Statistical approaches for the inference of drift and diffusion coefficients for a model describing directed cell migration

Jon Devlin

Department of Mathematics and Statistics,

University of Strathclyde,

Glasgow, U.K.

A thesis submitted for the degree of

Doctor of Philosophy

December 2021

Copyright Declaration

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by the University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Signed:

Date:

Acknowledgements

I would first like to say a huge thank you to my supervisors John Mackenzie and Dirk Husmeier for all their support, patience and encouragement throughout my PhD. I would never have gotten this far without you both, especially your support and understanding during many difficult times, and for that I am extremely grateful. I would also like to thank Robert Insall for his witty and informative conversations and to everyone from the Beatson Institute for Cancer Research for being so welcoming and friendly during my visits. My thanks also go to Cancer Research UK for funding my PhD scholarship without which this thesis would not be possible.

I would also like to thank all my colleagues at the Department of Mathematics and Statistics for their support and friendship over these past 5 years. In particular, I would like to thank the members of the Lunch Group for the many outings and happy memories we have shared together, my friends in LT1010 for being fantastic office mates and truly deserving of the name Office 10/10, Louise Kelly for her friendship and guidance during our outreach work together, and finally Irene Spencer, Ann Lynch, Mary McAuley and Sandra Miller for the incredible amount of work they have done for me (and who we know the department would never manage without!)

Lastly, I would like to thank my parents, Twiggy and Cha, my sister, Aikida, and my brother-in-law, Colin, who have always pushed me to do my best and supported me no matter what. I would also like to thank the Main Crew for their continued love and support, as well as for all our hilarious trips together, which helped lessen the stress of PhD life.

Abstract

The use of mathematical models has become a widespread and important aspect of cellular biology to describe cell migration and chemotaxis. Many of these models give results which qualitatively match experimental data well, but most are not calibrated to the data to quantitatively estimate the unknown parameters of interest. This thesis focuses on using statistical inference approaches to estimate the drift velocity and diffusion coefficient of a simple drift-diffusion stochastic differential equation model describing directed cell movement. All approaches make use of the mean square displacement as a summary statistic of the trajectory data. When using least squares regression, the quality of the inference depends on the number of regression fitting points or the measurement time interval over which experiments are made, depending on the experimental protocol. Simple and efficient iterative algorithms are presented to estimate the optimal number of fitting points and measurement time interval, along with estimates of the drift and diffusion coefficients. For inference using approximate Bayesian computation, the quality of the inference is again shown to depend crucially on the measurement time interval over which experiments are made. A number of different approximate Bayesian computation approaches are presented and compared, showing that the best approach changes depending on the value of the measurement time interval. Finally, a hybrid model describing cell migration and chemical diffusion is presented to investigate a process called self-generated gradient chemotaxis. Numerical simulation from the generative model using physiologically relevant parameter values produces data which agrees well with experimental data.

Contents

Chapter 1

Introduction

1.1 Cell migration and chemotaxis

Cell migration is an essential component of several important biological processes such as wound healing [1], collective cell migration in embryonic development [2], the movement of leukocytes (white blood cells) to infections in immune response [3] and cancer metastasis [4]. Most of these processes depend on a type of cell migration known as chemotaxis, the movement of cells along chemical gradients in response to a chemical stimulus. For example, it is well established that chemotaxis plays a key role in cancer metastasis [5]. Despite the obvious importance of chemotaxis, the sources of chemoattractants, and how these chemical gradients evolve in response to their depletion from cells, are often unknown [6].

In this thesis, we are mainly interested in eukaryotic cells, those which have a nucleus, rather than prokaryotic cells, such as bacteria. Eukaryotic cell mobility is controlled by the formation of pseudopods which are protrusions of the cell membrane through actin polymerization at the front of the cell, which drives the cell forward [7]. To do this, actin, a globular protein, polymerizes into actin filaments (F-actin) which is a major component of the cell cytoskeleton. However, actin polymerization creates new filaments at a slow rate and so the assembly of F-actin can be stimulated by nucleation factors. F-actin in most cells is concentrated directly beneath the cell membrane forming part of the cell cortex. Myosin II and other motor proteins then bind to the F-actin in the cell cortex, cross-linking and contracting the filaments, which causes cortical tension and mechanical resistance. This causes actin polymerization to occur between the cell cortex and membrane which pushes the cell membrane out from the pseudopod while cortical tension moves the rest of the cell along.

New pseudopods can be created from a process called pseudopod splitting, where a currently formed pseudopod splits in two. In fact, it has been observed that in many cell types, around 90% of new pseudopods formed are due to the bifurcation of existing pseudopods, for example, in human leukocytes, mouse embryonic fibroblasts and *Dictyostelium discoideum* cells [8, 9]. Pseudopods play a key role in eukaryotic cell movement as they determine the direction and speed of a cell's migration [10]. For directed motility, ligands bind to membrane bound receptors to identify signiling cues, then pseudopods are formed, either as new protrusions or by splitting, which guides the cell in the direction of attraction. Groups of cells can also coordinate their pseudopod formation, resulting in collective cell migration [11].

1.2 Self-generated gradients

Although cells often chemotax by detecting a high concentration of chemoattractant and then moving along the created chemical gradient, some cell types have also been observed to create their own chemical gradients [12]. When these cells are near an area of higher chemoattractant concentration, they begin to degrade the chemoattractant, creating local, steep chemical gradients. Cells then migrate towards the region of higher chemoattractant concentration and continue to degrade and migrate in this fashion. This process where cells essentially create their own chemical gradients is called self-generated chemotaxis.

There are several visual characteristics which can help identify whether cells are moving according to self-generated gradients [6]. The most obvious characteristic is a leading wave of cells. As the cells degrade the chemoattractant, those closest will detect the chemical gradient and migrate forward. The cells which are furthest away from the chemoattractant will migrate slower and eventually will be unable to detect the chemical gradients. They will then fall behind while the leading wave of cells continue to migrate forward. Physical experiments in an Insall Chamber [13] have shown that *Dictyostelium discoideum* cells can move according to self-generated gradients by degrading folate under-agarose [6].

It has been proposed that self-generated gradients could be a main driver for cancer cell metastasis over long distances. It has been shown that melanoma cells can degrade high levels of lysophosphatidic acid (LPA), creating their own chemical gradients which allow the cells to chemotax outwards from the tumour [14]. Recent experimental work shows that cells have the ability to break down local chemoattractants in order to solve maze problems [15]. In this study, mathematically modelled cells were able to navigate through mazes of different complexities with the inclusion of dead ends. Cells were released into the maze from one end with an attractant well of chemoattractant diffusing into the maze from the other side. By degrading the chemoattractant and creating steep, local chemical gradients, cells were able to migrate forward and ‘sense’ around maze junctions before reaching them. They were also able to make clever decisions, taking the optimal route through the maze and mostly avoiding dead ends. Physical experiments with *Dictyostelium discoideum* and pancreatic cancer cells showed good agreement with simulated experiments. Given that chemotaxis is known to be inefficient over long distances, these results could help explain how cancer cells are able to metastasise away from the tumour: by degrading chemoattractants, thus creating self-generated chemical gradients by which long distance migration can occur. For a recent discussion on self-generated gradients, see [16].

1.3 Mathematical models

Mathematical models have become an important and often essential tool in understanding complex biological processes, evidenced by the abundance of models in the literature, for example, [17–19]. These models can be used to help interpret experimental data and better understand the underlying mechanisms which led to the results. They can also be used to formulate hypotheses, make predictions under perturbations and allow certain aspects of the model to be added or removed to see its effect on the overall process, all of which can then be verified experimentally.

We concentrate on quantitative models, those which describe and interpret results by linking mathematical models to quantitative data. There are many different types of quantitative models used within biology. For example, hybrid models aim to combine different mathematical modelling approaches to try and account for often complicated biological behaviours [20–23]. Whole-cell modelling aims to understand the inner working of cells by accounting for every gene and molecule within a cell [24–26]. These models are often very high-dimensional and computationally expensive but are very realistic when modelling the underlying mechanisms by which cells behave.

Neilson *et al.* [27] proposed a pseudopod-centered model for the simulation of eukaryotic cell migration and chemotaxis by coupling pseudopod formation with the movement of the cell membrane. The governing equations which drives the formation of pseudopods is based on a discrete model developed by Meinhardt [28]. This uses a set of reaction-diffusion equations to describe the interaction between an autocatalytic activator, a rapidly distributed inhibitor and a local inhibitor. These equations depend on the cell boundary which is assumed to evolve with time in two dimensions. Therefore, an Arbitrary Lagrangian-Eulerian finite element method (ALE-FEM) approach was developed to solve the reaction-diffusion equations on an evolving curve. To computationally model the evolving surface of the cell membrane, a level set method (LSM) was used. They showed that this hybrid ALE-FEM/LSM model could be used to model persistent random cell movement, which is important to capture in the absence of chemoattractants, as well as chemotaxis, with simulations matching well the migratory behaviour seen in real experiments with a group of *Dictyostelium discoideum* cells.

Subsequent developments were made to improve the performance of the model. The level set method used to move the cell boundary was computationally expensive and so an alternative approach was considered which used the parameterised finite element method (PFEM) [29]. This method was found to be far more robust and computationally efficient for evolving the cell boundary than the level set method originally proposed. A change in the model to consider positive and negative cell feedback accurately predicted multiple aspects of cell behaviour in response to chemotaxis [30]. In [31], coupled bulk-surface reaction-diffusion equations were used to model cell migration and chemotaxis. This new approach included the solution of the chemotactic field. The bulk region corresponded with the extracellular domain and the surface corresponded to the cell membrane. A moving mesh finite element method was used to approximate the solutions of the bulk-surface reaction-diffusion equations. This required the generation of meshes for the bulk and surface which were created using a moving mesh partial differential equation (MMPDE) approach. More recently, the model in [31] was extended to couple the effects of extracellular processes to intracellular processes through membrane bound receptors [32]. For a review of mathematical models for cell motility, see [33].

In terms of models for self-generated gradients, Tweedy *et al.* [6] used a hybrid individual-continuum modelling approach. They assume that cells move in a two-dimensional chamber with cells starting from the left-hand side and chemoattractant diffusing from the right-hand side. The chemoattractant profile was modelled using the diffusion equation with constant diffusion coefficient and an additional term to denote the degradation of the chemoattractant by the cells. This latter term sums the degradation contribution from all the cells by using the Kronecker delta function at a rate which depends on the cell position and takes a Michaelis-Menten form. Numerical simulation of the chemoattractant diffusion was done on a background grid using a central differences approximation of the diffusion equation and the Euler method for the cell degradation term. Since the cells do not move on this background grid, their location could take intermediary positions. Therefore, the degradation of the chemoattractant was taken from surrounding grid points, using linear interpolation to retrieve the rate of degradation based on their distance from the cell. For numerical simulation of the movement of the cells, they begin by randomly placing the cells in a small well at the left-hand side of the chamber. Cells are then moved according to a persistent random walk with a specified direction and constant speed. The angle of persistence was taken as a weighed circular average of a persistent random walk and a bias introduced by local chemical cues. The persistent direction was taken to be a random variable sampled from a wrapped normal distribution, centred on the previous angle of persistence. The bias term was taken to be the difference of occupancy for a receptor between the very front and back of the cell. Their model also accounts for contact inhibition of locomotion (CIL) [34], a process where cells that come into contact with each other repolarise and migrate away from one another. Good agreement was seen between the simulated cell movement and those from real experiments with *Dictyostelium discoideum* cells.

McLennan *et al.* [35] considered a group of neural crest (NC) cells migrating in the growing domain of a developing embryo. In their paper, the NC cells enter the domain and travel long distances by internalizing vascular endothelial growth factor (VEGF), a chemoattractant for NC cells, in order to reach a target. They formulate a two-dimensional off-lattice individual-based model for the NC cell migration. The cells are modelled as circular discs which move with a constant speed in the direction of the maximum chemoattractant gradient. For simplicity, their computational experiments are performed on a 2D growing rectangular domain. Cells are continually introduced into the domain at the left hand boundary and movement only occurs if there is space for the cells to move into. The chemoattractant VGEF was modelled using a reaction-diffusion equation and was assumed to be produced logistically throughout the domain. Their model assumes that the NC cells internalize VEGF by surface bound receptors, which has the effect of depleting the amount of VEGF is the local vicinity of the NC cells. They found that if there was no chemoattractant at the left hand boundary, then cells introduced into the domain would migrate randomly, leading to a clog up and stopping or delaying new cell intake. From this, they postulate that there must be two types of cells, so-called leading and trailing cells. The leader cells follow the chemoattractant gradients, while trailing cells use cell-cell contacts to direct their movement forward. Their model successfully matches data collected on NC cell migration.

In a later paper, McLennan *et al.* [36] refined their model in light of new experimental findings. Some computational changes include allowing a wider stream of cells to better represent multicellular stream migration, and introducing a range of intracellular distances during cell-cell contact, which improved stream cohesion and reduced stream breakup in model migration. A fruther modification to their earlier model limits the accuracy of the cells to be able to determine the chemoattractant direction. This sensing accuracy depends on the term $\Delta c/\bar{c}$, where Δc is the spatial difference in the concentration and $\bar{c}$ is the background (average) chemoattractant concentration. If this term is small enough, then the NC cells will not be able to chemotax and will instead migrate randomly. This modification was motivated by the work of Berg and Purcell [37].

Ferguson *et al.* [38] looked to draw conclusions about the drivers of movement in two cell types by applying model fitting methodologies to data and carrying out model comparison. The data consisted of cell trajectories of *Dictyostelium discoideum* cells collected in [6] and human melanoma cells collected in [14], both moving according to self-generated gradient chemotaxis. They considered six advection-diffusion partial differential equation (PDE) models for the cell density which depended on the attractant concentration where the advection term was adapted for each new model to investigate different hypotheses for the drivers of cell movement. The cell density and attractant concentration were obtained by solving the PDEs numerically by the method of lines. Ferguson *et al.* also considered three additional PDE models in a related paper to better understand the movement of the *Dictyostelium discoideum* cells collected in Tweedy *et al.* [6].

The hybrid individual-continuum model considered by Tweedy *et al.* [6] can model each cell individually and account for cell heterogeneity by allowing for different diffusion coefficients and chemical degradation rates for each cell. While extensive simulation of these models can be used to exam the emergent behaviour of the individual models, this is typically computationally expensive and makes analysis difficult. Individual-based models can be coarse-grained to give rise to differential equation models. This approach can make analysis tractable and allow for investigations into population-level properties of individual-based models. Master equations for the individual models can be derived and simplified to give differential equation models. One of the most common simplifications is called the mean-field assumption, which assumes that partitions of the modelled quantities are independent from one another. One of the main disadvantages of these mean-field models is that they only give accurate results for certain parameter regimes. A very recent paper by Nardini *et al.* [39] examines coarse-grained differential equation models and compares these with equation learned differential equation models, which have the advantage of being both accurate and analytically tractable.

PDE models, such as those considered by Ferguson *et al.* [38, 40], are based on a density of cells, and can be cheaper to implement and allow for easier analysis. However, they cannot be used to study individual cell behaviour. Additionally, these PDE models do not capture the fluctuations resulting from random individual cell behaviour seen in small populations, as they are based on the assumption of a large population size, where all fluctuations resulting from random individual cell behaviour average out.

In this thesis, we will consider using stochastic differential equations (SDEs) to model cell movement. SDEs can be used to describe the migration of individual cells, similar to individual-based models. SDE models can also be used to describe collective migration but work better for small population sizes. When the population size is taken much larger, they can become computationally expensive and so PDE models woud be better in that case. Stochastic modelling of cell movement has been explored in, for example, [41–44].

1.4 Statistical inference

Using mathematical models with physiologically relevant parameter values with the aim of replicating the results of an experiment is often called the forward problem. Equally important is the opposite: being able to estimate parameter values of a model from experimental data. This is known as the inverse problem or statistical inference. This is seldom done in this area due to the complexity of the models and availability of the data. However, there is a history of statisical inference being used for biological problems, for example, [45–48].

In the paper by Ferguson *et al.* [38], they used statistical inference to estimate the parameters of their PDE models. A total weighted log-likelihood was calculated from the cell densities at varying time points. For the inference of the parameters, they sampled with replacement the cell density data for each time point to obtain many bootstrap datasets. The total weighted-loglikehood was then maximised for each of these bootstrap datasets for each new model. This was done using the quasi-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm for the *Dictyostelium discoideum* data and the Nelder-Mead algorithm for the melanoma data; the algorithms which worked best for the two different datasets. By optimising over many re-samples of the data, values for the posterior distributions were obtained. The widely applicable information criterion (WAIC) was used to compare the different models for the two cell types. They found that the receptor saturation model – which accounts for high concentrations of the chemoattractant saturating cell receptors, hindering cell migration up the chemical gradient – was the best model for the *Dictyostelium discoideum* data; while the overcrowding model – which accounts for a slowing of cell migration when the density of cells become large due to less physical space and the effect of CIL – was the best model for the human melanoma data.

In their related paper [40], Ferguson *et al.* looked more closely at the *Dictyostelium discoideum* cell movement data collected in [6] to identify the mechanisms that are involved in their movement. For this, they considered nine advection-diffusion PDE models, similar to the models tested in [38]. The numerical simulation of the PDEs and calculation of the log-likelihood were conducted similarly as in [38]. Inference of the parameters was obtained by using Markov chain Monte Carlo (MCMC) sampling; specifically, the delayed rejection adaptive

Metropolis (DRAM) algorithm. The WAIC was again used for model comparison between the nine candidate models. From this study, they made three conclusions about the movement of the *Dictyostelium discoideum* cells that were observed in the data. First, self-generated gradients play a key role in producing the observed movement pattern: the formation of the leading wave and random migration of cells behind the wave. Secondly, allowing for the interactions between cells gives an improvement in the model performance. Lastly, spatio-temporal changes in cell behaviour have a big effect on the produced movement patterns.

To our knowledge, the papers by Ferguson *et al.* were the first and only attempt at using inference to estimate the parameters of a model describing self-generated gradient chemotaxis. While the hybrid individual-continuum model considered by Tweedy *et al.* [6] was shown to replicate self-generated gradient chemotaxis well, they did not use inference to estimate the parameters of their model.

In this thesis, we will use Bayesian inference to estimate parameter values from data. Bayesian statistics is dependant on the calculation of the posterior distribution, the distribution of the unknown parameter vector $\boldsymbol{\theta}$ of a statistical model given some observed data $\boldsymbol{y}$, through use of Bayes' theorem,

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})\,\pi(\boldsymbol{\theta})}{p(\boldsymbol{y})}, \tag{1.1}$$

where $p(\boldsymbol{y}|\boldsymbol{\theta})$, called the likelihood function, is the probability of observing the data $\boldsymbol{y}$ given some parameter values $\boldsymbol{\theta}$; $\pi(\boldsymbol{\theta})$, called the prior distribution, is the initial distribution of the unknown parameters $\boldsymbol{\theta}$; and $p(\boldsymbol{y})$ is the marginal probability of $\boldsymbol{y}$. Since models describing cell movement and chemotaxis are usually complex, calculation of the likelihood function for use in Bayesian inference is computationally expensive or intractable. In this case, likelihood-free methods can be employed, for example, approximate Bayesian computation (ABC) which we will implement in later chapters. In ABC, the posterior distribution is found not by calculation of the likelihood function, but by repeatedly sampling from the prior distribution to generate simulated data which is then compared with observed data through chosen metrics. Often the output data from these models are high-dimensional and so summary statistics of the simulated and observed data are usually compared instead. The decision of which summary statistics to use and how these can be calculated and combined optimally is a complicated matter in its own right. ABC has been widely used for parameter inference in the biological sciences [49–52] including in cell biology [53–57].

1.5 Outstanding questions

One outstanding question in this area is how to optimally infer parameters of drift-diffusion SDEs based on the regression of the MSD. This problem is of interest in a number of areas and not just cell biology. Detailed analysis shows how to optimise the number of points used in weighted least squares regression and how the quality of the parameter estimates depends on the time step used to measure the time evolution of the data. With a view to tackling parameter estimation for a complex model of self-generated gradient chemotaxis, we then considered the use of ABC methods for parameter estimation for a model drift-diffusion SDE. The key question with this approach is the correct choice of summary statistics. We looked at the use of the MSD initially and then considered more sophisticated techniques based on multivariate least squares and Gaussian Processes. We discovered that one approach (GP with residual) delivered good estimation of both model parameters and was relatively insensitive to the time interval over which the data was collected. We then applied this ABC approach to a hybrid-continuum model for self-generated chemotaxis. The model for the movement of each cell is a SDE of drift-diffusion type so experience from previous chapters can be used to devise suitable summary statistics. We demonstrate that the ABC method can be used to infer parameters of the model based on pilot summary statistics based on the ensemble MSD. We also used Morris screening method to investigate the identifiability of parameters using scalar outputs from your model and these could be used to quickly test a range of summary statistics before they are used with complex models in biology.

1.6 Outline of thesis

In this thesis, we concentrate on investigating approaches to optimally infer the parameters of a simple SDE model for directed cell movement. We also present a hybrid model describing self-generated gradient cell movement.

In Chapter 2, we use weighted least squares to estimate the diffusion coefficient and drift magnitude of a drift-diffusion SDE model. We consider two different experimental protocols: one where the data cannot be re-collected and so the optimisation is done with respect to the number of regression fitting points; and the other where experiments can be repeated and so the optimisation is done with respect to the measurement time interval over which experiments are measured. Iterative algorithms to calculate the optimal number of fitting points and the optimal measurement time interval are presented. We investigate the robustness of these algorithms to infer the diffusion coefficient and drift magnitude for different parameter choices, as well as for simulated experiments with a single particle and an ensemble of particles. Note that the work in this chapter has been published in Physical Review E as the reference [58].

In Chapter 3, we introduce in more detail the ABC method mentioned above and review a range of different ABC procedures. This is in preparation of using some of the ABC procedures in Chapter 4.

In Chapter 4, we use the ABC approaches of Prangle [59] and Fearnhead & Prangle [60] described in Chapter 3 to infer the diffusion coefficient and drift velocity of a drift-diffusion SDE model. We investigate the quality of the posterior distributions obtained using both procedures with respect to the measurement time interval and compare these with the exact posterior distributions calculated using the likelihood function. Adaptations of the Fearnhead & Prangle approach are also considered and included in the comparison.

In Chapter 5, we introduce a hybrid model to describe self-generated cell movement. The movement of the cells is governed by a drift-diffusion SDE where the drift term relates to the chemical gradient and chemoattractant concentration. The chemical profile is modelled using the diffusion equation with an additional term to model the degradation of the chemoattractant by the cells. We solve numerically the drift-diffusion SDE by the Euler-Maruyama method and the chemical equation by an implicit-explicit finite difference scheme. A sensitivity analysis of our hybrid model is provided and ABC is used to try and infer the random motility of the cells and the chemotactic velocity.

Finally, we present our conclusions and future work in Chapter 6.

Chapter 2

Inferring the drift and diffusion coefficients using regression

2.1 The drift-diffusion model

We will start by considering a simple drift-diffusion SDE model for directed cell migration. Drift-diffusion models have been used in cell biology, for example, to model leuckocyte migration [53] and for the detection of biased motion of leukocytes [61] and T cells [62]. Drift-diffusion models have also been used in other disciplines, such as understanding the causes of animal mass migration in ecology [63], monitoring crowd behaviour in social science [64, 65], and studying rumour diffusion in social networks [66]. Since these models can be used generally, for this chapter, we will refer to an entity which moves as a particle.

This chapter aims to extend the work of Michalet [67] to include drift in the analysis. To do this, a weighted least squares (WLS) fit to the MSD will be used to estimate the diffusion and drift coefficients for both single-particle and ensemble data. For this, we are required to calculate theoretical expressions for the variance and the covariance of the MSD to use as weights in the fitting process. We will then look to optimally infer the diffusion and drift coefficients by optimising over the number of regression points used in the fitting, as well as the measurement time interval over which the experiments are made.

We will assume that all the particles move in two dimensions. The true location of a particle at time t will be denoted by the random variable $\tilde{\boldsymbol{X}}_t$ and it will be assumed that it evolves according to the drift-diffusion SDE

$$\mathrm{d}\tilde{\boldsymbol{X}}_t = \boldsymbol{\alpha}\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}\boldsymbol{W}_t, \tag{2.1}$$

where the drift velocity $\boldsymbol{\alpha} = \alpha(\cos(\theta_d), \sin(\theta_d))$, where α is the drift magnitude and θ_d is the drift direction; for simplicity, we assume that α and θ_d are fixed and so do not depend on time. The diffusion coefficient is denoted by D and $\mathrm{d}\boldsymbol{W}_t = (\mathrm{d}W_1, \mathrm{d}W_2)$, where $\mathrm{d}W_1, \mathrm{d}W_2$ are independent Wiener processes. Here, the domain of $\tilde{\boldsymbol{X}}_t$ is $\mathbb{R}^2$, with the initial condition $\tilde{\boldsymbol{X}}_0 = \boldsymbol{0}$. Since the cells are assumed to move in an infinite domain, we have no boundary conditions.

Typically, the position of a particle is extracted from a sequence of digital images. The observed trajectory is the path observed using a device such as a microscope connected to a video camera. The observed trajectory can be subject to two different types of localisation error, usually referred to as static error and dynamic error [68]. Static error is the difference between the measured and true position of an immobile particle or the instantaneous position of a moving particle. The source of static error therefore comes from the spatial resolution of the measuring instrument. Dynamic errors are inaccuracies which arise when measuring particles which move in time. An example of dynamic error is motion blur which can occur due to the camera shutter being left open to maximise the number of photons being recorded in any one frame. For transport by pure diffusion it has been shown [69] that the precision of determining the diffusion constant is negligibly effected by motion blur and hence in this chapter we will assume that it can be ignored (we present some numerical simulations of motion blur in Section 2.9.4 to investigate the effect of this assumption). We will however assume that the measured position of a particle is subject to additive independent and identically distributed static error of the form $\mathcal{N}(\mathbf{0}, \eta^2 I)$, where η^2 is the variance of the static error and I is the identity matrix. Throughout this chapter, we will assume that the static error is independent of time. Note that we do not consider experimental factors which affect the level of static error such as finite frame duration and pixelization of video images; the interested reader can find these issues addressed in Savin and Doyle [68].

2.2 The mean-square displacement curve

The analysis of the resulting trajectory data has traditionally been obtained using the mean-square displacement (MSD) [68, 70–72]. The MSD is a sensible choice in this context since, as we will see later in this section, its theoretical form depends explicitly on the parameters of interest. However, the inference of parameters using MSD data has never been considered for a drift-diffusion model with the inclusion of static error. Michalet [67] and Berglund [73] used MSD analysis to infer the parameters of a pure diffusion model in the presence of static error but did not extend their analysis to include a drift term in the model. Qian *et al.* [70] were the first to consider inference using the MSD for a drift-diffusion model but did not consider the situation where static error is present in the data collection.

Assuming that the true displacement of the particles follow the drift-diffusion equation (2.1), the probability density function (PDF) for their displacement at time t is given by [74]

$$\tilde{p}(\tilde{\boldsymbol{x}}, t) = \frac{1}{4\pi D t} \exp\left(\frac{-|\tilde{\boldsymbol{x}} - \boldsymbol{\alpha} t|^2}{4Dt} \right). \tag{2.2}$$

The observed displacement – the displacement observed and recorded under a microscope – of a particle from the origin at time t will be denoted by the random variable $\boldsymbol{X}_t^o$. Since we assume that the displacements contain static error, we know that $\boldsymbol{X}_t^o = \tilde{\boldsymbol{X}}_t + \boldsymbol{Z}$, where $\boldsymbol{Z}$ is the random variable denoting the static error with PDF

$$\tilde{p}^n(\boldsymbol{z}) = \frac{1}{2\pi\eta^2} \exp\left(\frac{-|\boldsymbol{z}|^{\boldsymbol{2}}}{2\eta^2}\right), \tag{2.3}$$

then the PDF of $\boldsymbol{X}_t^o$ can be obtained by the convolution

$$p^o(\boldsymbol{x}^o, t) = \int_{\mathbb{R}^2} \tilde{p}(\boldsymbol{x^o} - \boldsymbol{z}, t)\, \tilde{p}^n(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}. \tag{2.4}$$

This is a standard result and can be easily calculated since the convolution of two Gaussian random variables is again Gaussian with mean and variance given by the sum of the corresponding means and variances [75]. Therefore, the PDF is

$$p^o(\boldsymbol{x}^o, t) = \frac{1}{2\pi(2Dt + \eta^2)} \exp\left(\frac{-|\boldsymbol{x}^o - \boldsymbol{\alpha}t|^2}{2(2Dt + \eta^2)}\right). \tag{2.5}$$

The measured displacement – the displacement extracted from the sequence of digital images – of the particles is made relative to the origin, with the addition of static error. If $\boldsymbol{X}_t$ denotes the random variable for the measured displacement, then $\boldsymbol{X}_t = \boldsymbol{X}_t^o - \boldsymbol{Z}$, and hence its PDF can be found by evaluating the convolution

$$p(\boldsymbol{x}, t) = \int_{\mathbb{R}^2} p^o(\boldsymbol{x} - \boldsymbol{z}, t)\, \tilde{p}^n(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}. \tag{2.6}$$

Again this is a standard result, giving a Gaussian random variable whose mean and variance are the difference of the two means and sum of the two variances, respectively. Therefore, the PDF is given by

$$p(\boldsymbol{x}, t) = \frac{1}{2\pi(2Dt + 2\eta^2)} \exp\left(\frac{-|\boldsymbol{x} - \boldsymbol{\alpha}t|^2}{2(2Dt + 2\eta^2)}\right). \tag{2.7}$$

The measured MSD is defined as

$$\rho(t) \equiv \mathbb{E}(|\boldsymbol{X}_t|^2) = \int_{\mathbb{R}^2} |\boldsymbol{x}|^2 p(\boldsymbol{x}, t)\, \mathrm{d}\boldsymbol{x}. \tag{2.8}$$

Using the PDF for the observed displacement (2.7) we have that

$$\rho(t) = \frac{1}{2\pi(2Dt + 2\eta^2)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x^2 + y^2) \exp\left(\frac{-|\boldsymbol{x} - \boldsymbol{\alpha} t|^2}{2(2Dt + 2\eta^2)}\right) \mathrm{d}x\mathrm{d}y. \tag{2.9}$$

With $\boldsymbol{\alpha} = (\alpha_1, \alpha_2) = (\alpha\cos(\theta_d), \alpha\sin(\theta_d))$, we use the change of variables $x = \alpha_1 t + r\cos\theta$ and $y = \alpha_2 t + r\sin\theta$ for $0 \le r < \infty$ and $0 \le \theta \le 2\pi$, then

$$\mathrm{d}x\mathrm{d}y = \begin{vmatrix} x_r & x_\theta \\ y_r & y_\theta \end{vmatrix} \mathrm{d}r\mathrm{d}\theta = \begin{vmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{vmatrix} \mathrm{d}r\mathrm{d}\theta = r\,\mathrm{d}r\mathrm{d}\theta, \tag{2.10}$$

and

$$x^2 + y^2 = \alpha^2 t^2 + 2\alpha_1 tr\cos\theta + 2\alpha_2 tr\sin\theta + r^2. \tag{2.11}$$

When these expressions are substituted into (2.9), any terms involving $\cos\theta$ or $\sin\theta$ will vanish since $\int_0^{2\pi}\cos\theta\,\mathrm{d}\theta = \int_0^{2\pi}\sin\theta\,\mathrm{d}\theta = 0$. For simplification let $\gamma = 2Dt + 2\eta^2$, giving

$$\begin{aligned} \rho(t) = \alpha^2 t^2 &\left(\frac{1}{2\pi\gamma}\int_0^{2\pi}\int_0^{\infty} r\exp\left(\frac{-r^2}{2\gamma}\right)\mathrm{d}r\mathrm{d}\theta\right) \\ &+ \frac{1}{2\pi\gamma}\int_0^{2\pi}\int_0^{\infty} r^3\exp\left(\frac{-r^2}{2\gamma}\right)\mathrm{d}r\mathrm{d}\theta. \end{aligned} \tag{2.12}$$

The two integrals evaluate to give 1 and 2γ respectively and so substituting back in the original expression for γ, we obtain the measured MSD

$$\rho(t) = \alpha^2 t^2 + 4Dt + 4\eta^2. \tag{2.13}$$

This result has been derived previously without static error; for example, by [70] and [74]. Note that $\rho(t)$ is independent of the drift angle θ_d. If this is to be determined from experimental data then a separate procedure must be used, and we outline such an approach in Section 2.9.5.

