers representing detectors are overlaid.

Figure 4.9: The true proportion of calls, $\mathbf{Y}$, and the density of plausible points for Survey 4, conducted at Gunnislake, Devon between 08/08/2016 and 15/08/2016.

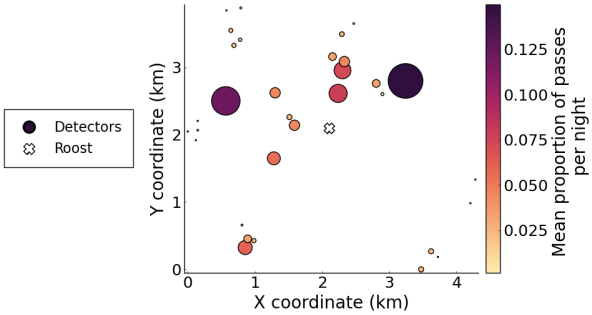
a local minimum in the distance metric.

The results for Survey 6 are shown in Figure 4.11. The roost estimate is located between the detector placed at $(x, y) = (2.65, 5.18)$km and the true roost, and 0.81 km from the true roost. This detector recorded 40% of passes throughout the survey, significantly higher than any other single detector, and it is important to note here that, given the data recorded, an ecologist searching for the roost would most likely assume that the roost is located close to this detector, and therefore in this case the model certainly does not perform worse than a human.

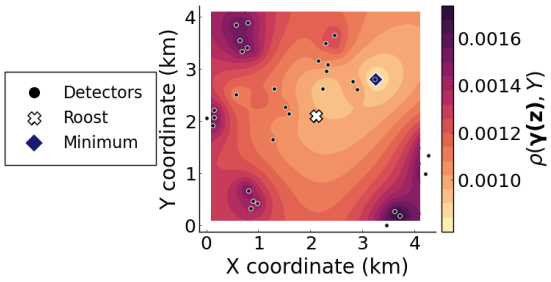
There are a number of possible reasons for the error in estimate for this survey. Firstly, in 2018, a large second roost was found in a cave system close to the Abbey Inn, Buckfastleigh, around 1km north of the main roost. In August 2018, several counts were taken of GHS bats exiting this roost, and on each occasion more than 800 bats were seen emerging. The location of the cave roost is shown, along with the locations of hedgerows in Figure 4.11c, and the roost estimate is only 0.43km from the cave roost. Unfortunately, we cannot know if this roost was used in 2016 (or indeed if there is another, unknown, roost in the area), however it is certainly a possibility that not all of the bats recorded during this survey were from the main roost. The second cave roost was first discovered in August, and may therefore only be used later in the season. Survey 6 was conducted very late in the season, and was the only survey at Buckfastleigh to be conducted after July. It is possible that by September, bats had already moved from the summer roost to the cave roost, or to another hibernation roost by this point in the season.

Another possible contribution to the the error is that the distribution of detectors in Figure 4.1a shows that the lowest distance to the roost is around 1.6km, significantly further than any of the other surveys. As we are assuming that bats move via diffusion, the expected number of bats recorded decreases as we move away from the roost, and therefore the number of passes recorded in this survey is lower than for the other surveys at Buckfastleigh.

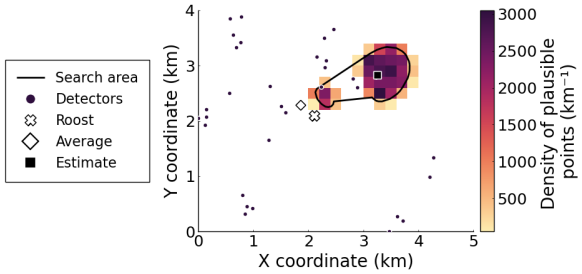
We will discuss possible sources of error in more detail and suggest some improvements in Section 4.



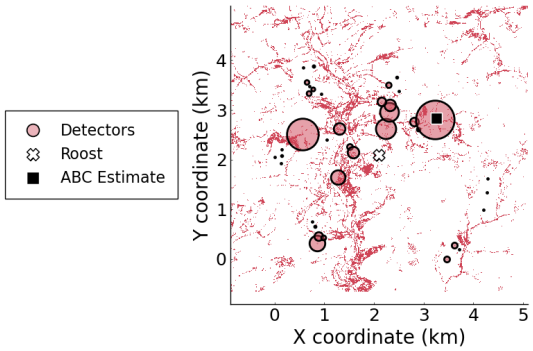
(a) The true proportion of calls recorded at each detector, $\mathbf{Y}$.



(b) A contour plot of the distance metric, $\rho(\gamma(\mathbf{z}), \mathbf{Y})$, calculated for a grid of simulated roost locations, $\mathbf{z}$.

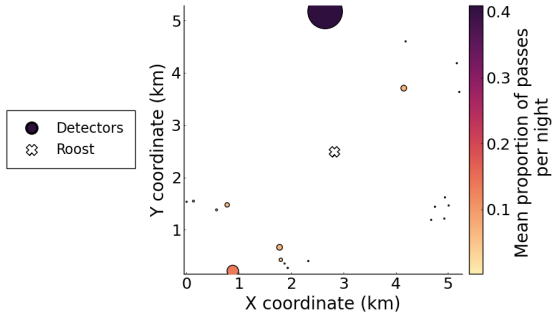


(c) The density of plausible points, shown as a heatmap overlaid on the scatter plot of detectors. The error in roost estimate is 0.81km and the search area is 1.02km^2.

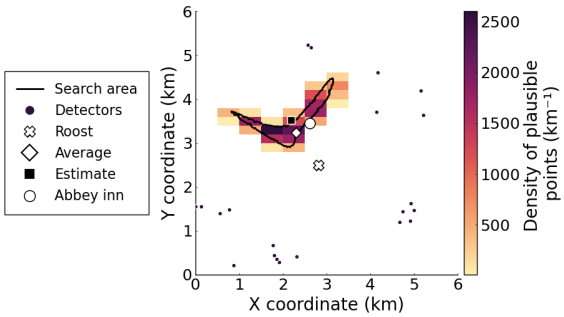


(d) The locations of hedgerows in the survey area. Hedges are coloured red, and markers representing detectors are overlaid.

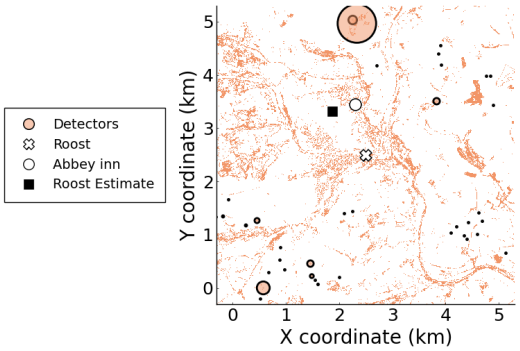
Figure 4.10: The true proportion of calls, $\mathbf{Y}$, and the density of plausible points for Survey 5, conducted at High Marks Barn, Devon between 22/08/2016 and 29/08/2016.



(a) The true proportion of calls recorded at each detector, $\mathbf{Y}$.



(b) The density of plausible points, shown as a heatmap overlaid on the scatter plot of detectors. The error in roost estimate is 0.85km and the search area is 0.65 km^2.



(c) The locations of hedgerows in the survey area. Hedges are coloured red, and markers representing detectors are overlaid.

Figure 4.11: The true proportion of calls, $\mathbf{Y}$, the density of plausible points and location of hedgerows for survey 6, conducted at Buckfastleigh, Devon between 05/09/2016 and 12/09/2016.

Date	Roost	Error in roost estimate (km)
26/6/16	Buckfastleigh	0.21
11/7/16	Braunton	0.33
25/7/16	Buckfastleigh	0.79
8/8/16	Gunnislake	0.64
22/8/16	High Marks Barn	0.81
5/9/16	Buckfastleigh	0.85

Table 4.2: The error in roost estimate, defined as the Euclidean distance between estimate and true roost location, for each of the six surveys.

4 Discussion

We have shown that the method is highly successful for some surveys, whilst it may not provide useful estimates for others. We will discuss a number of reasons for these discrepancies, including differences in behaviour due to the time of year in which the survey was conducted, differences in the landscape around different roosts as well as survey design and imperfections in the survey methods.

All 6 of the surveys discussed here were conducted at maternity roosts, during the summer months. Maternity roosts are populated by females and their young, whilst mature males generally use separate roosting sites and therefore do not disperse from the maternity roost [99]. However, it is not possible to distinguish between the calls of males and females, and therefore calls from males are recorded in the same way, adding noise to the number of passes at each detector. If some detectors are placed particularly close to male roosting sites, this could skew the number of passes recorded and therefore negatively affect the roost estimate. Additionally, it is always possible that there are multiple roosts in the area; as noted with Survey 6 at Buckfastleigh, a second roost was found within the survey area two years later. In this case, we assume that the roost estimate is a weighted average of all present roost locations.

It is important to note that bats do not use the roost and landscape equally throughout the year; during the summer, females use maternity roosts in which they give birth and raise their young. Maternity colonies are generally formed in early summer, from May

onwards. On average, Greater Horseshoes in south-west England give birth between late June and July [99], however birth timing can vary wildly depending on the temperatures throughout Spring and early Summer. Mothers generally care for their pups for around 8 weeks before they are weaned and ready to forage alone. It is therefore highly likely that foraging behaviour changes after this point, as the mother no longer needs to feed her young as well as herself.

For simplicity, we have assumed that each night bats disperse from the roost after sunset regardless of the temperature and weather conditions that night. However, this is not the case. Bats are small animals, using much of their energy to maintain stable body temperatures, and are therefore considered to be particularly sensitive to climatic conditions [100]. Differences in temperature or occurrence of rain or heavy winds are therefore likely to cause changes in foraging behaviour.

Additionally, Greater Horseshoes start mating from late August onwards, and therefore tend to travel further in search of mates and may abandon the maternity roost to move closer to mating sites [101]. It is therefore possible that the diffusion model fit to data from the start of the maternity season in May-June is not generalisable to the end of the maternity season in late August-September. As shown in Figure 4.1b, the number of passes recorded per detector is significantly lower in the September survey than the previous Buckfastleigh surveys, suggesting that the maternity roost could have already been abandoned in favour of a mating site. The error in roost estimate for each survey is shown in Table 4.2. Clearly, the error is higher in the surveys conducted towards the end of the summer than in those conducted in June or July, although with only 6 surveys it is difficult to generalise these results. A second radio tracking survey later on in the season would be extremely useful to verify if bats still move via diffusion, and if so, if the diffusion coefficient varies throughout the summer. However each radio tracking survey requires hundreds of person hours and is therefore beyond the scope of this thesis.

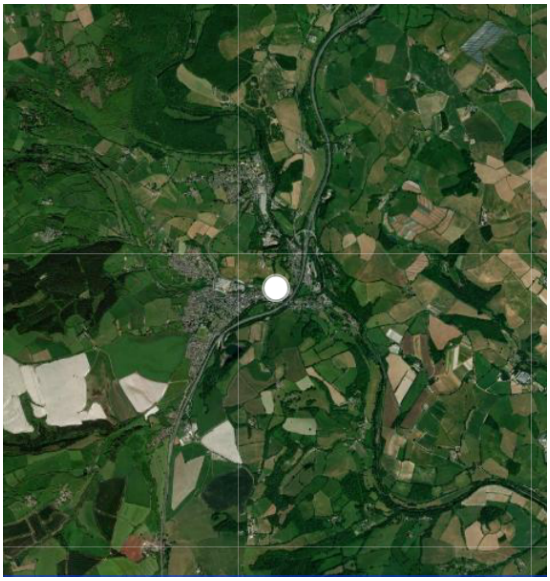
As noted in Section 1, these 6 surveys were conducted to study landscape use around the roost [11]. Detectors were deliberately placed to cover a variety of terrain types, some of which (for example, hedgerows, where insects are abundant and there is cover from

predators) are known to be favourable for bat movement and some of which are detrimental to bat movement (for example, busy roads, where there are bright lights, loud noises and few insects to feed on). Aerial photographs of the Buckfastleigh and Braunton roosts are shown in Figure 4.12. The terrain is clearly not homogeneous in either case, including a mixture of rural farmland and urban areas. It is almost certain therefore that the number of passes recorded at each detector has been affected by the surrounding terrain. It is highly likely that including landscape data in the model would improve roost estimates.

Unfortunately, the addition of this extra information would greatly increase the complexity of the method. Firstly, collecting the data required to map out landscape data is not trivial: the data for different landscape features is collected by different organisations (for example, the locations of roads and buildings can be extracted from Ordnance Survey maps [102], whilst data such as the location of streetlights is often only available on request from local authorities). In addition, much of this data is only available through subscription services that an ecologist searching for a roost may not necessarily have access to. Including extra data and parameters in the movement model would also increase the complexity of the model, and the computation time of a roost estimate. As the aim of this roost estimation method is to decrease the time required to locate roosts, adding extra steps of data collection and increasing the complexity of computation is not necessarily a helpful modification.

A simple way to reduce the effect of landscape effects would be to place all detectors in favourable terrain, for example only on hedgerows. This could have the added benefit of increasing the expected number of recordings at each detector, thereby increasing the quantity of data recorded. However, the number of passes recorded at each detector is affected not just by the landscape surrounding that detector, but also by the functional connectivity between the roost and the detector: bats are unlikely to cross motorways each night to reach a hedgerow, regardless of how many insects are available in the hedgerow. To further improve the validity of the diffusion model, we would need to take into account the complex effects of landscape connectivity on bat movement.

In addition to variations in bat behaviour, there are likely imperfections in data



(a) Aerial photography of the area around the Buckfastleigh roost.

(b) Aerial photography of the area around the Braunton roost.

Figure 4.12: Aerial photography of the landscape around the Buckfastleigh and Braunton roosts. The roost is marked with a white circle.

collection during the surveys. Detectors are placed outdoors, often in locations with public right of way or on farmland. They are therefore open to theft, vandalism and damage caused by animals, as well as subject to limited battery life, software failure and human error due to inaccurate detector calibration or data recording. As a result, some detectors fail before the end of the survey and stop recording, and some detectors are only operational for 1-2 days whilst others work for the full 7 days.

A further imperfection in the methods comes from the use of deterministic models to describe stochastic behaviour. At higher distances from the roost, stochasticity in movement plays a larger part in the number of passes recorded, as illustrated in Figure 4.13 for stochastic simulations of colonies of 9000 bats, chosen to be equivalent to the bat count of the Buckfastleigh roost of approximately 1800 bats over 5 nights (the mean number of nights over which detectors were active in this survey). The Coefficient of Variation (CV) is defined as the ratio of the standard deviation to the mean,

$$CV = \frac{\sigma}{\mu}, \tag{4.16}$$

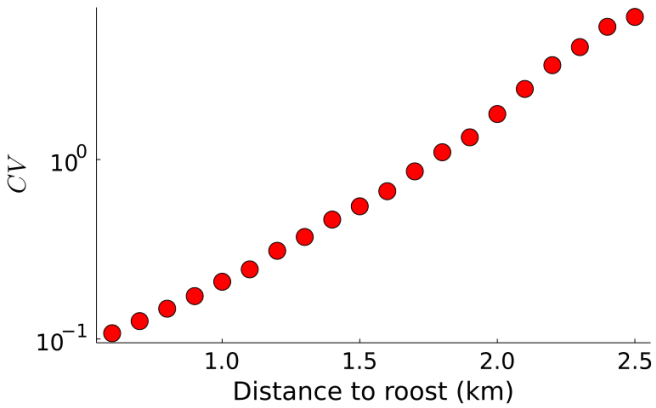


Figure 4.13: The Coefficient of Variation, CV, defined by equation (4.16), where $\sigma = \sigma(\mathrm{z})_i$, and $\mu = \gamma(\mathrm{z})_i$, over 5000 stochastic simulations of bat movement. The stochastic simulation process is described in Section 3.3. Each bat moves according to a random walk, and detectors are checked at intervals of $dt = 10$ seconds: if a bat is inside the range of detector i, a pass is recorded. 30 detectors are placed at intervals of 100m distance from the roost and the simulation parameters are $D = 63.4\mathrm{m}^2\mathrm{s}^{-1}$, $n = 9000$ bats (equivalent to the Buckfastleigh roost count of 1800 bats over the mean number of active nights for detectors of 5 nights) and $T = 90$ minutes.

and shows the extent of variability in relation to the mean. As the distance to the roost increases, the mean number of passes recorded decreases, but the standard deviation in the number of passes increases at a larger rate, and therefore stochasticity plays a larger role in the number of expected passes at higher distances. Additionally, it is important to note that the colony using the Buckfastleigh roost is significantly larger than an average colony, and stochasticity would increase with smaller numbers of bats.

This stochasticity could, in part, explain the discrepancy found in Survey 6. For a detector placed at 1.6km from the roost, as is the closest detector in this survey, the CV is 0.67, meaning that the standard deviation is 67% the size of the mean. At the detector which recorded the highest number of passes, 1.86km from the roost, the standard deviation is 110% the size of the mean. As a result, the uncertainty in the expected number of passes at these further detectors is huge, and adding detectors closer to the true roost could almost certainly improve the estimate. However, in a roost finding survey, the location of the true roost would not be known and we therefore would not know how to place detectors near to the roost.

To examine further the effect of stochasticity, the roost location was estimated using a stochastic simulation, shown in Figure 4.14a. For this roost estimate, ABC was used to fit the roost location parameter to the detector data. The stochastic simulation process described in Section 3.3 was used to generate trajectories for 9000 bats, starting from a particular roost location, $\boldsymbol{z}$. We use a uniform prior for the roost location, $f(z) \sim \mathcal{U}([0,6] \times [0,6])$. Detectors are checked at intervals of $dt = 10$ seconds: if a bat is inside the range of detector i, a pass is recorded. We generate a vector of the proportion of simulated calls at each detector, given roost location $\boldsymbol{z}$, denoted $\boldsymbol{\gamma(z)}$, and the number of passes at each detector recorded in the bat survey is denoted by $\boldsymbol{Y}$. We use the Euclidean distance between simulated and observed data as the distance metric,

$$\rho(\boldsymbol{\gamma}(\boldsymbol{z}), \boldsymbol{Y}) = \|\boldsymbol{\gamma}(\boldsymbol{z}) - \boldsymbol{Y}\|_2. \tag{4.17}$$

A total of 10^4 samples were drawn from the prior distribution, and the posterior was calculated by choosing the 2% of samples with the lowest distance metric, $\rho(\boldsymbol{\gamma}(\boldsymbol{z}), \boldsymbol{Y})$.