2.3 The variance of the square displacement

The variance of the measured square displacement is given by

$$\mathrm{Var}(|\boldsymbol{X}_t|^2) \equiv \mathbb{E}(|\boldsymbol{X}_t|^4) - (\mathbb{E}(|\boldsymbol{X}_t|^2))^2. \tag{2.14}$$

The latter term is simply the square of the MSD while the former term is calculated as

$$\begin{aligned}\mathbb{E}(|\boldsymbol{X}_t|^4) &= \int_{\mathbb{R}^2} |\boldsymbol{x}|^4 \, p(\boldsymbol{x}, t) \, \mathrm{d}\boldsymbol{x} \\ &= \frac{1}{2\pi\gamma} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x^2 + y^2)^2 \, \exp\left(\frac{-|\boldsymbol{x} - \boldsymbol{\alpha} t|^2}{2\gamma} \right) \mathrm{d}x\mathrm{d}y. \end{aligned} \tag{2.15}$$

Using the same change of variables as before, let $x = \alpha_1 t + r\cos\theta$ and $y = \alpha_2 t + r\sin\theta$ for $0 \le r < \infty$ and $0 \le \theta \le 2\pi$, then $\mathrm{d}x\mathrm{d}y = r\,\mathrm{d}r\mathrm{d}\theta$ and

$$\begin{aligned}(x^2 + y^2)^2 = {} & \alpha^4 t^4 + 2\alpha^2\alpha_1 t^3 r\cos\theta + 2\alpha^2\alpha_2 t^3 r\sin\theta + \alpha^2 t^2 r^2 \\ & + 2\alpha^2\alpha_1 t^3 r\cos\theta + 4\alpha_1^2 t^2 r^2 \cos^2\theta + 4\alpha_1\alpha_2 t^2 r^2 \cos\theta\sin\theta \\ & + 2\alpha_1 t r^3 \cos\theta + 2\alpha^2\alpha_2 t^3 r \sin\theta + 4\alpha_1\alpha_2 t^2 r^2 \cos\theta\sin\theta \\ & + 4\alpha_2^2 t^2 r^2 \sin^2\theta + 2\alpha_1 t r^3 \sin\theta + \alpha^2 t^2 r^2 + 2\alpha_1 t r^3 \cos\theta \\ & + 2\alpha_2 t r^3 \sin\theta + r^4. \end{aligned} \tag{2.16}$$

As before, any terms involving $\cos\theta$, $\sin\theta$ and additionally $\sin\theta\cos\theta$ will be equal to zero once integrated. This leaves

$$\begin{aligned}
\mathbb{E}(|\boldsymbol{X}_t|^4) = \alpha^4 t^4 &\left(\frac{1}{2\pi\gamma}\int_0^{2\pi}\int_0^{\infty} r\exp\left(\frac{-r^2}{2\gamma}\right)\mathrm{d}r\mathrm{d}\theta\right)\\
&+ 2\alpha^2 t^2\left(\frac{1}{2\pi\gamma}\int_0^{2\pi}\int_0^{\infty} r^3\exp\left(\frac{-r^2}{2\gamma}\right)\mathrm{d}r\mathrm{d}\theta\right)\\
&+ 4\alpha_1^2 t^2\left(\frac{1}{2\pi\gamma}\int_0^{2\pi}\int_0^{\infty} \cos^2\theta\, r^3\exp\left(\frac{-r^2}{2\gamma}\right)\mathrm{d}r\mathrm{d}\theta\right)\\
&+ 4\alpha_2^2 t^2\left(\frac{1}{2\pi\gamma}\int_0^{2\pi}\int_0^{\infty} \sin^2\theta\, r^3\exp\left(\frac{-r^2}{2\gamma}\right)\mathrm{d}r\mathrm{d}\theta\right)\\
&+ \frac{1}{2\pi\gamma}\int_0^{2\pi}\int_0^{\infty} r^5\exp\left(\frac{-r^2}{2\gamma}\right)\mathrm{d}r\mathrm{d}\theta. \qquad (2.17)
\end{aligned}$$

These integrals are equal to 1, 2γ, γ, γ and $8\gamma^2$, respectively. Therefore, substituting in these quantities gives

$$\begin{aligned}
\mathbb{E}(|\boldsymbol{X}_t|^4) &= \alpha^4 t^4 + 4\alpha^2 t^2\gamma + 4\alpha_1^2 t^2\gamma + 4\alpha_2^2 t^2\gamma + 8\gamma^2\\
&= \alpha^4 t^4 + 8\alpha^2 t^2\gamma + 8\gamma^2. \qquad (2.18)
\end{aligned}$$

Therefore, the variance of the square displacement is given by

$$\begin{aligned}
\mathrm{Var}(|\boldsymbol{X}_t|^2) &= \mathbb{E}(|\boldsymbol{X}_t|^4) - (\mathbb{E}(|\boldsymbol{X}_t|^2))^2\\
&= \alpha^4 t^4 + 8\alpha^2 t^2\gamma + 8\gamma^2 - (\alpha^2 t^2 + 2\gamma)^2\\
&= 4\alpha^2 t^2\gamma + 4\gamma^2\\
&= 4\alpha^2 t^2(2Dt + 2\eta^2) + 4(2Dt + 2\eta^2)^2. \qquad (2.19)
\end{aligned}$$

To our knowledge, this result has never been explicitly stated before, but has previously been derived without drift [67]. In the absence of drift it is clear that $\mathrm{Var}(|\boldsymbol{X}_t|^2) = (\rho(t))^2$ as the PDF for the measured squared displacement is an exponential distribution [67]. However, when drift is present then $\mathrm{Var}(|\boldsymbol{X}_t|^2) \neq (\rho(t))^2$ and hence the PDF for the squared displacements cannot be exponential. It is interesting to note that the variance of the squared displacement grows cubically in time when drift is present, whereas it only grows quadratically in the absence of drift. This observation has important implications when we consider which time interval to take measurements over as time intervals which are too large may result in extremely noisy estimates of the MSD.

2.4 Quantities retrieved from experimental data

In terms of the experimental data, we will assume that there are N_S observed trajectories, each comprising of particle coordinates using equal time interval between frames $t_n = (n-1)T/N = (n-1)\Delta t$, $n = 1, \dots N+1$, covering the measurement time range $[0, T]$. The entire observed experimental data will therefore be denoted as

$$\boldsymbol{x}_n^{(j)} = (x_n^{(j)}, y_n^{(j)})^T, \qquad 1 \leq n \leq N+1, \qquad 1 \leq j \leq N_S. \tag{2.20}$$

There are many possible ways to estimate the MSD [67] but the most widely used is the time-average overlapping MSD. This way of estimating the MSD extracts the most amount of information from the trajectory data, compared with the other methods, although it results in the samples being correlated which can make analysis more difficult. Since we have access to an ensemble of trajectories, we will consider the ensemble time-average overlapping MSD. This is constructed by first calculating N_S time-averaged MSDs

$$\rho_n^{(j)} = \frac{1}{N+1-n} \sum_{i=1}^{N+1-n} |\boldsymbol{x}_{i+n}^{(j)} - \boldsymbol{x}_i^{(j)}|^2, \quad n = 1, \dots, N, \quad j = 1 \dots, N_S, \tag{2.21}$$

then averaging over the trajectories to obtain

$$\rho_n = \frac{1}{N_S}\sum_{j=1}^{N_S} \rho_n^{(j)}, \quad n = 1, \ldots, N. \tag{2.22}$$

We assume that the MSD is always calculated starting from the initial time point, as shown in (2.21). This means that the value of the MSD at time point t_n is calculated using a time lag of n.

2.5 The variance of the MSD

We will use a weighted least squares (WLS) fit to the ρ_n values later in the chapter to estimate the parameters in the model and for this we require the variance σ_n^2 of ρ_n.

First of all, we calculate the variance of the MSD for a single particle, denoted by $\sigma_n^{2(S)}$. Then, due to the independence of the N_S trajectories, the variance of the MSD for an ensemble of particles is given by

$$\sigma_n^2 = \frac{\sigma_n^{2(S)}}{N_S}. \tag{2.23}$$

By denoting

$$\rho_n^{(1)} = \frac{1}{N+1-n}\sum_{i=1}^{N+1-n} |\boldsymbol{x}_{i+n} - \boldsymbol{x}_i|^2, \quad n = 1, \ldots, N, \tag{2.24}$$

as the overlapping MSD for a single particle, then the variance of the MSD is defined as

$$\sigma_n^{2(S)} = \mathbb{E}((\rho_n^{(1)})^2) - (\mathbb{E}(\rho_n^{(1)}))^2. \tag{2.25}$$

Letting $K = N + 1 - n$ be the number of samples of squared displacements of length $n\Delta t$, then, due to the correlation between overlapping displacements, we have that

$$\begin{aligned} \sigma_n^{2(S)} &= \sum_{i=1}^{K}\sum_{j=1}^{K} \mathrm{Cov}(i,j) \\ &= \frac{\mathrm{Var}(|\boldsymbol{X}_t|^2|)(n\Delta t)}{K} + \frac{2}{K^2}\sum_{i=1}^{K}\sum_{j=i+1}^{K} \mathrm{Cov}(i,j), \quad (i < j) \end{aligned} \tag{2.26}$$

where $\mathrm{Var}(|\boldsymbol{X}_t|^2|)(n\Delta t)$ is the variance of the square displacement at time point $n\Delta t$ and

$$\mathrm{Cov}(i,j) = \mathbb{E}(|\boldsymbol{x}_{i+n} - \boldsymbol{x}_i|^2|\boldsymbol{x}_{j+n} - \boldsymbol{x}_j|^2) - \mathbb{E}(|\boldsymbol{x}_{i+n} - \boldsymbol{x}_i|^2)\mathbb{E}(|\boldsymbol{x}_{j+n} - \boldsymbol{x}_j|^2) \tag{2.27}$$

is the covariance between square displacements. The first term in (2.26) can be calculated from (2.19) and so in order to calculate the variance of the MSD we need only to calculate the covariance of the square displacements. For this, we have to consider the different ways the displacements $|\boldsymbol{x}_{i+n} - \boldsymbol{x}_i|^2$ and $|\boldsymbol{x}_{j+n} - \boldsymbol{x}_j|^2$ can overlap. This can be split into two cases as shown in Figure 2.1, where $\boldsymbol{R}_1, \boldsymbol{R}_2$ and $\boldsymbol{R}_3$ are the displacements between the given vertices. We can then write the covariance as

$$\begin{aligned} &(a) \quad \mathrm{Cov}(i,j) = \mathbb{E}(\boldsymbol{R}_1^2\boldsymbol{R}_3^2) - \mathbb{E}(\boldsymbol{R}_1^2)\mathbb{E}(\boldsymbol{R}_3^2), && j > i + n, \\ &(b) \quad \mathrm{Cov}(i,j) = \mathbb{E}((\boldsymbol{R}_1 + \boldsymbol{R}_2)^2(\boldsymbol{R}_2 + \boldsymbol{R}_3)^2) \\ &\qquad\qquad - \mathbb{E}((\boldsymbol{R}_1 + \boldsymbol{R}_2)^2)\,\mathbb{E}((\boldsymbol{R}_2 + \boldsymbol{R}_3)^2), && j \leq i + n. \end{aligned} \tag{2.28}$$

Notice that when $j > i + n$ (case (a) in Figure 2.1) the trajectories do not overlap and so $\boldsymbol{R}_2 = \boldsymbol{0}$. Since the remaining displacements $\boldsymbol{R}_1$ and $\boldsymbol{R}_3$ are independent then for (2.28a) we have that $\mathrm{Cov}(i,j) = 0$. Therefore, we need only consider

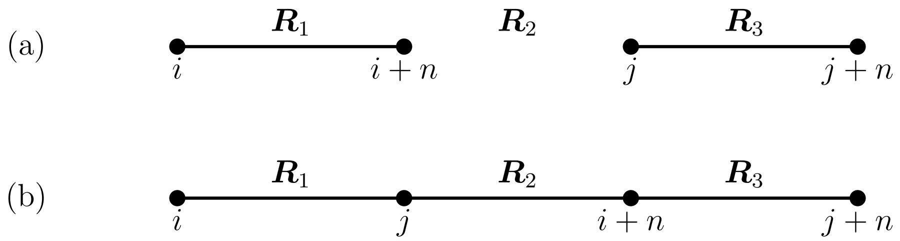

Figure 2.1: Diagram showing the two different combinations for overlapping displacements used to calculate the variance of the MSD. The two cases are (a) $j > i + n$, and (b) $j \leq i + n$.

the case when $j \leq i + n$ (case (b) in Figure 2.1). In this case, we can write the displacement vectors as

$$\boldsymbol{R}_1 = \boldsymbol{\alpha}\, \mathrm{d}t_1 + \sqrt{2D}\, \mathrm{d}\boldsymbol{W}_1 + \boldsymbol{\psi}_1, \tag{2.29}$$

$$\boldsymbol{R}_2 = \boldsymbol{\alpha}\, \mathrm{d}t_2 + \sqrt{2D}\, \mathrm{d}\boldsymbol{W}_2 + \boldsymbol{\psi}_2, \tag{2.30}$$

$$\boldsymbol{R}_3 = \boldsymbol{\alpha}\, \mathrm{d}t_3 + \sqrt{2D}\, \mathrm{d}\boldsymbol{W}_3 + \boldsymbol{\psi}_3, \tag{2.31}$$

where $t_1 = t_j - t_i = (j - i)\Delta t$, $t_2 = t_{i+n} - t_j = (i + n - j)\Delta t$, $t_3 = t_{j+n} - t_{i+n} = (j - i)\Delta t$, $\mathrm{d}\boldsymbol{W}_1$, $\mathrm{d}\boldsymbol{W}_2$, $\mathrm{d}\boldsymbol{W}_3$ are Weiner increments over the time length t_1, t_2, t_3, respectively, and $\boldsymbol{\psi}_1 = \boldsymbol{\eta}_j - \boldsymbol{\eta}_i$, $\boldsymbol{\psi}_2 = \boldsymbol{\eta}_{i+n} - \boldsymbol{\eta}_j$, $\boldsymbol{\psi}_3 = \boldsymbol{\eta}_{j+n} - \boldsymbol{\eta}_{i+n}$, where $\boldsymbol{\eta}_i$ is the static error at time t_i, etc. Note that when $j = i + n$, we have that $\boldsymbol{R}_2 = \mathbf{0}$.

Expanding the right hand side of equation $(2.28b)$ results in

$$\begin{aligned}\mathrm{Cov}(i,j) &= \mathbb{E}(|\boldsymbol{R}_1|^2|\boldsymbol{R}_2|^2) + 2\mathbb{E}(|\boldsymbol{R}_1|^2(\boldsymbol{R}_2\cdot\boldsymbol{R}_3)) + \mathbb{E}(|\boldsymbol{R}_1|^2|\boldsymbol{R}_3|^2)\\ &+ 2\mathbb{E}(|\boldsymbol{R}_2|^2(\boldsymbol{R}_1\cdot\boldsymbol{R}_2)) + 4\mathbb{E}((\boldsymbol{R}_1\cdot\boldsymbol{R}_2)(\boldsymbol{R}_2\cdot\boldsymbol{R}_3)) + 2\mathbb{E}(|\boldsymbol{R}_3|^2(\boldsymbol{R}_1\cdot\boldsymbol{R}_2))\\ &+ \mathbb{E}(|\boldsymbol{R}_2|^4) + 2\mathbb{E}(|\boldsymbol{R}_2|^2(\boldsymbol{R}_2\cdot\boldsymbol{R}_3)) + \mathbb{E}(|\boldsymbol{R}_2|^2|\boldsymbol{R}_3|^2)\\ &- \Big(\mathbb{E}(|\boldsymbol{R}_1|^2)\mathbb{E}(|\boldsymbol{R}_2|^2) + 2\mathbb{E}(|\boldsymbol{R}_1|^2)\mathbb{E}(\boldsymbol{R}_2\cdot\boldsymbol{R}_3) + \mathbb{E}(|\boldsymbol{R}_1|^2)\mathbb{E}(|\boldsymbol{R}_3|^2)\\ &+ 2\mathbb{E}(|\boldsymbol{R}_2|^2)\mathbb{E}(\boldsymbol{R}_1\cdot\boldsymbol{R}_2) + 4\mathbb{E}(\boldsymbol{R}_1\cdot\boldsymbol{R}_2)\mathbb{E}(\boldsymbol{R}_2\cdot\boldsymbol{R}_3)\\ &+ 2\mathbb{E}(|\boldsymbol{R}_3|^2)\mathbb{E}(\boldsymbol{R}_1\cdot\boldsymbol{R}_2) + (\mathbb{E}(|\boldsymbol{R}_2|^2))^2 + 2\mathbb{E}(|\boldsymbol{R}_2|^2)\mathbb{E}(\boldsymbol{R}_2\cdot\boldsymbol{R}_3)\\ &+ \mathbb{E}(|\boldsymbol{R}_2|^2)\mathbb{E}(|\boldsymbol{R}_3|^2)\Big). \end{aligned} \tag{2.32}$$

The first term is calculated as

$$\begin{aligned}\mathbb{E}(|\boldsymbol{R}_1|^2|\boldsymbol{R}_2|^2) &= \mathbb{E}(|\boldsymbol{\alpha}\,\mathrm{d}t_1 + \sqrt{2D}\,\mathrm{d}\boldsymbol{W}_1 + \boldsymbol{\psi}_1|^2|\boldsymbol{\alpha}\,\mathrm{d}t_2 + \sqrt{2D}\,\mathrm{d}\boldsymbol{W}_2 + \boldsymbol{\psi}_2|^2)\\ &= \mathbb{E}\Big(\big(|\boldsymbol{\alpha}|^2\mathrm{d}t_1^2 + 2\sqrt{2D}\mathrm{d}t_1(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_1) + 2\mathrm{d}t_1(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_1) + 2D|\mathrm{d}\boldsymbol{W}_1|^2\\ &+ 2\sqrt{2D}(\mathrm{d}\boldsymbol{W}_1\cdot\boldsymbol{\psi}_1) + |\boldsymbol{\psi}_1|^2\big)\big(|\boldsymbol{\alpha}|^2\mathrm{d}t_2^2 + 2\sqrt{2D}\mathrm{d}t_2(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_2)\\ &+ 2\mathrm{d}t_2(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_2) + 2D|\mathrm{d}\boldsymbol{W}_2|^2 + 2\sqrt{2D}(\mathrm{d}\boldsymbol{W}_2\cdot\boldsymbol{\psi}_2) + |\boldsymbol{\psi}_2|^2\big)\Big). \end{aligned} \tag{2.33}$$

Multiplying out the brackets and taking the expectation of each term gives

$$
\begin{aligned}
\mathbb{E}(|\boldsymbol{R}_1|^2|\boldsymbol{R}_2|^2) &= \alpha^4\mathbb{E}(\mathrm{d}t_1^2\mathrm{d}t_2^2) + 2\sqrt{2D}\alpha^2\mathbb{E}(\mathrm{d}t_1^2\mathrm{d}t_2(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_2)) \\
&+ 2\alpha^2\mathbb{E}(\mathrm{d}t_1^2\mathrm{d}t_2(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_2)) + 2D\alpha^2\mathbb{E}(\mathrm{d}t_1^2|\mathrm{d}\boldsymbol{W}_2|^2) \\
&+ 2\sqrt{2D}\alpha^2\mathbb{E}(\mathrm{d}t_1^2(\mathrm{d}\boldsymbol{W}_2\cdot\boldsymbol{\psi}_2)) + \alpha^2\mathbb{E}(\mathrm{d}t_1^2|\boldsymbol{\psi}_2|^2) \\
&+ 2\sqrt{2D}\alpha^2\mathbb{E}(\mathrm{d}t_1\mathrm{d}t_2^2(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_1)) \\
&+ 8D\mathbb{E}(\mathrm{d}t_1\mathrm{d}t_2(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_1)(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_2)) \\
&+ 4\sqrt{2D}\mathbb{E}(\mathrm{d}t_1\mathrm{d}t_2(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_1)(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_2)) \\
&+ 4D\sqrt{2D}\mathbb{E}(\mathrm{d}t_1|\mathrm{d}\boldsymbol{W}_2|^2(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_1)) \\
&+ 8D\mathbb{E}(\mathrm{d}t_1(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_1)(\mathrm{d}\boldsymbol{W}_2\cdot\boldsymbol{\psi}_2)) + 2\sqrt{2D}\mathbb{E}(\mathrm{d}t_1|\boldsymbol{\psi}_2|^2(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_1)) \\
&+ 2\alpha^2\mathbb{E}(\mathrm{d}t_1\mathrm{d}t_2^2(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_1)) + 4\sqrt{2D}\mathbb{E}(\mathrm{d}t_1\mathrm{d}t_2(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_1)(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_2)) \\
&+ 4\mathbb{E}(\mathrm{d}t_1\mathrm{d}t_2(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_1)(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_2)) + 4D\mathbb{E}(\mathrm{d}t_1|\mathrm{d}\boldsymbol{W}_2|^2(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_1)) \\
&+ 4\sqrt{2D}\mathbb{E}(\mathrm{d}t_1(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_1)(\mathrm{d}\boldsymbol{W}_2\cdot\boldsymbol{\psi}_2)) + 2\mathbb{E}(\mathrm{d}t_1|\boldsymbol{\psi}_2|^2(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_1)) \\
&+ 2D\alpha^2\mathbb{E}(\mathrm{d}t_2^2|\mathrm{d}\boldsymbol{W}_1|^2) + 4D\sqrt{2D}\mathbb{E}(\mathrm{d}t_2|\mathrm{d}\boldsymbol{W}_1|^2(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_2)) \\
&+ 4D\mathbb{E}(\mathrm{d}t_2|\mathrm{d}\boldsymbol{W}_1|^2(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_2)) + 4D^2\mathbb{E}(|\mathrm{d}\boldsymbol{W}_1|^2|\mathrm{d}\boldsymbol{W}_2|^2) \\
&+ 4D\sqrt{2D}\mathbb{E}(|\mathrm{d}\boldsymbol{W}_1|^2(\mathrm{d}\boldsymbol{W}_2\cdot\boldsymbol{\psi}_2)) + 2D\mathbb{E}(|\mathrm{d}\boldsymbol{W}_1|^2|\boldsymbol{\psi}_2|^2) \\
&+ 2\sqrt{2D}\alpha^2\mathbb{E}(\mathrm{d}t_2^2(\mathrm{d}\boldsymbol{W}_1\cdot\boldsymbol{\psi}_1)) + 8D\mathbb{E}(\mathrm{d}t_2(\mathrm{d}\boldsymbol{W}_1\cdot\boldsymbol{\psi}_1)(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_2)) \\
&+ 4\sqrt{2D}\mathbb{E}(\mathrm{d}t_2(\mathrm{d}\boldsymbol{W}_1\cdot\boldsymbol{\psi}_1)(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_2)) \\
&+ 4D\sqrt{2D}\mathbb{E}(|\mathrm{d}\boldsymbol{W}_2|^2(\mathrm{d}\boldsymbol{W}_1\cdot\boldsymbol{\psi}_1)) \\
&+ 8D\mathbb{E}((\mathrm{d}\boldsymbol{W}_1\cdot\boldsymbol{\psi}_1)(\mathrm{d}\boldsymbol{W}_2\cdot\boldsymbol{\psi}_2)) + 2\sqrt{2D}\mathbb{E}(|\boldsymbol{\psi}_2|^2(\mathrm{d}\boldsymbol{W}_1\cdot\boldsymbol{\psi}_1)) \\
&+ \alpha^2\mathbb{E}(\mathrm{d}t_2^2|\boldsymbol{\psi}_1|^2) + 2\sqrt{2D}\mathbb{E}(\mathrm{d}t_2|\boldsymbol{\psi}_1|^2(\boldsymbol{\alpha}\cdot\mathrm{d}\boldsymbol{W}_2)) \\
&+ 2\mathbb{E}(\mathrm{d}t_2|\boldsymbol{\psi}_1|^2(\boldsymbol{\alpha}\cdot\boldsymbol{\psi}_2)) + 2D\mathbb{E}(|\mathrm{d}\boldsymbol{W}_2|^2|\boldsymbol{\psi}_1|^2) \\
&+ 2\sqrt{2D}\mathbb{E}(|\boldsymbol{\psi}_1|^2(\mathrm{d}\boldsymbol{W}_2\cdot\boldsymbol{\psi}_2)) + \mathbb{E}(|\boldsymbol{\psi}_1|^2|\boldsymbol{\psi}_2|^2).
\end{aligned}
\tag{2.34}
$$

From here, we note that $\mathbb{E}(\mathrm{d}t_i) = t_i$ for $i \in \{1, 2, 3\}$, and since the Weiner processes are independent, then $(\mathrm{d}\boldsymbol{W}_i \cdot \mathrm{d}\boldsymbol{W}_j) = 0$ for $i \neq j$. Furthermore, since the Weiner processes and static error terms are independent then $(\mathrm{d}\boldsymbol{W}_i \cdot \boldsymbol{\psi}_j) = 0$ for $i, j \in \{1, 2, 3\}$. However, due to the static error terms containing variances of the static error which can share a common vertex, then $(\boldsymbol{\psi}_i \cdot \boldsymbol{\psi}_j) \neq 0$ for $|i-j| = 1$. Using these properties, many of the terms in (2.34) will be zero. Therefore, we are left with

$$\begin{aligned}\mathbb{E}(|\boldsymbol{R}_1|^2|\boldsymbol{R}_2|^2) &= \alpha^4 t_i^2 t_2^2 + 2D\alpha^2 t_1^2 \mathbb{E}(|\mathrm{d}\boldsymbol{W}_2|^2) + \alpha^2 t_1^2 \mathbb{E}(|\boldsymbol{\psi}_2|^2) \\ &\quad + 4t_1 t_2 \mathbb{E}((\boldsymbol{\alpha} \cdot \boldsymbol{\psi}_1)(\boldsymbol{\alpha} \cdot \boldsymbol{\psi}_2)) + 2D\alpha^2 t_2^2 \mathbb{E}(|\mathrm{d}\boldsymbol{W}_1|^2) \\ &\quad + 4D^2 \mathbb{E}(|\mathrm{d}\boldsymbol{W}_1|^2 |\mathrm{d}\boldsymbol{W}_2|^2) + 2D\mathbb{E}(|\mathrm{d}\boldsymbol{W}_1|^2 |\boldsymbol{\psi}_2|^2) \\ &\quad + \alpha^2 t_2^2 \mathbb{E}(|\boldsymbol{\psi}_1|^2) + 2D\mathbb{E}(|\mathrm{d}\boldsymbol{W}_2|^2 |\boldsymbol{\psi}_1|^2) + \mathbb{E}(|\boldsymbol{\psi}_1|^2 |\boldsymbol{\psi}_2|^2). \qquad (2.35)\end{aligned}$$

We have that $\mathrm{d}\boldsymbol{W}_i = (\mathrm{d}W_i^1, \mathrm{d}W_i^2)$ for $i \in \{1, 2, 3\}$, where $\mathrm{d}W_i^1, \mathrm{d}W_i^2$ are independent and identically distributed (i.i.d.) normal random variables with zero mean and variance $\sqrt{t_i}$, i.e. $\mathrm{d}W_i^1, \mathrm{d}W_i^2$ are of the form $\mathcal{N}(0, \sqrt{t_i})$. Similarly for the static error term, we have, for example, $\boldsymbol{\psi}_1 = \boldsymbol{\eta}_j - \boldsymbol{\eta}_i$, where $\boldsymbol{\eta}_i = (\eta_i^1, \eta_i^2)$ and η_i^1, η_i^2 are also i.i.d. random variables of the form $\mathcal{N}(0, \eta^2)$. These can be used to calculate the final expectation terms left in (2.35). This gives

$$\begin{aligned}\mathbb{E}(|\boldsymbol{R}_1|^2|\boldsymbol{R}_2|^2) &= \alpha^4 t_1^2 t_2^2 + 4D\alpha^2 t_1^2 t_2 + 4\alpha^2\eta^2 t_1^2 \\ &\quad - 4\alpha^2\eta^2 t_1 t_2 + 4D\alpha^2 t_1 t_2^2 + 16D^2 t_1 t_2 + 16D\eta^2 t_1 \\ &\quad + 4\alpha^2\eta^2 t_2^2 + 16D\eta^2 t_2 + 20\eta^4. \qquad (2.36)\end{aligned}$$

The rest of the expectations will not be shown here but use similar algebra. Instead, we will simply present the results.

$$
\begin{aligned}
&\mathbb{E}(\boldsymbol{R}_i \cdot \boldsymbol{R}_j) = \alpha^2 t_i t_j - 2\eta^2, && |i-j| = 1,\\
&\mathbb{E}(\boldsymbol{R}_1 \cdot \boldsymbol{R}_3) = \alpha^2 t_1 t_3, &&\\
&\mathbb{E}(|\boldsymbol{R}_i|^2) = \alpha^2 t_i^2 + 4Dt_i + 4\eta^2, && i \in \{1,2,3\},\\
&\mathbb{E}(|\boldsymbol{R}_2|^4) = \alpha^4 t_2^4 + 16D\alpha^2 t_2^3 + 16\alpha^2\eta^2 t_2^2 &&\\
&\qquad + 32D^2 t_2^2 + 64D\eta^2 t_2 + 32\eta^4, &&\\
&\mathbb{E}(|\boldsymbol{R}_i|^2|\boldsymbol{R}_j|^2) = \alpha^4 t_i^2 t_j^2 + 4D\alpha^2 t_i^2 t_j + 4\alpha^2\eta^2 t_i^2 &&\\
&\qquad - 4\alpha^2\eta^2 t_i t_j + 4D\alpha^2 t_i t_j^2 + 16D^2 t_i t_j + 16D\eta^2 t_i &&\\
&\qquad + 4\alpha^2\eta^2 t_j^2 + 16D\eta^2 t_j + 20\eta^4, && |i-j| = 1,\\
&\mathbb{E}(|\boldsymbol{R}_1|^2|\boldsymbol{R}_3|^2) = \alpha^4 t_1^2 t_3^2 + 4D\alpha^2 t_1^2 t_3 + 4\alpha^2\eta^2 t_1^2 &&\\
&\qquad + 4D\alpha^2 t_1 t_3^2 + 16D^2 t_1 t_3 + 16D\eta^2 t_1 + 4\alpha^2\eta^2 t_3^2 &&\\
&\qquad + 16D\eta^2 t_3 + 16\eta^4 + (4\eta^4 - 4\alpha^2\eta^2 t_1 t_3)\delta_{j,\,i+n}, &&\\
&\mathbb{E}(|\boldsymbol{R}_i|^2(\boldsymbol{R}_i \cdot \boldsymbol{R}_j)) = \alpha^4 t_i^3 t_j - 4\alpha^2\eta^2 t_i^2 + 8D\alpha^2 t_i^2 t_j &&\\
&\qquad - 16D\eta^2 t_i + 8\alpha^2\eta^2 t_i t_j - 16\eta^4, && i \neq j,\\
&\mathbb{E}(|\boldsymbol{R}_i|^2(\boldsymbol{R}_j \cdot \boldsymbol{R}_k)) = \alpha^4 t_i^2 t_j t_k + 4D\alpha^2 t_i t_j t_k + 4\alpha^2\eta^2 t_j t_k &&\\
&\qquad - 8D\eta^2 t_i - 8\eta^4 - 2\alpha^2\eta^2 t_i^2 - 2\alpha^2\eta^2 t_i t_k, && i \neq j \neq k,\, i \neq 2,\\
&\mathbb{E}(|\boldsymbol{R}_2|^2(\boldsymbol{R}_1 \cdot \boldsymbol{R}_3)) = \alpha^4 t_1 t_2^2 t_3 + 4D\alpha^2 t_1 t_2 t_3 + 4\alpha^2\eta^2 t_1 t_3 &&\\
&\qquad + 4\eta^4 - 2\alpha^2\eta^2 t_1 t_3 - 2\alpha^2\eta^2 t_2 t_3, &&\\
&\mathbb{E}((\boldsymbol{R}_1 \cdot \boldsymbol{R}_2)(\boldsymbol{R}_2 \cdot \boldsymbol{R}_3)) = \alpha^4 t_1 t_2^2 t_3 - 3\alpha^2\eta^2 t_1 t_2 - 3\alpha^2\eta^2 t_2 t_3 &&\\
&\qquad + 2\alpha^2\eta^2 t_1 t_3 + 2D\alpha^2 t_1 t_2 t_3 + 6\eta^4. && (2.37)
\end{aligned}
$$

Here,

$$\delta_{j,\,i+n} = \begin{cases} 0, & \text{when } j \neq i+n, \\ 1, & \text{when } j = i+n, \end{cases} \tag{2.38}$$

represents the Kronecker delta function. In the absense of drift, we recover the same expressions as found in the supplementary material of Michalet's paper [67]. Substituting these results into (2.32) and simplifying yields

$$\begin{aligned} \mathrm{Cov}(i,j) \equiv \mathrm{Cov}(l) = 16D^2((n-l)\Delta t)^2 + 8D\alpha^2 n^2 (n-l)(\Delta t)^3 \\ + (4\eta^4 - 4(\alpha\eta n\Delta t)^2)\delta_{j,\,i+n}, \end{aligned} \tag{2.39}$$

where $l = j - i$. Now that the covariance has been calculated, to obtain the variance of the MSD we must sum all the non-zero terms appearing in the double sum in (2.26). This can be split into two cases. When $n \leq K$ we sum the non-zero terms along the diagonals $l = 1, ..., n-1$ of the covariance matrix. The l^{th} diagonal contains $K - l$ identical elements, and therefore the non-zero terms are

$$\sum_{l=1}^{n-1} (K-l)\mathrm{Cov}(l), \tag{2.40}$$

where $\mathrm{Cov}(l) = 16D^2((n-l)\Delta t)^2 + 8D\alpha^2 n^2(n-l)(\Delta t)^3$. When $l = n$ this corresponds to the case when $j = i + n$ and so the Kronecker delta term will now be non-zero. This diagonal will contain $K - n = N + 1 - 2n$ elements, and so will equal $(N + 1 - 2n)(4\eta^4 - 4(\alpha\eta n\Delta t)^2)$. When $n > K$ we don't have any segments with no overlaps. Therefore, we only have overlapping contributions to the covariance matrix. This time the upper limit in the sum (2.40) is $N - n$ rather than $n - 1$. So when $n > K$ we have

$$\sum_{l=1}^{N-n} (K-l)\mathrm{Cov}(l), \tag{2.41}$$

where $\mathrm{Cov}(l) = 16D^2((n-l)\Delta t)^2 + 8D\alpha^2 n^2 (n-l)(\Delta t)^3$. Summing all these terms in either case ($n \leq K$ and $n > K$) as above allows us to calculate σ_n^2. The final result is

$$\sigma_n^2 = \begin{cases} \begin{aligned} &\Big[\tfrac{n}{6K^2}(4n^2K + 2K - n^3 + n)(4D\Delta t)^2 \\ &+8\alpha^2 D(\Delta t)^3\Big(\tfrac{n^3}{3K^2}(3Kn + 1 - n^2)\Big) \\ &+\tfrac{8\eta^2}{K^2}\Big((K-n)(\eta^2 - (\alpha n\Delta t)^2) \\ &+K((\alpha n\Delta t)^2 + 4Dn\Delta t + 2\eta^2)\Big)\Big]/N_S, \end{aligned} & n \leq K, \\ \\ \begin{aligned} &\Big[\tfrac{1}{6K}(6n^2K - 4nK^2 + 4n + K^3 - K)(4D\Delta t)^2 \\ &+8\alpha^2 D(\Delta t)^3\Big(\tfrac{n^2}{3K}(3nK - K^2 + 1)\Big) \\ &+\tfrac{8\eta^2}{K}((\alpha n\Delta t)^2 + 4Dn\Delta t + 2\eta^2)\Big]/N_S, \end{aligned} & n > K. \end{cases} \tag{2.42}$$

To our knowledge, this result has never been derived before with the inclusion of drift and static error. Note that in the absence of drift, the formulae above reduces to that appearing in [67]. In Figure 2.2 we plot the theoretical variance of the MSD (2.42) along with an empirical estimate of the variance of (2.22) averaged over 1000 and 10,000 samples for the parameter values $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $\eta = 2\,\mu\mathrm{m}$, $N_S = 10$, $N = 100$ and $T = 100\,\mathrm{s}$. We can see that as we increase the number of samples, the empirical estimate gets closer to the theoretical expression (2.42).

To investigate the behaviour of the MSD (2.9), as well as the quality of the ensemble time-averaged estimate (2.22), simulated data was obtained by solving numerically the drift-diffusion SDE (2.1) by the Euler-Maruyama method using a time step of Δt with $N_S = 10$ trajectories and $N = 100$ time points. Note that this choice for the time step is sufficient due to the simple form of our SDE. Figure 2.3 shows a plot of the theoretical MSD $\rho(t)$ compared with the estimate

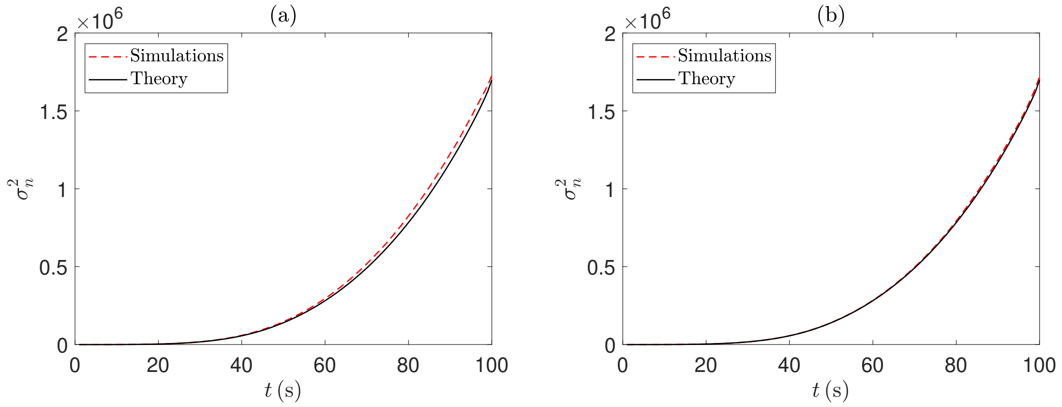

Figure 2.2: A plot of the theoretical variance of the MSD (2.42) (solid black line) along with an empirical estimate of the variance of (2.22) (dashed red line) averaged over 1000 samples (a) and 10,000 samples (b) for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$, $N = 100$ and $T = 100\,\text{s}$.

ρ_n. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$ and $\eta = 2\,\mu\text{m}$. To illustrate the uncertainty in ρ_n, Figure 2.3 also includes $\rho_n \pm \sigma_n$, where σ_n is the standard deviation of the estimated MSD. Both the theoretical σ_n given by (2.42) and an empirical estimate of σ_n, obtained using 100 independent sample values of ρ_n, are shown. Figure 2.3(a) shows simulations with a time interval of $T = 4\,\text{s}$, while Figure 2.3(b) shows simulations with the same parameter values but with a larger time interval of $T = 100\,\text{s}$. We can see that as time increases the size of the uncertainty in ρ_n increases, and for small times ρ_n does not approximate $\rho(t)$ well. This suggests a sufficiently large T is required in order to approximate the MSD accurately. We have also observed that choosing T too small lowers the accuracy of inferring the drift velocity, while taking the interval too large lowers the accuracy of inferring the diffusion coefficient. This is due to the quadratic form of the MSD, giving rise to two different time scales for the diffusive and drift processes.

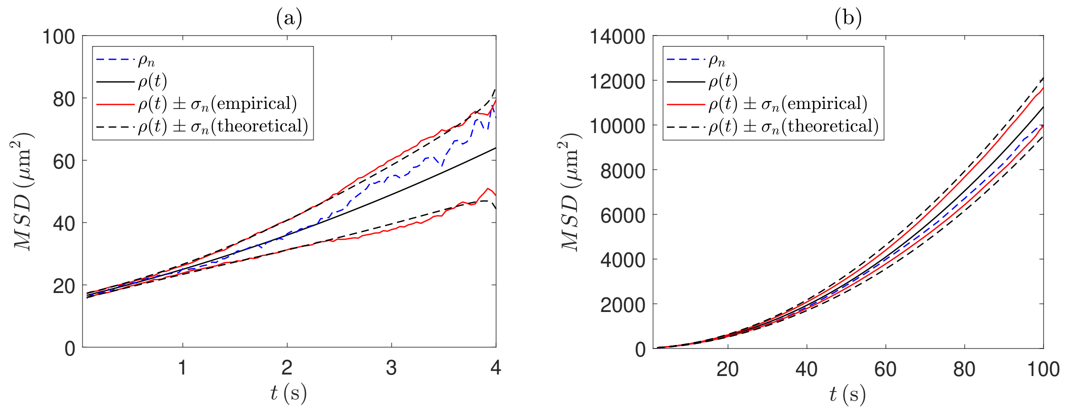

Figure 2.3: A plot of the theoretical MSD curve (2.9) (solid black line), the ensemble time-averaged estimate ρ_n (2.22) (dashed blue line), along with $\rho(t)\pm\sigma_n$, where σ_n is estimated empirically using 100 samples (solid red line) and $\rho(t)\pm\sigma_n$ where σ_n is given by (2.42) (dashed black line) for $n = 1, \ldots, N$, $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $\eta = 2\,\mu\mathrm{m}$, $N_S = 10$ and $N = 100$, for an end time of $T = 4\,\mathrm{s}$ (a) and $T = 100\,\mathrm{s}$ (b).

2.6 Variance of the regression coefficients

We look to use WLS regression to infer the model parameters from the MSD curve. Since $\rho(t) = a + bt + ct^2$, where $a = 4\eta^2$, $b = 4D$ and $c = \alpha^2$, the coefficients can be inferred by quadratic regression [76]. Let σ_n^2 be the variance of ρ_n at the time point $t_n = nT/N$, $1 \leq n \leq N$, and $\sigma_{nm}^2 = \mathbb{E}(\rho_n\rho_m) - \mathbb{E}(\rho_n)\mathbb{E}(\rho_m)$ be the covariance between ρ_n and ρ_m, where $1 \leq n, m \leq N$. For a quadratic polynomial of the form $\mu(t) = a+bt+ct^2$, the variance of the regression coefficients, calculated by fitting the first p MSD points, can be estimated by [67]

$$\sigma_a^2 \approx \sum_{n=1}^{p} \sigma_n^2 \left(\frac{\partial a}{\partial \mu_n}\right)^2 + 2\sum_{n=1}^{p}\sum_{m=1}^{n-1} \sigma_{nm}^2 \left(\frac{\partial a}{\partial \mu_n}\right)\left(\frac{\partial a}{\partial \mu_m}\right), \quad 3 \leq p \leq N, \tag{2.43}$$

$$\sigma_b^2 \approx \sum_{n=1}^{p} \sigma_n^2 \left(\frac{\partial b}{\partial \mu_n}\right)^2 + 2\sum_{n=1}^{p}\sum_{m=1}^{n-1} \sigma_{nm}^2 \left(\frac{\partial b}{\partial \mu_n}\right)\left(\frac{\partial b}{\partial \mu_m}\right), \quad 3 \leq p \leq N, \tag{2.44}$$

$$\sigma_c^2 \approx \sum_{n=1}^{p} \sigma_n^2 \left(\frac{\partial c}{\partial \mu_n}\right)^2 + 2\sum_{n=1}^{p}\sum_{m=1}^{n-1} \sigma_{nm}^2 \left(\frac{\partial c}{\partial \mu_n}\right)\left(\frac{\partial c}{\partial \mu_m}\right), \quad 3 \leq p \leq N, \tag{2.45}$$

where

$$\frac{\partial a}{\partial \mu_n} = \frac{S_2S_4 - S_3^2 - S_1S_4t_n + S_1S_3t_n^2 + S_2S_3t_n - S_2^2t_n^2}{\sigma_n^2\Delta}, \tag{2.46}$$

$$\frac{\partial b}{\partial \mu_n} = \frac{S_0S_4t_n - S_0S_3t_n^2 - S_1S_4 + S_2S_3 + S_1S_2t_n^2 - S_2^2t_n}{\sigma_n^2\Delta}, \tag{2.47}$$

$$\frac{\partial c}{\partial \mu_n} = \frac{S_0S_2t_n^2 - S_0S_3t_n - S_1^2t_n^2 + S_1S_2t_n + S_1S_3 - S_2^2}{\sigma_n^2\Delta}, \tag{2.48}$$

and

$$S_k = \sum_{n=1}^{p} \frac{(t_n)^k}{\sigma_n^2}, \quad k = 0, \ldots, 4, \quad 3 \leq p \leq N, \quad \Delta = \begin{vmatrix} S_0 & S_1 & S_2 \\ S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \end{vmatrix}. \tag{2.49}$$

Note that the lower limit for p reflects the minimum number of points needed to fit a quadratic polynomial, while the upper limit corresponds to fitting using all the MSD points.

2.7 The covariance of the MSD

The only quantity which is not known in (2.43)–(2.49) is the covariance of the MSD σ_{nm}^2. As with the calculation of the variance of the MSD, we consider the covariance of the MSD for a single particle, denoted by $\sigma_{nm}^{2(S)}$. Then, the covariance of the MSD for an ensemble of particles will be

$$\sigma_{nm}^2 = \frac{\sigma_{nm}^{2(S)}}{N_S}. \tag{2.50}$$

From the definition of the covariance, we have that

$$\sigma_{nm}^{2(S)} = \mathbb{E}(\rho_n^{(1)}\rho_m^{(1)}) - \mathbb{E}(\rho_n^{(1)})\mathbb{E}(\rho_m^{(1)}). \tag{2.51}$$

The process of calculating the covariance of the MSD is similar to that of deriving the formula for the variance of the MSD. First, we assume that $m > n$, then letting $K = N + 1 - n$ and $P = N + 1 - m$ we have that

$$\sigma_{nm}^{2(S)} = \frac{1}{KP} \sum_{i=1}^{K} \sum_{j=1}^{P} \mathrm{Cov}(i, j), \tag{2.52}$$

where

$$\mathrm{Cov}(i,j) = \mathbb{E}(|\boldsymbol{x}_{i+n} - \boldsymbol{x}_i|^2 |\boldsymbol{x}_{j+m} - \boldsymbol{x}_j|^2) - \mathbb{E}(|\boldsymbol{x}_{i+n} - \boldsymbol{x}_i|^2)\mathbb{E}(|\boldsymbol{x}_{j+m} - \boldsymbol{x}_j|^2). \tag{2.53}$$

As before, we have to consider the different ways the trajectories can overlap. These correspond to the five cases shown in Figure 2.4. The covariance for each case can be written as

$$\begin{aligned}
&(a) \quad \mathrm{Cov}(i,j) = \mathbb{E}(\boldsymbol{R}_1^2 \boldsymbol{R}_3^2) - \mathbb{E}(\boldsymbol{R}_1^2)\mathbb{E}(\boldsymbol{R}_3^2), && i < i+n < j < j+m.\\
&(b) \quad \mathrm{Cov}(i,j) = \mathbb{E}((\boldsymbol{R}_1 + \boldsymbol{R}_2)^2 (\boldsymbol{R}_2 + \boldsymbol{R}_3)^2) \\
&\qquad\qquad - \mathbb{E}((\boldsymbol{R}_1 + \boldsymbol{R}_2)^2)\, \mathbb{E}((\boldsymbol{R}_2 + \boldsymbol{R}_3)^2), && i < j \le i+n < j+m.\\
&(c) \quad \mathrm{Cov}(i,j) = \mathbb{E}(|\boldsymbol{R}_2^2|(\boldsymbol{R}_1 + \boldsymbol{R}_2 + \boldsymbol{R}_3)^2) \\
&\qquad\qquad - \mathbb{E}(|\boldsymbol{R}_2^2|)\mathbb{E}((\boldsymbol{R}_1 + \boldsymbol{R}_2 + \boldsymbol{R}_3)^2), && j \le i < i+n < j+m.\\
&(d) \quad \mathrm{Cov}(i,j) = \mathbb{E}((\boldsymbol{R}_1 + \boldsymbol{R}_2)^2 (\boldsymbol{R}_2 + \boldsymbol{R}_3)^2) \\
&\qquad\qquad - \mathbb{E}((\boldsymbol{R}_1 + \boldsymbol{R}_2)^2)\, \mathbb{E}((\boldsymbol{R}_2 + \boldsymbol{R}_3)^2), && j < i < j+m \le i+n.\\
&(e) \quad \mathrm{Cov}(i,j) = \mathbb{E}(\boldsymbol{R}_1^2 \boldsymbol{R}_3^2) - \mathbb{E}(\boldsymbol{R}_1^2)\mathbb{E}(\boldsymbol{R}_3^2), && j < j+m \le i < i+n.
\end{aligned} \tag{2.54}$$

These expectations can be calculated using the results of (2.37). From this we obtain

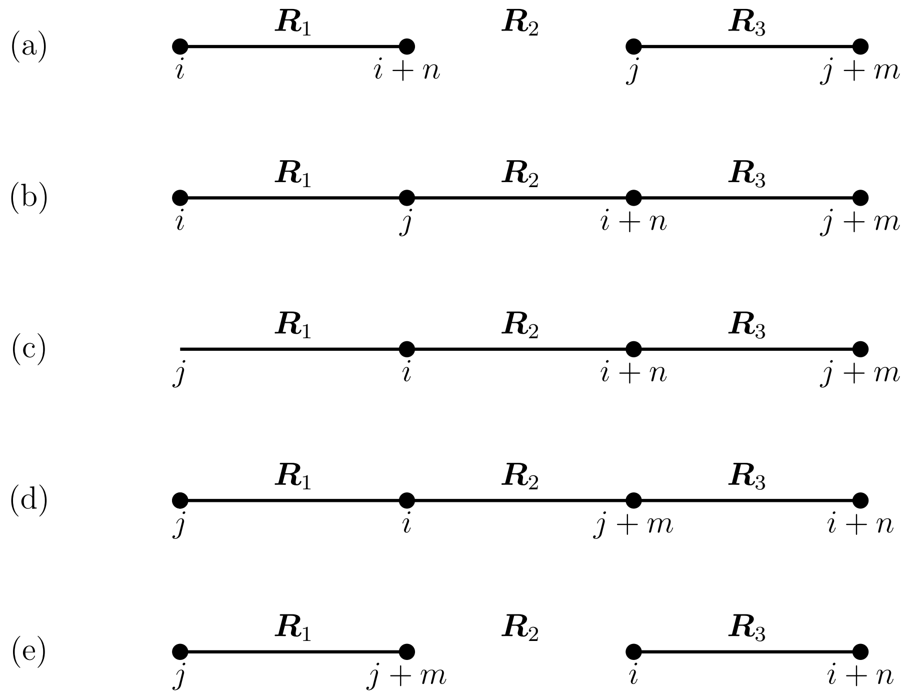

Figure 2.4: Diagram showing the five different combinations for overlapping displacements used to calculate the covariance of the MSD. The five cases are (a) $i < i+n < j < j+m$, (b) $i < j \leq i+n < j+m$, (c) $j \leq i < i+n < j+m$, (d) $j < i < j+m \leq i+n$, and (e) $j < i < j+m \leq i+n$.

$$
\begin{aligned}
&(a) \quad \operatorname{Cov}(l_1) = 0. \\[1em]
&(b) \quad \operatorname{Cov}(l_2) = 16D^2((n-l_2)\Delta t)^2 + 8D\alpha^2 mn(n-l_2)(\Delta t)^3 \\
&\qquad\qquad + (4\eta^4 - 4\alpha^2\eta^2 mn(\Delta t)^2)\delta_{j,\,i+n}, \quad \text{where } l_2 = j - i. \\[1em]
&(c) \quad \operatorname{Cov}(l_3) = 16D^2(n\Delta t)^2 + 8D\alpha^2 n^2 m(\Delta t)^3 \\
&\qquad\qquad + (16D\eta^2 n\Delta t + 4\eta^4 + 4\alpha^2\eta^2 mn(\Delta t)^2)\delta_{i,j}. \\[1em]
&(d) \quad \operatorname{Cov}(l_4) = 16D^2(l_4\Delta t)^2 + 8D\alpha^2 mnl_4(\Delta t)^3 \\
&\qquad\qquad + (16D\eta^2 n\Delta t + 4\eta^4 \\
&\qquad\qquad + 4\alpha^2\eta^2 mn(\Delta t)^2)\delta_{i+n,\,j+m}, \quad \text{where } l_4 = j + m - i. \\[1em]
&(e) \quad \operatorname{Cov}(l_5) = (4\eta^4 - 4\alpha^2\eta^2 mn(\Delta t)^2)\delta_{i,\,j+m}.
\end{aligned}
\tag{2.55}
$$

We need to sum these different covariance values to obtain the overall covariance of the MSD in the two cases $m + n \leq N$ and $m + n > N$. As before, this is done by summing the non-zero terms along diagonals in the covariance matrix. Also, note the Kronecker delta terms must be considered separately. The sums are done as follows:

$m + n \leq N$

$$
\begin{aligned}
&(a) \quad 0. \\
&(b) \quad \left(\sum_{l_2=1}^{n-1} (P - l_2)\mathrm{Cov}(l_2) \right) + (P - n)(4\eta^4 - 4\alpha^2\eta^2 mn(\Delta t)^2). \\
&(c) \quad \left(\sum_{l_3=1}^{m-n+1} P\mathrm{Cov}(l_3) \right) + P(16D\eta^2 n\Delta t + 4\eta^4 + 4\alpha^2\eta^2 mn(\Delta t)^2). \\
&(d) \quad \left(\sum_{l_4=1}^{n-1} (P - (n-1))\mathrm{Cov}(l_4) \right) + P(16D\eta^2 n\Delta t + 4\eta^4 + 4\alpha^2\eta^2 mn(\Delta t)^2). \\
&(e) \quad (P - n)(4\eta^4 - 4\alpha^2\eta^2 mn(\Delta t)^2).
\end{aligned}
\tag{2.56}
$$

$m + n > N$

$$
\begin{aligned}
&(a) \quad 0. \\
&(b) \quad \left(\sum_{l_2=1}^{P-1} (P - l_2)\mathrm{Cov}(l_2) \right) + (P - n)(4\eta^4 - 4\alpha^2\eta^2 mn(\Delta t)^2). \\
&(c) \quad \left(\sum_{l_3=1}^{m-n+1} P\mathrm{Cov}(l_3) \right) + P(16D\eta^2 n\Delta t + 4\eta^4 + 4\alpha^2\eta^2 mn(\Delta t)^2). \\
&(d) \quad \left(\sum_{l_4=n+1-P}^{n-1} (P - (n - l_4))\mathrm{Cov}(l_4) \right) + P(16D\eta^2 n\Delta t + 4\eta^4 + 4\alpha^2\eta^2 mn(\Delta t)^2). \\
&(e) \quad 0.
\end{aligned}
\tag{2.57}
$$

We used the software Maple to evaluate these sums. Finally, we obtain

$$\sigma_{nm}^2 = \begin{cases} \begin{array}{l} \Big[\frac{16nD^2(\Delta t)^2}{6KP}\Big(- n^3 - 2Pn^2 + n(1 - 6m^2 \\ +6(N+1)m) + 2P\Big) + \frac{8\alpha^2(\Delta t)^3 mn^2 D}{3KP}\Big(- n^2 \\ -3m^2 + 3m(N+1) + 1\Big) + \frac{8\eta^2}{KP}\Big(4nPD\Delta t \\ -n\eta^2 + 2P\eta^2 + \alpha^2 mn^2(\Delta t)^2\Big)\Big]/N_S, \end{array} & m + n \leq N, \\ \\ \begin{array}{l} \Big[\frac{8D^2(\Delta t)^2}{3K}\Big(- m^3 + (3 + 3N - 4n)m^2 + (8(N \\ +1)n - 2 - 3N^2 - 6N)m - 6n^3 + 6(N+1)n^2 \\ -(4N^2 + 8N)n + N(N+2)(N+1)\Big) \\ +\frac{8\alpha^2 D(\Delta t)^3 mn}{3K}\Big(m^2 - 2(N+1)m + 3n^2 \\ -3(N+1)nN^2 + 2N\Big) + \frac{8\eta^2}{K}\Big(\alpha^2(\Delta t)^2 mn \\ +4Dn\Delta t + \eta^2\Big)\Big]/N_S, \end{array} & m + n > N. \end{cases} \tag{2.58}$$

To our knowledge, this result has never been derived before. Figure 2.5 shows a comparison between the theoretical covariance of the MSD (2.58) and an empirical estimate of the covariance of the MSD averaged over 1000 and 10,000 samples, along with a cross section of both quantities, for the parameters values $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$, $N = 100$ and $T = 100\,\text{s}$. Note that we only provide one plot of the full covariance as the plots over the two different sample sizes are very similar. We see good agreement between the theoretical covariance of the MSD (2.58) and its empirical estimate.

In this chapter, we are interested in the optimal estimation of the diffusion coefficient D and the drift magnitude α. Since these are related to the regression coefficients b and c, we look to minimise $\sigma_b/b + \sigma_c/c$, the relative errors in b and c. This can be done in two ways depending on the experimental protocol.

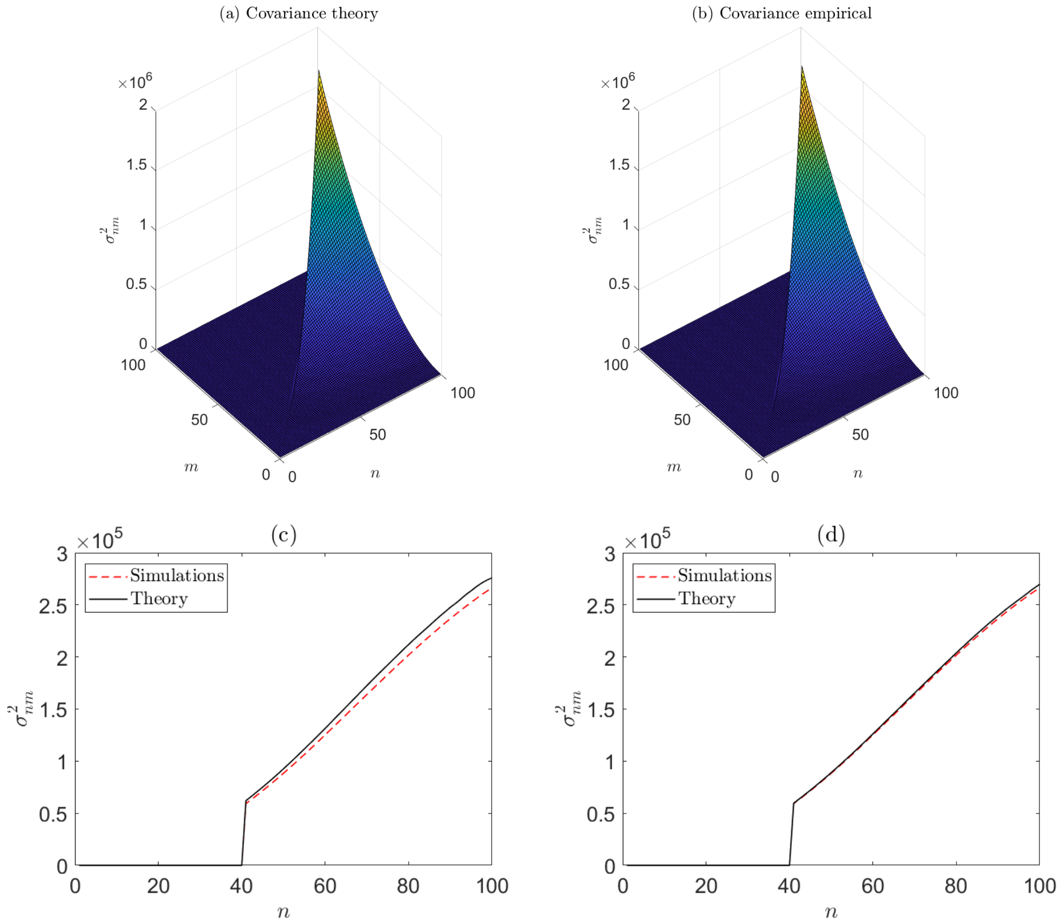

Figure 2.5: A plot comparing the theoretical covariance of the MSD (2.58) (a) and its empirical estimate averaged over 1000 samples (b), along with a plot of a cross section along $m = 40$ of the theoretical covariance of the MSD (solid black line) with its empirical estimate (dashed red line) averaged over 1000 samples (c) and 10,000 samples (d) for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$, $N = 100$ and $T = 100\,\text{s}$. Note that the jump in plots (c) and (d) are a consequence of only calculating the upper triangular part of the covariance matrix.

2.8 Results using the optimal number of fitting points

2.8.1 Existence of an optimal number of fitting points

If experiments cannot be repeated then the optimal estimates of the model parameters may be obtained by fitting a subset of the MSD points. For this, we assume that the MSD is calculated using all N time points as in (2.21) and (2.22), then fit using a subset of these points.

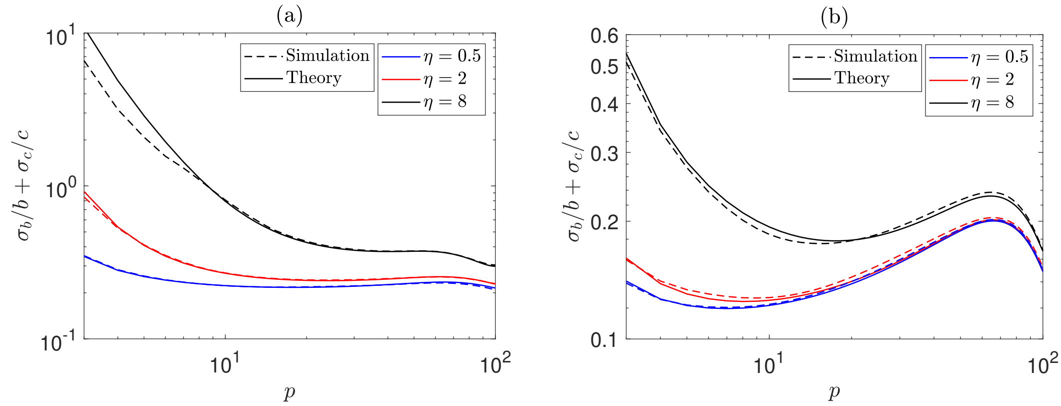

Figure 2.6: A plot of the theoretical value of $\sigma_b/b + \sigma_c/c$ (solid lines) and its empirically estimated value using 1000 samples (dashed lines) when fit with the first p MSD points for $\eta = 0.5\,\mu$m (blue lines), $2\,\mu$m (red lines) and $8\,\mu$m (black lines). These experiments were for $D = 2\,\mu\text{m}^2$/s, $\alpha = 1\,\mu$m/s, $N_S = 10$ and $N = 100$, for $\Delta t = 1$ s giving $T = 100$ s (a) and $\Delta t = 10$ s giving $T = 1000$ s (b). For $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m, the optimal number of fitting points are 100 for each curve in (a) and 7, 8 and 100, respectively, for the curves in (b).

To investigate optimising the number of fitting points, we look at the theoretical value of the uncertainty $\sigma_b/b+\sigma_c/c$ using (2.43)–(2.49) when fitting with the first p MSD points and compare this with an empirical estimate calculated from simulations. For the estimated uncertainty, we calculate the MSD data points then use WLS regression to obtain estimates for b and c by fitting with the first p points, where $3 \leq p \leq N$. This was repeated 1000 times to empirically estimate the values of σ_b and σ_c. Figure 2.6 shows the theoretical and simulated value of $\sigma_b/b + \sigma_c/c$ as a function of the number of fitting points p for two different Δt values for $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m. These experiments were for $D = 2\,\mu\text{m}^2$/s, $\alpha = 1\,\mu$m/s, $\eta = 2\,\mu$m, $N_S = 10$ and $N = 100$, with $\Delta t = 1$ s giving $T = 100$ s for Figure 2.6(a), while $\Delta t = 10$ s giving $T = 1000$ s for Figure 2.6(b). We denote the optimal number of fitting points which minimises $\sigma_b/b+\sigma_c/c$ by p_{opt}. Although it is difficult to see, when Δt is small, corresponding to Figure 2.6(a), the optimal estimation of the parameters is obtained by fitting with all of the MSD points, for all the values of η tested. On the other hand, if Δt is taken to be larger,

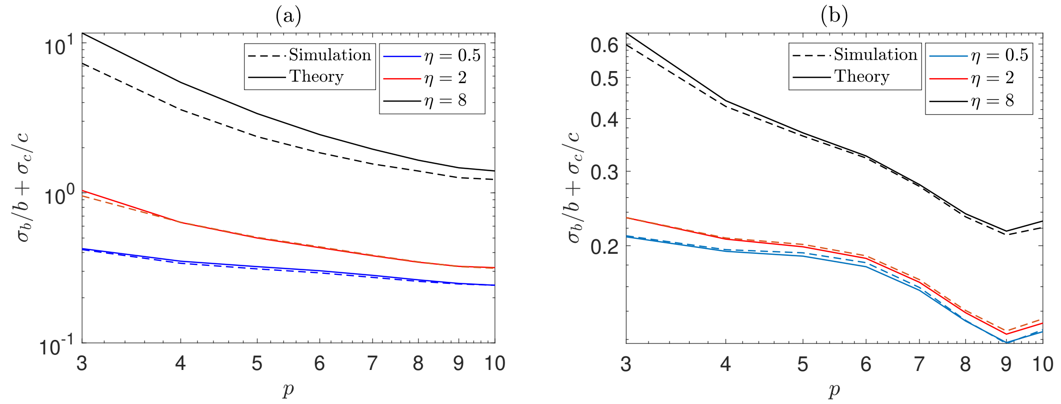

Figure 2.7: A plot of the theoretical value of $\sigma_b/b + \sigma_c/c$ (solid lines) and its empirically estimated value using 1000 samples (dashed lines) when fit with the first p MSD points for $\eta = 0.5\,\mu$m (blue lines), $2\,\mu$m (red lines) and $8\,\mu$m (black lines). These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu$m/s, $N_S = 100$ and $N = 10$, for $\Delta t = 1\,$s giving $T = 10\,$s (a) and $\Delta t = 10\,$s giving $T = 100\,$s (b). For $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m, the optimal number of fitting points are 10 for each curve in (a) and 9 for each curve in (b).

corresponding to Figure 2.6(b), then there may be an optimal number of fitting points which is less than N. In this case, for $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m, we have that the optimal number of fitting points are $p_{opt} = 7$, 8 and 100, respectively.