There is little qualitative difference between the deterministic and stochastic roost estimates: in both cases, the estimate is to the North West of the true roost, and in a region far from any detectors. This is unsurprising, given that the detector to the North recorded huge numbers of passes, and all of the detectors in the cluster to the South West each recorded small numbers of passes.

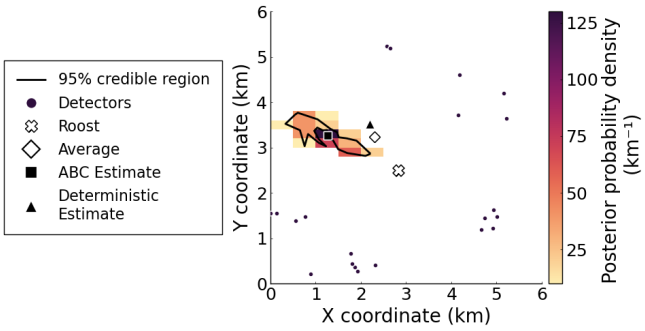
However, the stochastic estimate depends on the number of bats in the roost, since stochasticity increases with fewer bats. When we estimate the roost location using simulations with very few bats, as shown in Figure 4.14b, the results are very different. This time, trajectories for only 50 bats were used in each simulation. The roost estimate is to the South West of the true roost, close to the cluster of detectors in the South West. This is because stochasticity dominates as the number of bats is reduced. When low numbers of bats are simulated, the probability of recording a call at a given detector is significantly reduced, and the majority of samples with roost locations in the North West region surrounding the deterministic estimate do not record any passes at any of the

detectors. Since we use the proportion of calls at each detector as our summary statistics, these samples are clearly unsuitable, and will not be added to the posterior distribution. In contrast, the samples very close to multiple detectors are far more likely to record at least a small number of passes at these detectors. As there are multiple detectors in the South West, which each recorded a significant proportion of calls in the bat survey, the distance metric for samples in this area is low compared to those far from any detectors, which do not record any passes.

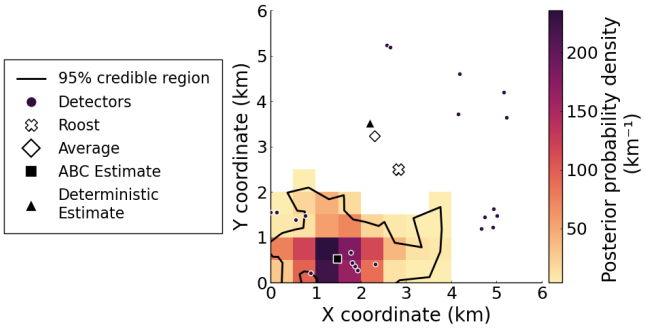
From this analysis, we conclude that stochasticity plays an important role in roost estimates, especially when there are no detectors close to the true roost. Unfortunately, there are disadvantages to using stochastic simulations for estimating roosts. Firstly, as mentioned earlier, stochastic simulations are highly computationally expensive when compared to deterministic simulations: the estimate shown in Figure 4.14a took over 3 hours to calculate with 10^4 samples, compared to just 2 seconds for a deterministic estimate with 10^5 samples. For ecologists looking to reduce the time required to locate a roost, fast calculations are clearly preferable. Whilst 3 hours may not seem a prohibitively long computation time, the target user for these roost finding methods is significantly more likely to use software that gives them an answer in seconds compared to multiple hours. Additionally, few ecologists have access to powerful computers with many cores, and therefore a parallel computing approach may not be usable by a large proportion of target users.

Additionally, estimating the number of bats in a roost may be difficult. As we have shown in Figure 4.14, the number of bats can significantly change the estimates. However, the only way to know the size of a colony is to first find the roost, and then conduct a roost count survey, physically counting each bat inside the roost. Further work into the effect of the roost size would be valuable in improving our understanding of the results of detector surveys, but this is beyond the scope of this thesis.

A more accurate method would take into account all of these variables: landscape effects, time of year, solitary males, weather conditions, stochasticity, roost size and survey design. However, the complexity of the model increases rapidly with the inclusion of



(a) Stochastic roost estimates with 9000 simulated bats.



(b) Stochastic roost estimates with 50 simulated bats.

Figure 4.14: A roost estimate for Survey 6 using stochastic simulations and ABC. For each, 10^4 samples were drawn from the prior, $f(z)$ ($[0,6] \times [0,6]$) for the roost location, and the posterior is chosen as the 2% of samples with the lowest distance metric, $\rho(\boldsymbol{\gamma}(\boldsymbol{z}), \boldsymbol{Y})$. The 95% credible region is calculated as the 95% of the posterior with the lowest distance metric.

each new variable and without huge quantities of data, which would likely be extremely time-consuming and expensive to collect, it is difficult to study the interdependence of each variable.

Next, in Chapter 5, we will aim to improve survey design by using Bayesian Global Optimisation to optimise the placement of detectors, such that the data recorded is the most useful for estimating roost locations. We will discuss an iterative process to update detector locations based on the data collected in initial surveys in order to hone in on the roost location.

Chapter Five

Survey design

In Chapter 4 we showed how we can estimate the location of the roost given data from static detector surveys. The next step is to design surveys such that the information we get from detectors is optimised for finding roosts. The placement of detectors is important, as the proximity of detectors to the roost will determine the number of bats recorded. However, as we do not know the location of the roost before conducting the survey, we need to determine a configuration of detectors that provides the best data for any possible roost location in the area. As the overall goal of these surveys is to locate bat roosts, we can define the optimal detector locations as those that minimise the expected error in roost location estimation.

We will initially use simulated data to design a survey to locate a hypothetical roost as proof of concept. Later we will use the same method to design a survey to estimate the location of a real roost.

1 The detector placement method

The method will have two main steps. First an initial survey to gain some information about the presence of possible roosts in the area followed by a second survey to hone in on the roost location and improve the estimate. As the method is an iterative process, it may be necessary to conduct further surveys after this second survey if more information is required. We assume that we have n detectors available to be placed for each survey. An

overview of the method is detailed below:

1. **Initial survey**

    - We first assume that we have limited information about the location of a possible roost and we aim to design an initial survey to gain more information.
    - Use a uniform prior for the roost location. At this point, we do not have an estimate for the roost location and we therefore assume that the roost is equally likely to be found anywhere in the search area.
    - Use the detector placement algorithm described in Section 1.2 to determine the optimal locations for the n detectors.
    - Conduct the survey. Place detectors at the optimised locations and collect data.
    - Use the data collected from the survey, along with the roost estimation algorithm from Chapter 4, to produce a plausible region for the roost location.

2. **Second survey**

    - Now that we have some information about the location of the roost, we want to hone in closer to the roost, to provide a better estimate. Again, we assume that we have n detectors to be placed, and we want to decide on the best locations to place them given the plausible region calculated from the initial survey.
    - Use the plausible region calculated from the initial survey as the basis for a prior distribution for the placement of detectors during the second survey. For example, we could take a uniform distribution on the plausible region.
    - Use the detector placement algorithm, this time with the new prior, to determine the optimal locations for the n detectors.
    - Conduct the survey. Place detectors at the optimised locations and collect data.
    - Using the data collected from the survey, along with the roost estimation algorithm, to produce a second plausible region for the roost location. This time, we have the data from the n detectors in the initial survey as well as the

data from the n detectors in the second survey, so we can use the data from all $2n$ detectors in the roost estimate.

3. **Further surveys**

   - If more information is required, repeat step 2 to plan further surveys.
   - Each further survey will use the plausible region for roost location calculated using data collected in all previous surveys as a basis for the prior.

Next we will describe the detector placement algorithm.

1.1 The loss function

The goal of the detector placement algorithm is to determine the detector configurations which minimise the expected error in roost location. In order to quantify this, we define a loss function that describes the expected error in roost estimate given a prior distribution for the roost location. We let the roost location be a parameter $\mathbf{z} \in \mathbb{R}^2$. The prior distribution will be denoted $f(\mathbf{z})$ and the location of detectors will be a $2 \times n$ array, $\mathbf{X}$. Using the roost estimation algorithm set out in Chapter 4, given $\boldsymbol{Y}(\boldsymbol{z})$, the proportion of counts at each detector, we can determine the expected error in a roost estimate given a specific $\mathbf{z}$. For $\boldsymbol{Y}(\boldsymbol{z})$ we will use synthetic data generated using a stochastic diffusion process centred at $\boldsymbol{z}$. The roost estimate is denoted $\zeta(\mathbf{X}, \mathbf{Y}(\boldsymbol{z}))$, as defined in equation (4.14). The estimation error is given by the Euclidean distance between $\mathbf{z}$ and $\zeta(\mathbf{X}, \mathbf{Y}(\boldsymbol{z}))$,

$$\rho(\mathbf{z}, \zeta(\mathbf{X}, \mathbf{Y}(\boldsymbol{z}))) = \|\mathbf{z} - \zeta(\mathbf{X}, \mathbf{Y}(\boldsymbol{z}))\|. \tag{5.1}$$

The Euclidean distance chosen here as it is the natural way most people think of distance, particularly in rural areas in the absence of roads. Equally, we could choose the squared distance metric, as chosen for the distance metric for the roost finding method in equation (4.12), however this is not a natural choice to describe distance in the real world, particularly for ecologists without mathematical training.

The loss function is defined as the average expected estimation error over all possible roost locations given roost prior $f(\mathbf{z})$,

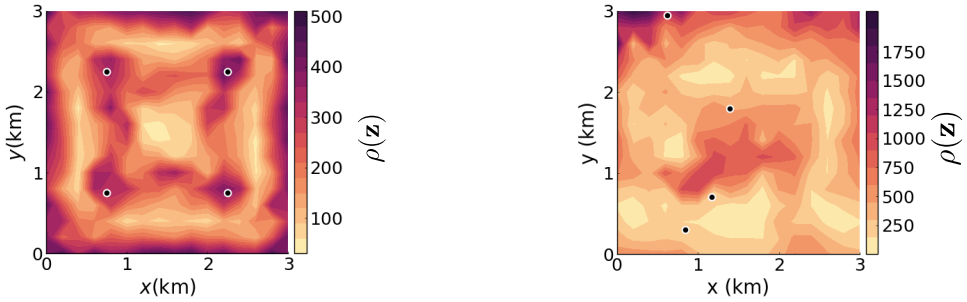
$$L(\mathbf{X}) = \int_{\mathbf{X}} \rho(\mathbf{z}, \zeta(\mathbf{X}, \mathbf{Y}(\boldsymbol{z})))f(\mathbf{z})d\mathbf{z}. \tag{5.2}$$

To evaluate the loss function, we need to calculate the expected error in roost estimate. To do this calculation, we generate synthetic data. $\boldsymbol{Y}(\boldsymbol{z})$, the proportion of counts at detectors at $\mathbf{X}$, given a roost location $\mathbf{z}$, is simulated using a stochastic diffusion simulation. We then use the roost finding algorithm to estimate the location of the simulated roost, and determine the error. The loss function is then calculated by numerically integrating the error over simulated roost locations, $\mathbf{z}$, uniformly placed on a 2D square grid with spacing 200m using Simpson's rule [94]. Therefore, $L(\mathbf{X})$ is a random function, with uncertainty coming from simulations of $\mathbf{Y}$. As such, the calculation of the loss function is repeated 3 times and the mean is taken in order to reduce the impact of noise on the results.

Contour plots for the expected error in roost estimate for 2 different configurations of detectors is shown in Figure 5.1. The error is clearly dependent on the configuration of the detectors, as well as on the location of the hypothetical roost. The loss functions calculated using equation (5.2) for Figure 5.1a and Figure 5.1b give average expected errors in roost estimate of $L(\mathbf{X}) = 215$m and $L(\mathbf{X}) = 482$m, respectively. As such, the location of detectors clearly has an important impact on the error in roost estimate. Next we will describe a procedure to iteratively optimise the detector placement in order to minimise the loss function.

1.2 The detector placement algorithm

We will use a greedy algorithm to determine a detector arrangement [103, 104]. A greedy algorithm is an optimisation approach which iteratively builds a solution by selecting the local optimimum at each iteration. The final solution then often, but not always, approximates the global optimum solution. In this case, we choose a greedy algorithm because the state space for multiple detector locations can be huge, and increases with



(a) The expected error when 4 detectors are spaced evenly in the domain. The loss function was integrated over the domain using equation (5.2), $L(\mathbf{X}) = 215$m.

(b) The expected error when 4 detectors are placed randomly in the domain. The loss function was integrated over the domain using equation (5.2), $L(\mathbf{X}) = 482$m.

Figure 5.1: Contour plots of the estimation error in roost estimate for 2 different detector configurations, corresponding to the initial survey in which we have limited information about the location of a possible roost.

each detector added. Using an iterative process greatly reduces the computational cost by considering only a single location at each iteration.

To implement the detector placement algorithm, at each iteration, we estimate the optimal location to place the next detector, leading to a final detector configuration which approximates a globally optimal solution. First, we define our domain as the square $[0, h] \times [0, h]$. We choose the locations for an initial array of 4 detectors, denoted by $\mathbf{X}$, at $(h/3, h/3)$, $(2h/3, h/3)$, $(h/3, 2h/3)$ and $(2h/3, 2h/3)$. Note that the number of detectors and their location in this initial array can be changed: an ecologist following the method can choose their own locations for the initial array based on their own expertise and knowledge of the local area. We then define a parameter, $\mathbf{x}$, as the location of the next detector, and implement Bayesian global optimisation using the Julia packages Hyperopt.jl and BayesianOptimization.jl to find the value of $\mathbf{x}$ which minimises the loss function [105, 106]. Hyperopt.jl and BayesianOptimization.jl are software packages which together implement the Bayesian global optimisation process set out in Chapter 2.

We define $[\mathbf{X}, \mathbf{x}]$ as the concatenation of $\mathbf{X}$ and $\mathbf{x}$, such that $[\mathbf{X}, \mathbf{x}]$ defines the location of the initial array of detectors $\mathbf{X}$ as well as the next detector $\mathbf{x}$. The objective function to be minimised is given by the loss function for the detector configuration $[\mathbf{X}, \mathbf{x}]$, calculated

from equation (5.2), $f(\mathbf{x}) = L([\mathbf{X}, \mathbf{x}])$. We assume that there are a total of n detectors available to be placed. Pseudocode for the iterative process is shown below:

Set $i = 4$, and $\mathbf{X}$ as the initial array of 4 detectors

Set a budget of function evaluations, N, for each of the $n - 4$ remaining detectors

Set n_0, the initial number of trial points, $\mathbf{x}$ at which we evaluate $f(\mathbf{x})$

while $i < n$ **do**

Define $\mathbf{x}$ as a parameter for the location of the next detector

Place a Gaussian process prior on the objective function $f(\mathbf{x})$, where $f(\mathbf{x}) = L([\mathbf{X}, \mathbf{x}])$

Evaluate $f(\mathbf{x})$ at n_0 trial points, $\mathbf{x}$, distributed uniformly within the domain

Set $j = n_0$

while $j < N$ **do**

Update the conditional distribution on $f(\mathbf{x})$ using the current vector of function values, $f(\mathbf{x}_{1:j-1})$ and equation (2.30)

Calculate the acquisition function using the current conditional distribution and equation (2.45)

Determine f_j^*, the maximum of the acquisition function over $\mathbf{x}_{1:j-1}$, and $\mathbf{x}_j$, the point which gives f_j^*

Evaluate $f(\mathbf{x}_j)$ at the next point $\mathbf{x}_j$

Increment j

end while

Determine the optimum point for the next detector, the point with the minimum posterior mean x_i^*

Add the optimum point to the array of detectors, $\mathbf{X} = [\mathbf{X}, \mathbf{x}_i^*]$

Increment i

end while

Examples of the process are shown for 2 different prior distributions in figures 5.2 and 5.3 for the roost location, uniform and normal, corresponding to the initial survey and second survey respectively. Both were solved with Hyperopt.jl and BayesianOptimization.jl

[105, 106] using Gaussian kernel and constant mean functions. The solution for a uniform prior, $f(\mathbf{z}) \sim \mathcal{U}([0,3] \times [0,3])$, where we take the unit of measurement to be km, is shown in Figure 5.2. In this case, the first detector to be added should be placed at $\mathbf{x_0} = (1.2, 2.9)$. Due to the stochasticity in the loss function, this "best" solution is not the only possible solution. Additionally, it should be noted that there is rotational symmetry in the initial detector array, and therefore the location of this first detector could be rotated around the domain. Further, by adding the 5th point to $\mathbf{X}$, the rotational symmetry of the detector placement is broken. After 10 iterations, detectors are distributed throughout the domain.

The solution for a normal prior, $f(\mathbf{z}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with

$$\boldsymbol{\mu} = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix} \text{ and } \boldsymbol{\Sigma} = \begin{bmatrix} 0.75^2 & 0 \\ 0 & 0.75^2 \end{bmatrix}, \tag{5.3}$$

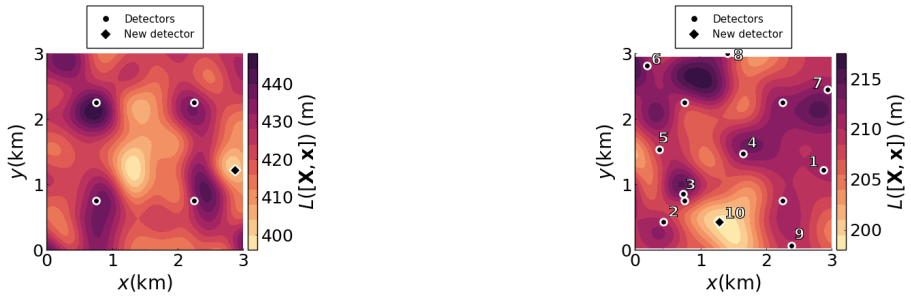
is shown in Figure 5.3. In this case, the algorithm clearly suggests that the first detector should be placed close to the centre of the domain, close to the centre of the prior distribution where the roost is most likely to be. The expected error at each iteration for both priors is shown in Figure 5.4a, showing that in both cases, the error decreases with each iteration. The error is always lower for the normal prior because we already have some idea of where the roost is.