To test the robustness of our calculations, we also tested with $N_S = 100$ and $N = 10$. Figure 2.7 shows the theoretical and simulated uncertainty $\sigma_b/b + \sigma_c/c$ as a function of the number of fitting points for the two different Δt values for $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu$m/s, with $\Delta t = 1\,$s giving $T = 10\,$s for Figure 2.7(a), while $\Delta t = 10\,$s giving $T = 100\,$s for Figure 2.7(b). Here we see a similar result to that in Figure 2.6. We find that, for a small value of Δt, corresponding to Figure 2.7(a), the optimal estimate of the parameters comes from fitting with all the MSD points. As Δt is increased, corresponding to Figure 2.7(b), we see that the optimal inference comes from fitting with a subset of points; in this case, all values of η tested had $p_{opt} = 9$.

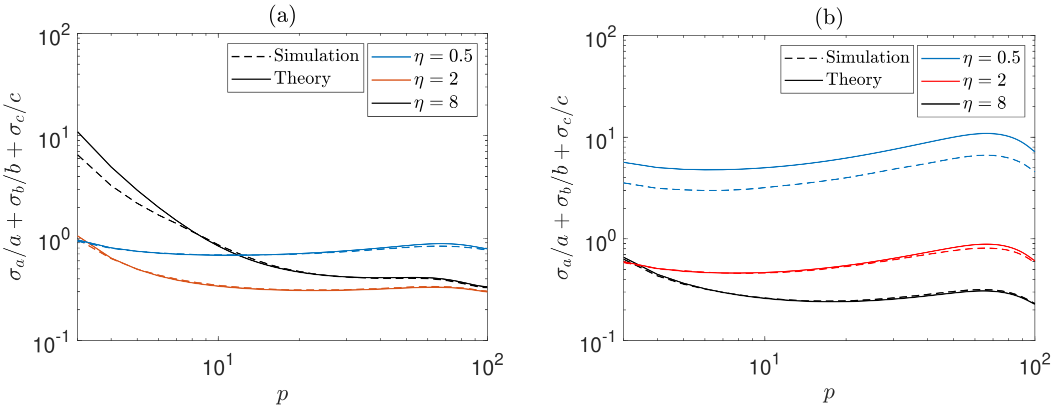

Figure 2.8: A plot of the theoretical value of $\sigma_a/a + \sigma_b/b + \sigma_c/c$ (solid lines) and its empirically estimated value using 1000 samples (dashed lines) when fit with the first p MSD points for $\eta = 0.5\,\mu$m (blue lines), $2\,\mu$m (red lines) and $8\,\mu$m (black lines). These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu$m/s, $N_S = 10$ and $N = 100$, for $\Delta t = 1\,$s giving $T = 100\,$s (a) and $\Delta t = 10\,$s giving $T = 1000\,$s (b). For $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m, the optimal number of fitting points are 11, 100 and 100, respectively, for the curves in (a) and 6, 8 and 100, respectively, for the curves in (b).

We also looked for the existence of an optimal number of fitting points when the static error η is included in the inference. Analogous to the two parameter case, we look to minimise the uncertainty $\sigma_a/a+\sigma_b/b+\sigma_c/c$ since the static error term is related to the regression coefficient a. Figure 2.8 shows the theoretical and simulated value of $\sigma_a/a + \sigma_b/b + \sigma_c/c$ as a function of the number of fitting points for the two different Δt values for $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu$m/s, $N_S = 10$ and $N = 100$, with $\Delta t = 1\,$s giving $T = 100\,$s for Figure 2.8(a), while $\Delta t = 10\,$s giving $T = 1000\,$s for Figure 2.8(b). Here we see that, with the inclusion of the static error term, the value of the uncertainty $\sigma_a/a + \sigma_b/b + \sigma_c/c$ does not follow an obvious trend like the value of $\sigma_b/b + \sigma_c/c$ from Figures 2.6 and 2.7.

A MATLAB routine to calculate the optimal number of regression points p_{opt} is presented in Appendix A

2.8.2 Iterative algorithm to calculate optimal number of fitting points

The difficulty with using $p_{opt}(D, \alpha, \eta, \Delta t, N)$ to infer the model parameters is we require the values of D, α and η themselves in order to calculate it. We therefore consider the following iterative technique for determining p_{opt}. The iterative algorithm initially estimates D, α and η by fitting all N MSD points. The weighting used in the fitting is initially taken to be uniform, then, for all future iterations, we estimate the variance of the MSD by substituting the current parameter estimates into (2.42). The algorithm then adapts the number of fitting points according to Algorithm 1. The tolerance τ determines the stopping criterion depending on the relative differences between two successive p_i values.

We tested the iterative algorithm for the parameter values $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$ and $\eta = 2\,\mu\text{m}$ for three different time steps, $\Delta t = 1\,\text{s}$, $\Delta t = 10\,\text{s}$ and $\Delta t = 100\,\text{s}$. Each simulation run uses $N = 1000$ time points and $N_S = 10$ trajectories to create the MSD data and a quadratic fit. Since simulations are likely to end after a different number of iterations, Steps 9 – 11 of Algorithm 1 will be ignored and instead all simulations are stopped after 10 iterations. These simulations were then repeated 100 times. By denoting the mean value of a quantity by the angular brackets $\langle \cdot \rangle$, we indicate the performance of the algorithm by plotting $\langle p_i \rangle$, $\langle |D_i/D - 1| \rangle$ and $\langle |\alpha_i/\alpha - 1| \rangle$ in Figure 2.9. The first thing to notice is that the algorithm converges to p_{opt} in a couple of iterations for the cases considered, with most being after just one iteration. We do not see much improvement in $\langle |\alpha_i/\alpha - 1| \rangle$ when fit with the optimal number of fitting points, compared with all the MSD points, for any value of Δt. However, we do see a decrease in its value as we increase Δt. This is due to the value of the measurement time interval T increasing as we increase Δt. This increase in T moves us into the drift time scale where the inference of α is better. The value of

$\langle |D_i/D-1| \rangle$ decreases after one iteration in all cases, with a larger decrease for larger values of Δt. The final value of $\langle |D_i/D-1| \rangle$ are similar for the three values of Δt. This example shows that the choice of Δt is important for the optimal inference of both parameters, particularly the diffusion coefficient D. Fitting with the optimal number of fitting points leads to a reduction in the uncertainty in D, particularly for large value of Δt where the reduction is more prominent.

Algorithm 1 Iterative algorithm to find p_{opt} and estimates of D, α and η

```
    Input: MSD data found at $N$ fixed time points with time step $\Delta t = T/N$,
    and convergence parameter $\tau$.
    Output: Estimates of optimal number of fitting points $p_{opt}$ and parameters
    $D$, $\alpha$ and $\eta$.
 1: Set the number of fitting points $p_0 = N$ and set $i = 0$.
 2: if $i = 0$ then
 3:     $\sigma_n^{2\,(i)} = 1, \quad 1 \leq n \leq p_i,$
 4: else
 5:     $\sigma_n^{2\,(i)} = \sigma_n^2(D_{i-1}, \alpha_{i-1}, \eta_{i-1}, \Delta t)$ using (2.42), $\quad 1 \leq n \leq p_i.$
 6: end if
 7: Use WLS regression with weights $1/\sigma_n^{2\,(i)}$ on the first $p_i$ points of the MSD to
    get the parameter estimates $D_i$, $\alpha_i$ and $\eta_i$.
 8: Update $p_{i+1} = p_{opt}(D_i, \alpha_i, \eta_i, \Delta t, N)$.
 9: if $|p_{i+1} - p_i|/p_{i+1} < \tau$ then
10:     end algorithm,
11: else
12:     Set $i = i + 1$ and go back to Step 2.
13: end if
```

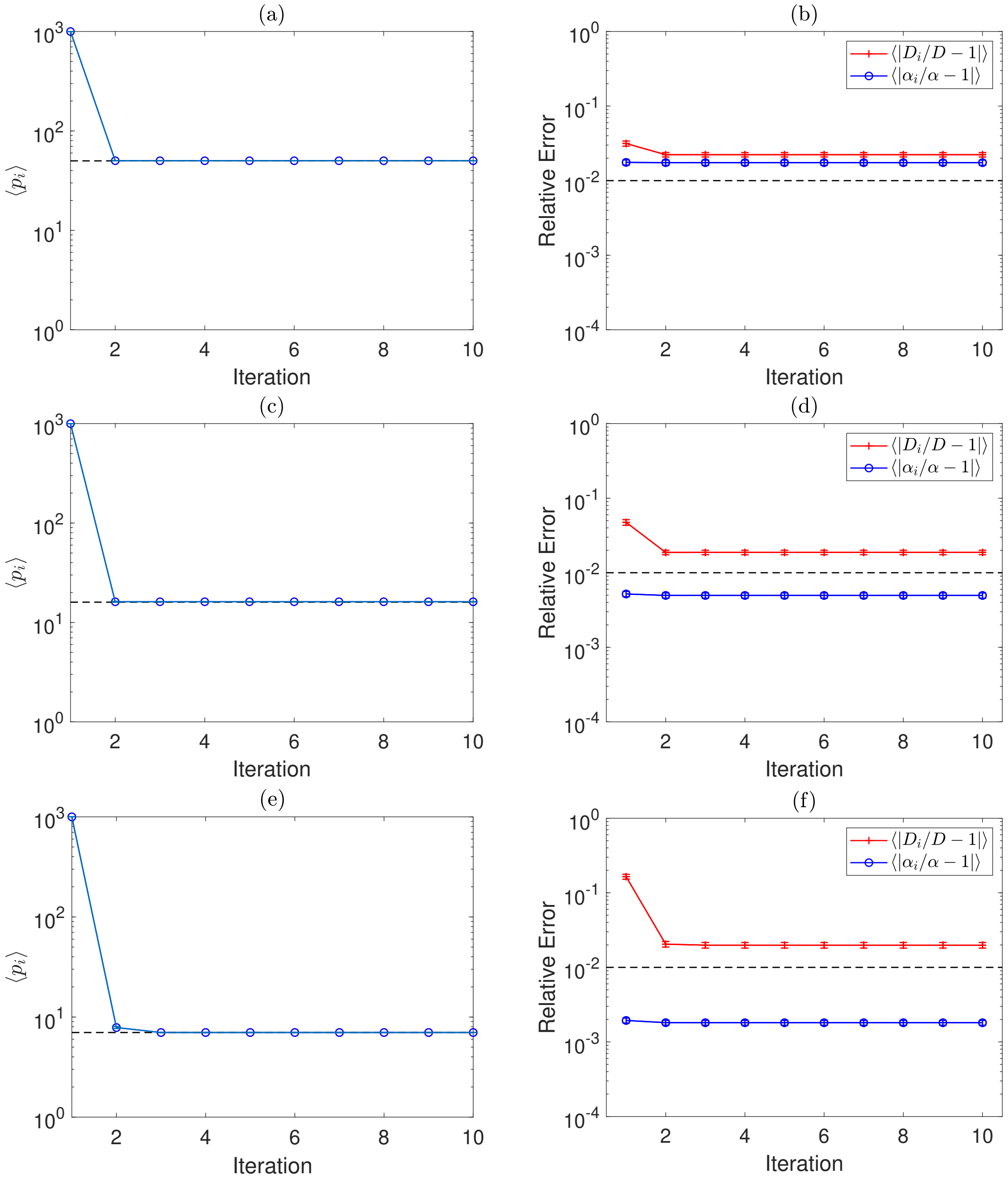

Figure 2.9: A plot of the value of $\langle p_i \rangle$ for each iteration with standard error bars [(a), (c), and (e)], along with a plot of the value of $\langle |D_i/D - 1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha - 1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b), (d) and (f)]. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$ and $N = 1000$, for $\Delta t = 1\,\text{s}$ [(a) and (b)], $\Delta t = 10\,\text{s}$ [(c) and (d)] and $\Delta t = 100\,\text{s}$ [(e) and (f)]. The dashed line in the plots of $\langle p_i \rangle$ correspond to $p_{opt} = 50$ (a), $p_{opt} = 16$ (c), and $p_{opt} = 7$ (e), while the dashed line in the plots of the relative errors correspond with the value 10^{-2}, indicating a 1% error.

We test further examples of the iterative algorithm for more parameter values. First we test for $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 7\,\mu\mathrm{m/s}$, $\eta = 2\,\mu\mathrm{m}$, $N_S = 10$ and $N = 1000$, for $\Delta t = 1\,\mathrm{s}$, $\Delta t = 10\,\mathrm{s}$ and $\Delta t = 100\,\mathrm{s}$. The results are shown in Figure 2.10. These are similar to the case shown in Figure 2.9. We see that the value of $\langle p_i \rangle$ converges to p_{opt} in no more than two iterations for all the cases shown. The value of $\langle |\alpha_i/\alpha - 1| \rangle$ does not decrease between iterations but decreases as Δt increases. The value of $\langle |D_i/D - 1| \rangle$ decreases when we fit with p_{opt} points rather than using all the points, with a larger decrease in the uncertainty for larger values of Δt, and the final value of $\langle |D_i/D - 1| \rangle$ is similar for all values of Δt.

The algorithm was also tested for $D = 6\,\mu\mathrm{m}^2/\mathrm{s}$ and $\alpha = 1\,\mu\mathrm{m/s}$. The results are shown in Figure 2.11. Again, we find very similar results. The value of $\langle p_i \rangle$ converges quickly to p_{opt}, in this case, after one iteration each time. The value of $\langle |\alpha_i/\alpha - 1| \rangle$ remains similar between iterations but again decreases as Δt increases. The value of $\langle |D_i/D - 1| \rangle$ decreases after one iteration in each case with the final value of $\langle |D_i/D - 1| \rangle$ being similar.

2.8.3 Single particle parameter estimation using the optimal number of fitting points

While the analysis and results presented so far assume the availability of data for an ensemble of particles, in some situations only single particle data are available. We now consider how the results we have perform in the single particle case. An important point to note is that the optimal number of fitting points for both the single particle case and ensemble of particles case are identical. This is because when calculating the variance and covariance of the MSD in the ensemble particle case, we simply take the single particle variance and covariance and divide by N_S, as stated in Sections 2.5 and 2.7. Hence, when calculating the variance of the regression coefficients in (2.43)–(2.45) for the ensemble case, we can take out

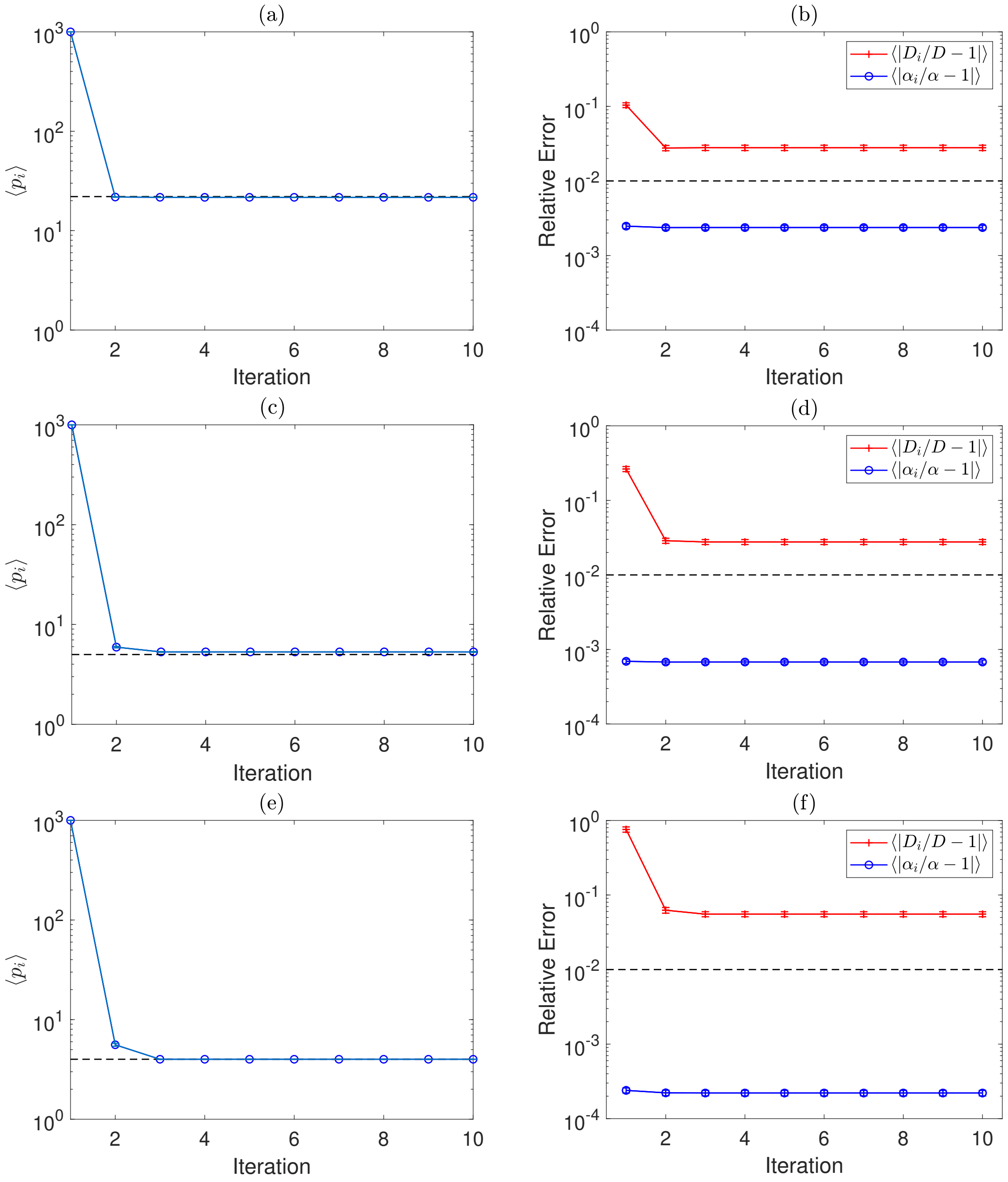

Figure 2.10: A plot of the value of $\langle p_i \rangle$ for each iteration with standard error bars [(a), (c), and (e)], along with a plot of the value of $\langle |D_i/D - 1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha - 1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b), (d) and (f)]. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 7\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$ and $N = 1000$, for $\Delta t = 1\,\text{s}$ [(a) and (b)], $\Delta t = 10\,\text{s}$ [(c) and (d)] and $\Delta t = 100\,\text{s}$ [(e) and (f)]. The dashed line in the plots of $\langle p_i \rangle$ correspond to $p_{opt} = 22$ (a), $p_{opt} = 5$ (c), and $p_{opt} = 4$ (e), while the dashed line in the plots of the relative errors correspond with the value 10^{-2}, indicating a 1% error.

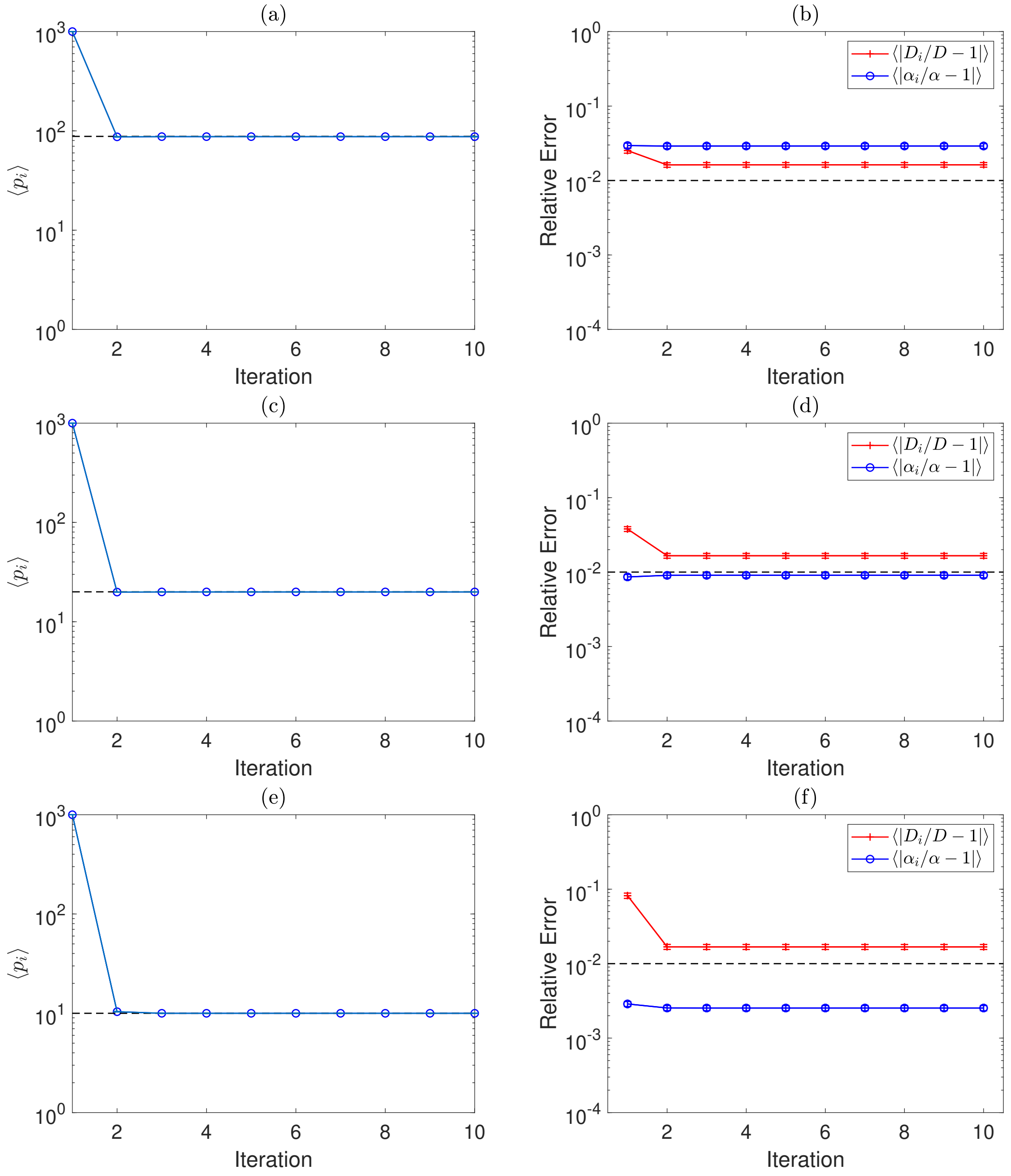

Figure 2.11: A plot of the value of $\langle p_i \rangle$ for each iteration with standard error bars [(a), (c), and (e)], along with a plot of the value of $\langle |D_i/D - 1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha - 1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b), (d) and (f)]. These experiments were for $D = 6\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$ and $N = 1000$, for $\Delta t = 1\,\text{s}$ [(a) and (b)], $\Delta t = 10\,\text{s}$ [(c) and (d)] and $\Delta t = 100\,\text{s}$ [(e) and (f)]. The dashed line in the plots of $\langle p_i \rangle$ correspond to $p_{opt} = 88$ (a), $p_{opt} = 20$ (c), and $p_{opt} = 10$ (e), while the dashed line in the plots of the relative errors correspond with the value 10^{-2}, indicating a 1% error.

a factor of $1/N_S$ from σ_n^2 and σ_{nm}^2. Therefore, the value of $\sigma_b/b + \sigma_c/c$ in the ensemble case will be a factor of $\sqrt{N_S}$ smaller than the single particle case but the shape of the curve will be the same in both cases.

When using Algorithm 1 with an ensemble of particles, Steps 2 – 6 could be ignored and the variance of the MSD can be estimated empirically from the data. This obviously cannot be done for the single particle case. This stresses the importance of having the theoretical expression for the variance of the MSD (2.42) as WLS regression can be still be done using single particle data.

Figure 2.12 shows the results of the iterative algorithm for the same parameter values as in Figure 2.9 but for $N_S = 1$. Since we only have a single particle, we expect the relative errors to be higher. Therefore, in each right plot, the dashed line will now correspond to a 10% error. Notice that the value of $\langle p_i \rangle$ takes a couple more iterations to converge but still does so in a small number of iterations. We often see the relative errors converge before $\langle p_i \rangle$, which is a result of the shallow minimum around p_{opt} in Figure 2.6(b). We also observe similar behaviour for repeatable experiments tested later in the chapter; for example, later in Figures 2.15 and 2.23. We see the same trend for $\langle |\alpha_i/\alpha - 1| \rangle$ as before, namely that fitting with the optimal number of fitting points does not improve its value much, but using a large value of Δt does. However, we see that the value of $\langle |D_i/D - 1| \rangle$ is significantly improved; for example, looking at the case where $\Delta t = 100\,\mathrm{s}$, we start with around a 10,000% error and end below a 10% error. This is a considerable improvement compared with the ensemble case seen in Figure 2.9.

We also tested further examples of single particle experiments for different parameter values. First we look at $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 7\,\mu\mathrm{m/s}$, $\eta = 2\,\mu\mathrm{m}$, $N_S = 1$ and $N = 1000$, for $\Delta t = 1\,\mathrm{s}$, $\Delta t = 10\,\mathrm{s}$ and $\Delta t = 100\,\mathrm{s}$. The results are shown in Figure 2.13. We see similar dynamics to the results in Figure 2.12. In all three cases, the value of $\langle p_i \rangle$ converges close to p_{opt}, the value of $\langle |\alpha_i/\alpha - 1| \rangle$

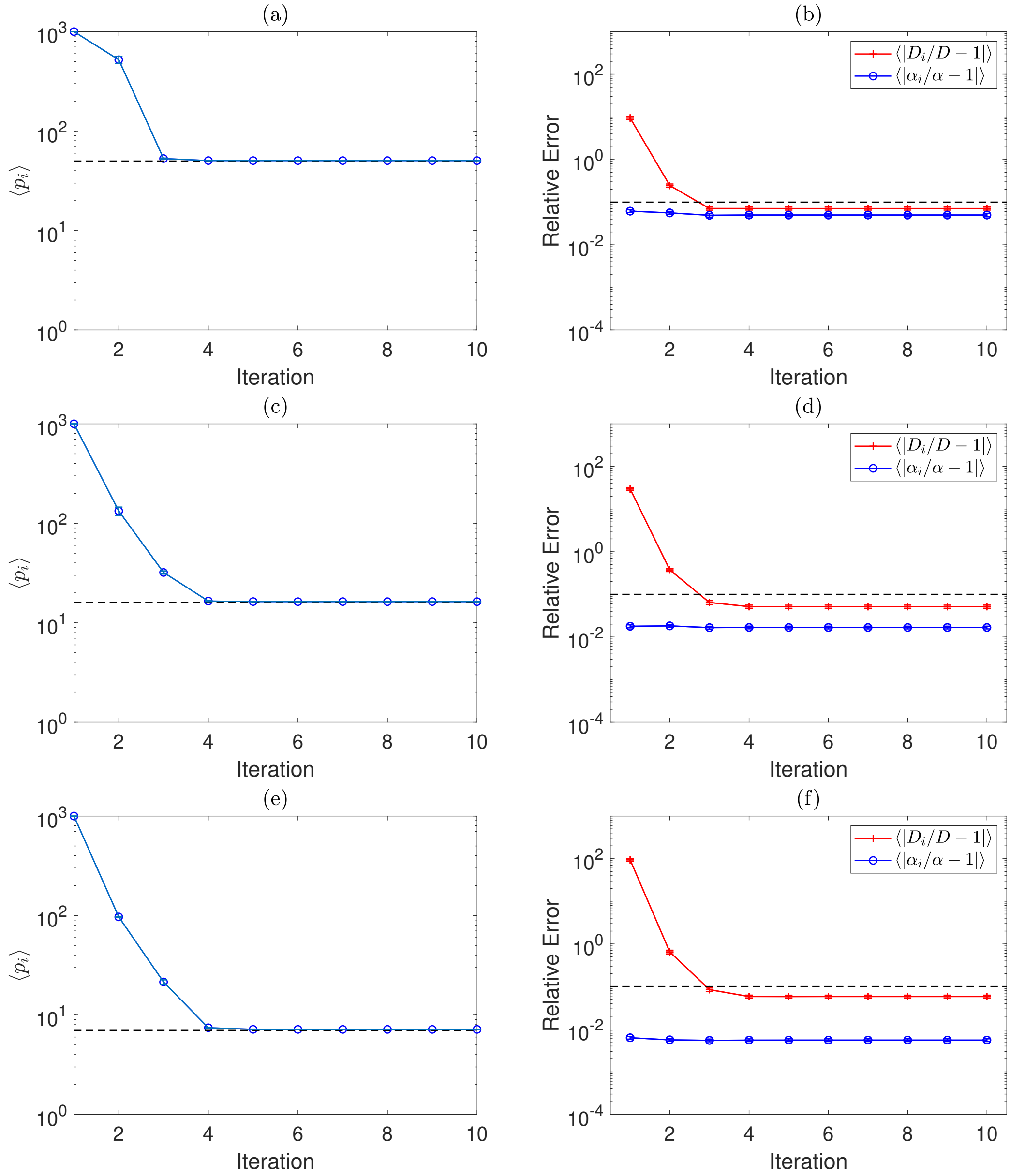

Figure 2.12: A plot of the value of $\langle p_i \rangle$ for each iteration with standard error bars [(a), (c), and (e)], along with a plot of the value of $\langle |D_i/D - 1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha - 1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b), (d) and (f)]. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 1$ and $N = 1000$, for $\Delta t = 1\,\text{s}$ [(a) and (b)], $\Delta t = 10\,\text{s}$ [(c) and (d)] and $\Delta t = 100\,\text{s}$ [(e) and (f)]. The dashed line in the plots of $\langle p_i \rangle$ correspond to $p_{opt} = 50$ (a), $p_{opt} = 16$ (c), and $p_{opt} = 7$ (e), while the dashed line in the plots of the relative errors correspond with the value 10^{-1}, indicating a 10% error.

only decreases significantly when Δt is increased and the value of $\langle|D_i/D-1|\rangle$ is significantly improved when fit with p_{opt} points rather than all the points. Notice, however, for corresponding Δt values, that the final value of $\langle|\alpha_i/\alpha-1|\rangle$ are always smaller, while the final value of $\langle|D_i/D-1|\rangle$ are always larger, than those given in Figure 2.12. This is expected as the larger value of α will cause the MSD to increase quicker for larger times, and so a smaller value of Δt would be required for better inference of D.

The case where $D = 6\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 1$ and $N = 1000$, for $\Delta t = 1\,\text{s}$, $\Delta t = 10\,\text{s}$ and $\Delta t = 100\,\text{s}$ was also tested and is shown in Figure 2.14. Again, the results show the same dynamics as those in Figure 2.12. This time, however, we see that the final value of $\langle|\alpha_i/\alpha-1|\rangle$ are always larger for corresponding values of Δt. Again, this is is expected as the larger value of D will in turn require a larger value of Δt to optimise the inference of α.

2.9 Results using the optimal measurement interval

2.9.1 Existence of an optimal measurement time interval

If experiments are able to be repeated then the optimisation can be done with respect to the measurement time interval T, rather than the number of MSD fitting points. This has the advantage that the optimal measurement time interval could help inform future experiments. For this method we assume that the MSD is calculated from all N time points and that all ρ_n data points are used in the fitting process. Note that since all the MSD points are used in the fitting, a new value of T will correspond with a new value of Δt. As before, we concentrate on the optimal inference of the diffusion coefficient D and drift magnitude α.

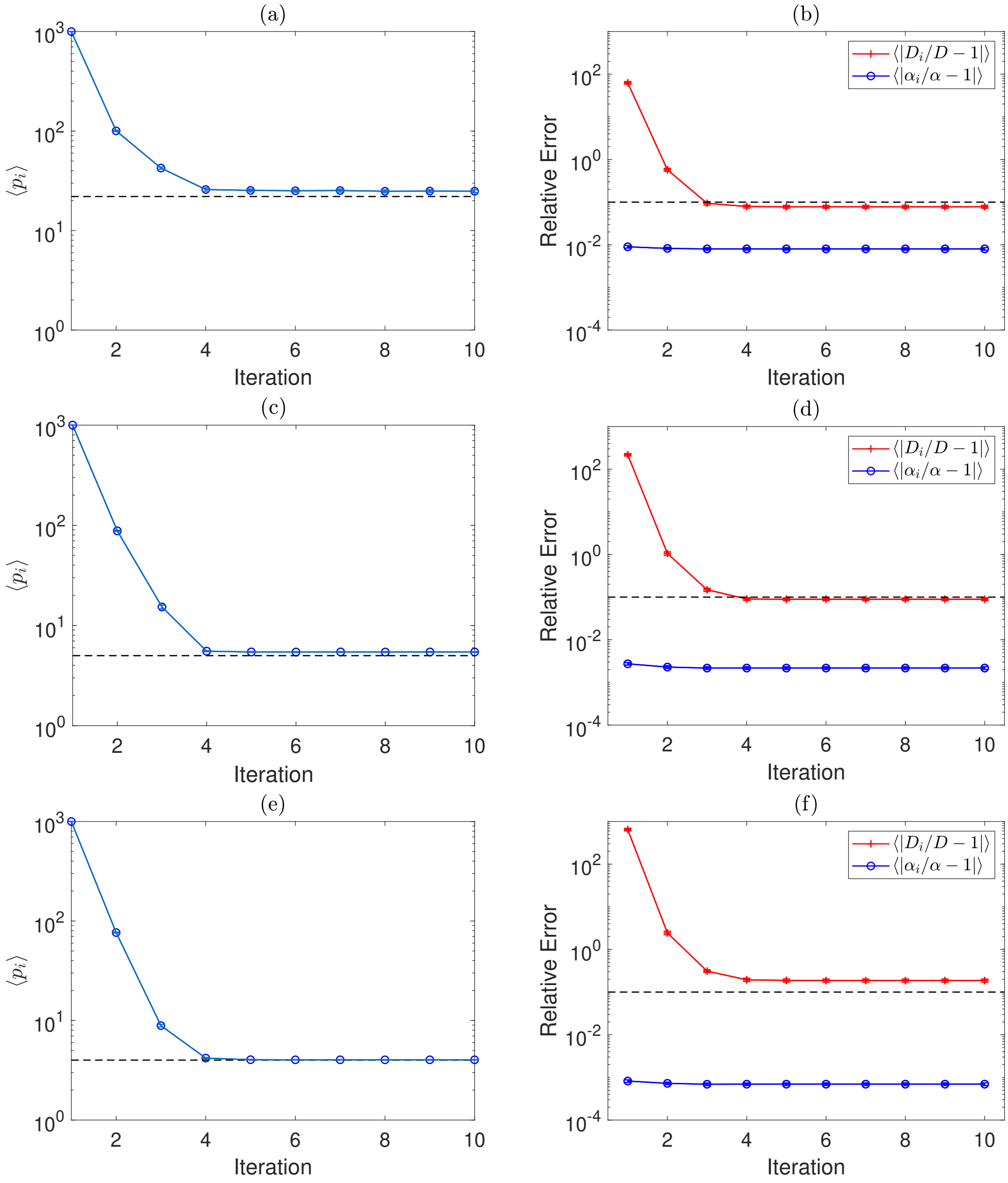

Figure 2.13: A plot of the value of $\langle p_i \rangle$ for each iteration with standard error bars [(a), (c), and (e)], along with a plot of the value of $\langle |D_i/D - 1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha - 1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b), (d) and (f)]. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 7\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 1$ and $N = 1000$, for $\Delta t = 1\,\text{s}$ [(a) and (b)], $\Delta t = 10\,\text{s}$ [(c) and (d)] and $\Delta t = 100\,\text{s}$ [(e) and (f)]. The dashed line in the plots of $\langle p_i \rangle$ correspond to $p_{opt} = 22$ (a), $p_{opt} = 5$ (c), and $p_{opt} = 4$ (e), while the dashed line in the plots of the relative errors correspond with the value 10^{-1}, indicating a 10% error.

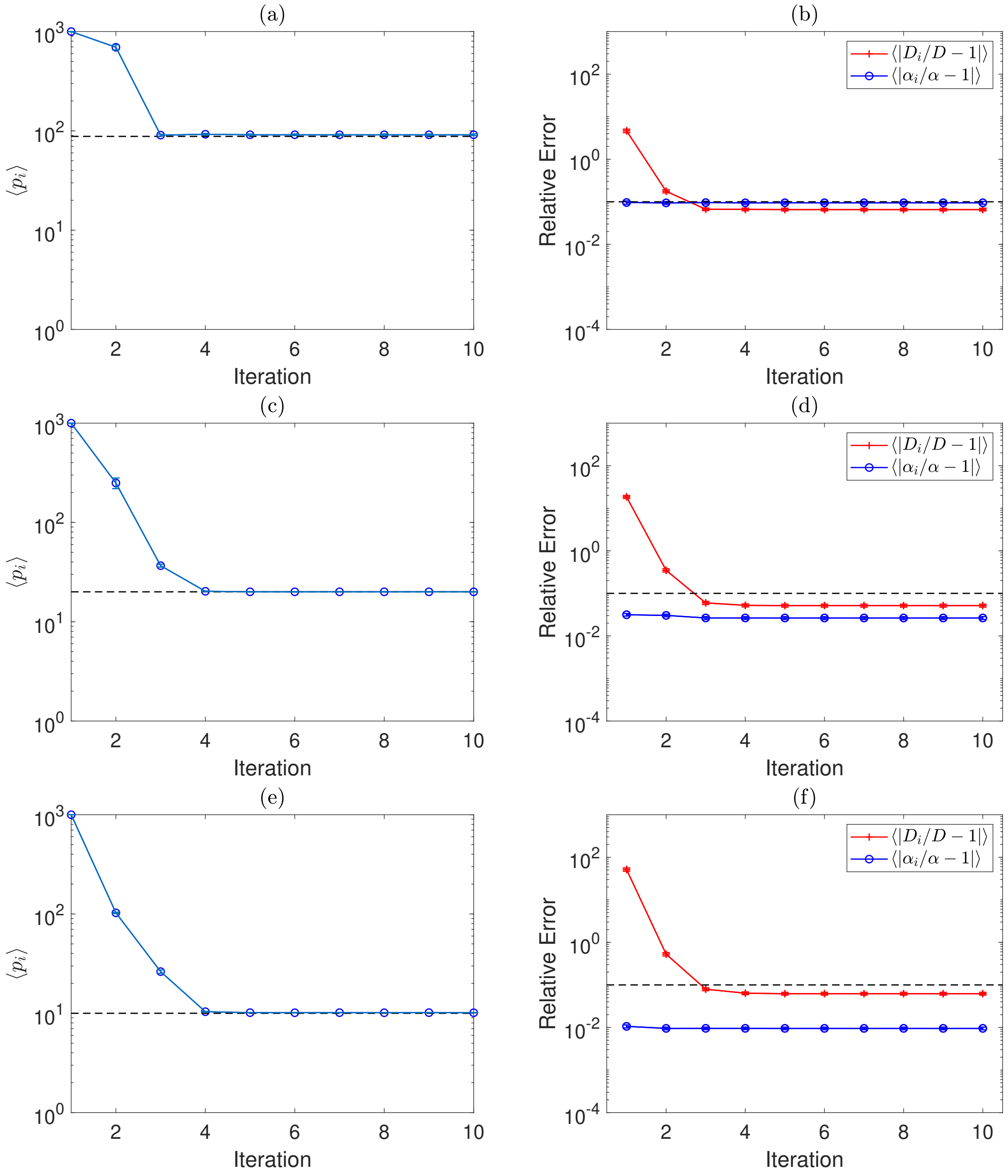

Figure 2.14: A plot of the value of $\langle p_i \rangle$ for each iteration with standard error bars [(a), (c), and (e)], along with a plot of the value of $\langle |D_i/D - 1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha - 1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b), (d) and (f)]. These experiments were for $D = 6\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 1$ and $N = 1000$, for $\Delta t = 1\,\text{s}$ [(a) and (b)], $\Delta t = 10\,\text{s}$ [(c) and (d)] and $\Delta t = 100\,\text{s}$ [(e) and (f)]. The dashed line in the plots of $\langle p_i \rangle$ correspond to $p_{opt} = 88$ (a), $p_{opt} = 20$ (c), and $p_{opt} = 10$ (e), while the dashed line in the plots of the relative errors correspond with the value 10^{-1}, indicating a 10% error.

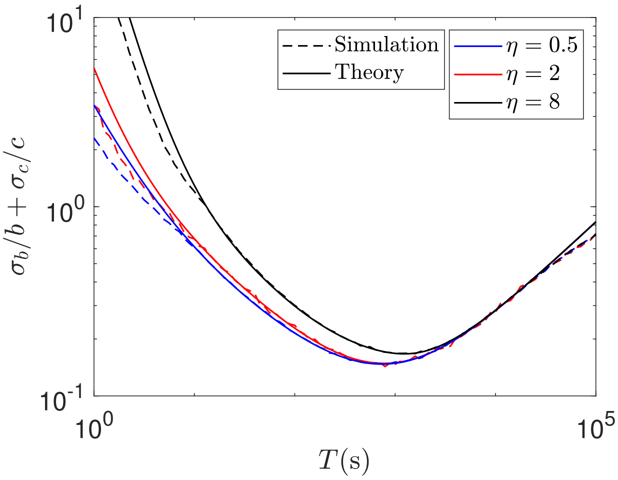

Figure 2.15: A plot of the theoretical value of $\sigma_b/b + \sigma_c/c$ (solid line) and its empirically estimated value using 1000 samples (dashed line) against T for $\eta = 0.5\,\mu$m (blue lines), $2\,\mu$m (red lines) and $8\,\mu$m (black lines). These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu$m/s, $N_S = 10$ and $N = 100$. For $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m, the optimal measurement time intervals are $T_{opt} \approx 735$ s, 780 s and 1216 s, respectively.

Here, the theoretical uncertainty $\sigma_b/b + \sigma_c/c$ is calculated over many different values of T using (2.43)–(2.49) with $p = N$ so that all the MSD points are used in the fitting, and is compared with simulations. The simulated results were found by calculating the MSD and using WLS regression to obtain estimates of b and c. This was repeated 1000 times to obtain estimates of σ_b and σ_c. Figure 2.15 shows the comparison between the theoretical and simulated value of $\sigma_b/b + \sigma_c/c$ over many different values of T for $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu$m/s, $N_S = 10$ and $N = 100$. We denote the value of T which minimises the uncertainty $\sigma_b/b + \sigma_c/c$ by T_{opt}. We see good agreement between the theory and simulations, particularly that both have their minimum's at the same measurement time intervals. For $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m, these optimal measurement time intervals are $T_{opt} \approx 735$ s, 780 s and 1216 s, respectively.

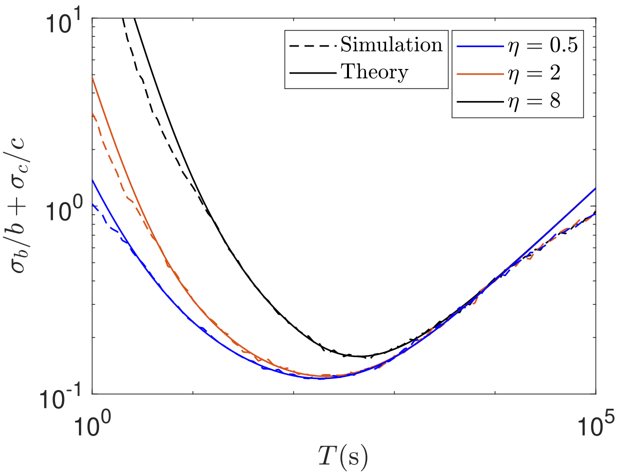

Figure 2.16: A plot of the theoretical value of $\sigma_b/b + \sigma_c/c$ (solid line) and its empirically estimated value using 1000 samples (dashed line) against T for $\eta = 0.5\,\mu$m (blue lines), $2\,\mu$m (red lines) and $8\,\mu$m (black lines). These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu$m/s, $N_S = 100$ and $N = 10$. For $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m, the optimal measurement time intervals are $T_{opt} \approx 189\,$s, $212\,$s and $445\,$s, respectively.

As before, to test the robustness of our calculations, we now test for the existence of an optimal measurement time interval with $N_S = 100$ and $N = 10$. Figure 2.16 shows the value of $\sigma_b/b + \sigma_c/c$, again with a comparison between theoretical expressions and simulations. We also notice that the optimal measurement time intervals have decreased to $T_{opt} \approx 189\,$s, $212\,$s and $445\,$s. We have observed in general that as the value of N is increased, the optimal measurement time interval T_{opt} also increases.

Again, we also look for the existence of an optimal measurement time interval which minimises the uncertainty $\sigma_a/a + \sigma_b/b + \sigma_c/c$ in order to additionally infer the static error η. The results are shown in Figure 2.17. Even when we include the value of η in the inference, we still find a distinct minimum in the value of $\sigma_a/a + \sigma_b/b + \sigma_c/c$. For $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m, we have corresponding optimal measurement time intervals $T_{opt} \approx 32\,$s, $178\,$s and $708\,$s, respectively.

A MATLAB routine to calculate the optimal measurement time interval T_{opt} is presented in Appendix B.

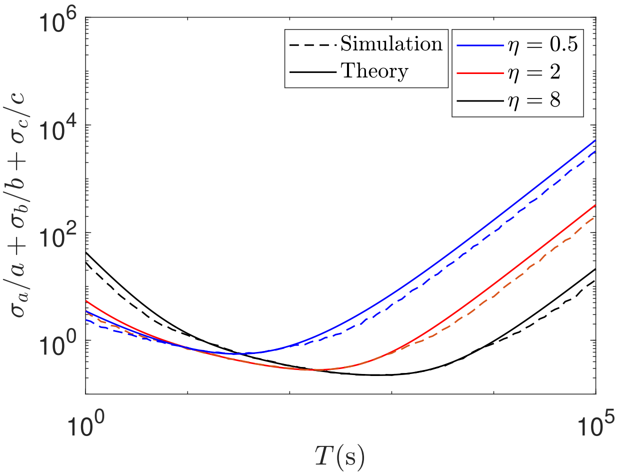

Figure 2.17: A plot of the theoretical value of $\sigma_a/a+\sigma_b/b+\sigma_c/c$ (solid line) and its empirically estimated value using 1000 samples (dashed line) against T for $\eta = 0.5\,\mu$m (blue lines), $2\,\mu$m (red lines) and $8\,\mu$m (black lines). These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu$m/s, $N_S = 100$ and $N = 10$. For $\eta = 0.5\,\mu$m, $2\,\mu$m and $8\,\mu$m, the optimal measurement time intervals are $T_{opt} \approx 32$ s, 178 s and 708 s, respectively.

2.9.2 Iterative algorithm to calculate the optimal measurement time interval

As before, since the function to calculate T_{opt} depends on the model parameters, another iterative algorithm was created. Note that in this case, each new iteration corresponds with repeating the experiment with a new measurement time interval T. To begin the iteration we need to provide an initial guess for T_{opt}, which we denote by T_0, with corresponding time interval between frames $\Delta t_0 = T_0/N$. The algorithm then adapts the measurement time interval according to Algorithm 2. The role of the under-relaxation parameter ω_i is to improve the robustness of the algorithm by reducing oscillations; this is effectively a low-pass filter for the time series of adjustments. For example, if the initial guess T_0 is far from the optimal value T_{opt}, then the values T_i will quickly be adapted towards the optimal time. Close to the optimal time the algorithm can display oscillations in the convergence behaviour i.e. $(T_{i+1} - T_i) \times (T_i - T_{i-1}) < 0$. When this occurs the relaxation parameter ω_i is decreased to smooth out the difference between iterates. Here, the tolerance τ determines the stopping criterion depending on the relative differences between two successive T_i values. The rate at which the value of ω_i is decreased in Step 10 is determined by the adjustment parameter ψ where $0 < \psi \leq 1$. In the experiments that follow, the value of $\psi = 0.8$ has been used; however, further experiments which use different values of ψ are tested below.

The iterative algorithm was tested for the two different initial measurement time intervals; a very large initial guess $T_0 = 10^7\,$s and a very small initial guess $T_0 = 10^{-3}\,$s. Both experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$ and $N = 100$; for these parameters $T_{opt} \approx 780\,$s. Similar to Algorithm 1, Steps 14 – 16 of Algorithm 2 will be ignored and instead all simulations are stopped after 10 iterations. These simulations were then repeated 100 times. The quantities $\langle T_i \rangle$, $\langle |D_i/D-1| \rangle$ and $\langle |\alpha_i/\alpha-1| \rangle$ are shown in Figure 2.18. Notice that the initial guess $T_0 = 10^7\,$s significantly overestimates the true value of T_{opt} but that $\langle T_i \rangle$ rapidly converges close to T_{opt}. While the value of $\langle |\alpha_i/\alpha - 1| \rangle$ becomes less accurate as we progress, the value of $\langle |D_i/D - 1| \rangle$ quickly falls from around a 1000% error to under a 10% error in a small number of iterations. When using a much smaller initial time of $T_0 = 10^{-3}\,$s, we see that $\langle T_i \rangle$ continues to rapidly converge to T_{opt}. Initially the value of $\langle |D_i/D-1| \rangle$ is of the order of magnitude 10^2 while $\langle |\alpha_i/\alpha-1| \rangle$ is of the order of magnitude 10^3, corresponding to a 10,000% and 100,000% error respectively. This highlights the fact that an incorrect choice of T can lead to very large inaccuracies in the value of inferred parameters. However, using the iterative algorithm we see that as the $\langle T_i \rangle$ values get closer to T_{opt}, the errors both reduce to under 10%. This stresses the importance of using T_{opt} when inferring D and α using all the MSD points in the fitting.

Algorithm 2 Iterative algorithm to find T_{opt} and estimates of D, α and η

```
    Input: Initial estimate of measurement time interval $T_0$ and measurement interval between frames $\Delta t_0$, number of time points $N$, adaptation parameter $\psi$ and convergence parameter $\tau$.
    Output: Estimates of optimal time $T_{opt}$ and parameters $D$, $\alpha$ and $\eta$.
 1: Guess an initial time $T_0$ with corresponding $\Delta t_0$ and set the relaxation parameter $\omega_0 = 1$ and set $i = 0$.
 2: if $i = 0$ then
 3:     $\sigma_n^{2\,(i)} = 1, \quad 1 \leq n \leq N,$
 4: else
 5:     $\sigma_n^{2\,(i)} = \sigma_n^2(D_{i-1}, \alpha_{i-1}, \eta_{i-1}, \Delta t_i)$ using (2.42), $\quad 1 \leq n \leq N.$
 6: end if
 7: Calculate the MSD at the $N$ time points with interval $\Delta t_i$ up to $T_i$ and use WLS on all the points with weights $1/\sigma_n^{2\,(i)}$ to get the parameter estimates $D_i$, $\alpha_i$ and $\eta_i$.
 8: Update $T_{i+1} = (1-\omega_i)T_i + \omega_i T_{opt}(D_i, \alpha_i, \eta_i, N)$ and calculate $\Delta t_{i+1} = T_{i+1}/N$.
 9: if $i \geq 2$ and $(T_{i+1} - T_i) \times (T_i - T_{i-1}) < 0$ then
10:     $\omega_{i+1} = \psi \times \omega_i, \quad 0 < \psi \leq 1$
11: else
12:     $\omega_{i+1} = \omega_i$
13: end if
14: if $|T_{i+1} - T_i|/T_{i+1} < \tau$ then
15:     end algorithm
16: else
17:     Set $i = i + 1$ and go back to Step 2
18: end if
```

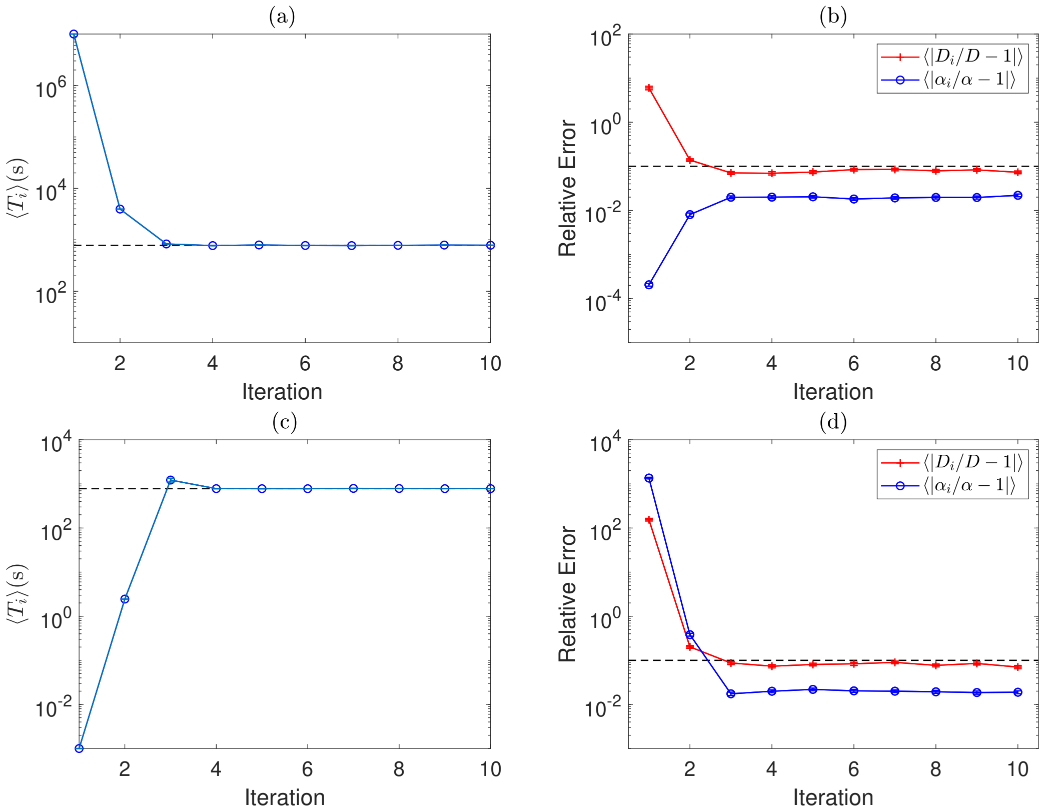

Figure 2.18: A plot of the value of $\langle T_i \rangle$ for each iteration with standard error bars [(a) and (c)], along with a plot of the value of $\langle |D_i/D - 1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha - 1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b) and (d)]. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$ and $N = 100$, with a starting time of $T_0 = 10^7\,\text{s}$ [(a) and (b)] and $T_0 = 10^{-3}\,\text{s}$ [(c) and (d)]. The dashed line in the plots of $\langle T_i \rangle$ correspond to $T_{opt} \approx 780\,\text{s}$, while the dashed line in the plots of the relative errors correspond with the value 10^{-1}, indicating a 10% error.

The iterative algorithm was tested for other values of the adaptation parameter ψ, as well as more values of D and α, to test the robustness of the algorithm. First, we look at testing the iterative algorithm but changing the adaptation parameter (Algorithm 2, Step 10) to $\psi = 0.5$ and $\psi = 0.2$. Both of these experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$ and $\eta = 2\,\mu\text{m}$ with the two initial measurement time intervals, $T_0 = 10^7\,\text{s}$ and $T_0 = 10^{-3}\,\text{s}$. The results are shown in Figures 2.19 and 2.20, respectively. The reduction of the adaptation parameter

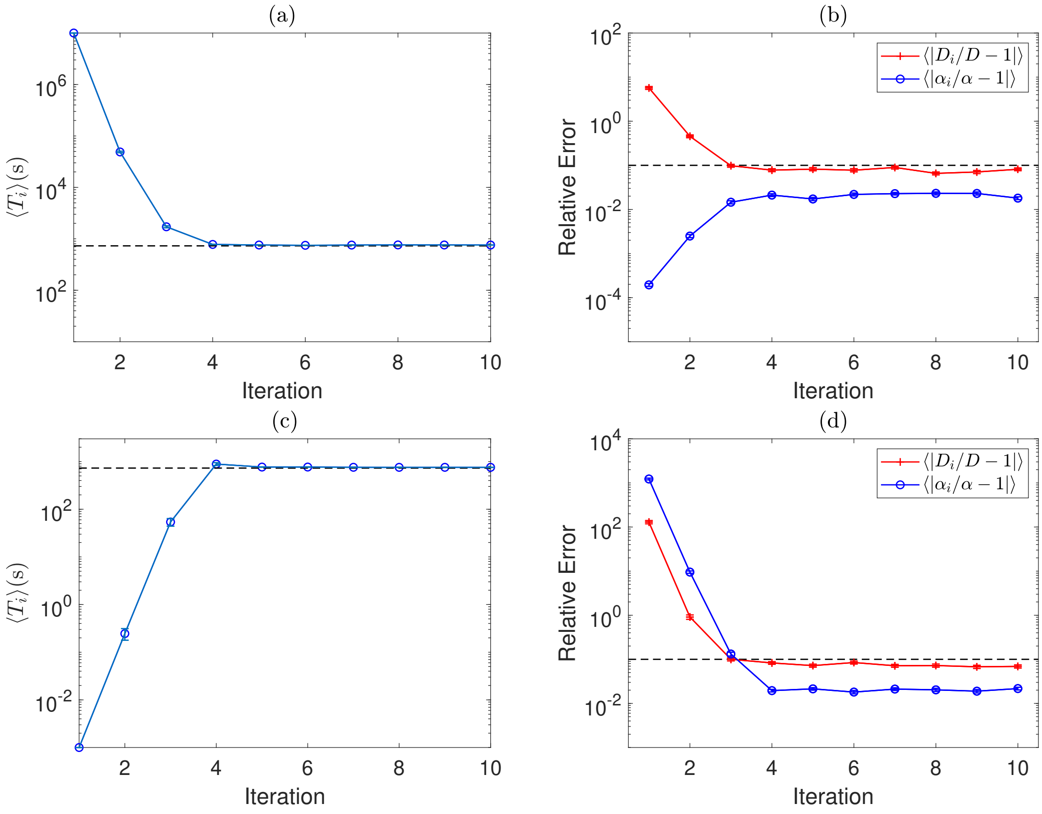

Figure 2.19: A plot of the value of $\langle T_i \rangle$ for each iteration with standard error bars [(a) and (c)], along with a plot of the value of $\langle |D_i/D-1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha-1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b) and (d)]. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$, $N = 100$ and $\psi = 0.5$, with a starting time of $T_0 = 10^7\,\text{s}$ [(a) and (b)] and $T_0 = 10^{-3}\,\text{s}$ [(c) and (d)]. The dashed line in the plots of $\langle T_i \rangle$ correspond to $T_{opt} \approx 780\,\text{s}$, while the dashed line in the plots of the relative errors correspond with the value 10^{-1}, indicating a 10% error.

does not appear to have affected the inference in either case. The value of $\langle T_i \rangle$ still converges to a time close to T_{opt} and the final relative error of both $\langle |D_i/D-1| \rangle$ and $\langle |\alpha_i/\alpha-1| \rangle$ are under 10% in both cases.

We now test adapting the values of D and α. The first experiment is for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 7\,\mu\text{m/s}$ and $\eta = 2\,\mu\text{m}$; for these parameters $T_{opt} \approx 32\,\text{s}$. The second experiment is for $D = 6\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$ and $\eta = 2\,\mu\text{m}$; for these parameters $T_{opt} \approx 2195\,\text{s}$. The value of the adaptation parameter for both experiments was set back to $\psi = 0.8$. The results are plotted in Figure 2.21 and 2.22, respectively. As before, the results are very similar to what has been seen

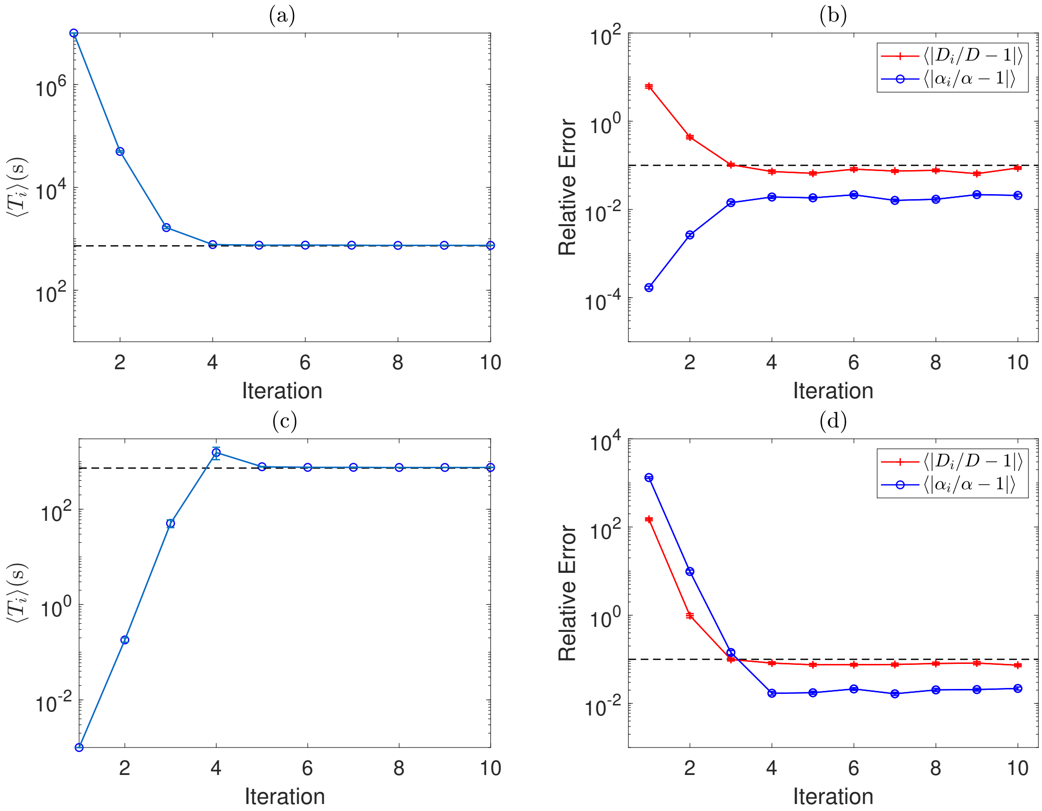

Figure 2.20: A plot of the value of $\langle T_i \rangle$ for each iteration with standard error bars [(a) and (c)], along with a plot of the value of $\langle |D_i/D-1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha-1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b) and (d)]. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$, $N = 100$ and $\psi = 0.2$, with a starting time of $T_0 = 10^7\,\text{s}$ [(a) and (b)] and $T_0 = 10^{-3}\,\text{s}$ [(c) and (d)]. The dashed line in the plots of $\langle T_i \rangle$ correspond to $T_{opt} \approx 780\,\text{s}$, while the dashed line in the plots of the relative errors correspond with the value 10^{-1}, indicating a 10% error.

before. As both cases converge to T_{opt} the relative errors are reduced, particularly for a smaller value of the measurement time interval where the reduction is more significant.

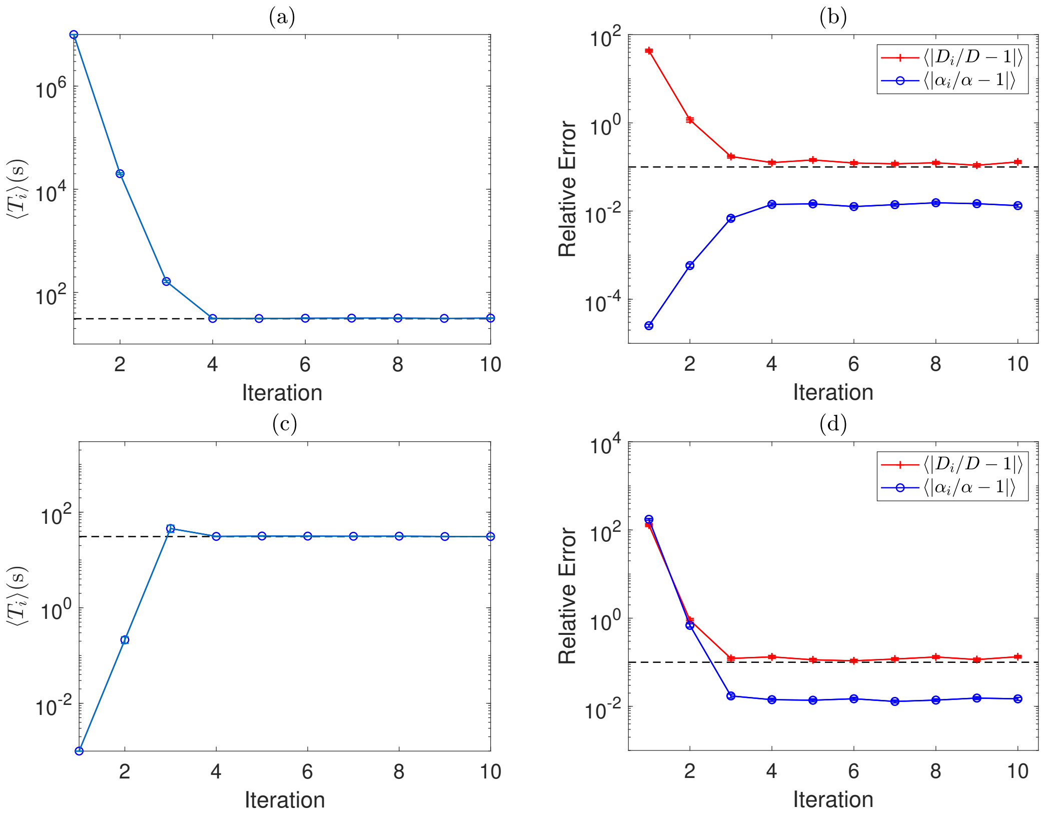

Figure 2.21: A plot of the value of $\langle T_i \rangle$ for each iteration with standard error bars [(a) and (c)], along with a plot of the value of $\langle |D_i/D-1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha-1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b) and (d)]. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 7\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$ and $N = 100$, with a starting time of $T_0 = 10^7\,\text{s}$ [(a) and (b)] and $T_0 = 10^{-3}\,\text{s}$ [(c) and (d)]. The dashed line in the plots of $\langle T_i \rangle$ correspond to $T_{opt} \approx 32\,\text{s}$, while the dashed line in the plots of the relative errors correspond with the value 10^{-1}, indicating a 10% error.