A comparison of the distributions in distance from each detector to the centre of the domain is shown in Figure 5.4b. This shows that for the normal prior, the algorithm places detectors closer to the centre of the domain and the centre of the prior distribution, because placing detectors closer to the roost reduces the error in roost estimate.

Next we will extend the method to account for the inhomogeneity of landscape, ensuring that the algorithm only selects viable detector locations.

2 Incorporating landscape features

Landscape is generally not homogeneous, and bats have preferences for the landscape features they tend to travel through. Therefore, it is preferable to place detectors in



(a) The loss function at the first iteration $L([\mathbf{X}, \mathbf{x}])$ for an initial detector array $\mathbf{X}$ when adding a new detector at location $\mathbf{x} = (x, y)$.

(b) The loss function at the tenth iteration $L([\mathbf{X}, \mathbf{x}])$ for a detector array when adding a new detector at location $\mathbf{x} = (x, y)$. The iteration number of each detector is labelled.

Figure 5.2: The detector placement for a uniform prior after 1 and 10 iterations. GP regression was used to interpolate between samples. Detectors are shown as black circles outlined in white, and the minimum of the loss function is shown as a black diamond.

locations where bats are likely to travel to maximise the number of recordings collected from a survey. Additionally, some landscape features are more difficult to survey due to access issues. We can use GIS (geographic information system) data to determine the locations of undesirable features to avoid. GIS databases contain geographic data, and can be used to generate maps of man-made structures such as buildings and roads, as well as natural landscape features such as rivers and forests.

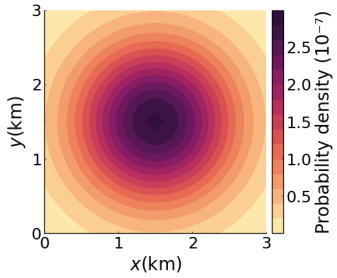
The features that are undesirable for bats include:

1. Streetlights. Bats avoid bright lights, so we would expect that detectors close to streetlights would record fewer bats.

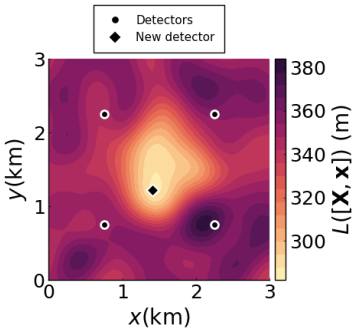
2. Motorways and A roads. Bats avoid busy roads due to bright lights and noise. These roads are also difficult to place detectors on.

3. Lakes. It is generally not possible to place detectors in a lake.

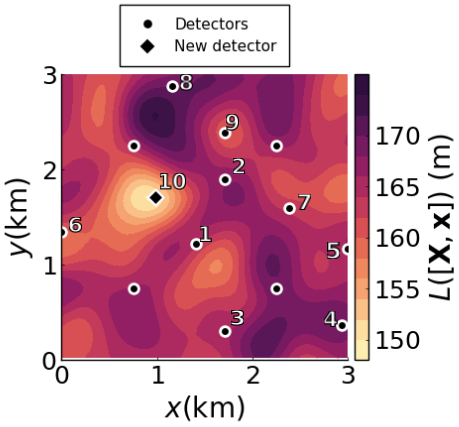
4. Buildings. Most buildings are privately owned, and therefore placing detectors on, or very close to, buildings requires permission from the owner. Seeking permission can be difficult, time-consuming, and often not possible.



(a) The probability density function for a normal prior, $f(\mathbf{z}) \sim \mathcal{N}(1.5\text{km}, 0.75\text{km})^2$, truncated to $[0, 3] \times [0, 3]$.

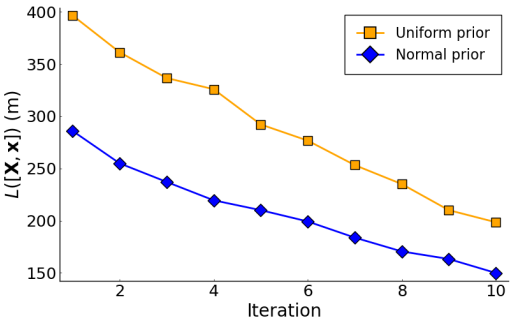


(b) The loss function at the first iteration $L([\mathbf{X}, \mathbf{x}])$ for an initial detector array $\mathbf{X}$ when adding a new detector at location $\mathbf{x} = (x, y)$.

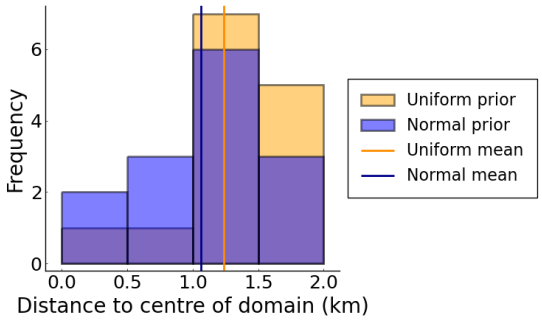


(c) The loss function at the tenth iteration $L([\mathbf{X}, \mathbf{x}])$ for a detector array when adding a new detector at location $\mathbf{x} = (x, y)$. The iteration number of each detector is labelled.

Figure 5.3: The detector placement for a normal prior after 1 and 10 iterations. GP regression was used to interpolate between samples. Detectors are shown as black circles outlined in white, and the minimum of the loss function is shown as a black diamond.



(a) The reduction in expected error as each detector is iteratively added to the array.

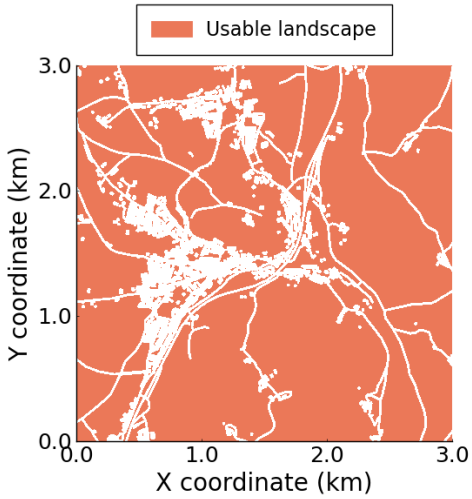


(b) Histograms of the distance from each detector to the centre of the domain. The distributions for uniform and normal priors are overlaid. The mean for each distribution is shown as a solid vertical line.

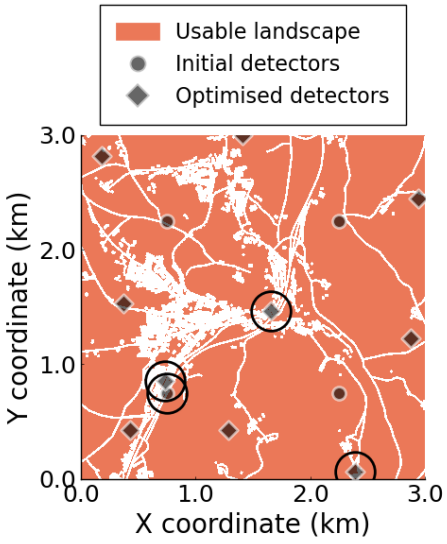
Figure 5.4: Comparing uniform and normal prior distributions.

Using GIS layers for each landscape feature, we generate a mask which describes the presence of features, where any grid square within a buffer distance of 10m of an undesirable feature is assigned a value of ∞ and all other grid squares are assigned a value of 1. An example of a mask generated this way is shown in Figure 5.5a. We can then multiply this mask by a matrix calculated from the loss function. Then, all locations containing undesirable features are assigned a loss value of ∞ and therefore will not be chosen when searching for the location with minimum loss. The detector configuration determined for a uniform prior from Figure 5.2 is shown overlaid on the landscape mask in Figure 5.5b. In this case, one of the initial four detectors is located in an unusable location, and the algorithm suggests placing a further three detectors on unusable locations if the landscape mask is not used. To rectify this, first we move the unusable initial detector to the nearest usable location. In this case, the detector moves 43m to the East. Since changing the location of the initial detectors changes the loss function, the location of detectors placed in future iterations will also change.

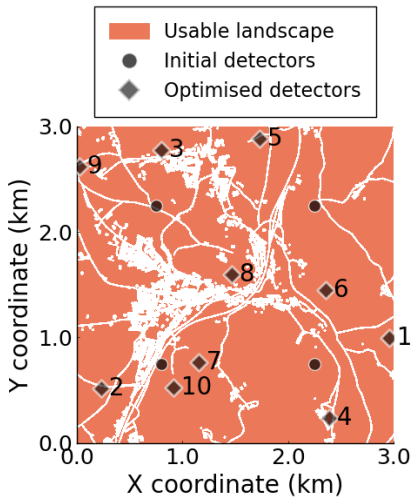
We run the detector placement algorithm once again, this time incorporating the landscape mask when calculating the loss function. The final detector configuration is shown in Figure 5.5c. Detectors are located throughout the domain as before, but this time none are placed in unusable locations.



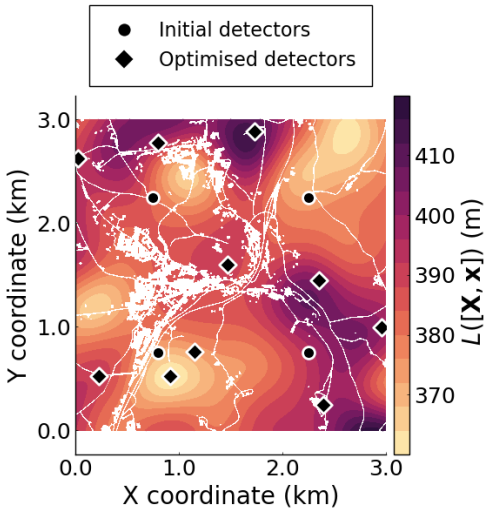
(a) A landscape mask showing the presence of streetlights, roads, lakes and buildings.



(b) The detector configuration determined for a uniform prior from Figure 5.2 overlaid on the landscape mask. Four detectors, including one of the initial four, are placed on unusable landscape, and are marked with a circle.



(c) The detector configuration determined for a uniform prior when avoiding undesirable features determined by the landscape mask.



(d) The final loss function at the 10th iteration.

Figure 5.5: Landscape mask, loss function and detector placement for a domain with undesirable landscape features.

Next, we will discuss an application of the detector placement algorithm, at a survey site at the Goodwood Estate in Sussex.

3 Survey at Goodwood, Sussex

To test the detector placement method, a survey was conducted at the Goodwood Estate in West Sussex between 21/08/2021 and 24/08/2021. This area is dominated by woodland, and therefore the majority of the space is unsuitable for placing detectors due to access issues. Additionally, as noted in Section 1, many bat species prefer to travel along linear features, and therefore more bat calls are likely to be recorded along trails than elsewhere in woodland. As a result, possible detector locations are restricted to footpaths and trails through the woodland.

To generate a landscape mask, the trails were manually selected from Ordnance Survey maps and converted to a GIS raster layer. A buffer of 10m around the selected locations was added to account for any inaccuracies in trail selection and ensure that all parts of the trail are included in the layer. An image of the final landscape mask is shown in Figure 5.6a.

Note that the geometry of the usable areas appears similar to a network graph, and it is likely that graph optimisation algorithms would provide a more efficient framework for optimising detector placement on a graph-like landscape. For this particular landscape mask, the usable areas could be converted from a two-dimensional search space to a one dimensional network, reducing the state space of possible detector locations. Graph optimisation algorithms are often used for integer programming problems, including sensor placement problems [107, 108]. Integer programming describes a mathematical optimisation problem in which variables are restricted to integers. The special case of binary integer programming can describe placement problems in which all nodes on the network are assigned values of 0 (no sensor) or 1 (sensor). There are a number of techniques for solving integer programming problems [109], of which one of the most common is the branch-and-bound algorithm [110]. The branch-and-bound algorithm consists of a state space search, in which the set of candidate solutions is split into subsets or branches. The objective function is then minimised separately on each branch, and those branches that do not produce a better solution than some estimated bound on the optimal solution are

discarded.

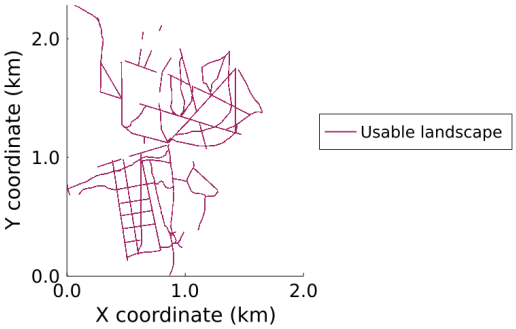
Similar methods based on graph optimisation could be used here to optimise detector placement on graph-like landscapes. However, these methods often involve some variation on a state-space search, and with many detectors, the state-space is large. Additionally, these methods would likely only be useful on landscapes which can be reduced to a graph-like path. For a survey in an area where the majority of the landscape is usable, for example, the mask shown in Figure 5.5a, the landscape is better represented as a 2D region. As a result, a graph optimisation algorithm would not be suitable for all landscapes.

For this survey, a total of 28 detectors were available. We start with an initial array of 7 detectors, chosen manually by an ecologist, shown in Figure 5.6b. Detectors were placed at the intersections of paths, as these are generally considered to be the best locations because there are more chances for recording bats, as bats travelling along any of the intersected paths will eventually reach the intersection. The chosen locations were placed on points where multiple paths intersect, and spread out throughout the survey area.

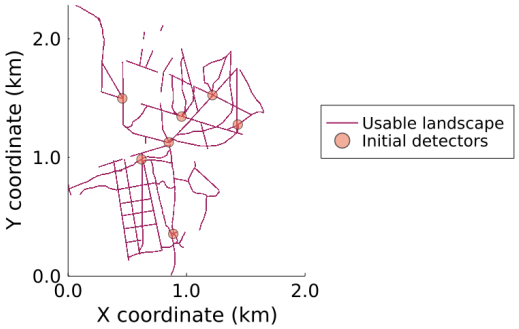
The iterative process described in Section 1.2 was used to determine the locations of the remaining 21 detectors. The loss surface at the first iteration is shown in Figure 5.7. The area with the maximum loss is located in the centre of 4 of the initial detectors, whilst the minimum loss is found in the south west, far away from the initial detectors. The point with the minimum loss function moves when including the locations of tracks and trails, as shown in Figure 5.6b. The final planned detector configuration is shown in Figure 5.8, and includes detectors spread across the domain.

3.1 Survey results

Every attempt was made to place detectors in the planned locations, but in some cases detectors were placed at slightly different locations due to differences in the locations of trails or a lack of suitable locations to place detectors. One of the 28 detectors failed during the survey, and as such this detector was excluded from the analysis. The planned and actual detector locations are shown in Figure 5.9.

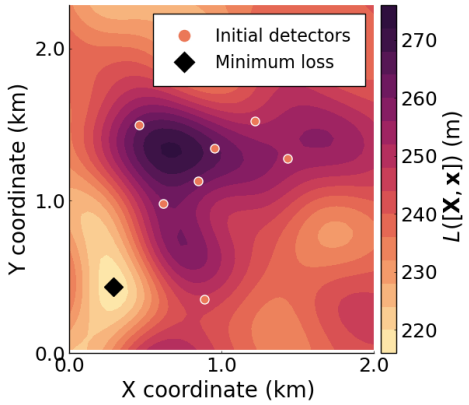


(a) The landscape mask calculated from the locations of tracks and trails in the survey area.

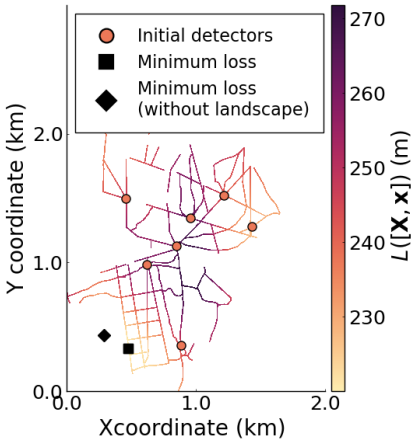


(b) The locations of the initial 7 detectors superimposed over the landscape mask.

Figure 5.6: The landscape mask and initial 7 detector location for a survey at Goodwood, Sussex.



(a) The loss function, $L([\mathbf{X}, \mathbf{x}])$, at the first iteration for a detector array when adding a new detector at location $\mathbf{x} = (x, y)$.



(b) The loss function shown in Figure 5.7a multiplied by the landscape mask as shown in Figure 5.6a.

Figure 5.7: The loss function, $L([\mathbf{X}, \mathbf{x}])$, at the first iteration given 7 initial detectors.

Multiple species were recorded throughout the survey, and the count over the first 90 minutes from sunset for each species recorded is shown in Figure 5.10a. A total of 15 species were identified, whilst in 215 recordings the species could not be identified (due to the recording being corrupted or distorted by noise) and so is labelled X. The count for the top 4 species, for which more than 50 calls were recorded during the first 90 minutes, is shown in Figure 5.10b. The most recorded species by far is the Common Pipistrelle (Pippip), followed by Soprano Pipistrelle (Pippyg), Serotine bat, (Eptser) and Western Barbastelle (Barbar). Very few calls were recorded for the other species. The roost finding

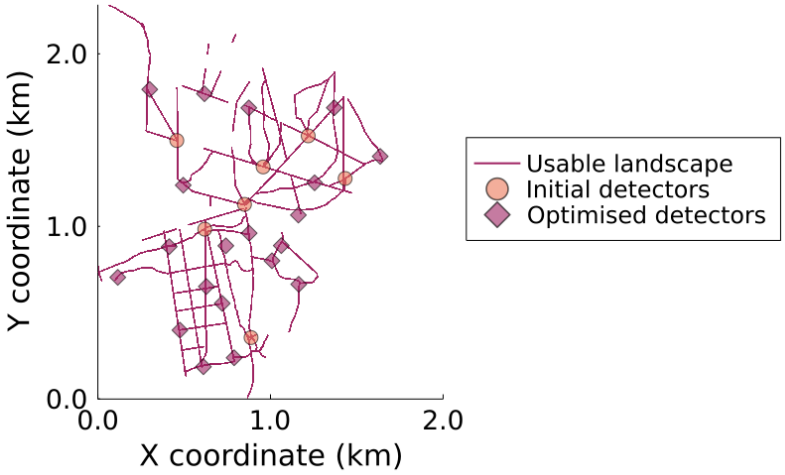


Figure 5.8: The initial 7, manually chosen, detector locations and final 21 detector locations chosen by optimisation.