2.9.3 Single particle parameter estimation using the optimal measurement time interval

The results for T_{opt} can also extend to the single particle case for the same reasons as the p_{opt} method. As before, the optimal measurement time interval will be the same for the ensemble of particles and the single particle case. Figure 2.23 compares the performances using the same initial measurement time intervals, $T_0 = 10^7\,\text{s}$ and $T_0 = 10^{-3}\,\text{s}$, for the same parameter values as those in Figure 2.18 but with $N_S = 1$. As expected, since we have less data in the single particle case,

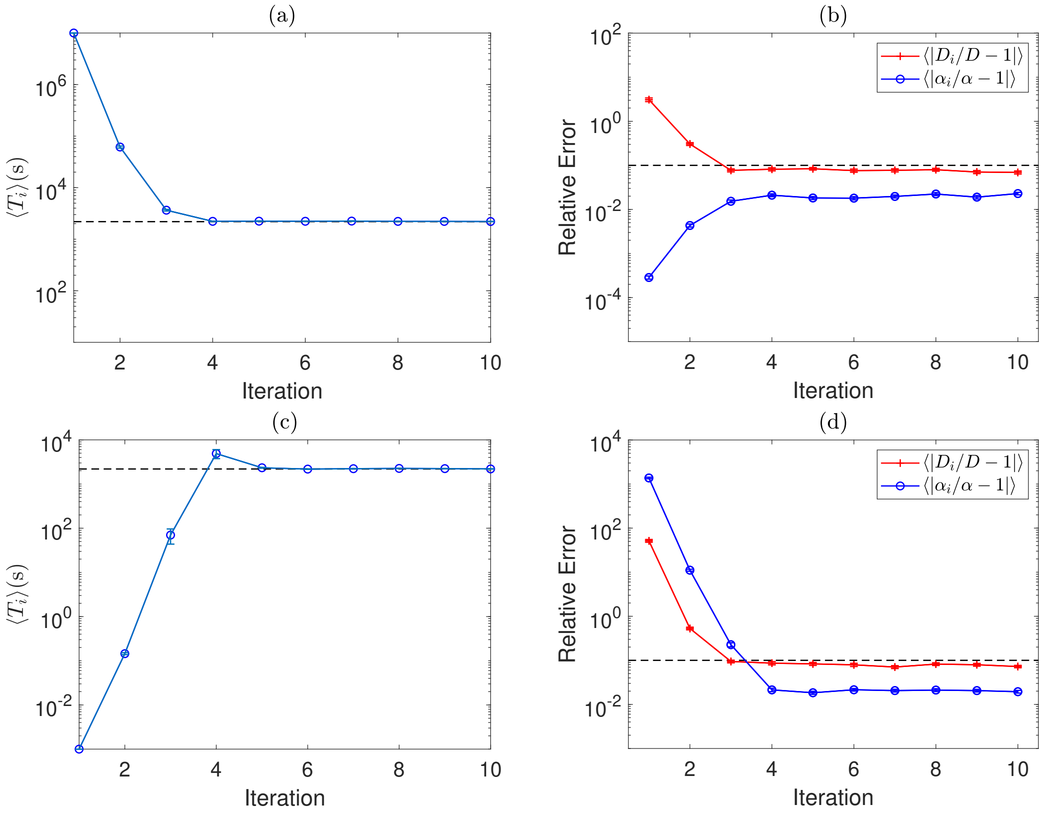

Figure 2.22: A plot of the value of $\langle T_i \rangle$ for each iteration with standard error bars [(a) and (c)], along with a plot of the value of $\langle |D_i/D - 1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha - 1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b) and (d)]. These experiments were for $D = 6\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$ and $N = 100$, with a starting time of $T_0 = 10^7\,\text{s}$ [(a) and (b)] and $T_0 = 10^{-3}\,\text{s}$ [(c) and (d)]. The dashed line in the plots of $\langle T_i \rangle$ correspond to $T_{opt} \approx 2195\,\text{s}$, while the dashed line in the plots of the relative errors correspond with the value 10^{-1}, indicating a 10% error.

the relative errors are higher than in the ensemble of particles case. However, the value of $\langle T_i \rangle$ continues to converge to T_{opt} in a small number of iterations and we observe that the results for $\langle |D_i/D-1| \rangle$ and $\langle |\alpha_i/\alpha-1| \rangle$ have similar dynamics to the ensemble case. This shows the strength of the iterative algorithm as we obtain good results even in the single particle case where we have less information.

The practical feasibility of this procedure to change the measurement time interval depends on the chosen application domain. For instance, in environmental statistics, where the task is, for example, to monitor the spread of pollutants and contaminants in ground water, it is common practice to repeatedly estimate the

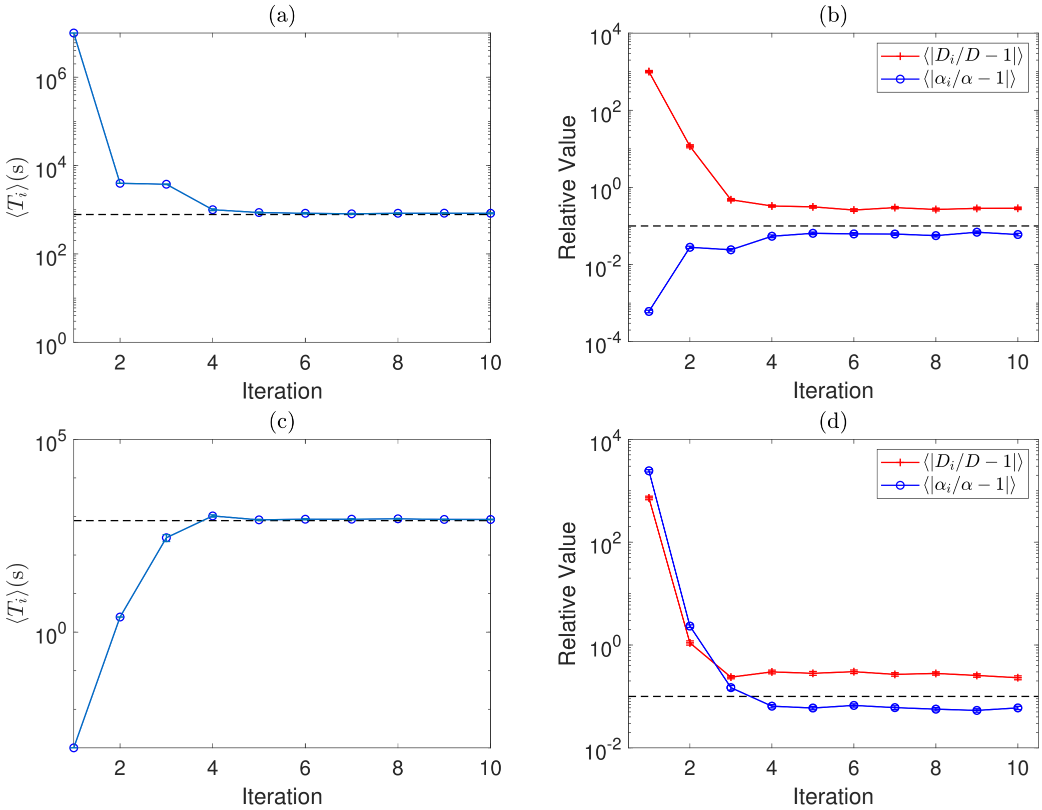

Figure 2.23: A plot of the value of $\langle T_i \rangle$ for each iteration with standard error bars [(a) and (c)], along with a plot of the value of $\langle |D_i/D - 1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha - 1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b) and (d)]. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$ and $\eta = 2\,\mu\text{m}$, $N_S = 1$ and $N = 100$, with a starting time of $T_0 = 10^7\,\text{s}$ [(a) and (b)] and $T_0 = 10^{-3}\,\text{s}$ [(c) and (d)]. The dashed line in the plots of $\langle T_i \rangle$ correspond to $T_{opt} \approx 780\,\text{s}$, while the dashed line in the plots of the relative errors correspond with the value 10^{-1}, indicating a 10% error.

same physical quantities. This setting therefore naturally lends itself to the integration of the proposed iterative adjustment scheme. For other applications, like the study of collective cell movement with high-resolution microscopy, a change of the experimental protocol may be required, to allow (and budget) for a series of experiments that enable iterative adjustments of the measurement time intervals.

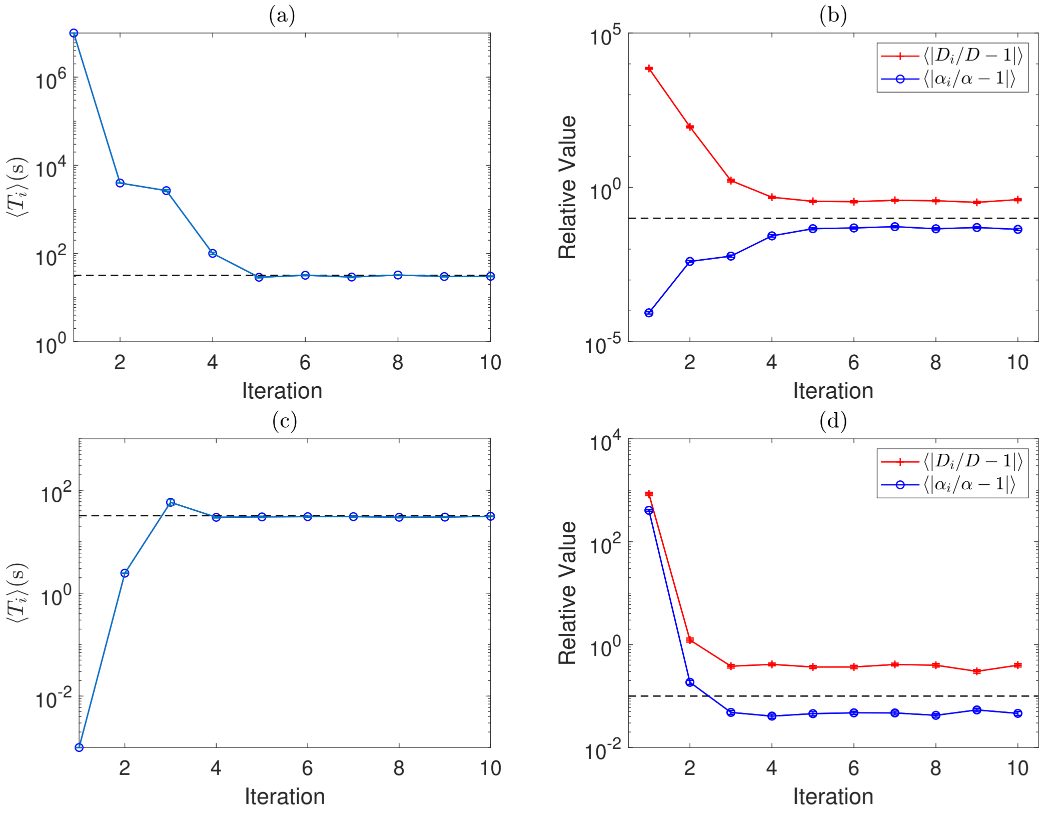

Figure 2.24: A plot of the value of $\langle T_i \rangle$ for each iteration with standard error bars [(a) and (c)], along with a plot of the value of $\langle |D_i/D-1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha-1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b) and (d)]. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 7\,\mu\text{m/s}$ and $\eta = 2\,\mu\text{m}$, $N_S = 1$ and $N = 100$, with a starting time of $T_0 = 10^7\,\text{s}$ [(a) and (b)] and $T_0 = 10^{-3}\,\text{s}$ [(c) and (d)]. The dashed line in the plots of $\langle T_i \rangle$ correspond to $T_{opt} \approx 32\,\text{s}$, while the dashed line in the plots of the relative errors correspond with the value 10^{-1}, indicating a 10% error.

We look at using the iterative algorithm for the single particle case for different values of D and α. First we test for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 7\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 1$ and $N = 100$, again for both $T_0 = 10^7\,\text{s}$ and $T_0 = 10^{-3}\,\text{s}$. For these parameter values, we have that $T_{opt} \approx 32\,\text{s}$. The results are given in Figure 2.24. The algorithm takes a few more iterations to converge compared with the case in Figure 2.23 but this is likely due to the value of T_{opt} being further from the initial guess T_0. However, we continue see convergence and similar dynamics in the values of $\langle |D_i/D-1| \rangle$ and $\langle |\alpha_i/\alpha-1| \rangle$.

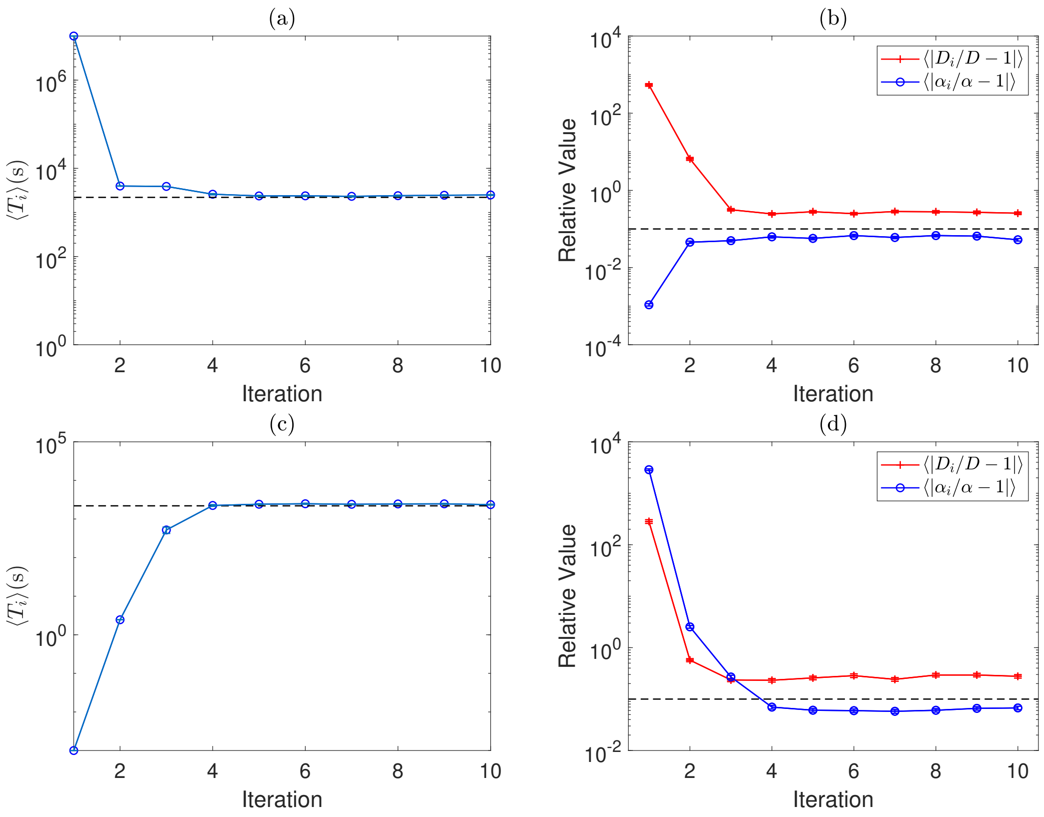

Figure 2.25: A plot of the value of $\langle T_i \rangle$ for each iteration with standard error bars [(a) and (c)], along with a plot of the value of $\langle |D_i/D - 1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha - 1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b) and (d)]. These experiments were for $D = 6\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$ and $\eta = 2\,\mu\text{m}$, $N_S = 1$ and $N = 100$, with a starting time of $T_0 = 10^7\,\text{s}$ [(a) and (b)] and $T_0 = 10^{-3}\,\text{s}$ [(c) and (d)]. The dashed line in the plots of $\langle T_i \rangle$ correspond to $T_{opt} \approx 2195\,\text{s}$, while the dashed line in the plots of the relative errors correspond with the value 10^{-1}, indicating a 10% error.

The final parameter values tested are $D = 6\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 1$ and $N = 100$, again for both $T_0 = 10^7\,\text{s}$ and $T_0 = 10^{-3}\,\text{s}$. For these parameter values, we have that $T_{opt} \approx 2195\,\text{s}$. The results are given in Figure 2.25. The algorithm continues to converge in a small number of iterations and reduce the errors significantly in both cases.

2.9.4 Motion blur

It is of interest what effect motion blur has on the MSD. To answer this question we follow the approach of Goulian and Simon [77], who considered the effect of full-frame motion blur on the MSD of a purely diffusive motion. We assume that the particle position in the nth image, $\boldsymbol{x}(t)$ with $t = n\Delta t$, is taken to be the combination of the average of the true position of the particle $\tilde{\boldsymbol{x}}(t)$ over the time frame and the addition of static Gaussian measurement error. If the time interval Δt is divided by M smaller microsteps of size $\delta t = \Delta t/M$, then we assume that the measured position

$$\boldsymbol{x}(t) = \frac{1}{M+1}\sum_{m=0}^{M} \tilde{\boldsymbol{x}}(t - m\delta t) + \boldsymbol{\eta}_t \equiv \bar{\boldsymbol{x}}(t) + \boldsymbol{\eta}_t. \tag{2.59}$$

We then have for $t' \geq t + M\delta t$,

$$\begin{aligned}\mathbb{E}((\boldsymbol{x}(t) - \boldsymbol{x}(t'))^2) &= \mathbb{E}((\bar{\boldsymbol{x}}(t) + \boldsymbol{\eta}_t - \bar{\boldsymbol{x}}(t') - \boldsymbol{\eta}_{t'})^2)\\ &= \mathbb{E}((\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}(t'))^2) + 2\mathbb{E}((\boldsymbol{\eta}_t - \boldsymbol{\eta}_{t'})(\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}(t')))\\ &+ \mathbb{E}((\boldsymbol{\eta}_t - \boldsymbol{\eta}_{t'})^2)\\ &= \mathbb{E}((\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}(t'))^2) + 4\eta^2. \end{aligned} \tag{2.60}$$

Furthermore, we have

$$
\begin{aligned}
&\mathbb{E}((\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}(t'))^2)\\
&= \frac{1}{(M+1)^2} \sum_{m_1,m_2=0}^{M} \mathbb{E}([\tilde{\boldsymbol{x}}(t - m_1\delta t) - \tilde{\boldsymbol{x}}(t' - m_1\delta t)][\tilde{\boldsymbol{x}}(t - m_2\delta t) - \tilde{\boldsymbol{x}}(t' - m_2\delta t)]\\
&= \frac{1}{2(M+1)^2} \sum_{m_1,m_2=0}^{M} 2\mathbb{E}([\tilde{\boldsymbol{x}}(t - m_1\delta t) - \tilde{\boldsymbol{x}}(t' - m_2\delta t)])\\
&- \mathbb{E}(\tilde{\boldsymbol{x}}(t - m_1\delta t) - \tilde{\boldsymbol{x}}(t - m_2\delta t)) - \mathbb{E}(\tilde{\boldsymbol{x}}(t' - m_1\delta t) - \tilde{\boldsymbol{x}}(t' - m_2\delta t))\\
&= \frac{1}{(M+1)^2} \sum_{m_1,m_2=0}^{M} \big[\alpha^2(t' - t - (m_2 - m_1)\delta t)^2 + 4D(t' - t - (m_2 - m_1)\delta t)\\
&-(\alpha|m_2 - m_1|\delta t)^2 + 4D|m_2 - m_1|\delta t)\big]\\
&= \alpha^2(t' - t)^2 + 4D\left(t' - t - \frac{(M+1)^2 - 1}{3(M+1)}\delta t\right). \qquad (2.61)
\end{aligned}
$$

For the MSD we take $t' - t = n\Delta t$, and using (2.60) and (2.61) and letting $M \to \infty$, we finally get

$$
MSD(n\Delta t) = \alpha^2(n\Delta t)^2 + 4Dn\Delta t + 4\left(\eta^2 - \frac{D\Delta t}{3}\right). \qquad (2.62)
$$

To our knowledge, this result has never been derived before. It is clear that motion blur only effects the offset of the MSD curve for zero time lags. In particular, if the MSD is fitted by a quadratic polynomial then the quadratic term can be used to estimate the drift velocity and the linear term used to estimate the diffusion coefficient.

The effect of motion blur and static error was simulated by first generating true particle trajectories using microsteps of size

$$
\delta t = \frac{\Delta t}{M}, \qquad (2.63)
$$

where Δt is the time between frames and M is the number of microsteps between frames. For the jth particle the displacement was updated as follows:

$$\tilde{\boldsymbol{x}}_{i+1}^{(j)} = \tilde{\boldsymbol{x}}_{i}^{(j)} + \boldsymbol{\alpha}\delta t + \sqrt{2D\delta t}\,\mathbf{N}(0,1), \qquad i = 1, \ldots, NM, \tag{2.64}$$

with $\tilde{\boldsymbol{x}}_1 = 0$. The measured displacement at $t = (n-1)\Delta t$ was obtained by averaging the true position of the particle over the previous M microsteps to simulate full-frame motion blur, and a Gaussian static error was then added so that

$$\boldsymbol{x}_{n}^{(j)} = \frac{1}{M+1} \sum_{i=(n-2)M+1}^{(n-1)M+1} \tilde{\boldsymbol{x}}_{i}^{(j)} + \boldsymbol{\eta}_n, \qquad n = 2, \ldots, N+1, \tag{2.65}$$

and $\boldsymbol{x}_1^{(j)} = \boldsymbol{\eta}_1$.

The following results are for calculating the optimal number of fitting points. First, we look for the existence of an optimal number of fitting points which minimises the value of $\sigma_b/b + \sigma_c/c$. Figure 2.26 shows a comparison between the theoretical value of $\sigma_b/b + \sigma_c/c$ assuming no motion blur and its simulated value which includes motion blur. We see that for small Δt, corresponding to using all the MSD points in the fitting, motion blur does not seem to have an effect on the results. However, when Δt is increased, motion blur appears to effect the value of $\sigma_b/b + \sigma_c/c$. While the theoretical optimal number of fitting points is 8, the simulated optimal number of fitting points is 5.

Since it can be seen that motion blur has an effect on the value of p_{opt}, we look to see whether using our theoretical value for p_{opt} on data which includes motion blur leads to a change in results. To test this, we will use the iterative algorithm to see the effect of motion blur on the inference of D and α. For this, the simulated data will include motion blur but will be driven towards the theoretical p_{opt} which does not include motion blur (see Appendix A). In the absence of motion blur, the regression coefficient $c = 4\eta^2$. However, when motion blur is present, the

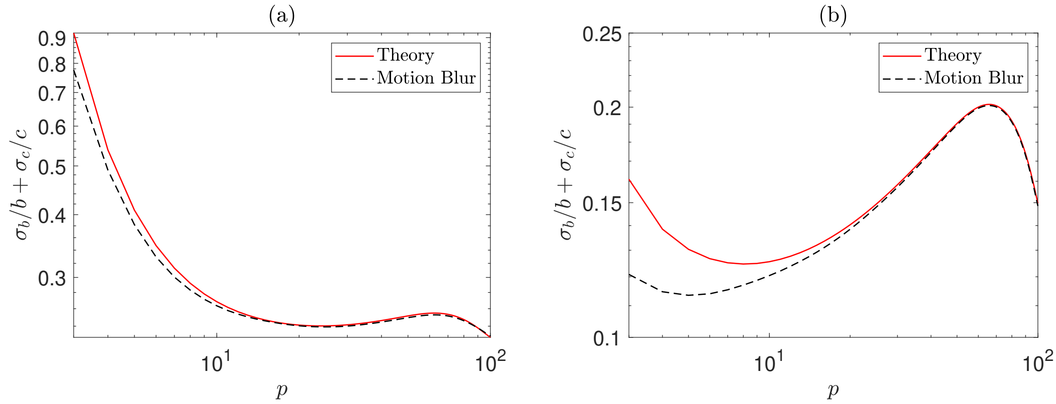

Figure 2.26: A plot of the theoretical value of $\sigma_b/b + \sigma_c/c$ assuming no motion blur (solid red line) and an empirically estimated value using 1000 samples which include motion blur (dashed black line) when fit with the first p MSD points for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$ and $N = 100$, for $\Delta t = 1\,\text{s}$ giving $T = 100\,\text{s}$ (a), and $\Delta t = 10\,\text{s}$ giving $T = 1000\,\text{s}$ (b). The optimal number of fitting points for the curves in (a) are both 100, while for the curves in (b) it is 8 for the theoretical curve and 5 for the empirically estimated curve.

regression coefficient becomes $c = 4\eta^2 - \frac{4}{3}D\Delta t$ and so the value of η will be inferred with this in mind. The results are shown in Figure 2.27. For this we have $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$ and $N = 1000$ for $\Delta t = 1\,\text{s}$, $\Delta t = 10\,\text{s}$ and $\Delta t = 100\,\text{s}$. Note that these plots are comparable with Figure 2.9. Even though we are driving p_i to the theoretical optimal number of fitting points which does not include motion blur, we do not see a significant difference in the inference of D and α when motion blur is included. This suggests that our theory for the optimal number of fitting points can still be used when motion blur is present.

To further test the effects of motion blur, we also tested the iterative algorithm for a single particle. The parameters values were chosen to be the same as those above but for $N_S = 1$. The results are shown in Figure 2.28. Note that these plots are comparable with Figure 2.12. Again, we do not see a significant difference with the inclusion of motion blur, further suggesting that our theoretical p_{opt} can be used when motion blur is present.

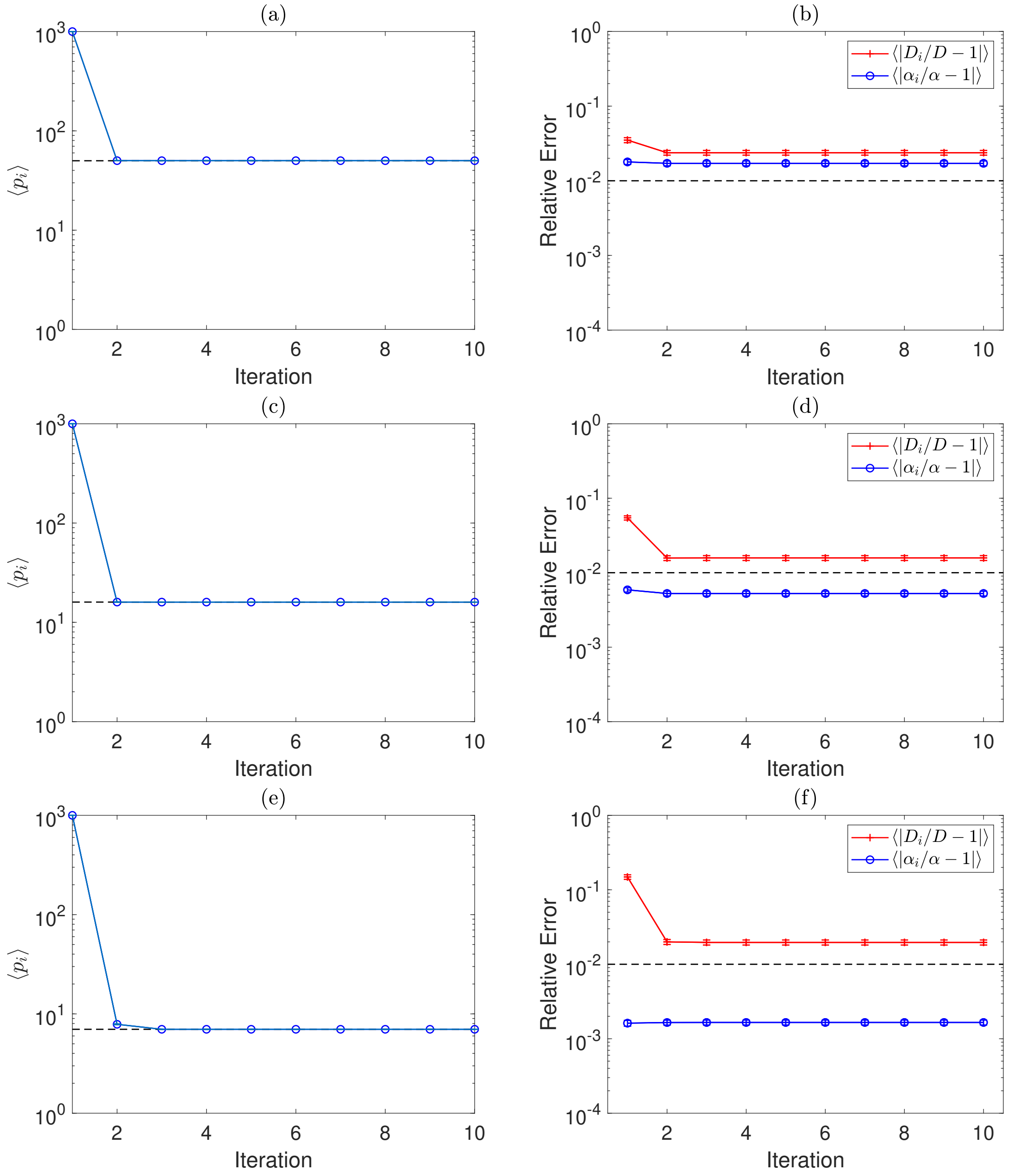

Figure 2.27: A plot of the value of $\langle p_i \rangle$ for each iteration with standard error bars [(a), (c), and (e)], along with a plot of the value of $\langle |D_i/D - 1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha - 1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b), (d) and (f)]. The simulated values include motion blur. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 10$ and $N = 1000$, for $\Delta t = 1\,\text{s}$ [(a) and (b)], $\Delta t = 10\,\text{s}$ [(c) and (d)] and $\Delta t = 100\,\text{s}$ [(e) and (f)]. The dashed line in the plots of $\langle p_i \rangle$ correspond to $p_{opt} = 50$ (a), $p_{opt} = 16$ (c), and $p_{opt} = 7$ (e), while the dashed line in the plots of the relative errors correspond with the value 10^{-2}, indicating a 1% error.

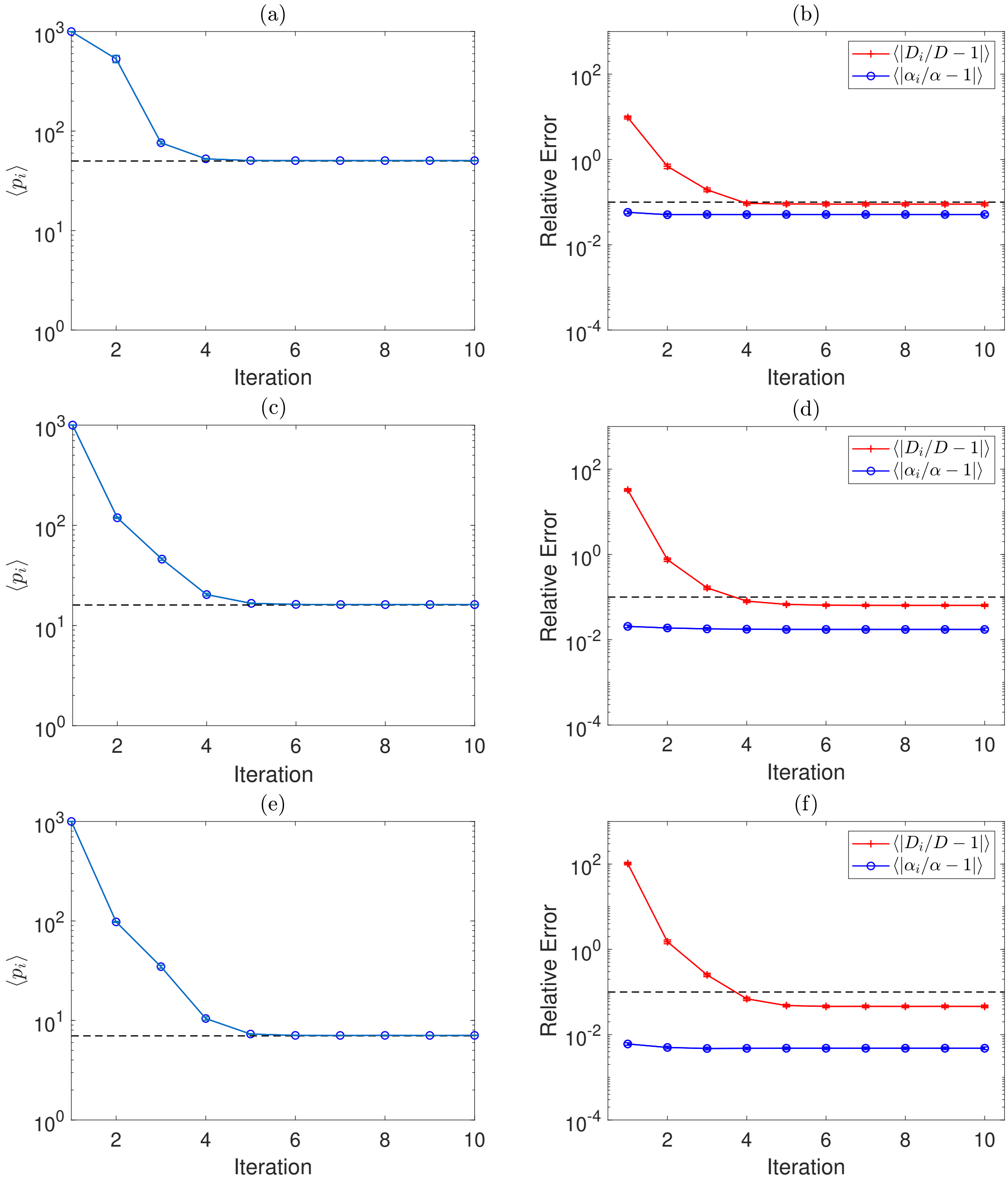

Figure 2.28: A plot of the value of $\langle p_i \rangle$ for each iteration with standard error bars [(a), (c), and (e)], along with a plot of the value of $\langle |D_i/D - 1| \rangle$ (red line with crosses) and $\langle |\alpha_i/\alpha - 1| \rangle$ (blue line with circles) for each iteration with standard error bars [(b), (d) and (f)]. The simulated values include motion blur. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $N_S = 1$ and $N = 1000$, for $\Delta t = 1\,\text{s}$ [(a) and (b)], $\Delta t = 10\,\text{s}$ [(c) and (d)] and $\Delta t = 100\,\text{s}$ [(e) and (f)]. The dashed line in the plots of $\langle p_i \rangle$ correspond to $p_{opt} = 50$ (a), $p_{opt} = 16$ (c), and $p_{opt} = 7$ (e), while the dashed line in the plots of the relative errors correspond with the value 10^{-2}, indicating a 1% error.