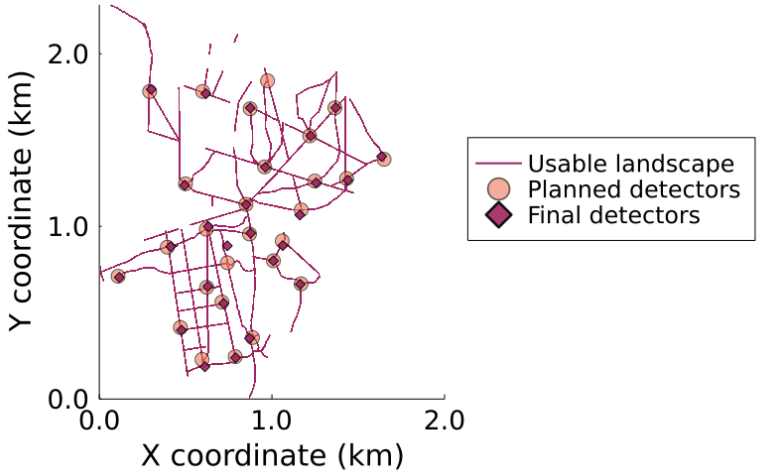
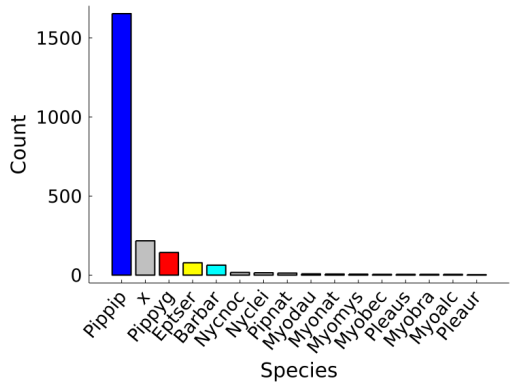


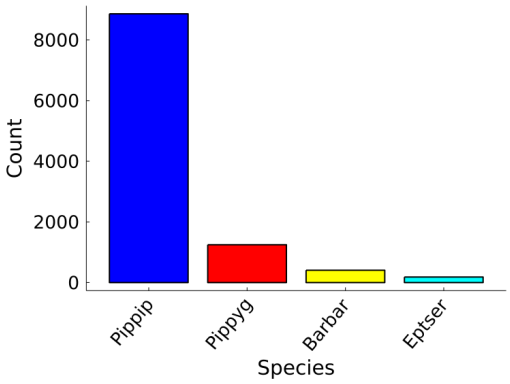
Figure 5.9: The locations of the 28 planned and the 27 actual detector locations.

method will be used to estimate the roost locations for the top 4 species and these results will be compared to an estimate assuming that all species are present in the same roost.

The results for all species grouped together are shown in Figure 5.11. Two detectors recorded significantly more calls than the other detectors, and the plausible region, shown in Figure 5.11b, is therefore clustered around these two detectors. Since this estimate uses data from 15 species, each of which has different roosting preferences, there will be multiple roosts in this area. However, the roost estimate assumes that only one roost is present, and we therefore assume that it estimates the average location of all roosts

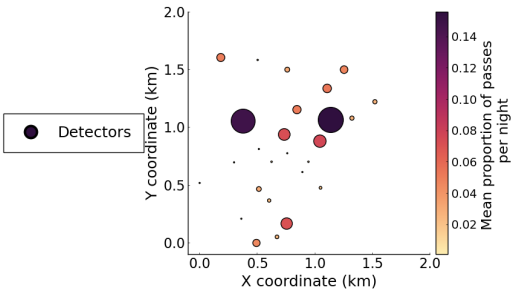


(a) All species recorded during the survey.

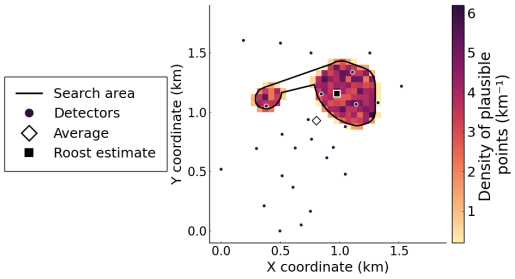


(b) The top 4 species, for which more than 50 calls were recorded during the first 90 minutes.

Figure 5.10: Count of each species over first 90 minutes of the night from sunset for a survey at Goodwood, Sussex between 21/08/2021 and 24/08/2021. Species are classified by a 6 letter code, as follows: Pippip = Common Pipistrelle, X = Species unidentified, Pippyg = Soprano Pipistrelle, Eptser = Serotine bat, Barbar = Western Barbastelle, Nycnoc = Noctule bat, Nyclei = Leisler's bat, Pipnat = Nathusius Pipistrelle, Myodau = Daubenton's bat, Myonat = Natterers bat, Myomys = Whiskered bat, Myoalc = Alcathoe's bat, Myobec = Bechstein's bat, Myobra = Brandt's bat, Pleaus = Grey long-eared bat and Pleaur = Brown long-eared bat.



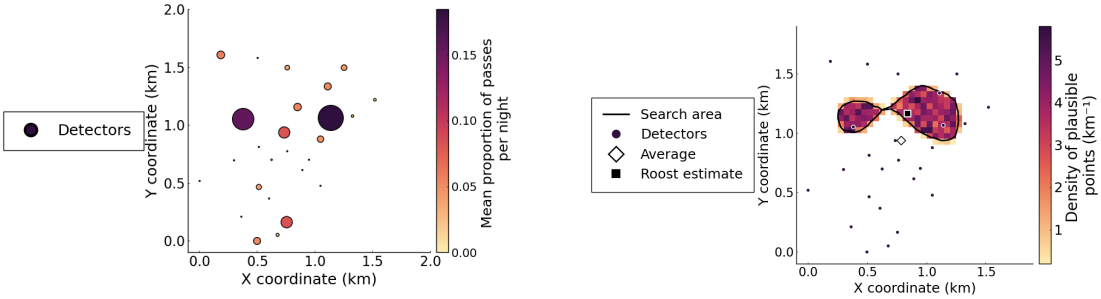
(a) The true proportion of calls recorded at each detector, $\mathbf{Y}$.



(b) The plausible region, shown as a heatmap overlaid on the scatter plot of detectors. The plausible region is defined as the 2% most likely subset of the search region.

Figure 5.11: The true proportion of calls, $\mathbf{Y}$, and the plausible region for all species recorded.

present, weighted by the size of the roosts.



(a) The true proportion of calls recorded at each detector, $\mathbf{Y}$.

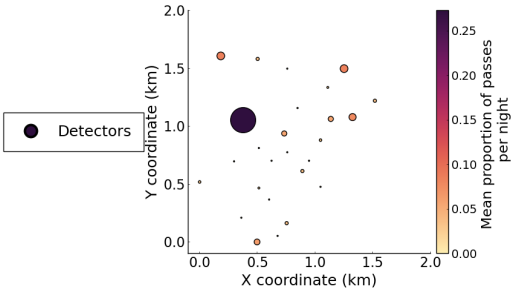
(b) The plausible region, shown as a heatmap overlaid on the scatter plot of detectors. The plausible region is defined as the 2% most likely subset of the search region.

Figure 5.12: The true proportion of calls, $\mathbf{Y}$, and the plausible region for Common Pipistrelles (Pippip).

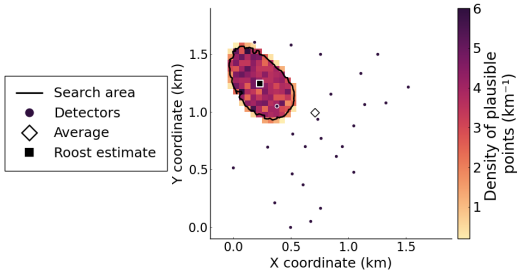
The results for Common Pipistrelles, shown in Figure 5.12, are qualitatively very similar to the case with all species, an unsurprising result given that the majority of calls recorded were from Common Pipistrelles. Common Pipistrelles are highly adaptable and their roosts are generally found in crevices in a variety of structures, either in buildings or trees [93].

As with the previous case, there are two detectors with significantly higher numbers of calls than the others, and therefore two distinct regions in the plausible region. In fact, Pipistrelle bats (both Common Pippistrelles and Soprano Pipistrelles) are known to form 'fission-fusion' societies, in which colonies are split between one or two main roosts and a variety of smaller roosts found close to one another, and individual bats switch between each of these roosts over a number of nights [111, 112]. Colonies of Common Pipistrelles have been found to 'fission', or break apart, when a small proportion of the colony decides to move to the next roost. Over the next few days, the two split colonies then 'fuse' together as the rest of the colony follows, moving to the new roost. As a result of these roost switching events, it is therefore possible that the Common Pipistrelle colony used more than one roost throughout the survey.

The results for the next 3 most recorded species are, however, very different. The results of the survey for Soprano Pipistrelles are shown in Figure 5.13. There was one detector



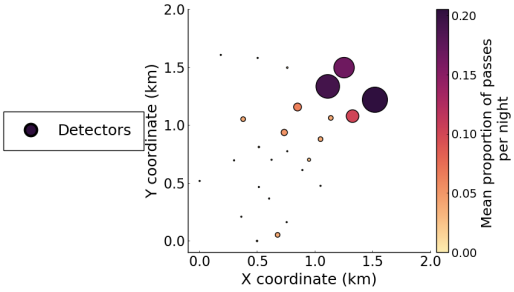
(a) The true proportion of calls recorded at each detector, **Y**.



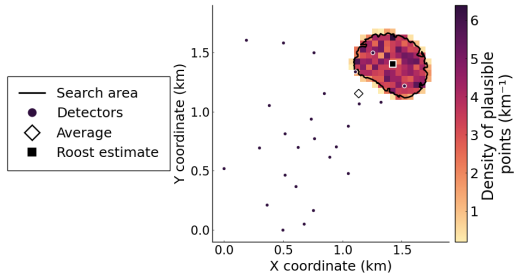
(b) The plausible region, shown as a heatmap overlaid on the scatter plot of detectors. The plausible region is defined as the 2% most likely subset of the search region.

Figure 5.13: The true proportion of calls, **Y**, and the plausible region for Soprano Pipistrelles (Pippyg).

that recorded significantly more bat passes than the rest, and unsurprisingly the plausible region is located close to this point. Soprano Pipistrelles and Common Pipistrelles are generally considered to have similar roosting characteristics, although the two species are unlikely to share roosts, and Soprano Pipistrelle roosts could therefore be found anywhere within the survey area [93].



(a) The true proportion of calls recorded at each detector, **Y**.



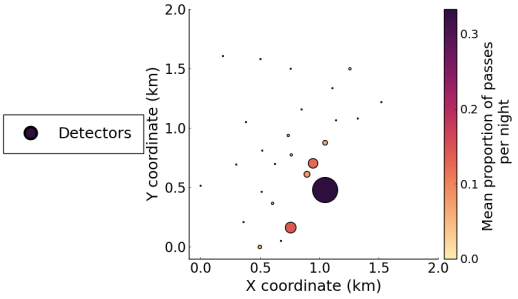
(b) The plausible region, shown as a heatmap overlaid on the scatter plot of detectors. The plausible region is defined as the 2% most likely subset of the search region.

Figure 5.14: The true proportion of calls, **Y**, and the plausible region for Serotine bats (Eptser).

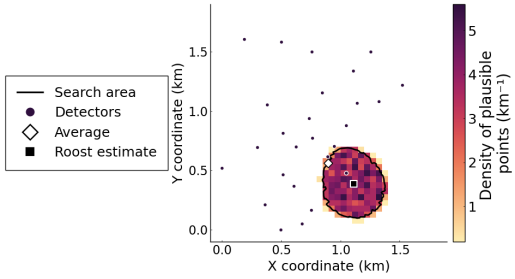
The results for Serotine bats, shown in Figure 5.14, show that the majority of Serotine calls were found at the north east of the survey area. The plausible region is located close to the most north-easterly detectors, partly outside of the survey area. This result suggests

that the Serotine roost may be found outside of the woodland in the Goodwood estate. In fact, Serotine bats often choose to roost in buildings during the summer [113], but they have a preference to roost close to woodland: 90% of Serotine roosts in the south of England are found within 440m of broadleaved woodland [114]. It is likely that the Serotine roost is found in buildings close to the border of the estate.

The results for Western Barbastelles, shown in Figure 5.15, show that the vast majority of calls, and therefore also the plausible region, are found in the south east corner of the survey area. Western Barbastelles prefer to hunt in wooded areas, and roosts are usually found in tree crevices and bat boxes in woodland, and therefore Barbastelle roosts could be found anywhere within the survey area.



(a) The true proportion of calls recorded at each detector, $\mathbf{Y}$.



(b) The plausible region, shown as a heatmap overlaid on the scatter plot of detectors. The plausible region is defined as the 2% most likely subset of the search region.

Figure 5.15: The true proportion of calls, $\mathbf{Y}$, and the plausible region for Western Barbastelles (Barbar).

3.1.1 Sensitivity to changes in the diffusion coefficient

Next, we will look at the effect of the diffusion coefficient on the results. Until now we have used a diffusion coefficient of $D = 63.4\text{m}^2\text{s}^{-1}$, as this was the value calculated during radio tracking surveys for Greater Horseshoe bats in Section 4. The flight speed of bats is known to vary between species [93], and therefore the diffusion coefficient is also likely to vary between species. Additionally, the radiotracking survey in Chapter 3 and all 6 surveys considered in Chapter 4 were all conducted in similar landscapes, dominated by

farmland and hedgerows and with little woodland. However, in this survey, we are looking at data from 4 different species, in a different area, in which the landscape was dominated by woodland.

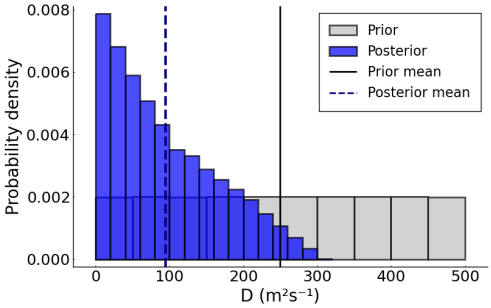
To study the effect of the diffusion coefficient on roost estimates, we will add the diffusion coefficient as an extra parameter to estimate in the roost finding algorithm and we will use a stochastic diffusion model, fit to the survey data using ABC. In each case, we will use uniform distributions for the prior for D and for the roost location, and assume a colony size of 1000 bats.

Firstly, we will look at the results for common Pipistrelles, shown in Figure 5.16.

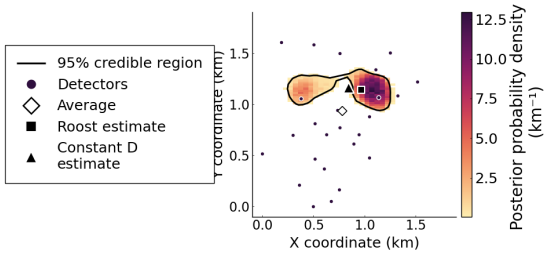
The prior for D was uniform, $D \sim \mathcal{U}(0, 500)\text{m}^2\text{s}^{-1}$, and the prior and posterior distributions are shown in Figure 5.16a. The posterior distribution is clearly skewed towards lower values of D when compared to the uniform prior, and the posterior mean, $D = 94\text{m}^2\text{s}^{-1}$, is much lower than the prior mean, $D = 250\text{m}^2\text{s}^{-1}$, although still higher than the Greater Horseshoe calculated value of $63.4\text{m}^2\text{s}^{-1}$. The posterior distribution for the roost location, however, is qualitatively similar to the distribution calculated for constant D in Figure 5.12b; there are 2 peaks in the distribution, located around the two detectors with the highest number of calls. The eastern peak shows a higher probability density than the Western peak. The joint probability distributions for D and the roost location, $\boldsymbol{z}$ are shown in Figure 5.16c. We can see that the majority of samples with high diffusion coefficients, $D > 120\text{m}^2\text{s}^{-1}$, correspond to samples in the eastern peak.

For Soprano Pipistrelles, the posterior distribution for D is highly skewed towards 0 when compared to the prior distribution of $D \sim \mathcal{U}(0, 600)\text{m}^2\text{s}^{-1}$, as shown in Figure 5.17a. Likewise, the roost distribution, shown in Figure 5.17b, is skewed towards the centre of the survey area. The joint distributions for D and the roost location, $\boldsymbol{z} = (x, y)$, are shown in Figure 5.17c. The peak in the roost posterior, in the south east of the posterior region, corresponds to samples with high D, whilst the long tail of the roost posterior, the region to the north west with low probability density, corresponds to samples with low D, $D < 40\text{m}^2\text{s}^{-1}$.

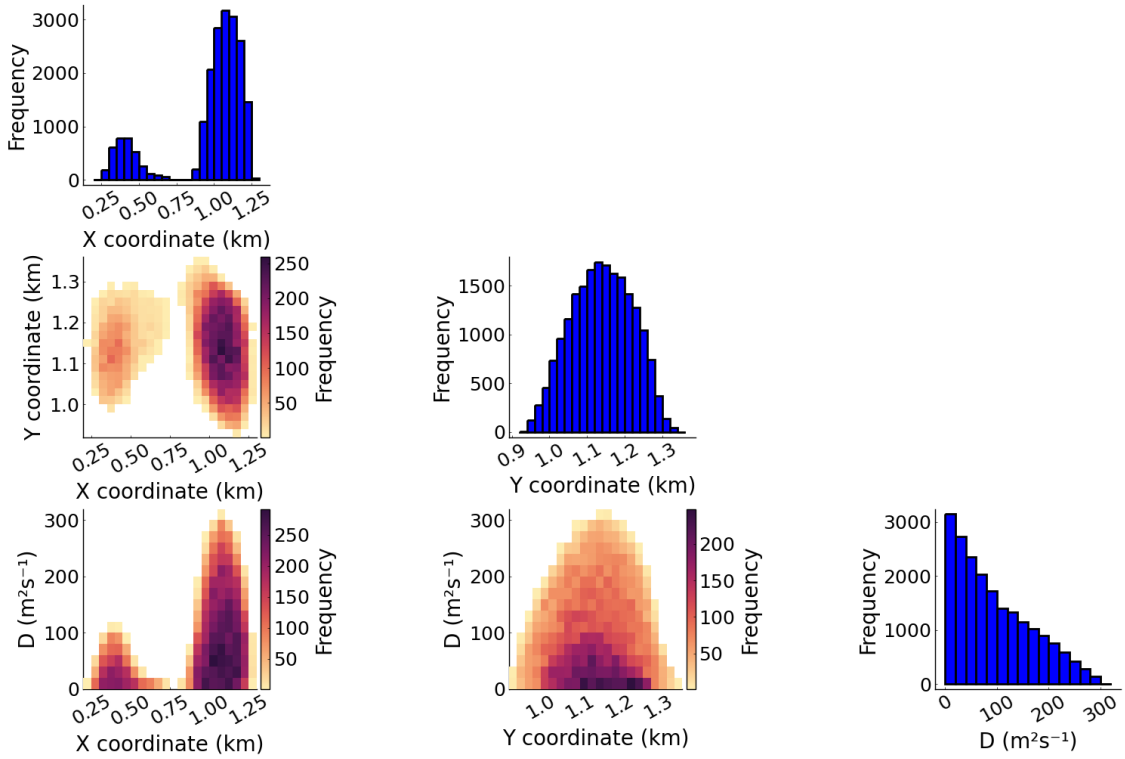
The posterior distributions for Serotine bats are shown in Figure 5.18. Clearly, the



(a) The prior and posterior distributions for the diffusion coefficient, D.

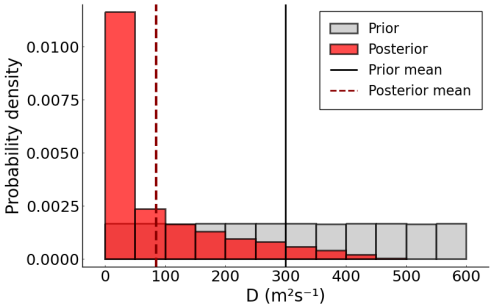


(b) The posterior probability distribution, shown as a heatmap overlaid on the scatter plot of detectors. The 95% credible region has an area of 0.22km^2.

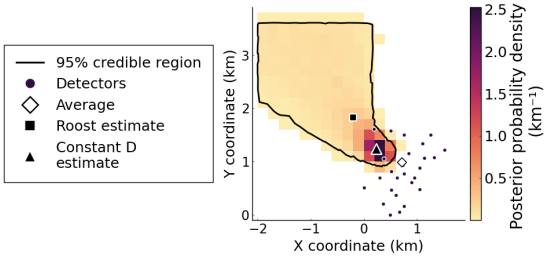


(c) The joint posterior distributions for diffusion coefficient, D, and the roost location.

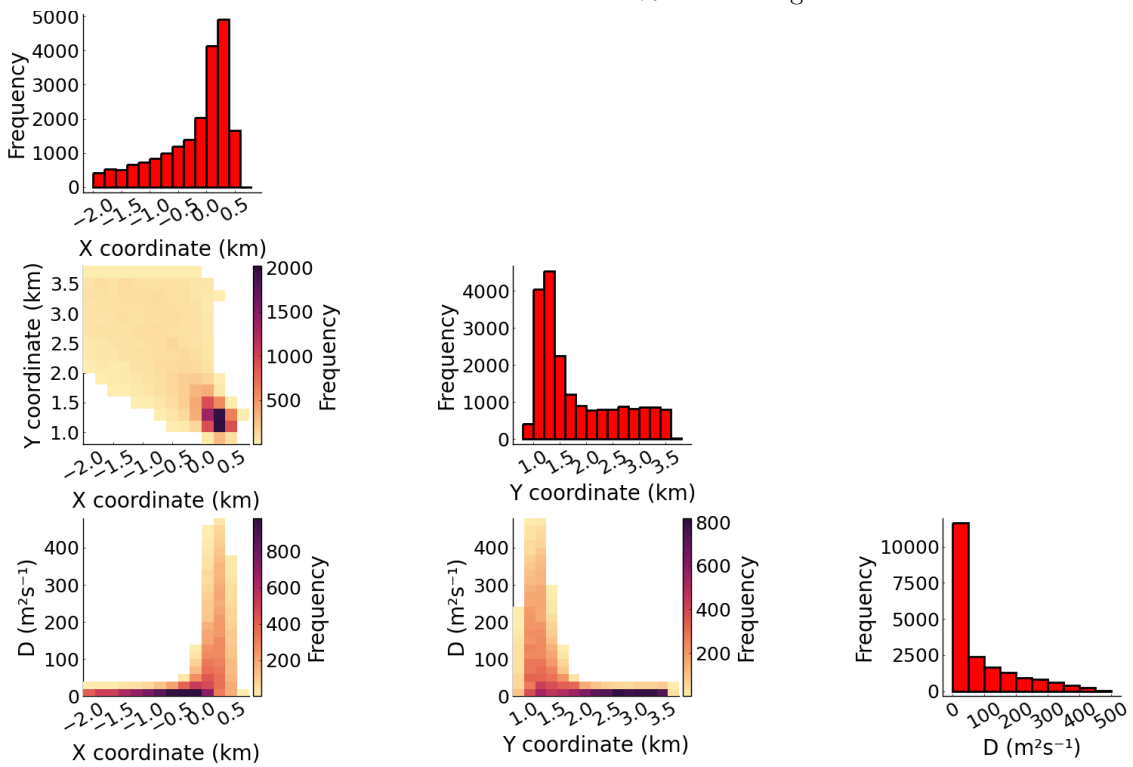
Figure 5.16: The posterior distributions for diffusion coefficient, D, and the roost location for a roost finding model with D as a parameter for Common Pipistrelles (Pippip).



(a) The prior and posterior distributions for the diffusion coefficient, D.



(b) The posterior probability distribution, shown as a heatmap overlaid on the scatter plot of detectors. The 95% credible region has an area of 4.75km^2.



(c) The joint posterior distributions for diffusion coefficient, D, and the roost location.

Figure 5.17: The posterior distributions for diffusion coefficient, D, and the roost location for a roost finding model with D as a parameter for Soprano Pipistrelles (Pippyg).

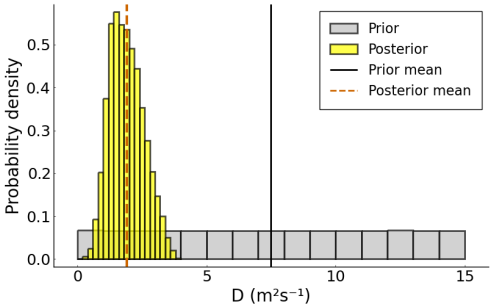
posterior distribution for the diffusion coefficient is much smaller than for the Pipistrelle species, as all samples have a diffusion coefficient of $D < 5\text{m}^2\text{s}^{-1}$. The posterior 95% credible region includes the region identified using fixed D in Figure 5.14, but this time the credible region covers a larger area, spread out further towards the north east. There is no clear peak in the probability density, instead the probability density is fairly even over the area. The joint distributions of D and roost location are shown in Figure 5.18c. These distributions show that the samples with higher values of D are found at locations further to the north east, and further away from the detectors.

The results for Barbastelles, shown in Figure 5.19, show that again the distribution for D is skewed towards low values, and the posterior distribution for the roost location is spread out over a larger area and further from the detector locations than for the constant D case in Figure 5.15. There is a peak in the probability density around the location of the estimate calculated with constant D, which from Figure 5.19c we can see is from the samples with higher D, whilst the long tail in the roost distribution comes from only the samples with $D < 10\text{m}^2\text{s}^{-1}$.

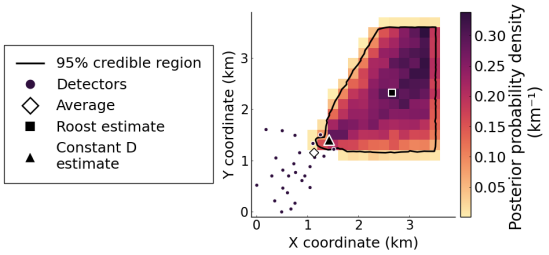
We have shown that for all species, the roost estimates are sensitive to the diffusion coefficient, D. The three species for which the roost estimate is outside, or close to the edge of, the detector arrays, Soprano Pipistrelles, Serotines and Barbastelles, show a clear relationship between D and $\boldsymbol{z}$ in the posterior distributions, with higher D corresponding to a larger distance between the sample $\boldsymbol{z}$ and the detector arrays. When we look at the expected number of calls at detector i (from equations (4.9) and (4.10)),

$$N_i(T) \approx \int_{t=0}^{t=T} \frac{1}{4Dt} e^{-\frac{x_i^2+y_i^2}{4Dt}} \left[R^2 + \frac{R^4}{4} \left(\frac{x_i^2+y_i^2}{8D^2t^2} - \frac{1}{2Dt} \right) \right] n dt, \tag{5.4}$$

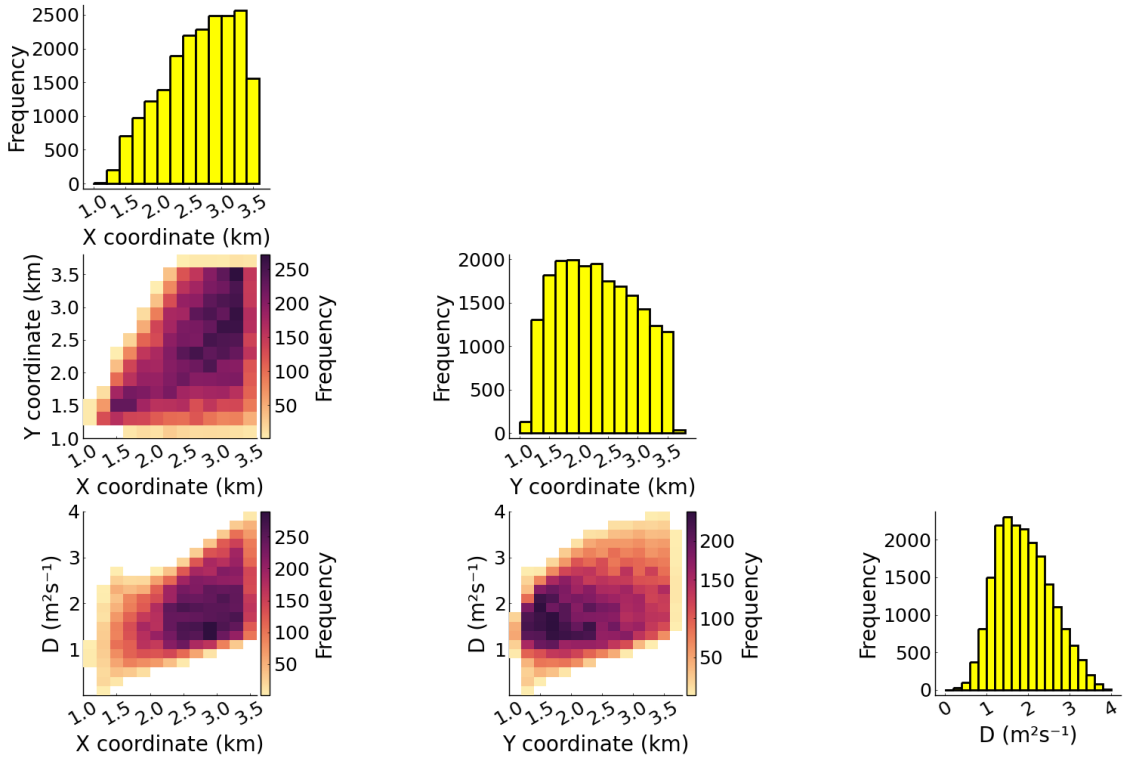
we see that the number of calls is dependent on both the distance to the roost and the diffusion coefficient. The expected proportion of calls for 3 simulated detectors with varying simulated roost locations and diffusion coefficients are shown in Figure 5.20. In each case, the proportion of calls at each detector is nearly identical, showing that for a roost located outside of the array of detectors, the roost estimate will be highly dependent



(a) The prior and posterior distributions for the diffusion coefficient, D.

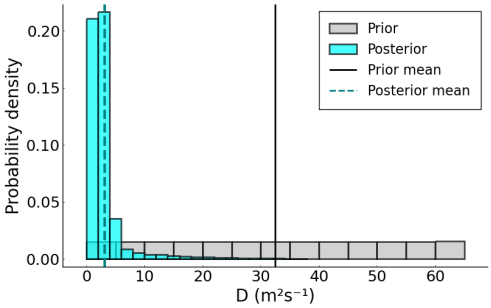


(b) The posterior probability distribution, shown as a heatmap overlaid on the scatter plot of detectors. The 95% credible region has an area of 4.32km^2.

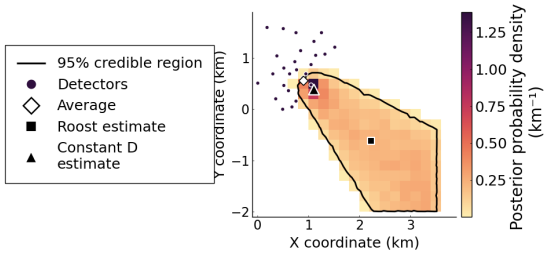


(c) The joint posterior distributions for diffusion coefficient, D, and the roost location.

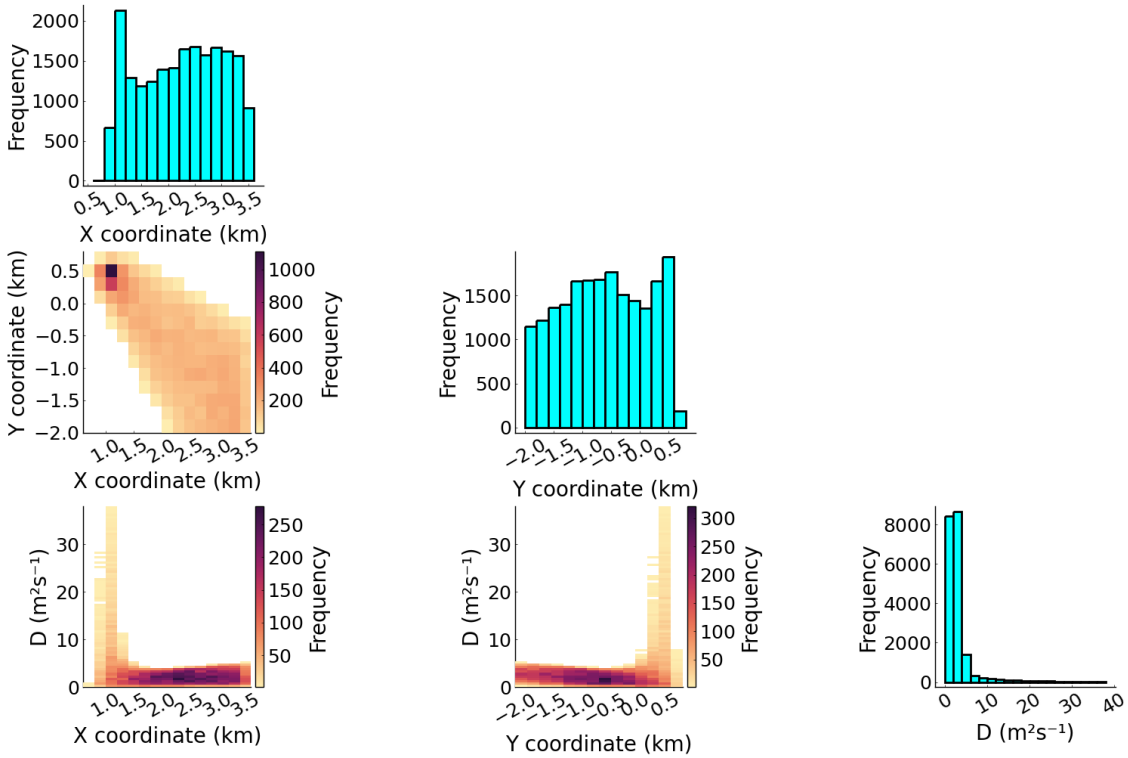
Figure 5.18: The posterior distributions for diffusion coefficient, D, and the roost location for a roost finding model with D as a parameter for Serotine bats (Eptser).



(a) The prior and posterior distributions for the diffusion coefficient, D.

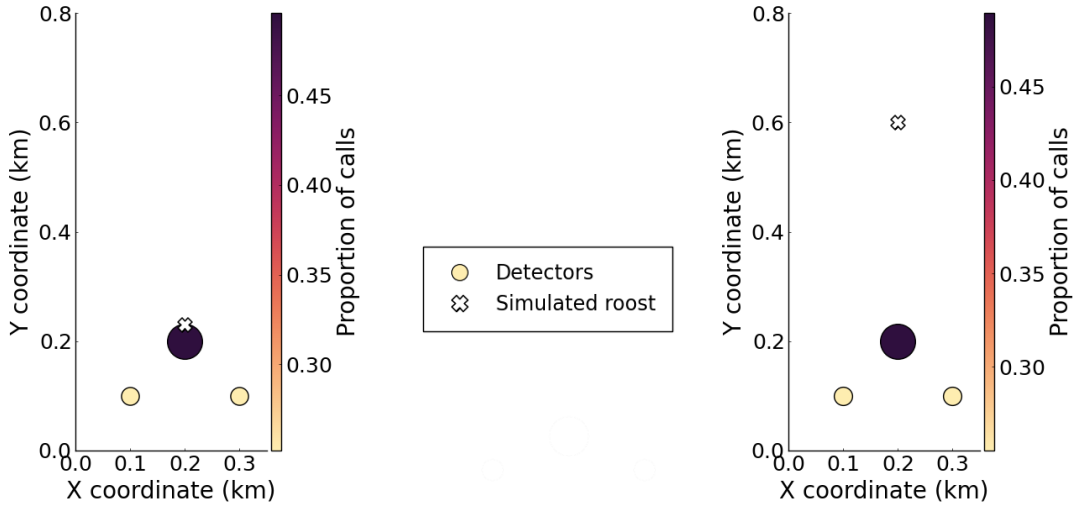


(b) The posterior probability distribution, shown as a heatmap overlaid on the scatter plot of detectors. The 95% credible region has an area of 4.44km^2.



(c) The joint posterior distributions for diffusion coefficient, D, and the roost location.

Figure 5.19: The posterior distributions for diffusion coefficient, D, and the roost location for a roost finding model with D as a parameter for Western Barbastelles (Barbar).



(a) $D = 30\text{m}^2\text{s}^{-1}$, simulated roost at 30m from closest detector.