2.9.5 Determination of the drift direction

Until now we have assumed that the main interest is the determination of the drift magnitude and the diffusion coefficient. If the drift angle is also of interest then this can also be inferred using the trajectory data. To calculate the mean direction and measures of the spread about the mean requires the use of circular statistics, details of which can be found in, for example, Fisher [78] or Mardia and Jupp [79]. Following in the same vain as the estimation of the MSD, we first calculate the ensemble overlapping time-averaged quantities

$$C_n = \sum_{j=1}^{N_S} \sum_{i=1}^{N+1-n} \cos\,\theta_{i,n}^{(j)} \quad \text{and} \quad S_n = \sum_{j=1}^{N_S} \sum_{i=1}^{N+1-n} \sin\,\theta_{i,n}^{(j)}, \tag{2.66}$$

where

$$\cos\,\theta_{i,n}^{(j)} = \frac{x_{i+n}^{(j)} - x_i^{(j)}}{|\boldsymbol{x}_{i+n}^{(j)} - \boldsymbol{x}_i^{(j)}|} \quad \text{and} \quad \sin\,\theta_{i,n}^{(j)} = \frac{y_{i+n}^{(j)} - y_i^{(j)}}{|\boldsymbol{x}_{i+n}^{(j)} - \boldsymbol{x}_i^{(j)}|}. \tag{2.67}$$

The resultant vector using data with a time lag of $n\Delta t$ is $R_n = \sqrt{C_n^2 + S_n^2}$ from which we can define the cosine and sine of the average angle using a time lag of n by

$$\cos\overline{\theta}_{d,n} = C_n/R_n \quad \text{and} \quad \sin\overline{\theta}_{d,n} = S_n/R_n. \tag{2.68}$$

The average angle using displacements with a time lag of $n\Delta t$ is given by

$$\overline{\theta}_{d,n} = \begin{cases} \tan^{-1}(S_n/C_n), & S_n > 0,\ C_n > 0, \\ \tan^{-1}(S_n/C_n) + \pi, & C_n > 0, \\ \tan^{-1}(S_n/C_n) + 2\pi, & S_n < 0,\ C_n > 0. \end{cases} \tag{2.69}$$

To help quantify the uncertainty in inferring the drift angle, the circular variance V_n is defined as

$$V_n = 1 - \overline{R}_n, \tag{2.70}$$

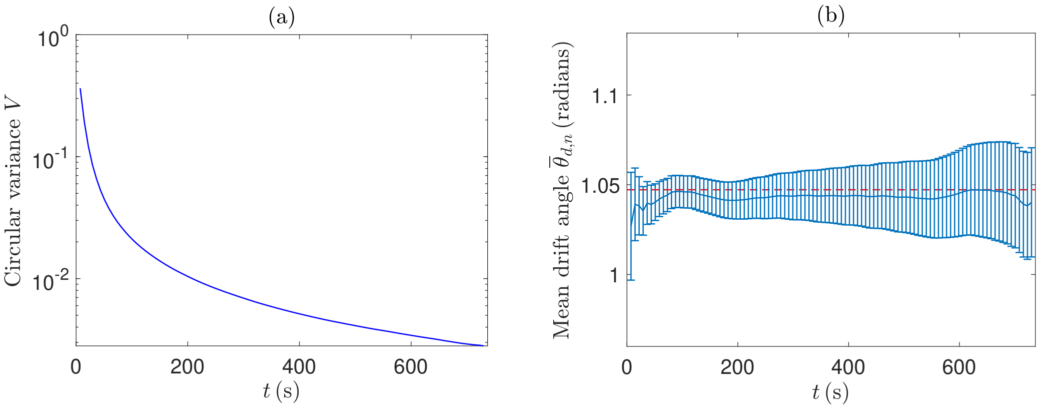

Figure 2.29: A plot of the circular variance V (a) and the mean drift angle $\overline{\theta}_{d,n}$ with standard error bars (b) for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $\eta = 2\,\mu\text{m}$, $\theta_d = \pi/3$ (1.047) radians, $N_S = 10$ and $N = 100$.

where

$$\overline{R}_n = R_n/(N_S(N+1-n)) \tag{2.71}$$

is the mean resultant length. Finally, the first and second central trigonometric moments

$$m_{1,n} = \frac{1}{N_S(N+1-n)} \sum_{j=1}^{N_S} \sum_{i=1}^{N+1-n} \cos\,(\theta_{i,n}^{(j)} - \overline{\theta}_n), \tag{2.72}$$

and

$$m_{2,n} = \frac{1}{N_S(N+1-n)} \sum_{j=1}^{N_S} \sum_{i=1}^{N+1-n} \cos\,(2(\theta_{i,n}^{(j)} - \overline{\theta}_n)), \tag{2.73}$$

can be used to define the sample circular dispersion

$$\hat{\delta}_n = (1 - m_{2,n})(2m_{1,n}^2). \tag{2.74}$$

The error bars in the following plots use the circular standard error

$$\hat{\sigma}_n = \sqrt{\hat{\delta}_n/(N_S(N+1-n))}. \tag{2.75}$$

Figure 2.29 shows the circular variance and mean drift angle using $N_S = 10$ and $N = 100$, where we have assumed that the optimal measurement time interval has been used. We can see that the variance in the estimation of the drift angle decreases rapidly over time. This would suggest that the optimal time to infer the drift angle would be at the maximal measurement time interval. However, due to the lack of samples in the time averaging using the large time lags, we find that there is very little difference in the quality of the inferred drift angle other than when very short lags are used.

2.10 Conclusions

In this chapter, we looked at the optimal estimation of the diffusion coefficient D and the drift magnitude α from the drift-diffusion SDE (2.1) using weighted least squares regression. We assumed that the trajectories are also subject to static localisation error with strength η^2. Optimisation of the fitting process was carried out with respect to the number of regression points p used and the time interval T over which measurements were made.

We estimated the model parameters using weighted least squares regression of the ensemble time-average overlapping MSD (2.22). For the regression weights, it is common to use the variance of the predictors, and so a theoretical expression for the variance of the MSD was calculated, as given by (2.42). Simulations indicate that the empirical estimate of the MSD is noisy for small values of T, while the uncertainty in the MSD increases as T increases. This latter feature is a result of the variance of the MSD increasing cubically with time. Also, the quadratic form of the MSD gives rise to two different time scales associated with the diffusive and drift processes. Therefore, we find that an intermediate value of T is needed to balance these features of the MSD.

Since the MSD is of the form $a + bt + ct^2$, where $a = 4\eta^2$, $b = 4D$ and $c = \alpha^2$, we use quadratic regression to infer D and α. Using theoretical estimates of the variance of the regression coefficients, σ_a^2, σ_b^2 and σ_c^2 (2.43)–(2.45), we looked to minimise $\sigma_b/b + \sigma_c/c$. To calculate the regression variances, we were required to find a theoretical expression for the covariance of the MSD, given by (2.58).

To optimise the inference with respect to the number of regression points p, we calculate the theoretical value of $\sigma_b/b + \sigma_c/c$ as a function of p. Figures 2.6 and 2.7 show the existence of p_{opt} which minimises the uncertainty in the inference of the model parameters. Additionally, Figure 2.8 shows that an optimal number of regression points exists which minimises $\sigma_a/a + \sigma_b/b + \sigma_c/c$ in order to optimally infer D, α and η together. Algorithm 1 describes a procedure to iteratively find the value of p_{opt}. We see, from Figures 2.9–2.11, that the algorithm converges quickly to p_{opt}. The relative error in the parameters decrease as we approach p_{opt}, in particular, the relative error in D. We also see that the choice of the timestep Δt is important in optimising the inference of D and α. When considering the single particle case, we showed that the relative error in D can decrease significantly when p_{opt} is used, in one case showing a 1000% improvement.

To optimise the inference with respect to the measurement time interval T, we calculated the theoretical value of $\sigma_b/b + \sigma_c/c$ as a function of T. The existence of T_{opt} is shown in Figures 2.15 and 2.16, and again, an optimal measurement time interval is shown to exist which minimises $\sigma_a/a + \sigma_b/b + \sigma_c/c$, as shown in Figure 2.17. Algorithm 2 describes a procedure to iteratively find the value of T_{opt}. Compared with the regression points case, we found that using a value of T other than T_{opt} can result in large errors in both D and α. Looking, for example, at Figure 2.18, we see that taking a value of T larger than T_{opt}, while giving a low relative error in α, gives a large relative error in D. When we converge to T_{opt}, we see a balance in the relative errors. When we use a value of T smaller than T_{opt}, the relative errors in D and α are very high but decrease dramatically as we converge to T_{opt}. This shows the importance of using the optimal value of T. We see similar results for the single particle case.

We showed that when motion blur is considered, only the constant term in the MSD changes, given by (2.62). We found, as seen in Figure 2.26, that the inclusion of motion blur can change the value of the optimal number of regression points p_{opt}. However, when we use our previous theoretical value of p_{opt} on data which includes motion blur, we do not see a significant difference in the inference of the model parameters, as seen in Figures 2.27 and 2.28.

Finally, we considered the optimal estimation of the drift direction θ_d. We found that, although the circular variance given by (2.70) decreases as t increases, the lack of samples for large time lags results in little difference in the quality of the average drift angle (2.69).

The approach described in this chapter provides a method for optimally estimating the coefficients from the drift-diffusion model (2.1) where analytical forms for the MSD and its properties can be found. When a more complex model is considered, where an analytical form for the summary statistics is not available, we require a more general approach for parameter inference.

Chapter 3

Approximate Bayesian computation

3.1 Bayes' theorem

Within Bayesian inference, it is of interest to calculate the distribution of the unknown parameters $\boldsymbol{\theta}$ of a statistical model given some observed data $\boldsymbol{y}$. This distribution, denoted by $\pi(\boldsymbol{\theta}|\boldsymbol{y})$, is called the posterior distribution, and can be calculated using Bayes' theorem

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})\,\pi(\boldsymbol{\theta})}{p(\boldsymbol{y})}, \tag{3.1}$$

where $p(\boldsymbol{y}|\boldsymbol{\theta})$ is the likelihood function, $\pi(\boldsymbol{\theta})$ is the prior distribution and $p(\boldsymbol{y})$ is the marginal probability. The likelihood function denotes the probability that we observe the data $\boldsymbol{y}$ given the parameter values $\boldsymbol{\theta}$. The prior distribution encapsulates the prior belief about the parameters. When no prior knowledge is known about the parameters, it is common to use uninformative priors. There are several methods for creating uninformative priors, for example, Jeffreys priors

[80]. The marginal probability, which for continuous data is given by

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{\theta})\,\pi(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}, \tag{3.2}$$

is a normalisation factor to ensure that the posterior distribution is a true probability distribution. This factor is often not of interest and so we can simply write

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\theta})\,\pi(\boldsymbol{\theta}). \tag{3.3}$$

3.2 Likelihood-free inference

Application of Bayes' theorem is dependent on having access to the likelihood function; however, this is not always available. For stochastic systems, for example, calculation of the likelihood depends on the solution of path integrals over all realisations of the latent state, which is analytically intractable. Likelihood-free methods are a common workaround for systems where the likelihood funciton is not available. The two classical likelihood-free approaches are density estimation methods, which approximates the likelihood function numerically, for example, the synthetic likelihood method [81], and ABC, which compares observed and simulated data, or statistics of the data, through use of a distance measure. In recent years, mathematical developments have allowed for methods which can overcome issues that are commonly faced in classical methods. For example, machine learning techniques allow us to make use of higher-dimensional data, while active learning methods can improve sample efficiency. For a review of recent developments in likelihood-free methods, see [82]. These newer methods can be computationally expensive and many of the concepts used are beyond the remit of the thesis. Therefore, we choose to investigate the performance of ABC schemes.

3.3 Introduction to ABC

Rubin (1984) [83] discusses the interpretation of Bayesian statements and promotes the applied statistician to use Bayesian statistics. He states that it is good to be able to use approximate Bayesian methods and Monte Carlo simulations to examine many different models, especially those which are too complicated for exact inference, rather than one analytically simplified and examined model. In Section 3.1 of [83], a hypothetical procedure to calculate the posterior distribution of an unknown parameter is given which matches the ideas of ABC. This is thought to be the beginnings of ABC. Rubin's hypothetical procedure is similar to the exact ABC rejection scheme given in Algorithm 3. Notice that rather than using the likelihood function, we instead make use of a generative model $f(\boldsymbol{y}|\boldsymbol{\theta})$ which can be used to generate data for a specified model. The accepted parameters $\boldsymbol{\theta}_i,\ i = 1, \dots, N_a$, are actually draws from the true posterior distribution. This can be seen because

$$\boldsymbol{\theta}_i = \int_Y p(\boldsymbol{y}^*|\boldsymbol{\theta}_i)\, \pi(\boldsymbol{\theta}_i)\, \mathbb{I}_{\boldsymbol{y}}(\boldsymbol{y}^*)\, \mathrm{d}\boldsymbol{y}^* = p(\boldsymbol{y}|\boldsymbol{\theta}_i)\, \pi(\boldsymbol{\theta}_i) \propto \pi(\boldsymbol{\theta}_i|\boldsymbol{y}), \tag{3.4}$$

where Y is the space in which $\boldsymbol{y}$ takes values and

$$\mathbb{I}_{\boldsymbol{y}}(\boldsymbol{y}^*) = \begin{cases} 1, & \text{if } \boldsymbol{y}^* = \boldsymbol{y}, \\ 0, & \text{otherwise}, \end{cases} \tag{3.5}$$

is an indicator function. As stated above, the advantage of this method is that all accepted parameter values will be from the true posterior distribution. However, the difficulty with this procedure is that it could take a long time to find an appropriate number of $\boldsymbol{y}^*$ values. For continuous data or large parameter spaces, it may be impossible to find any values at all. Therefore, the exact rejection algorithm is best used for problems involving discrete data.

Algorithm 3 ABC exact rejection algorithm

```
Input: The observed data $\boldsymbol{y}$, prior distribution for the parameters $\pi(\boldsymbol{\theta})$, a generative model $f(\boldsymbol{y}|\boldsymbol{\theta})$ and the number of parameter acceptances $N_a$.
Output: Samples $\boldsymbol{\theta}_i,\ i = 1, \ldots, N_a$, from the posterior distributions $\pi(\boldsymbol{\theta}|\boldsymbol{y})$.
Set the number of accepted parameter values $i = 0$.
while $i < N_a$ do
    Sample $\boldsymbol{\theta}^*$ from the prior $\pi(\boldsymbol{\theta})$.
    Simulate data using the generative model $\boldsymbol{y}^* \sim f(\boldsymbol{y}|\boldsymbol{\theta}^*)$.
    if $\boldsymbol{y}^* = \boldsymbol{y}$ then
        accept and store the parameters values $\boldsymbol{\theta}_i = \boldsymbol{\theta}^*$ and set $i = i + 1$.
    end if
end while
```

3.4 ABC rejection methods

Tavaré *et al.* (1997) [84] implemented the first ABC algorithm to estimate the coalescence time (time since the most recent common ancestor) from DNA sequences. Their rejection algorithm, similar to acceptance-rejection algorithms, calculated a summary statistic, the number of segregating sites, from simulated data and accepted the estimation of the coalescence time based on some probability, which was calculated using prior knowledge about the population demography. This method was specific to analysing genealogical data and could not be directly implemented in other areas. A more general algorithm, similar to their methods, is the ABC rejection scheme given in Algorithm 4. This time, we accept the parameter values if $||\boldsymbol{y}^* - \boldsymbol{y}|| < \varepsilon$. The norm $||\cdot||$ must be specified by the user. A consequence of this is that the accepted parameter values now come from approximate posterior distributions

$$\pi_\varepsilon(\boldsymbol{\theta}|\boldsymbol{y}) \propto \int_Y p(\boldsymbol{y}^*|\boldsymbol{\theta}_i)\,\pi(\boldsymbol{\theta}_i)\,\mathbb{I}_{S_{\varepsilon,\boldsymbol{y}}}(\boldsymbol{y}^*)\,\mathrm{d}\boldsymbol{y}^*, \tag{3.6}$$

where $S_{\varepsilon,\boldsymbol{y}} = \{\boldsymbol{y} \in Y : ||\boldsymbol{y}^* - \boldsymbol{y}|| < \varepsilon\}$. As ε tends to zero, the approximate posterior distribution approaches the exact posterior distributions. While we no longer require the simulated data to match exactly the observed data, if the data or parameter space is large then we can still run into issues where it will take a long time to accept parameter values.

Algorithm 4 ABC rejection algorithm 1

```
Input: The observed data $\boldsymbol{y}$, prior distribution for the parameters $\pi(\boldsymbol{\theta})$, a generative model $f(y|\boldsymbol{\theta})$, the number of parameter acceptances $N_a$ and the tolerance $\varepsilon$.
Output: Samples $\boldsymbol{\theta}_i,\ i = 1, \dots, N_a$, from the approximate posterior distributions $\pi_\varepsilon(\boldsymbol{\theta}|\boldsymbol{y})$.
Set the number of accepted parameter values $i = 0$.
while $i < N_a$ do
    Sample $\boldsymbol{\theta}^*$ from the prior $\pi(\boldsymbol{\theta})$.
    Simulate data using the generative model $\boldsymbol{y}^* \sim f(\boldsymbol{y}|\boldsymbol{\theta}^*)$.
    if $||\boldsymbol{y}^* - \boldsymbol{y}|| < \varepsilon$ then
        accept and store the parameters values $\boldsymbol{\theta}_i = \boldsymbol{\theta}^*$ and set $i = i + 1$.
    end if
end while
```

Pritchard *et al.* (1999) [49] created a similar algorithm to the one used by Tavaré *et al.* [84]; however, this algorithm accepts parameter values by looking at the distance between a vector of summary statistics, rather than the complete data. Specifically, a parameter value is accepted if $||S(\boldsymbol{y}^*) - S(\boldsymbol{y})|| < \varepsilon$. They used the relative l^1 norm in their algorithms. A general version of their algorithm is given in Algorithm 5. This time, the accepted parameter values are from the approximate posterior distributions

$$\pi_\varepsilon(\boldsymbol{\theta}|S(\boldsymbol{y})) \propto \int_Y p(\boldsymbol{y}^*|\boldsymbol{\theta}_i)\, \pi(\boldsymbol{\theta}_i)\, \mathbb{I}_{S_{\varepsilon,\boldsymbol{y}}}(\boldsymbol{y}^*)\, \mathrm{d}\boldsymbol{y}^*, \tag{3.7}$$

where $S_{\varepsilon,\boldsymbol{y}} = \{\boldsymbol{y} \in Y : ||S(\boldsymbol{y}^*) - S(\boldsymbol{y})|| < \varepsilon\}$. It is important to note that if the summary statistics used are sufficient, meaning that the conditional probability distribution of $\boldsymbol{y}$ given the statistic $S(\boldsymbol{y})$ does not depend on the parameters θ, then $\pi_\varepsilon(\boldsymbol{\theta}|S(\boldsymbol{y})) = \pi_\varepsilon(\boldsymbol{\theta}|\boldsymbol{y})$ [85]. However, it is not always possible to find a sufficient statistic for a given distribution. The Factorization theorem [86] states that a statistic is sufficient if and only if the joint PDF $f(\boldsymbol{y}, \boldsymbol{\theta})$ can be factored as

$$f(\boldsymbol{y}, \boldsymbol{\theta}) = u(\boldsymbol{y})v(\boldsymbol{y}, \boldsymbol{\theta}), \tag{3.8}$$

where u and v are non-negative functions. The exponential family of distributions satisfy the Factorization theorem, and so if a distribution is in the exponential family, then it is possible to find a sufficient statistic.

Beaumont *et al.* (2002) [50] were the first to use the now accepted name "approximate Bayesian computation". They adapted the algorithm presented by Pritchard *et al.* [49] in two ways. Firstly, they used the Euclidean norm instead of the relative l^1 norm, giving (for example, in two-dimensions) circular acceptance regions, rather than rectangular ones. Secondly, when N_a parameter values are accepted, $\boldsymbol{\theta}_i,\ i = 1, \ldots, N_a$, along with their associated summary statistics, $S_i(\boldsymbol{y}^*),\ i = 1, \ldots, N_a$, the values of $\boldsymbol{\theta}_i$ are weighted according to their value of $||S_i(\boldsymbol{y}^*) - S(\boldsymbol{y})||$, then these values are adjusted using local linear regression. They also set the value of ε to be a quantile of the $||S_i(\boldsymbol{y}^*) - S(\boldsymbol{y})||$ values. They coin this method the regression method, which follows the same procedure as Algorithm 5 up to Step 5. At Step 6, the smooth weighting and local linear regression techniques are used as described above. This method was shown to give more accurate results compared with the rejection method.

Algorithm 5 ABC rejection algorithm 2

```
   Input: Observed summary statistic(s) of the data $S(\boldsymbol{y})$, prior distribution
   for the parameters $\pi(\boldsymbol{\theta})$, a generative model $f(\boldsymbol{y}|\boldsymbol{\theta})$, the number of parameter
   acceptances $N_a$ and the tolerance $\varepsilon$.
   Output: Samples $\boldsymbol{\theta}_i,\ i = 1, \ldots, N_a$ from the approximate posterior distribu-
   tions $\pi_\varepsilon(\boldsymbol{\theta}|S(\boldsymbol{y}))$.
1: Set the number of accepted parameter values $i = 0$.
2: while $i < N_a$ do
3:     Sample $\boldsymbol{\theta}^*$ from the prior $\pi(\boldsymbol{\theta})$.
4:     Simulate data using the generative model $\boldsymbol{y}^* \sim f(\boldsymbol{y}|\boldsymbol{\theta}^*)$.
5:     Calculate the summary statistic(s) of the simulated data $S(\boldsymbol{y}^*)$.
6:     if $||S(\boldsymbol{y}^*) - S(\boldsymbol{y})|| < \varepsilon$ then
7:         accept and store the parameters values $\boldsymbol{\theta}_i = \boldsymbol{\theta}^*$ and set $i = i + 1$.
8:     end if
9: end while
```

Rejection methods are the most basic ABC approach. One advantage of these methods is that they often have a small number of initialisation parameters needed to run. The simplicity of these methods make it appealing for non-statisticians as they are easier to understand and work with. However, even with modifications to the simple rejection algorithm, for example, the adjustments made by Beaumont *et al.* [50], rejection methods are still often computationally expensive when dealing with high dimensional parameter spaces or continuous data [51]. These methods are also inefficient when the prior and posterior distribution are vastly different, requiring a much smaller tolerance value to obtain an accurate estimate of the posterior distribution, which will increase the computational cost. The subsequent ABC methods discussed are examples of approaches which aim to improve upon rejection methods.

3.5 ABC Sequential Monte Carlo methods

Sisson *et al.* (2007) [87] introduced an ABC version of Sequential Monte Carlo (SMC). One of the main differences from rejection algorithms is the introduction of a schedule of tolerance values $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{N_\varepsilon}$. Typically, the algorithm is started with a large tolerance value and then uses smaller values as the algorithm progresses. The reason for this is to allow the algorithm to easily find parameter acceptances initially and then progressively update with smaller tolerance values so that the posterior distributions become more accurate. Initially, the parameter samples are taken from the prior as normal, and at all future iterations, the parameter samples come from the previously found posterior distribution. To ensure that we do not sample the exact same values and create a bias in the posterior samples, importance sampling is implemented by weighting the parameter samples and then perturbing them using a kernel.

Importance sampling is a way of estimating an expectation of a distribution by sampling from a different distribution. Suppose that X is a random variable with probability density $p(x)$, and we are looking to calculate the expectation

$$\mathbb{E}(f(X)) = \int f(x)\, p(x)\, \mathrm{d}x. \tag{3.9}$$

Then given a probability density q such that $q(x) > 0$ whenever $f(x)p(x) \neq 0$, we can estimate (3.9) by

$$\mathbb{E}(f(X)) = \int \frac{f(x)\, p(x)}{q(x)}\, q(x)\, \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i) w(x_i), \tag{3.10}$$

where $w(x_i) = p(x_i)/q(x_i)$ are called the importance weights.

A typical choice for the kernel is the standard Gaussian kernel. However, the algorithm was created to be used generally which resulted in a bias. Sisson *et al.* corrected their original algorithm, as well as corrections and modifications made by, for example, Beaumont *et al.* (2009) [88], Toni *et al.* (2009) [89] and Beaumont (2010) [90]. A general version of Beaumont's algorithm (2010) is given in Algorithm 6. Beaumont uses a Gaussian kernel with a mean taken to be the current sampled parameter and variance corresponding to twice the empirical variance of the previous accepted parameters. However, there are still some issues in generalising the algorithm. One issue is that the choice of the tolerance values are chosen by the user and will likely need to be carefully specified for each new problem. Another issue is that if the summary statistics are out of scale with one another then the largest summary statistic can dominate over the others during the acceptance step. Note that all the previous algorithms discussed here share the first issue, while the algorithms presented by Pritchard *et al.* [49] and Beaumont *et al.* [50] share the second issue.

Algorithm 6 ABC SMC algorithm

```
Input: Observed summary statistic(s) of the data $S(\boldsymbol{y})$, prior distribution for the parameters $\pi(\boldsymbol{\theta})$, a generative model $f(\boldsymbol{y}|\boldsymbol{\theta})$, the number of parameter acceptances $N_a$ and a schedule of tolerances $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{N_\varepsilon}$.
Output: Samples $\boldsymbol{\theta}_i,\ i = 1, \dots, N_a$ from the approximate posterior distributions $\pi_\varepsilon(\boldsymbol{\theta}|S(\boldsymbol{y}))$.
for $t = 1, \dots, N_\varepsilon$ do
    Set the number of accepted parameter values $i = 0$.
    while $i < N_a$ do
        Sample $\boldsymbol{\theta}^*$ from
            $q_t(\boldsymbol{\theta}) = \begin{cases} \pi(\boldsymbol{\theta}), & \text{if } t = 1, \\ \sum_{i=1}^{N_a} w_i^{t-1} K_t(\boldsymbol{\theta}|\boldsymbol{\theta}_i^{t-1}) / \sum_{i=1}^{N_a} w_i^{t-1}, & \text{otherwise.} \end{cases}$
        Simulate data using the generative model $\boldsymbol{y}^* \sim f(\boldsymbol{y}|\boldsymbol{\theta}^*)$.
        Calculate the summary statistic(s) of the simulated data $S(\boldsymbol{y}^*)$.
        if $||(S(\boldsymbol{y}^*) - S(\boldsymbol{y}))|| < \varepsilon_t$ then
            accept and store the parameters values $\boldsymbol{\theta}_i^t = \boldsymbol{\theta}^*$ and set $i = i + 1$.
        end if
    end while
    Calculate the importance weights $w_i^t = \pi(\boldsymbol{\theta})/q_t(\boldsymbol{\theta}_i^t)$.
    Set $t = t + 1$.
end for
```

Prangle (2017) [59] collates different methods to help counteract these issues. His paper concentrates on weighting the distance function used during the acceptance step. Throughout the paper, a weighted Euclidean distance is used. The role of the weighting is to try and normalise the summary statistics to ensure that they are in scale with each other. While a popular choice for the weighting is an empirical estimate of standard deviation of the summary statistic, Csilléry *et al.* (2012) [91] recommend using the median absolute deviation (MAD). The advantage of the MAD is that it is robust to large outliers and better captures the variation of each summary statistic. To apply this weighting in the algorithm, during the first iteration, we give equal weights to the distance function, then, for all later iterations, we weight by the MAD of accepted summary statistics from the previous iteration. To counteract the issue of choosing the tolerance values, Prangle uses the method of Drovandi and Pettitt (2011) [92] which adaptively chooses a threshold value to be the ϕ quantile of the previous accepted distance values. The initial tolerance value is taken to be very large, essentially accepting all parameter values initially. The advantage of this approach is that the tolerance values are chosen in a range which have previously been observed, compared with choosing the values in an ad hoc manner prior to running the algorithm. The tolerance values are therefore chosen automatically which removes any user input when changing experiments. His algorithm is presented in Algorithm 7.

Algorithm 7 Prangle algorithm

```
    Input: Observed summary statistic(s) of the data $S(\boldsymbol{y})$, prior distribution for the parameters $\pi(\boldsymbol{\theta})$, a generative model $f(y|\boldsymbol{\theta})$, the number of parameter acceptances $N_a$, a stopping tolerance $\varepsilon_s$ and the quantile value $\phi$.
    Output: Samples $\boldsymbol{\theta}_i,\ i = 1, \ldots, N_a$ from the approximate posterior distributions $\pi_\varepsilon(\boldsymbol{\theta}|S(\boldsymbol{y}))$.
 1: Set $t = 1$, $\varepsilon_1 = \infty$ and $\omega_j^1 = 1$.
 2: while $\varepsilon_t > \varepsilon_s$ do
 3:     Set the number of accepted parameter values $i = 0$.
 4:     while $i < N_a$ do
 5:         Sample $\boldsymbol{\theta}^*$ from
            $q_t(\boldsymbol{\theta}) = \begin{cases} \pi(\boldsymbol{\theta}), & \text{if } t = 1 \text{ or } 2, \\ \sum_{i=1}^{N_a} w_i^{t-1} K_t(\boldsymbol{\theta}|\boldsymbol{\theta}_i^{t-1}) / \sum_{i=1}^{N_a} w_i^{t-1}, & \text{otherwise.} \end{cases}$
 6:         Simulate data using the generative model $\boldsymbol{y}^* \sim f(y|\boldsymbol{\theta}^*)$.
 7:         Calculate the summary statistic(s) of the simulated data $S(\boldsymbol{y}^*)$.
 8:         if $d^t = ||(S(\boldsymbol{y}^*) - S(\boldsymbol{y}))/\omega_j^t|| < \varepsilon_t$ then
 9:             accept and store the parameters values $\boldsymbol{\theta}_i^t = \boldsymbol{\theta}^*$ and set $i = i + 1$.
10:         end if
11:     end while
12:     Calculate the MAD, $M_1^t, M_2^t, \ldots$, for each summary statistic in Step 8.
13:     Calculate the distance function weights $\omega_j^{t+1} = M_j^t$.
14:     Calculate the importance weights $w_i^t = \pi(\boldsymbol{\theta})/q_t(\boldsymbol{\theta}_i^t)$.
15:     Let $\varepsilon_{t+1}$ be the $\phi$ quantile of the accepted $d^t$ values.
16:     Set $t = t + 1$.
17: end while
```

The main advantage of SMC methods over rejection methods is that they are usually more computationally efficient. The sequential nature of the procedure means that, at each iteration, parameter values are sampled from distributions which get closer to the true posterior. This reduces the number of parameter values which are sampled from low probability regions, resulting in faster computational efficiency. A disadvantage of SMC methods is that, for small tolerance values, the probability of accepting parameter values can become small, even if the proposal distribution is close to the true posterior distribution. This results in the algorithm being ran for longer than needed with little improvement in the inference [51]. A further disadvantage is, due to their reliance on more advanced statistical properties, SMC methods can be difficult for non-statisticians to understand, and could therefore be less appealing.

3.6 Semi-automatic ABC

Fearnhead and Prangle (2012) [60] proposed a method to select the summary statistics in a semi-automatic way. For the true parameter values $\boldsymbol{\theta}$, they define accuracy of the inferred estimates $\boldsymbol{\theta}^*$ in terms of loss functions, $L(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$, and in particular, consider the quadratic loss, defined in terms of a positive definite matrix A, as

$$L(\boldsymbol{\theta}, \boldsymbol{\theta}^*; A) = (\boldsymbol{\theta} - \boldsymbol{\theta}^*)^T A (\boldsymbol{\theta} - \boldsymbol{\theta}^*). \tag{3.11}$$

They found that for any choice of A that is full rank, the best summary statistic to minimise the quadratic loss (3.11) was an estimate of the posterior mean. In practice, this is not something that one has access to, and in fact is one of the quantities ABC hopes to find. In order to implement their method, an initial pilot run of ABC is used, with an arbitrarily chosen summary statistic, to find a region closer to the posterior than the prior. This step is only done if the prior is uninformative. Then, a set of M parameter values, $\boldsymbol{\theta}^{(j)}$, $j = 1, \ldots, M$, and data, $\boldsymbol{y}^{(j)},\ j = 1, \ldots, M$, is simulated to estimate the summary statistic, in this case, an estimate of the posterior mean. They found that a simple and effective method to estimate the posterior means was by using linear regression, with appropriate functions of the data as predictors. This function $f(\cdot)$ is a vector-valued function whose entries are transformations of the data. Then, to estimate the ith summary statistic, the parameter values $\theta_i^{(j)},\ j = 1, \ldots, M$, are used as the response variables, while the transformations $f(\boldsymbol{y}^{(j)}),\ j = 1, \ldots, M$, are used as the predictors, so that

$$\theta_i = \mathbb{E}(\theta_i|\boldsymbol{y}) + \varepsilon_i = \beta_0^{(i)} + (\boldsymbol{\beta}^{(i)} \cdot f(\boldsymbol{y})) + \varepsilon_i, \tag{3.12}$$

where ε_i is a zero-mean noise term. After using linear regression on these M data sets, we are left with $\hat{\beta}_0^{(i)} + (\hat{\boldsymbol{\beta}}^{(i)} \cdot f(\boldsymbol{y}))$, which is an estimate of the posterior mean of θ_i. These are then used as the new summary statistics within a second ABC run. To compare the summary statistics within the ABC framework, we must calculate the new observed summary statistics from the observed data $\theta_i = \hat{\beta}_0^{(i)} + (\hat{\boldsymbol{\beta}}^{(i)} \cdot f(\boldsymbol{y}))$, as well as the new simulated summary statistics from the simulated data $\theta_i^* = \hat{\beta}_0^{(i)} + (\hat{\boldsymbol{\beta}}^{(i)} \cdot f(\boldsymbol{y}^*))$.