(b) $D = 20\text{m}^2\text{s}^{-1}$, simulated roost at 400m from closest detector.

Figure 5.20: The number of calls expected at 3 simulated detectors with varying roost locations and diffusion coefficients.

on the diffusion coefficient. Moreover, if the roost is outside the detector array, we can only really point to the direction of the roost, and cannot necessarily estimate the distance to the roost without knowing the diffusion coefficient beforehand.

Without data from radio tracking surveys we cannot calculate a diffusion coefficient in the same way as we did for Greater Horseshoe bats, and so we will assume that the diffusion coefficient can take any value from the posterior distributions calculated here. In the next section, we will use these posterior distributions to plan a second survey at the Goodwood estate in order to hone in further on the roost locations.

4 Planning a second survey

Next, we will look at optimising detector locations for a second survey, given the posterior distributions calculated in Section 3.1.1. The second survey at Goodwood is currently ongoing as of July 2022, and the results are beyond the timeline of this thesis, but we will discuss our recommendations. As the posterior distributions were different for each of the four species studied, we will calculate the optimum detector locations for each species separately. In total, 40 detectors were available for the second round of surveys,

so we will allocate 10 detectors to each of the four species. As the results from the first survey suggest that roosts may be found outside of the original woodland, the survey area is expanded, to ensure that detectors can better cover possible roost locations. As with the first survey, possible detector locations are restricted to footpaths and trails through the woodland. The usable landscape for the second survey is shown, along with the area covered by the original survey, in Figure 5.21. The original survey covered an area of 3.28km^2, and the area covered by the second survey is 26.1km^2, around 8 times larger.

For each species, we use three-dimensional prior distributions for the roost location and diffusion coefficient, taking samples directly from the 95% credible regions calculated from the posterior distributions in Section 3.1.1. The areas covered by the spatial prior distributions for each species are shown in Figure 5.22. The priors for each of the four species clearly cover different areas of different sizes. From the results in Section 1.2, which suggest that the optimal detector placements include detectors close to the centre of the prior distribution, we can predict that the detector configurations for each species will be clustered around the prior distributions.

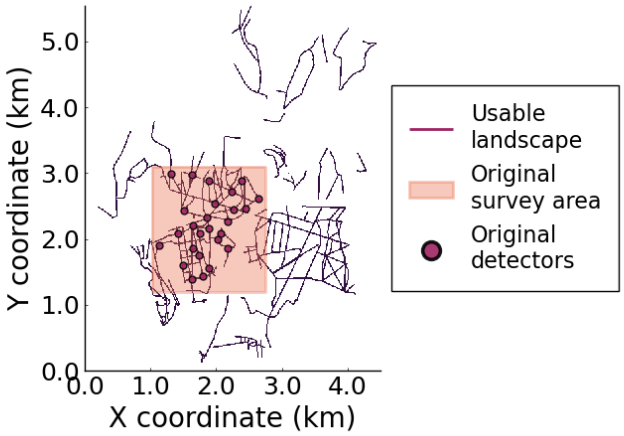


Figure 5.21: The usable landscape for the second survey at Goodwood compared to the area covered by the original survey.

The detector placement algorithm was run independently for each species, such that the placement of detectors for one species would not affect the placement for any of the other species. This is because it is likely that an ecologist searching for a bat roost would

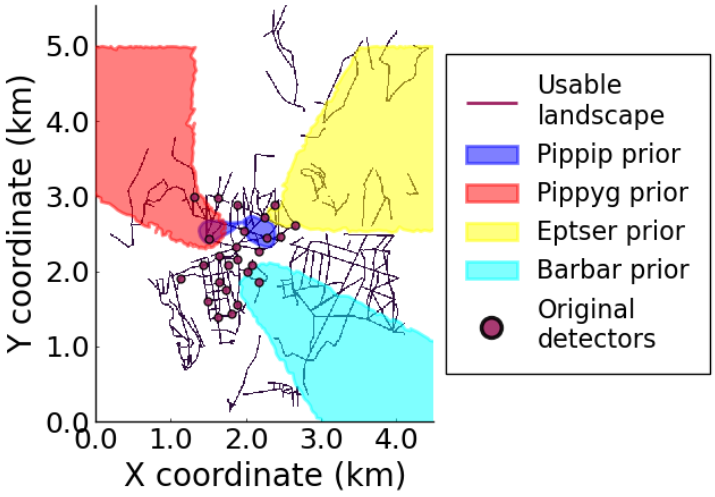


Figure 5.22: The area covered by the prior distributions for each species for the second survey at Goodwood.

likely be interested specifically in one, or two species, and therefore would be able to use only the most useful detectors for that particular species without wasting survey effort on potentially less useful detectors. For each species, we use the original set of 27 detector locations as the initial detector locations. As data from these locations has already been collected, we will not place new detectors in these locations, however we can still use the data from the first survey in the second round of analysis.

The final suggested detector placement for each species is shown in figures 5.23 and 5.24. It should be noted that since the placement for each species is independent, there are some detectors placed particularly close to one another. In particular, there are two pairs of detectors placed within 50m of each other. One of these pairs is located in the south east and includes one detector for Common Pipistrelles and one for Western Barbastelles, and the other pair is located in the north east and includes one detector for Soprano Pipistrelles and another for Serotine bats. Each of these pairs are marked with a circle on their respective plot. As the range of these detectors is a maximum of around 30m for these species, we would recommend that only one detector from each of these pairs be used as the differences in bat passes recorded at a second would be negligible.

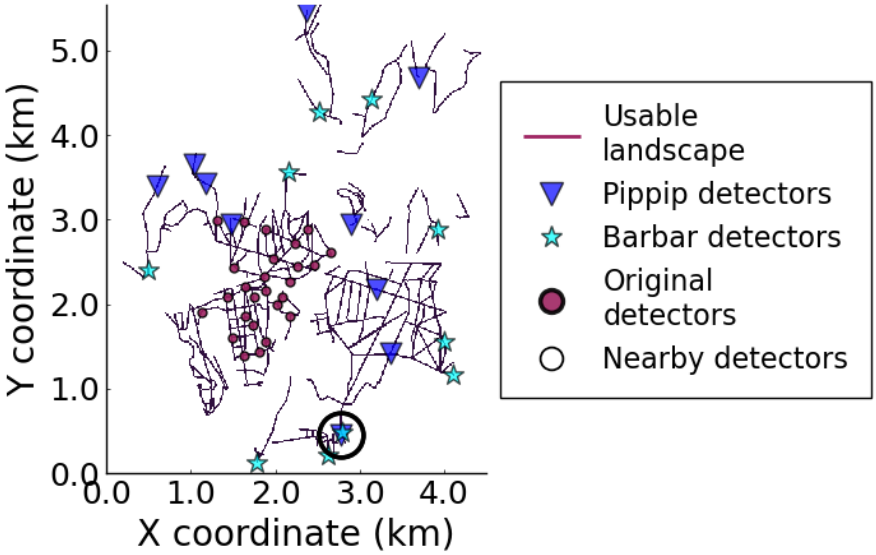


Figure 5.23: The locations of optimised detectors for Common Pipistrelles (Pippip) and Western Barbastelles (Barbar). Circles are plotted around two detectors located closer than 50m to one another.

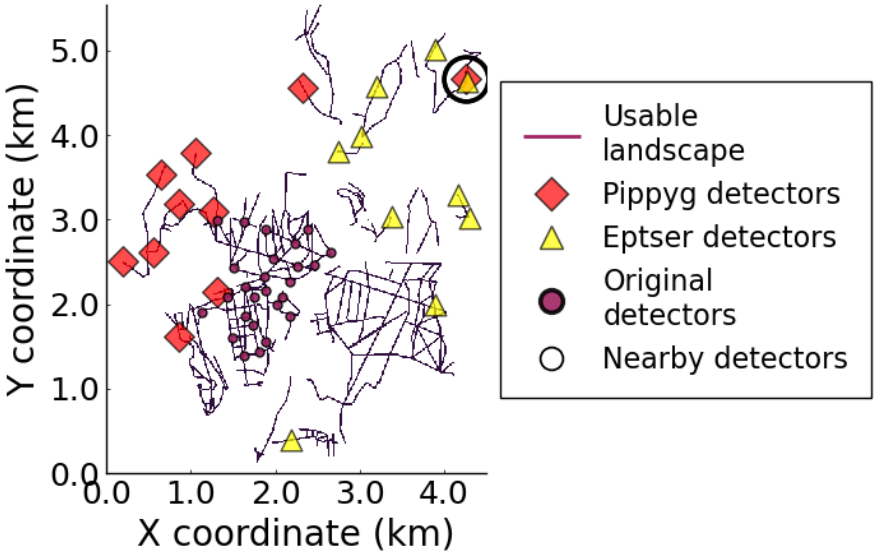
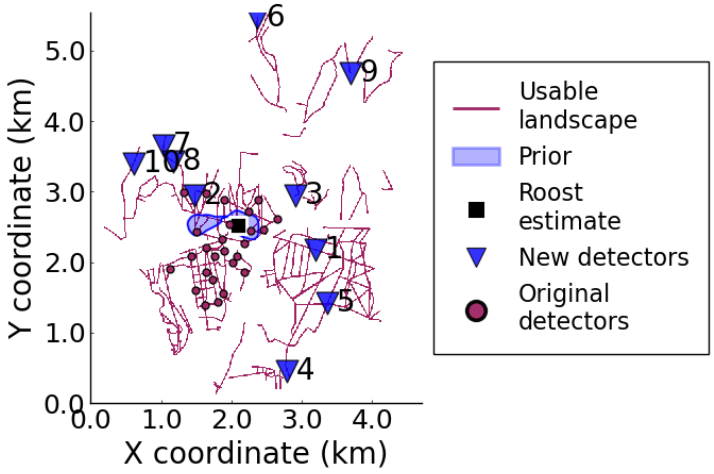


Figure 5.24: The locations of optimised detectors for Soprano Pipistrelles (Pippyg) and Serotine bats (Eptser). Circles are plotted around two detectors located closer than 50m to one another.

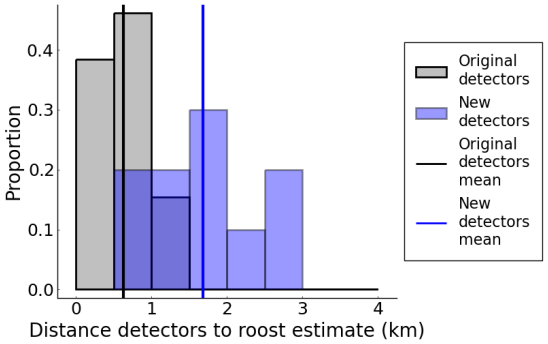
As expected, detector placements are different for each species, and Figure 5.24 shows that especially the detectors for Soprano Pipistrelles and Serotines appear to be clustered around their prior distributions. The results are shown individually for each species and in more detail in figures 5.25-5.28.

First we will look at the results for common Pipistrelles in Figure 5.25. We can see that the prior distribution is already very narrow, covering an area of just 0.22km^2, and it is also surrounded by multiple detectors from the original survey. As shown in Figure 5.25b, the distances between detectors and the roost estimate for the original detectors are low, with the maximum distance at less than 1.5km. The final detectors are significantly more spread out around the domain, and the distribution of distances is much higher than for the original detectors. The expected loss function at each iteration when adding new detectors is shown in Figure 5.25c. We see that fluctuations in the expected loss are high compared to the 95% confidence interval, and the expected loss does not appear to improve at later iterations when adding more detectors. This is likely due to high levels of noise compared to the low expected error in roost estimate. We conclude that the original detector locations already cover the best locations for this prior distribution and that the addition of extra detectors will not significantly improve roost estimates.

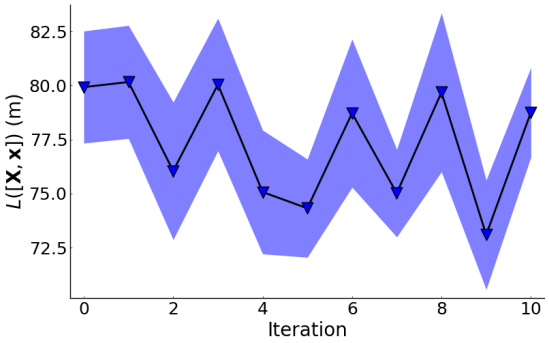
The results for the three other species studied are more interesting. We will next focus on the results for Soprano Pipistrelles, shown in Figure 5.26. The majority of the prior is located outside of the original survey area, and the prior search area is 4.75km^2, much larger than the common Pipistrelle search area. From Figure 5.26b, we see that whilst there are some detectors in the original survey closer than 1km to the posterior estimate, the vast majority are at a distance of between 1km and 2km. Most of the new detectors are clustered around the prior distribution, and we can see that 5 of the 10 new detectors are located within 1km of the roost estimate. The loss function at each iteration, shown in Figure 5.26c, shows a sharp decrease when adding the first two detectors, and this decrease begins to level off as more detectors are added. The loss function surface produced when detector locations are not restricted to trails and footpaths is shown in Figure 5.26d. The minimum loss here is found in the north west corner of the domain, at 1.2km from



(a) The prior distribution and locations of detectors from the first survey and proposed detectors for the second survey. The iteration number of each proposed detector is labelled.



(b) The distribution of distance between detectors and the roost estimate for the original survey and the final detector configuration.



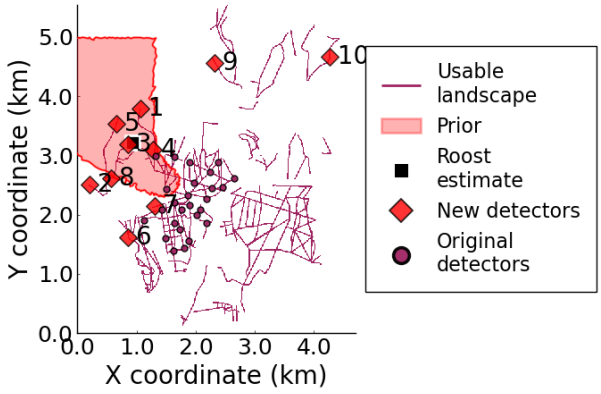
(c) The expected loss function at each iteration after placing suggested detectors. Markers are shown at the mean loss function calculated over 100 samples, and a 95% confidence interval, defined as $1.96SE$, where SE is the standard error, is shown as a ribbon around the mean.

Figure 5.25: Details of the proposed detectors and loss function for the second survey at Goodwood considering common Pipistrelles (Pippip).

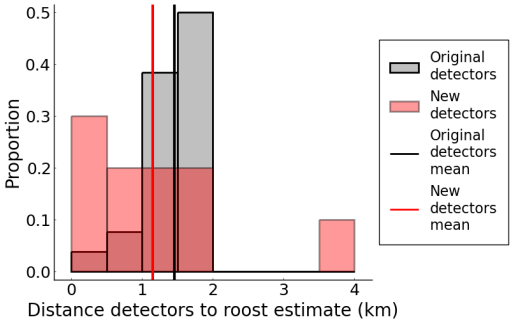
the nearest detectors. This point is close to the edge of the prior, but far from all other detectors.

Next we will consider the results for Serotine bats, shown in Figure 5.27. Again, the prior distribution is large at 4.32km^2, and in this case all of the detectors from the original survey are relatively far from the roost estimate, all between 1.5km and 3.5km. The suggested detector locations are mostly clustered around the prior in the north east of the survey area, with 9 detectors placed between 0.5km and 2km from the roost estimate. The loss function surface with no restrictions on landscape is shown in Figure 5.27d. In this case, the minimum loss was located in the south, far from other detectors. This point is located on a footpath and so the 10th detector was placed at this point. There is, however, a local minimum inside the main cluster of new detectors and close to the centre of the prior. In this case there are no footpaths in the centre of the prior, and therefore it is not possible to place a detector closer than 0.5km to the estimate. The loss function at each iteration, shown in Figure 5.27c, shows that the expected error decreases as new detectors are added, although the decrease levels off between iteration 8 and 10.

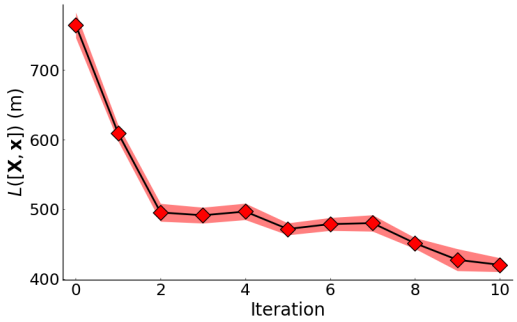
Finally, we will look at the results for Western Barbastelles, shown in Figure 5.28. The prior covers an area of 4.44km^2, and the majority is located outside the original survey area. The distance between the estimate and detectors is therefore high for the original survey, with all detectors located between 1.5km and 3.5km from the roost estimate, as shown in Figure 5.28b. The new detector locations include 4 detectors within 1.5km of the roost estimate, whilst 4 are located further than 3km from the roost estimate. The expected loss after each iteration, shown in Figure 5.28c, shows a sharp decrease with the addition of the first two detectors, which later levels off. After the 6th iteration, there is very little change in the loss function, and it is likely that the last detectors to be placed are essentially random.



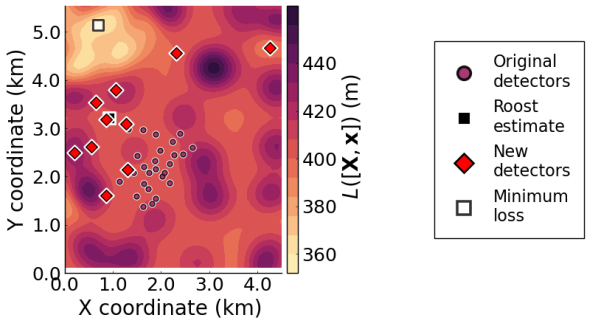
(a) The prior distribution and locations of detectors from the first survey and proposed detectors for the second survey. The iteration number of each proposed detector is labelled.



(b) The distribution of distance between detectors and the roost estimate for the original survey and the final detector configuration.

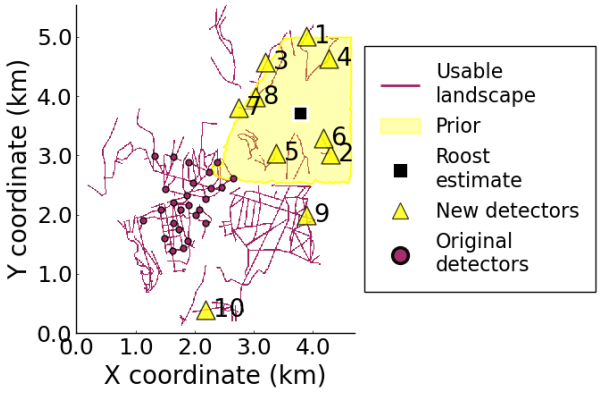


(c) The expected loss function at each iteration after placing suggested detectors. Markers are shown at the mean loss function calculated over 100 samples, and a 95% confidence interval, defined as $1.96SE$, where SE is the standard error, is shown as a ribbon around the mean.

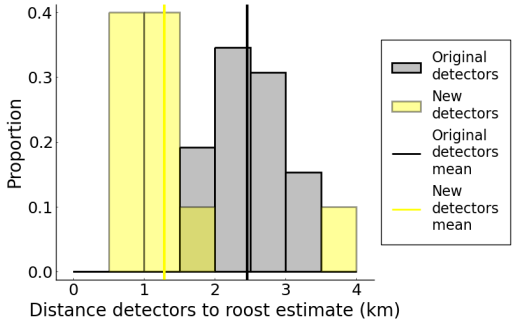


(d) A surface plot of the loss function at the 10th iteration.

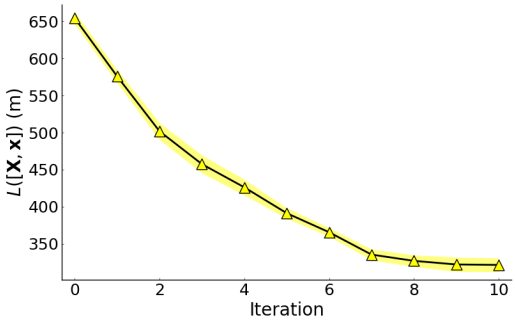
Figure 5.26: Details of the proposed detectors and loss function for the second survey at Goodwood considering Soprano Pipistrelles (Pippyg).



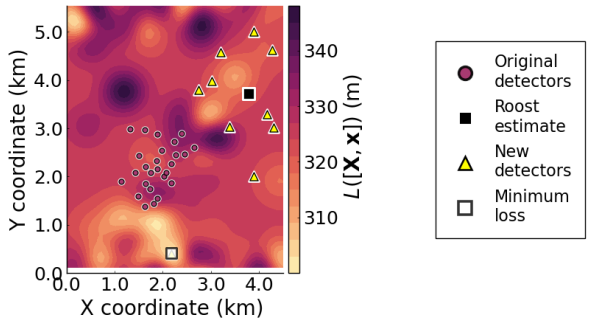
(a) The prior distribution and locations of detectors from the first survey and proposed detectors for the second survey. The iteration number of each proposed detector is labelled.



(b) The distribution of distance between detectors and the roost estimate for the original survey and the final detector configuration.

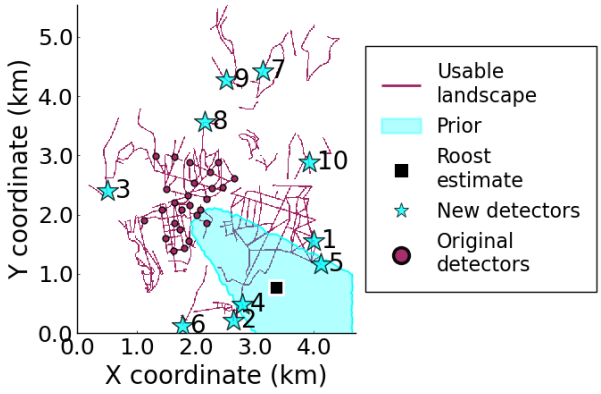


(c) The expected loss function at each iteration after placing suggested detectors. Markers are shown at the mean loss function calculated over 100 samples, and a 95% confidence interval, defined as $1.96SE$, where SE is the standard error, is shown as a ribbon around the mean.

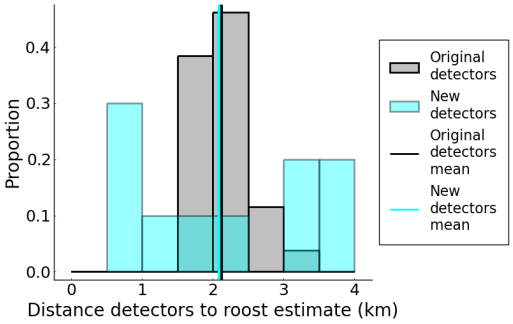


(d) A surface plot of the loss function at the 10th iteration.

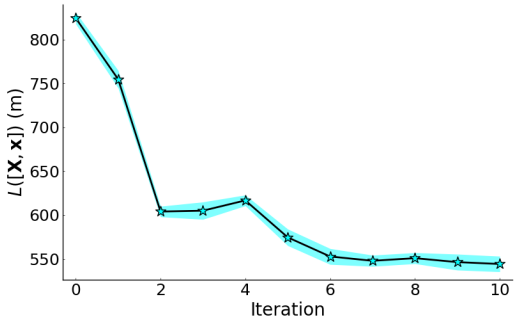
Figure 5.27: Details of the proposed detectors and loss function for the second survey at Goodwood considering Serotines (Eptser).



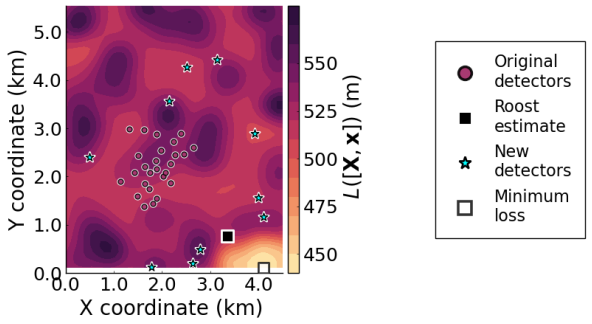
(a) The prior distribution and locations of detectors from the first survey and proposed detectors for the second survey. The iteration number of each proposed detector is labelled.



(b) The distribution of distance between detectors and the roost estimate for the original survey and the final detector configuration.



(c) The expected loss function at each iteration after placing suggested detectors. Markers are shown at the mean loss function calculated over 100 samples, and a 95% confidence interval, defined as $1.96SE$, where SE is the standard error, is shown as a ribbon around the mean.



(d) A surface plot of the loss function at the 10th iteration.

Figure 5.28: Details of the proposed detectors and loss function for the second survey at Goodwood considering western Barbastelles (Barbar).

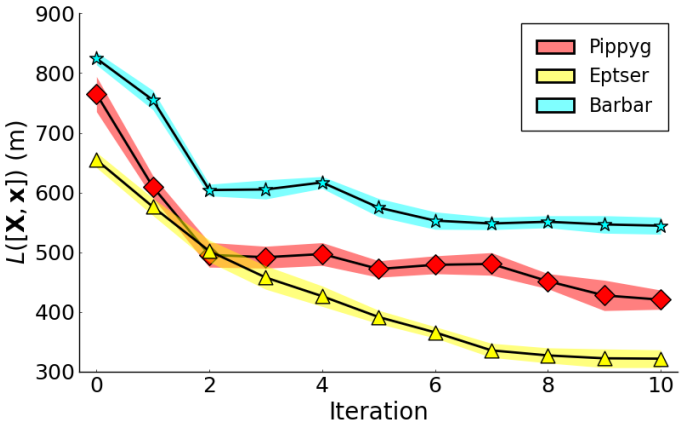


Figure 5.29: The loss function after each iteration for three species: Soprano Pipistrelles, Serotines and Barbastelles. Common Pipistrelles are not included here as fluctuations in loss function are dominated by noise for this species. Markers are shown at the mean loss function calculated over 100 samples, and a 95% confidence interval, defined as $1.96SE$, where SE is the standard error, is shown as a ribbon around the mean.

5 Discussion

We have discussed a method to choose optimal detector placements using an iterative process by minimising the expected error in roost estimate over the search area, and we have shown that the optimal detector configuration is highly dependent on the prior distribution for the roost location. In general, distributing detectors close to a peak in the prior distribution, where a roost is likely to be found, is desirable. However, placing detectors close to existing detectors is undesirable, since this will not add significantly more information about bat activity. In other words, there is a trade-off between placing detectors in locations of high entropy, where we know very little about bat activity, and locations of high probability density for the roost location.

The results for the initial survey at Goodwood show that there is a clear difference in the results for Common Pipistrelles and the other three species. Since the prior distribution for common Pipistrelles covers a small area, the expected error is already small, even before placing any new detectors. For each of the other species, the expected error in roost estimate decreases with added detectors (see Figure 5.29). The detector placement algorithm therefore appears to be more useful in situations when we have only a vague idea of the location of the roost. The algorithm could also be useful in deciding how

many detectors to use in a survey. Bat detectors can be expensive to purchase, and data collection is highly labour intensive, and so ecologists working on locating roosts generally prefer to use as few detectors as possible, whilst still ensuring they collect sufficient data. We have shown that the expected error in roost estimate does not decrease linearly with added detectors, instead the improvement levels off after adding a certain number of detectors. For Barbastelles, for example, the loss function does not appear to improve significantly after the 6th iteration, and perhaps only the first six detectors are needed to hone in further on the roost location.

It is important to note that whilst here we have discussed the method for the first two surveys, this process is iterative, and can be continued for multiple surveys, further increasing the accuracy of the estimate with each iteration. Multiple surveys could be particularly useful in situations in which few detectors are available, as an ecologist searching for a roost may not necessarily have 20-30 detectors available for a survey at a given time. If, for example, there are only 5 detectors available, a set of 4 surveys, optimising the locations for the next 5 detectors based on the previous results, would provide similar quantities of data as a single survey with 20 detectors. This has the added benefit that at each iteration we hone in further on the roost location, and therefore the detector locations for later surveys are optimised given all of the available data.

Of course, the best test of the method is more data. The results from the second survey at the Goodwood Estate will provide more insight into both the effectiveness of the method and the locations of roosts in the survey area.

Chapter Six

Discussion

Bats are incredibly important for ecosystems around the world, but human activity puts their habitats at risk[4, 5, 6].

To ensure that populations are protected, ecologists conduct surveys of bat roosts and provide conservation advice to local governments and developers. Unfortunately, locating roosts is highly labour intensive and difficult. In this thesis, we aimed to address this issue by using a combination of acoustic bat surveys and mathematical modelling to estimate the location of roosts and reduce the survey effort required to locate them.

In Chapter 3, we analysed location data from radiotracking surveys, and used this to inform a deterministic partial differential equation model for bat movement during the night. We found two distinct phases of movement, an initial dispersal from the roost, and a slow return to the roost throughout the remainder of the night. A diffusion model was fitted to the initial dispersal, whilst the second phase of movement was explained using a diffusion model on a shrinking domain. These models are consistent with possible mechanisms driving bat behaviour. Soon after sunset, bats disperse away from the roost, spreading out throughout the landscape to forage with minimal competition from the rest of the colony. Later on, bats start to return to the roost, gathering together at the edge of their foraging zone, as they eavesdrop on other bat's hunting calls to locate lucrative foraging grounds [88, 89].

In Chapter 4, we applied the diffusion model derived for the first phase of movement in

Chapter 3 to the problem of locating roosts. We used acoustic bat detectors to record bat calls and generate data describing the locations of bats throughout the survey. We then fitted the diffusion model to the data to estimate the position of the roost. The method was highly successful in some cases, however, for some surveys the method does not appear to provide a useful estimate. We discussed a number of reasons for this discrepancy, including the effect of landscape on foraging behaviour, the time of year in which the survey was conducted, additional roosts located close to the main roost, and the placement of detectors. In particular, we found that surveys conducted towards the start of the summer, during June and July, are most successful, whilst errors in roost estimates are significantly higher throughout August and September. We theorised that behaviour changes towards the end of the summer when pups are weaned and able to forage for themselves, and maternity colonies begin to break up as bats move to mating sites. We also found that the roost estimates for surveys which included detectors close to the roost were most accurate, and we suggested that when planning surveys, ecologists should ideally try to place detectors as close to a suspected roost location as possible. Of course, in practice, this is generally impossible, as if we wish to locate a roost, we generally do not know where it is.

In Chapter 5, we addressed the problem of detector placement, looking at the effect of different detector configurations on the quality of roost estimates. We developed a method to iteratively choose detector locations, given an initial prior probability distribution for the roost location, such that the expected error in roost estimate is minimised at each step. This method takes the terrain of the survey area into account, ensuring that detectors are not placed on unfavourable terrain. We found that, as expected based on the results from Chapter 4, the algorithm favours placing detectors close to the current roost estimate. However, spreading detectors out around the survey area by placing detectors far away from each other increases the information we gain about bat activity in the area. As such, there is a trade-off between placing detectors close to the current roost estimate and increasing the distance between detectors. We used the detector placement algorithm to plan two surveys at the Goodwood Estate in Sussex, an initial survey to gather information about the species present and make an initial estimate of roost locations, and a second

survey to further hone in on roost locations for the four most commonly recorded species. The second survey is still ongoing, and therefore results for this second survey are beyond the timeline of this thesis.

1 Further work

Whilst the roost finding method has been highly successful in some cases, further data is needed to improve upon the accuracy of the method and validate it in different scenarios.

Firstly, the models produced in Chapter 3 provide an excellent fit to radiotracking data from Greater Horseshoe bats. However, there are 18 species of bats present in the UK, and over 1270 worldwide [93, 2]. We already know that bat behaviour varies wildly between species, from the habitats they inhabit to their diets. It stands to reason that bats roosting in different habitats and consuming different foods would also be driven to move by different mechanisms. We do not know, therefore, if these models are generalisable to species other than Greater Horseshoe bats. Of course, the best way to test these models is collecting data from radiotracking surveys on bats of different species. With this data, it would be possible to confirm if diffusion models are suitable for bats of different species, and if so, estimate their diffusion coefficients.

As we have mentioned, the second survey at the Goodwood Estate is still ongoing. The data collected here will provide more of an insight into the generalisability of the method to different species and landscapes. Whilst we expect the results to improve estimates of the roost locations to some degree, it is likely that some combination of further acoustic surveys and radiotracking surveys may be required to hone in sufficiently to locate the specific structures bats are using as roosts.

Additionally, in the models developed in this thesis we assume that landscape is homogeneous. It is clear from previous research that landscape features have an important impact on bat movement [11]. In Chapter 5 we incorporated landscape features into the detector placement algorithm, ensuring that detectors are only placed on certain types of terrain. For the Goodwood Estate surveys, detector locations were restricted to

only footpaths and trails, such that the immediate surroundings of each detector were as homogeneous as possible. This goes some way in accounting for the effects of landscape on the data collected, however in reality bat activity is affected not just by the immediate surroundings but also the functional connectivity of the landscape.

Firstly, we know that many bat species prefer to travel along linear features such as hedgerows and rivers where insects are abundant and navigation is easy. Additionally, most bat species avoid bright lights and loud noises where possible, and therefore are unlikely to forage close to busy roads. To improve the model of bat movement, these landscape preferences could be built in.

The software Circuitscape [32] has been used to generate maps quantifying the connectivity of landscape surrounding a roost and predict the activity levels of bats at specific points throughout the landscape [11]. Circuitscape takes maps which represent the resistance to movement of the landscape in a region as input. It then produces a current map which shows the expected flow of movement between different points in the landscape. For bats, resistance is increased by proximity to undesirable features (roads, streetlights, buildings) and decreased by proximity to desirable features (rivers, hedgerows). The current map produced shows the expected flow of bats out of the roost as they disperse into the landscape.

Circuitscape could be used to improve upon the current model of bat movement and incorporate functional connectivity. In this case, instead of using a simple diffusion model to estimate the expected number of bat calls recorded at an array of detectors given a possible roost location, we would use Circuitscape current maps to estimate the expected number of calls. As previously mentioned, including the large quantities of data required to use Circuitscape is non-trivial and is likely to provide a barrier to use for many ecologists if they are required to source the data themselves.

Currently, a web application is being produced by Cardiff University and the Mammal Society which applies Circuitscape to study the relative importance of locations around a bat roost. The app will be accessible to any member of the public at batdispersion.mammal.org.uk when it is deployed and is designed to be simple to use for any

ecologist, or developer; all of the data required to produce landscape maps is available on a server, and therefore the user does not need to source anything themselves. Once the app is up and running, the aim is to include roost estimation. Since the landscape data required is already incorporated into the app, the roost estimation could use an improved model for bat movement incorporating functional connectivity.

Of course, the inclusion of extra landscape data and Circuitscape calculations will increase computation time, and it would therefore be important to test the sensitivity of detector data to landscape effects to ensure that these additions improve roost estimations enough to merit the extra computation time. Equally, it may be useful to have both options available: the simple diffusion model to gain a rough idea of where a roost might be, and a more sophisticated landscape dependent model to further hone in on the roost.

In this thesis we have improved our understanding the mechanisms that drive bat movement throughout the night and used this to design a method to estimate roost locations. Whilst there are still steps that can be taken to further improve the model and roost estimation algorithms, we have shown that the model has been successful in locating Greater Horseshoe maternity roosts in early summer and thus can be used to optimise bat survey design and execution.

Bibliography

[1] T. H. Kunz, E. Braun de Torrez, D. Bauer, T. Lobova, and T. H. Fleming. Ecosystem services provided by bats. *Annals of the New York Academy of Sciences*, 1223(1):1–38, Mar 2011.

[2] T. H. Kunz and M. B. Fenton. *Bat ecology*. University of Chicago Press, 2005.

[3] G. Jones, D. S. Jacobs, T. H. Kunz, M. R. Willig, and P. A. Racey. Carpe noctem: the importance of bats as bioindicators. *Endangered Species Research*, 8:93–115, Jul 2009.

[4] S. Rossiter, G. Jones, R. Ransome, and E. Barratt. Genetic variation and population structure in the endangered greater horseshoe bat rhinolophus ferrumequinum. *Molecular Ecology*, 9(8):1131–1135, 2000.

[5] R. Stebbings. *The conservation of European bats*. Christopher Helm Pub Limited, 1988.

[6] A. Hutson and S. Mickleburgh. *Microchiropteran bats: global status survey and conservation action plan*, volume 56. IUCN, 2001.

[7] F. Marnella and P. Presetnik. *Protection of overground roosts for bats*. UNEP/EUROBATS Secretariat, 2010.

[8] Wildlife and countryside act 1981. `https://www.legislation.gov.uk/ukpga/1981/69`. [Accessed 28-September-2021].

[9] Conservation of habitats and species regulations 2017. `https://www.legislation.gov.uk/uksi/2017/1012/regulation/1/made`. [Accessed 28-September-2021].

[10] C. C. Voigt, W. F. Frick, M. W. Holderied, R. Holland, G. Kerth, M. Mello, and R. K. Plowright. Principles and patterns of bat movements: from aerodynamics to ecology. *The Quarterly Review of Biology*, 92(3):267–287, 2017.

[11] D. Finch, D. P. Corbacho, H. Schofield, S. Davison, P. Wright, R. K. Broughton, and F. Mathews. Modelling the functional connectivity of landscapes for greater horseshoe bats rhinolophus ferrumequinum at a local scale. *Landscape Ecology*, 35(3):577–589, 2020.

[12] F. Mathews. Radiotracking study of greater horseshoe bats from beer and branscombe. Technical report, University of Exeter, 2009.

[13] J. Froidevaux, K. Boughey, K. Barlow, and G. Jones. Factors driving population recovery of the greater horseshoe bat (rhinolophus ferrumequinum) in the UK: implications for conservation. *Biodiversity and Conservation*, 26(7):1601–1621, 2017.

[14] A. Haarsma and D. Tuitert. An overview and evaluation of methodologies for locating the summer roosts of pond bats (myotis dasycneme) in the netherlands. *Lutra*, 52(1), 2009.

[15] J. Perks and A. Goodenough. Abiotic and spatiotemporal factors affect activity of european bat species and have implications for detectability for acoustic surveys. *Wildlife Biology*, 2020(2), 2020.

[16] F. Bontadina, H. Schofield, and B. Naef-Daenzer. Radio-tracking reveals that lesser horseshoe bats (rhinolophus hipposideros) forage in woodland. *Journal of Zoology*, 258(3):281–290, 2002.

[17] J. A. Encarnação, U. Kierdorf, D. Holweg, U. Jasnoch, and V. Wolters. Sex-related differences in roost-site selection by daubenton's bats myotis daubentonii during the nursery period. *Mammal Review*, 35(3-4):285–294, 2005.

[18] T. H. Kunz and S. Parsons. *Ecological and behavioral methods for the study of bats*. Number 599.4 E2. Smithsonian Institution Press Washington, DC, 1988.

[19] A. Kurta, T. H. Kunz, et al. Capture methods and holding devices. *Ecological and behavioral methods for the study of bats (T. H Kunz ed.). Smithsonian Institution Press, Washington, DC*, pages 1–30, 1988.

[20] R. M. Brigham. Load carrying and maneuverability in an insectivorous bat: a test of the 5% rule of radio-telemetry. *Journal of mammalogy*, 69(2):379–382, 1988.

[21] G. Jones. Flight performance, echolocation and foraging behaviour in noctule bats nyctalus noctula. *Journal of Zoology*, 237(2):303–312, 1995.

[22] M. O'Mara, M. Wikelski, and D. Dechmann. 50 years of bat tracking: device attachment and future directions. *Methods in Ecology and Evolution*, 5(4):311–319, 2014.

[23] S. Picardi, B. Smith, M. Boone, P. Frederick, J. Cecere, D. Rubolini, L. Serra, S. Pirrello, R. Borkhataria, and M. Basille. Analysis of movement recursions to detect reproductive events and estimate their fate in central place foragers. *Movement ecology*, 8:1–14, 2020.

[24] C. Bracis, K. Bildstein, and T. Mueller. Revisitation analysis uncovers spatio-temporal patterns in animal movement data. *Ecography*, 41(11):1801–1811, 2018.

[25] S. Le Comber, B. Nicholls, D. Rossmo, and P. Racey. Geographic profiling and animal foraging. *Journal of Theoretical Biology*, 240(2):233–240, May 2006.

[26] D. Rossmo. *Geographic profiling*. CRC press, 1999.

[27] A. Hill, P. Prince, E. Piña Covarrubias, P. Doncaster, J. Snaddon, and A. Rogers. Audiomoth: Evaluation of a smart open acoustic device for monitoring biodiversity and the environment. *Methods in Ecology and Evolution*, 9(5):1199–1211, 2018.

[28] D. Parker and R. Bernard. The use of acoustic detectors for assessing bat species richness and functional activity in a south african national park. *Mammalia*, 83(1):53–63, 2019.

[29] D. Scherrer, P. Christe, and A. Guisan. Modelling bat distributions and diversity in a mountain landscape using focal predictors in ensemble of small models. *Diversity and Distributions*, 25(5):770–782, 2019.

[30] J. Froidevaux, K. Boughey, C. Hawkins, G. Jones, and J. Collins. Evaluating survey methods for bat roost detection in ecological impact assessment. *Animal conservation*, 23(5):597–606, 2020.

[31] B. McRae, B. Dickson, T. Keitt, and V. Shah. Using circuit theory to model connectivity in ecology, evolution, and conservation. *Ecology*, 89(10):2712–2724, 2008.

[32] R. Anantharaman, K. Hall, V. Shah, and A. Edelman. Circuitscape in julia: High performance connectivity modelling to support conservation decisions. *Proceedings of the JuliaCon Conferences*, 1(1):58, 2020.

[33] P. Lintott, S. Davison, J. van Breda, L. Kubasiewicz, D. Dowse, J. Daisley, E. Haddy, and F. Mathews. Ecobat: An online resource to facilitate transparent, evidence-based interpretation of bat activity data. *Ecology and evolution*, 8(2):935–941, 2018.

[34] O. Ovaskainen, H. J. de Knegt, and M. del Mar Delgado. *Quantitative Ecology and Evolutionary Biology: Integrating models with data.* Oxford University Press, 2016.

[35] J. D. Murray. *Mathematical Biology.* Springer New York, 2011.

[36] T. Patterson, A. Parton, R. Langrock, P. Blackwell, L. Thomas, and R. King. Statistical modelling of individual animal movement: an overview of key methods and a discussion of practical challenges. *AStA Advances in Statistical Analysis*, 101(4):399–438, 2017.

[37] G. Uhlenbeck and L. Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

[38] J. Dunn and P. Gipson. Analysis of radio telemetry data in studies of home range. *Biometrics*, pages 85–101, 1977.

[39] R. Jennrich and F. Turner. Measurement of non-circular home range. *Journal of theoretical Biology*, 22(2):227–237, 1969.

[40] H. Preisler, A. Ager, B. Johnson, and J. Kie. Modeling animal movements using stochastic differential equations. *Environmetrics*, 15(7):643–657, 2004.

[41] F. Lenz, A. Chechkin, and R. Klages. Constructing a stochastic model of bumblebee flights from experimental data. *PloS one*, 8(3):e59036, 2013.

[42] J. Murray. *Mathematical biology II: spatial models and biomedical applications*, volume 3. Springer New York, 2001.

[43] J. Potts and M. Lewis. How do animal territories form and change? lessons from 20 years of mechanistic modelling. *Proceedings of the Royal Society B: Biological Sciences*, 281(1784):20140231, 2014.

[44] O. Ovaskainen, H. Rekola, E. Meyke, and E. Arjas. Bayesian methods for analyzing movements in heterogeneous landscapes from mark–recapture data. *Ecology*, 89(2):542–554, 2008.

[45] R. Eftimie, G. de Vries, and M. Lewis. Complex spatial group patterns result from different animal communication mechanisms. *Proceedings of the National Academy of Sciences*, 104(17):6974–6979, 2007.

[46] N. Cvikel, K. Berg, E.Levin, and E. Hurme. Bats aggregate to improve prey search but might be impaired when their density becomes too high. *Current Biology*, 25(2):206–211, 2015.

[47] P. Kerches-Rogeri, D.Ramos, J. Siren, B. de Oliveira Teles, R. Alves, C. Priante, M. Ribeiro, M. Araújo, and Otso O. Ovaskainen. Movement syndromes of a neotropical frugivorous bat inhabiting heterogeneous landscapes in brazil. *Movement ecology*, 9(1):1–12, 2021.

[48] J. Potts and U. Schlägel. Parametrizing diffusion-taxis equations from animal movement trajectories using step selection analysis. *Methods in Ecology and Evolution*, 11(9):1092–1105, 2020.

[49] E. Ferguson, J. Matthiopoulos, R. Insall, and D. Husmeier. Inference of the drivers of collective movement in two cell types: Dictyostelium and melanoma. *Journal of The Royal Society Interface*, 13(123):20160695, 2016.

[50] E. J. Crampin. *Reaction-Diffusion Patterns on Growing Domains*. PhD thesis, University of Oxford, 2000.

[51] E. J. Crampin, W. W. Hackborn, and P. K. Maini. Pattern formation in reaction-diffusion models with nonuniform domain growth. *Bulletin of Mathematical Biology*, 64(4):747–769, 2002.

[52] A. Gelman, J. B Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis, Third Edition*. Chapman and Hall/CRC, Nov 2013.

[53] P. Marjoram. Approximation bayesian computation. *OA genetics*, 1(3):853, 2013.

[54] M. Beaumont, W. Zhang, and D. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162:2025–2035, Dec 2002.

[55] S. A. Sisson and Y. Fan. Likelihood-free markov chain monte carlo. *arXiv*, 2010.

[56] I. J. Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.

[57] D. Prangle, P. Fearnhead, M. Cox, P. Biggs, and N. French. Semi-automatic selection of summary statistics for abc model choice. *Statistical applications in genetics and molecular biology*, 13(1):67–82, 2014.

[58] M. Blum. Approximate bayesian computation: a nonparametric perspective. *Journal of the American Statistical Association*, 105(491):1178–1187, 2010.

[59] D. Prangle. Summary statistics in approximate bayesian computation. *arXiv preprint arXiv:1512.05633*, 2015.

[60] E. Brochu, V. M. Cora, and N. De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[61] P. I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[62] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[63] J. Fairbrother, C. Nemeth, M. Rischard, J. Brea, and T. Pinder. Gaussianprocesses. jl: A nonparametric bayes package for the julia language. *arXiv preprint arXiv:1812.09064*, 2018.

[64] O. Bousquet, U. von Luxburg, and G. Rätsch. *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, volume 3176. Springer, 2011.

[65] C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[66] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[67] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[68] P. Billingsley. *Probability and measure*. John Wiley & Sons, 2008.

[69] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[70] Bat Conservation Trust. Core sustenance zones: Determining zone size. `https://www.bats.org.uk/our-work/landscapes-for-bats/core-sustenance-zones`, Feb 2016. [Accessed 28-September-2021].

[71] J. Curtiss. A note on the theory of moment generating functions. *The Annals of Mathematical Statistics*, 13(4):430–433, 1942.

[72] C. Rackauckas and Q. Nie. DifferentialEquations.jl – a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 2017.

[73] T. E. Woolley, R. E. Baker, E. A. Gaffney, and P. K. Maini. Stochastic reaction and diffusion on growing domains: understanding the breakdown of robust pattern formation. *Physical Review E*, 84(4):046216, 2011.

[74] V. Anh, N. Leonenko, and N. Shieh. Multifractal products of stationary diffusion processes. *Stochastic Analysis and Applications*, 27(3):475–499, 2009.

[75] Y. Yun. The moments of a diffusion process. *Statistics & Probability Letters*, 138:36–41, 2018.

[76] D. Grebenkov. A unifying approach to first-passage time distributions in diffusing diffusivity and switching diffusion models. *Journal of Physics A: Mathematical and Theoretical*, 52(17):174001, 2019.

[77] C. N. T. Mori and E. C. Romão. Numerical simulation by finite difference method of 2d convection-diffusion in cylindrical coordinates. *Applied Mathematical Sciences*, 9(123):6157–6165, 2015.

[78] C. Galeriu, L. C. Lew Yan Voon, R. Melnik, and M. Willatzen. Modeling a nanowire superlattice using the finite difference method in cylindrical polar coordinates. *Computer physics communications*, 157(2):147–159, 2004.

[79] S. Britt, S. Tsynkov, and E. Turkel. A compact fourth order scheme for the helmholtz equation in polar coordinates. *Journal of Scientific Computing*, 45(1-3):26–47, 2010.

[80] T. Stocker. *Introduction to climate modelling*. Springer Science & Business Media, 2011.

[81] S. Karlin and H. Taylor. *A second course in stochastic processes*. Elsevier, 1981.

[82] Lsqfit.jl. `https://github.com/JuliaNLSolvers/LsqFit.jl`. Accessed: 2020-05-13.

[83] E. J. Crampin, E. Gaffney, and P. K. Maini. Reaction and diffusion on growing domains: scenarios for robust pattern formation. *Bulletin of Mathematical Biology*, 61(6):1093–1120, 1999.

[84] D. J. Acheson. *Elementary fluid dynamics: Oxford University Press*. Oxford, England, 1990.

[85] A. Fick. On liquid diffusion. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 10(63):30–39, 1855.

[86] P. Duvergé and G. Jones. Greater horseshoe bats-activity, foraging behaviour and habitat use. *British Wildlife*, 6:69–69, 1994.

[87] G. Jones and J. Rayner. Foraging behavior and echolocation of wild horseshoe bats rhinolophus ferrumequinum and r. hipposideros (chiroptera, rhinolophidae). *Behavioral Ecology and Sociobiology*, 25(3):183–191, 1989.

[88] M. Roeleke, T. Blohm, U. Hoffmeister, L. Marggraf, U. E. Schlägel, T. Teige, and C.C. Voigt. Landscape structure influences the use of social information in an insectivorous bat. *Oikos*, 2020.

[89] K. Egert-Berg, E. R. Hurme, S. Greif, A. Goldstein, L. Harten, José J. Flores-Martínez, A.T. Valdés, D.S. Johnston, O. Eitan, I. Borissov, et al. Resource ephemerality drives social foraging in bats. *Current Biology*, 28(22):3667–3673, 2018.

[90] H. U. Schnitzler, C. F. Moss, and A. Denzinger. From spatial orientation to food acquisition in echolocating bats. *Trends in Ecology & Evolution*, 18(8):386–394, 2003.

[91] C. L. Walters, R. Freeman, A. Collen, C. Dietz, M. B. Fenton, G. Jones, M. K. Obrist, S. J. Puechmaille, T. Sattler, B. M. Siemers, S. Parsons, and K. E. Jones. A continental-scale tool for acoustic identification of european bats. *Journal of Applied Ecology*, 49(5):1064–1074, 2012.

[92] N. Vaughan, G. Jones, and S. Harris. Identification of british bat species by multivariate analysis of echolocation call parameters. *Bioacoustics*, 7(3):189–207, 1997.

[93] J. Russ. *British bat calls: a guide to species identification*. Pelagic publishing, 2012.

[94] A. Horwitz. A generalization of simpson's rule. *Approximation Theory and its Applications*, 9(2):71–80, 1993.

[95] J. Berntsen, T. O. Espelid, and A. Genz. An adaptive algorithm for the approximate calculation of multiple integrals. *ACM Transactions on Mathematical Software*, 17(4):437–451, Dec 1991.

[96] Concavehull.jl. https://github.com/lstagner/ConcaveHull.jl. [Accessed 28-September-2021].

[97] A. Moreira and M. Santos. Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points. *INSTICC Press*, 2007.

[98] J. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information processing letters*, 2(1):18–21, 1973.

[99] R. D. Ransome and T. P. McOwat. Birth timing and population changes in greater horseshoe bat colonies (rhinolophus ferrumequinum) are synchronized by climatic temperature. *Zoological journal of the Linnean Society*, 112(3):337–351, 1994.

[100] S. E. Lewis. Effect of climatic variation on reproduction by pallid bats (antrozous pallidus). *Canadian Journal of Zoology*, 71(7):1429–1433, 1993.

[101] S. J. Rossiter, G. Jones, R. D. Ransome, and E. M. Barratt. Relatedness structure and kin-biased foraging in the greater horseshoe bat (rhinolophus ferrumequinum). *Behavioral Ecology and Sociobiology*, 51(6):510–518, 2002.

[102] OS mastermap topography layer. `https://www.ordnancesurvey.co.uk/business-government/products/mastermap-topography`. [Accessed 03-February-2022].

[103] W. Miller and E. Myers. A file comparison program. *Software: Practice and Experience*, 15(11):1025–1040, 1985.

[104] E. Ukkonen. Algorithms for approximate string matching. *Information and control*, 64(1-3):100–118, 1985.

[105] Hyperopt.jl. `https://github.com/baggepinnen/Hyperopt.jl`. [Accessed 04-March-2022].

[106] Bayesianoptimization.jl. `https://github.com/jbrea/BayesianOptimization.jl`. [Accessed 04-March-2022].

[107] J. Berry, W. Hart, C. Phillips, J. Uber, and Jean J. Watson. Sensor placement in municipal water networks with temporal integer programming models. *Journal of water resources planning and management*, 132(4):218–224, 2006.

[108] W. Yuill, A. Edwards, and S. Chowdhury. Optimal pmu placement: A comprehensive literature review. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–8. IEEE, 2011.

[109] J. Smith and Z. Taskin. A tutorial guide to mixed-integer programming models and solution techniques. *Optimization in Medicine and Biology*, pages 521–548, 2008.

[110] G. Ross and R. Soland. A branch and bound algorithm for the generalized assignment problem. *Mathematical programming*, 8(1):91–103, 1975.

[111] F. Feyerabend and M. Simon. Use of roosts and roost switching in a summer colony of 45 khz phonic type pipistrelle bats (pipistrellus pipistrellus schreber, 1774). *Myotis*, 38:51–59, 2000.

[112] E. Stone, M. Zeale, S. Newson, W. Browne, S. Harris, and G. Jones. Managing conflict between bats and humans: the response of soprano pipistrelles (pipistrellus pygmaeus) to exclusion from roosts in houses. *PLoS One*, 10(8):e0131825, 2015.

[113] A. Martinoli, M. Mazzamuto, and M. Spada. Serotine eptesicus serotinus (schreber, 1774). *Handbook of the Mammals of Europe*, pages 1–17, 2020.

[114] K. Boughey, I. Lake, K. Haysom, and P. Dolman. Effects of landscape-scale broadleaved woodland configuration and extent on roost location for six bat species across the UK. *Biological Conservation*, 144(9):2300–2310, 2011.