3.7 Comparison with particle Markov chain Monte Carlo

ABC methods, such as the ABC SMC method given in Algorithm 7, target the approximate posterior distribution $\pi_\varepsilon(\boldsymbol{\theta}|S(\boldsymbol{y}))$. When sufficient summary statistics are used and we take $\varepsilon \to 0$, the approximate posterior distribution tends to the exact posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y})$. However, in practice, we typically do not have access to sufficient summary statistics and we are required to take $\varepsilon > 0$ for computational efficiency. This is sometimes referred to as an "approximate approximate" method.

A popular approach for sampling from the posterior distribution are Markov chain Monte Carlo (MCMC) methods. These involve creating Markov chains whose stationary distribution is the distribution of interest; in this case, the posterior distribution of the parameters. One of these algorithms is called the Metropolis-Hastings algorithm. For this, we first choose an initial value $\boldsymbol{\theta}^{(0)}$. We then propose a new value by sampling from a proposal distribution K. This new value is accepted according to some probability fraction. A version of the Metropolis-Hastings algorithm is given in Algorithm 8. As can be seen in Step 5, the acceptance step relies on knowing the likelihood function, which is often intractable for more complex problems.

An ABC version of MCMC was first proposed by Marjoram *et al.* [93] to deal with problems where the likelihood function is unavailable. For this, after we have sampled from the proposal distribution at Step 4 of Algorithm 8, data $\boldsymbol{y}^*$ is then generated using our generative function and we only proceed to Step 5 if $||(S(\boldsymbol{y}^*) - S(\boldsymbol{y}))|| < \varepsilon$, for some tolerance value ε. The probability fraction in Step 5 is then adapted by removing the likelihood function. This ABC MCMC algorithm will sample values from the approximate posterior distribution $\pi_\varepsilon(\boldsymbol{\theta}|S(\boldsymbol{y}))$, similar to ABC SMC.

Another MCMC algorithm which deals with problems where the likelihood function is intractable is Particle Markov chain Monte Carlo (pMCMC) [94]. This algorithm replaces the likelihood function in Step 5 of Algorithm 8 by a Monte Carlo estimate. Andrieu *et al.* show that replacing the likelihood function by a Monte Carlo estimate is still mathematically sound by proving that the stationary distribution found by pMCMC converges to the true posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y})$

asymptotically. This is an advantage over ABC MCMC which samples from an approximate posterior distribution. This is sometimes referred to as an "exact approximate" method.

Owen *et al.* [95] compared using pMCMC with ABC SMC for parameter inference of stochastic kinetic models. The pMCMC algorithm used is similar to Algorithm 8 where the likelihood function in Step 5 is replaced by a Monte Carlo estimate. This Monte Carlo estimation of the likelihood function is calculated using its own SMC algorithm. This approximation must be done at every iteration of the pMCMC algorithm. The ABC SMC algorithm used is similar to Algorithm 7. For fair comparison, however, a Euclidean distance over the full set of data points was taken for the norm in Step 8. This ensures that the SMC algorithm targets the true posterior distribution.

The overall results show that pMCMC was a better choice than ABC SMC, provided that it could be well initialised. The tuning parameters for this initialisation step include the choice of the initial parameter values used for the MCMC chains, the number of particles used during the SMC algorithm to estimate the likelihood function, and the covariance matrix used in the proposal distribution. The results which showed that pMCMC performed better than ABC SMC were found when these initialisation parameters were chosen cleverly with knowledge of the posterior distribution. When these values are chosen without any knowledge of the posterior distribution, which is more realistic to real life, the computational cost for pMCMC becomes very inefficient and it 'made pMCMC look completely uncompetitive.'

Algorithm 8 Metropolis-Hastings MCMC algorithm

```
Input: Prior distribution for the parameters $\pi(\boldsymbol{\theta})$, a proposal distribution $K$ and the number of parameter acceptances $N_a$.
Output: Samples $\boldsymbol{\theta}_i,\ i = 1, \ldots, N_a$, from the posterior distributions $\pi(\boldsymbol{\theta}|\boldsymbol{y})$.
Set the number of accepted parameter values $i = 0$.
Choose an initial value $\boldsymbol{\theta}^{(0)}$.
for $j = 1, \ldots,$ do
    Sample $\boldsymbol{\theta}^* \sim K(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(j-1)})$
    Calculate
        $r = \min\left(1, \frac{p(\boldsymbol{y}|\boldsymbol{\theta}^*)\pi(\boldsymbol{\theta}^*)K(\boldsymbol{\theta}^{(j-1)}|\boldsymbol{\theta}^*)}{p(\boldsymbol{y}|\boldsymbol{\theta}^{(j-1)})\pi(\boldsymbol{\theta}^{(j-1)})K(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(j-1)})}\right).$
    Sample $u \sim U(0, 1)$.
    Set
        $\boldsymbol{\theta}^{(t)} = \begin{cases} \boldsymbol{\theta}^*, & \text{if } u \leq r, \\ \boldsymbol{\theta}^{(t-1)}, & \text{if } u > r. \end{cases}$
end for
Stop the algorithm after a large number of iterations and select the last $N_a$ values as our samples.
```

3.8 Challenges in ABC

Although a lot of work has been done to advance the theory and application of ABC since its introduction by Tavaré *et al.*, a lot of challenges still remain.

One prominent question relates to the use of the summary statistics. As we have discussed, sufficient summary statistics would be the optimal choice within ABC, but this is not usually available in practice. Therefore, methods to construct informative summary statistics has become an important topic of study. Although work has been done to address this issue, for example, [60, 96], a lot of work still remains to be done.

A related topic looks at effective ways to weight and/or combine the summary statistics that are already availlable. We have already discussed the work of Prangle [59] who weighted the summary statistics by their MAD in order to address the issue of the summary statistics being out of scale with one another. The issue of how to efficiently combine summary statistics has been adressed by, for example, Harrison and Baker [97].

Another challenge is the need to adapt ABC approaches to accommodate for the rise of big data. As the size of the data continues to increase, there is a need to come up with ABC schemes which can effectively handle large dimensional data or summary statistics. Some papers which look to address this issue are [98, 99].

Chapter 4

Inferring the drift and diffusion coefficients using ABC

4.1 ABC experiments for drift-diffusion

Rather than using regression as in Chapter 2, we look to infer the drift and diffusion coefficients using ABC. This chapter aims to apply the ABC approaches in a general setting, rather than making heavy use of specific analytical results like in Chapter 2, to better reflect realistic situations where more complicated models are used. We will start by running simulations using the basic ABC rejection and SMC approaches, and will compare these with the Prangle [59] and Fearnhead & Prangle [60] approaches outlined in Chapter 3.

The model we will use is a one-dimensional version of the drift-diffusion SDE from Chapter 2. We choose this model as a way to look forward to Chapter 5, where we will use a one-dimensional drift-diffusion SDE to model self-generated gradient chemotaxis. In Ferguson *et al.* [40], differential equation models were fit to self-generated gradient cell movement data, which follow a similar set-up as in Figure 5.1. In the supplementary material of the paper, they used two tests to decide whether using a one-dimensional model would significantly misrepresent the experimental data. The first was a Kolmogorov-Smirnov test which confirmed that the y-coordinate data were not significantly different from uniform distributions, meaning that there were no significant biological features happening in this dimension. The second test confirmed that the x- and y-coordinate data are independent from one another. Both of these tests showed that a one-dimensional model would be appropriate to model the self-generated gradient data.

In addition, we choose to remove static error from our model. Exact posterior distributions can be calculated for our drift-diffusion SDE model, which cannot be found when static error is present. Therefore, in order to calculate exact posterior distributions for comparison with the simulated ABC posterior distributions, we choose to remove static error from our model.

4.2 Drift-diffusion model

We assume that particles move according to the one-dimensional drift-diffusion SDE

$$\mathrm{d}X_t = \alpha\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}W_t, \tag{4.1}$$

where α is the drift velocity, D is the diffusion coefficient, and $\mathrm{d}W_t$ is a Wiener process. As before, we assume that the data comprises of N_S trajectories, measured at the $N+1$ time points $t_n = (n-1)T/N = (n-1)\Delta t$, $n = 1, \ldots, N+1$, covering the measurement time range $[0, T]$.

The following quantities will be used for all experiments in this chapter, unless stated otherwise. The observed data will be generated by solving numerically (4.1) by the Euler-Maruyama method with $N_S = 100$ and $N = 100$. The true values of the model parameters will be $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$ and $\alpha = 1\,\mu\mathrm{m/s}$ and the number of parameter acceptances $N_a = 1000$. For the observed summary statistic, we will use the ensemble time-average MSD given by (2.22), where the data is created using the true values of D and α. The prior distribution for both parameters is taken to be the uniform distribution from 0 to 10, denoted $U(0, 10)$. Finally, we shall use the Euclidean norm during the acceptance step of each ABC algorithm.

We will be comparing the different ABC algorithms given in Chapter 3. Since we showed in Chapter 2 that the accuracy of regression depends crucially on the value of T, we will also see whether the performance of the ABC schemes are time dependent for our SDE model by testing the algorithms for the values $T = 0.05\,\mathrm{s}, 5\,\mathrm{s}$ and $500\,\mathrm{s}$.

4.3 Exact posterior distributions

To start, it is of interest to calculate the exact joint posterior distribution in D and α using the three different values of T to allow comparison with the posterior distributions that will arise from the ABC schemes. Due to the simplistic form of our model, we are able to generate exact joint posterior distributions through use of the likelihood function.

Equation (2.2) from Chapter 2 gives the PDF for the displacement of particles at time t assuming that the particles move according the model given by (2.1) with no static error. This is in fact the likelihood of observing the trajectory data given the parameter values. Notice that our current model (4.1) is simply a one-dimensional form of this model, and so the likelihood will be given by

$$p(x, t) = \frac{1}{\sqrt{4\pi D t}} \exp\left(\frac{-|x - \alpha t|^2}{4Dt}\right). \tag{4.2}$$

If we simulate data directly from our model (4.1), then we can calculate the likelihood function. For this, we generate $N_S = 100$ trajectories at $N = 100$ steps of size $\mathrm{d}t = T/N$, denoted by $x_n^{(j)}$, $n = 1, \ldots, N$, $j = 1, \ldots, N_S$. Here, we assume that each step from $x_n^{(j)}$ to $x_{n+1}^{(j)}$ is equivalent to taking a time step of size Δt starting from $x_n^{(j)} = 0$. Given this data, the likelihood function for trajectory j is given by

$$\begin{aligned} L &= \prod_{n=2}^{N} p(x_n, t|x_{n-1}) \\ &= \prod_{n=2}^{N} \frac{1}{\sqrt{4\pi D\mathrm{d}t}} \exp\left(\frac{-|\Delta x_n^{(j)} - \alpha \mathrm{d}t|^2}{4D\mathrm{d}t} \right) \\ &= (4\pi D\mathrm{d}t)^{-\frac{N}{2}} \exp\left(\frac{-\sum_{n=2}^{N} |\Delta x_n^{(j)} - \alpha \mathrm{d}t|^2}{4D\mathrm{d}t} \right), \end{aligned} \tag{4.3}$$

where $\Delta x_n^{(j)} = x_n^{(j)} - x_{n-1}^{(j)}$. From this, the log-likelihood is calculated as

$$\begin{aligned} l = \ln(L) &= \ln\left((4\pi D\mathrm{d}t)^{-\frac{N}{2}} \exp\left(\frac{-\sum_{n=2}^{N} |\Delta x_n^{(j)} - \alpha \mathrm{d}t|^2}{4D\mathrm{d}t} \right) \right) \\ &= -\frac{N}{2} \ln(4\pi D\mathrm{d}t) - \frac{\sum_{n=2}^{N} |\Delta x_n^{(j)} - \alpha \mathrm{d}t|^2}{4D\mathrm{d}t}. \end{aligned} \tag{4.4}$$

To plot the exact joint posterior distribution in D and α, we will split the prior distributions in D and α into 1000 uniformly distributed points each, creating a mesh of 1000×1000 points over the parameter domain. We then calculate the log-likelihood given by (4.4) at each of these points. To obtain a more accurate log-likelihood, we do this for all N_S trajectories and average over these. We then exponentiate the averaged log-likelihood to obtain the likelihood in D and α. The joint posterior distribution in D and α can then be found by using Bayes' theorem

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})\, \pi(\boldsymbol{\theta})}{p(\boldsymbol{y})}. \tag{4.5}$$

In fact, since we assume uniform prior distributions for both parameters, the shape of the joint posterior distribution will match that of the likelihood. The marginal probability $\pi(\boldsymbol{\theta})$ will only adapt the value of the posterior distribution and will not change the shape. Therefore, plots of the exact joint posterior distribution will correspond with plots of the likelihood.

Figure 4.1 compares contour plots of the likelihood and the log-likelihood for the values $T = 0.05\,\mathrm{s}, 5\,\mathrm{s}$ and $500\,\mathrm{s}$. We find that the log-likelihood and likelihood have very similar shapes for a small and middle value of T, but for a large value of T, the likelihood is more compact than the log-likelihood. As expected, the shape of both distributions change as we change T. Similar to Chapter 2, for a small value of T, α is difficult to infer accurately – illustrated here by the wide distributions in α. As we increase T, the distributions become more isometric. When we reach the largest value of T, the distributions in D begins to widen out.

4.4 ABC rejection method

4.4.1 Rejection algorithm

We will begin by using the rejection algorithm given in Algorithm 5 in Chapter 3 to infer the drift velocity α and the diffusion coefficient D for our three values of T. The only input parameter that still needs to be defined to run the algorithm is the tolerance ε. We will need to adapt the value of the tolerance for different values of T. As we increase the value of T, we will also need to increase the value of the tolerance. One of the reasons why larger values of T require a larger value of ε is due to the MSD. In one dimension, the analytical form of the MSD is

$$\rho(t) = \alpha^2 t^2 + 2Dt. \tag{4.6}$$

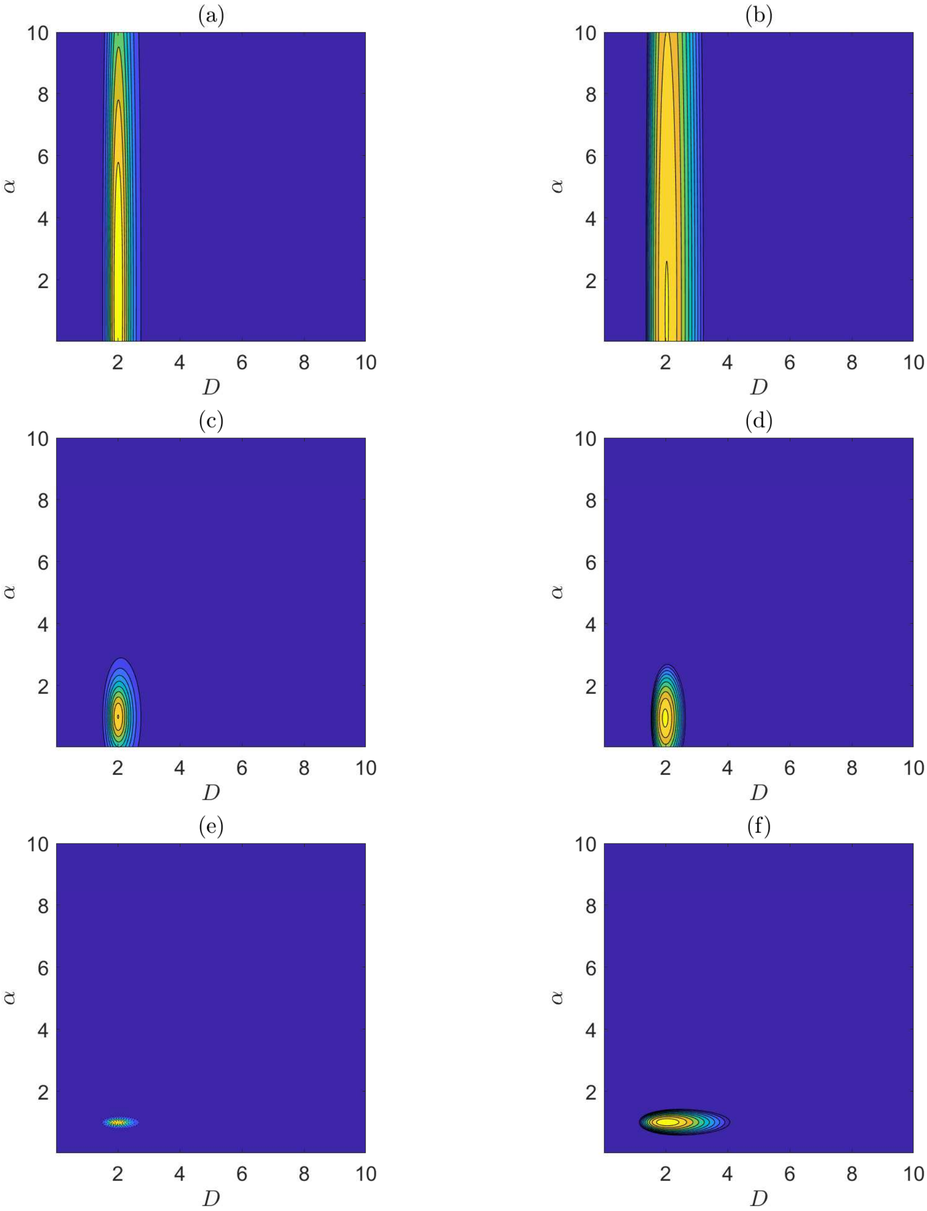

Figure 4.1: Contour plots of the likelihood [(a), (c) and (e)] and the log-likelihood [(b), (d) and (f)] in D and α for $T = 0.05\,\mathrm{s}$ [(a) and (b)], $T = 5\,\mathrm{s}$ [(c) and (d)] and $T = 500\,\mathrm{s}$ [(e) and (f)].

Therefore, as we increase T, the value of the MSD will increase quadratically and so the value of the summary statistics will increase similarly. Also, from (2.42) we see that the variance of the MSD increases cubically with time, and so the estimated MSD will be less accurate as time increases. Due to both of these properties, the value of the tolerance ε will need to be increased for a larger T.

To illustrate how the posterior distributions change as we decrease ε, we will test the algorithm for a schedule of decreasing tolerance values. For $T = 0.05$, we will take $\varepsilon = 2, 1, 0.5$ and 0.25; for $T = 5$, we will take $\varepsilon = 200, 100, 50$ and 25; and for $T = 500$, we will take $\varepsilon = 200000, 100000, 50000$ and 25000. Figures 4.2-4.4 show samples from the joint posterior distributions for $T = 0.05\,\mathrm{s}, 5\,\mathrm{s}$ and $500\,\mathrm{s}$, respectively. As we decrease ε, the joint posterior distributions get slowly closer to the exact posterior distributions for all values of T. We see a better improvement for a small and middle value of T, while for a large value of T, the joint posterior distribution in D covers the entire prior distribution.

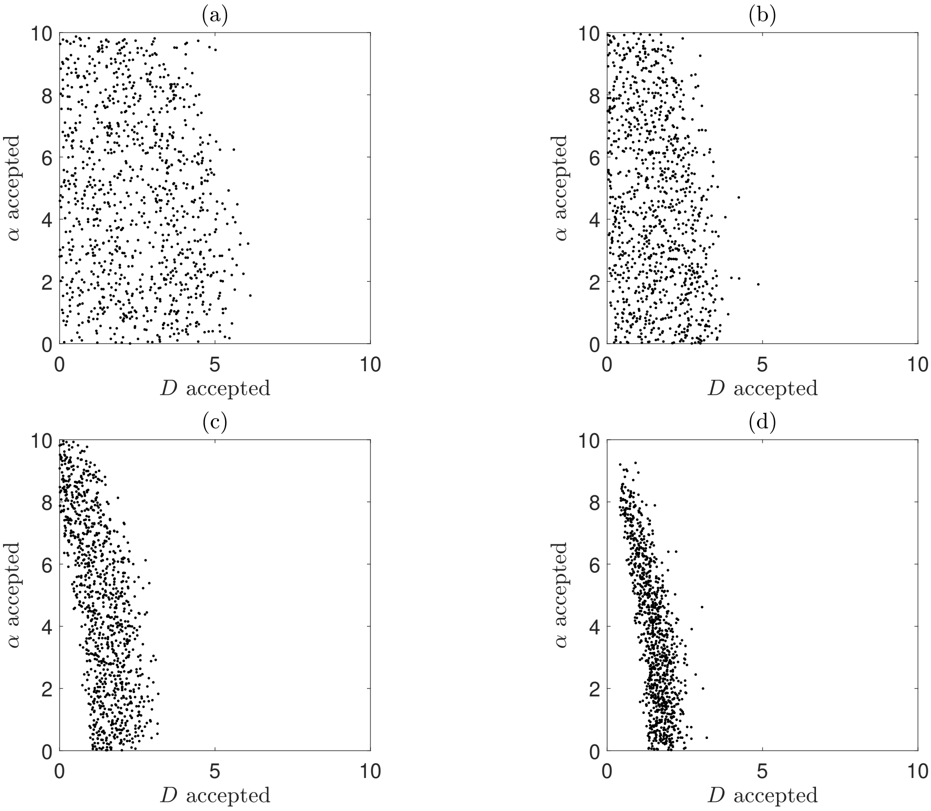

Figure 4.2: Samples from the joint posterior distributions for D and α using the rejection algorithm for $T = 0.05\,\mathrm{s}$ and $\varepsilon = 2$ (a), $\varepsilon = 1$ (b), $\varepsilon = 0.5$ (c) and $\varepsilon = 0.25$ (d). These experiments were for $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m}/\mathrm{s}$, $N_S = 100$ and $N = 100$.

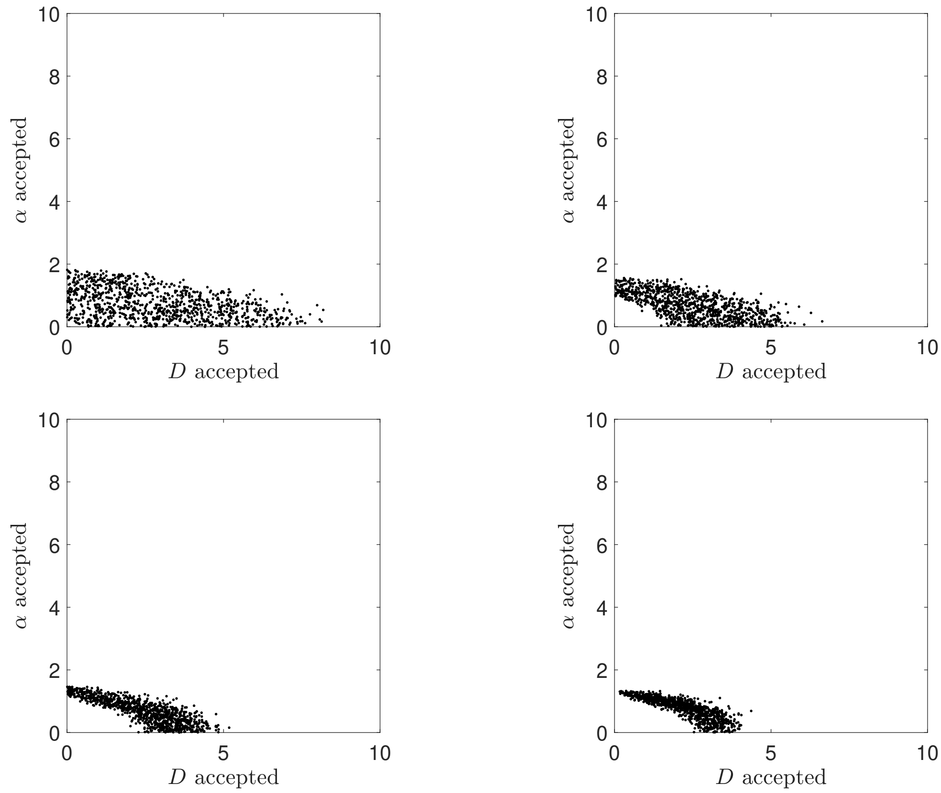

Figure 4.3: Samples from the joint posterior distributions for D and α using the rejection algorithm for $T = 5\,\text{s}$ and $\varepsilon = 200$ (a), $\varepsilon = 100$ (b), $\varepsilon = 50$ (c) and $\varepsilon = 25$ (d). These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $N_S = 100$ and $N = 100$.

To see how good the joint posterior distributions are from the rejection algorithm, we will create contour plots from the final ABC joint posterior distributions samples. To do this, we calculate the mean vector and covariance matrix from the ABC samples and create the contour plots assuming that the samples come from a multivariate normal distribution. We will then compare these with the exact joint posterior distributions superimposed with the final ABC joint posterior distributions samples. The results are shown in Figure 4.5. We find that the rejection joint posterior distributions are poor approximations of the exact joint posterior distributions. It is suspected that these poor joint posterior distributions are due to using the MSD as the summary statistics.

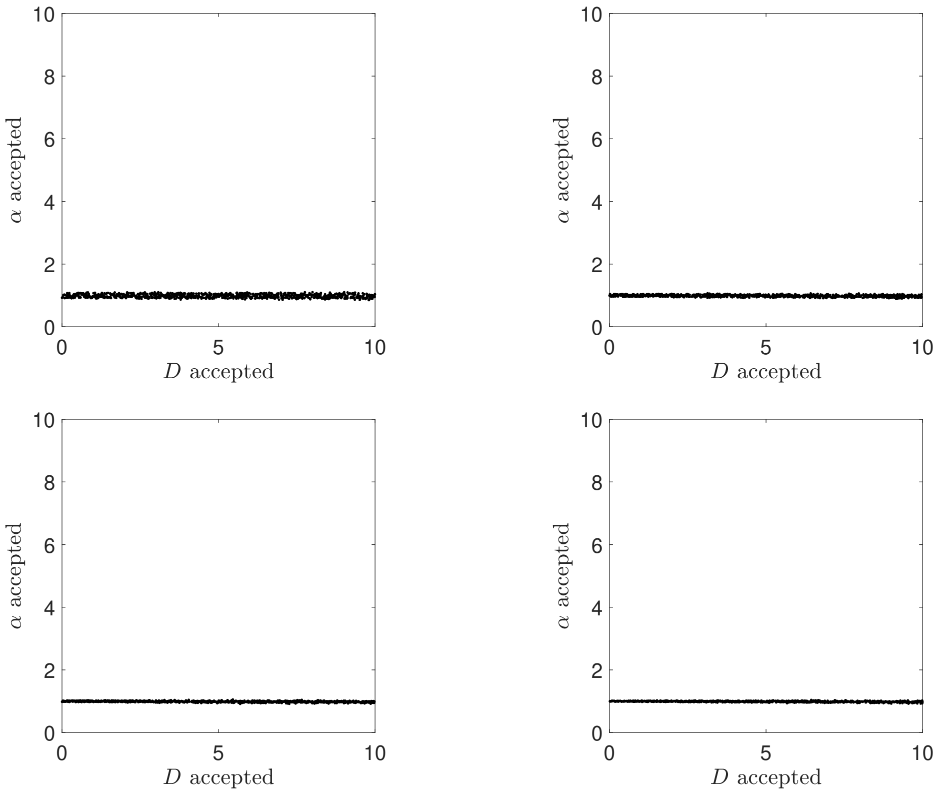

Figure 4.4: Samples from the joint posterior distributions for D and α using the rejection algorithm for $T = 500\,\text{s}$ and $\varepsilon = 200000$ (a), $\varepsilon = 100000$ (b), $\varepsilon = 50000$ (c) and $\varepsilon = 25000$ (d). These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $N_S = 100$ and $N = 100$.

Depending on the value of T, one of the parameters can become very difficult to infer accurately. For example, for a small value of T, the posterior distributions remain wide in α as the algorithm progresses, and so it becomes challenging to accurately infer α. In this case, we will say that α is weakly identifiable. However, the posterior distributions narrow at a much faster rate in D, making it easier to infer. For this case, we will say that D is strongly identifiable.

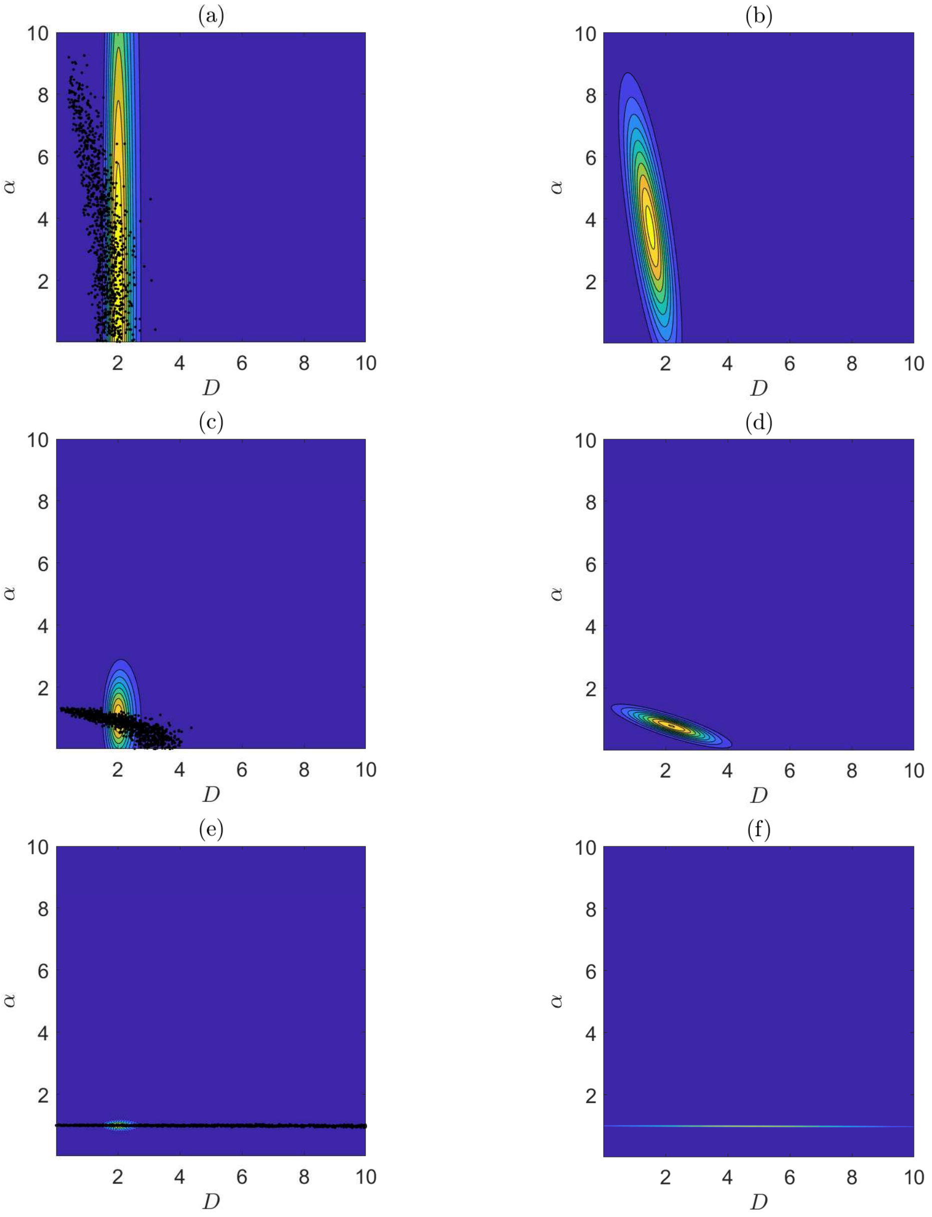

Figure 4.5: Contour plots of the exact joint posterior distributions with the ABC samples [(a), (c) and (e)] and the contour plots created from the ABC samples [(b), (d) and (f)] using the rejection algorithm for $T = 0.05\,\mathrm{s}$ (a), $T = 5\,\mathrm{s}$ (b) and $T = 500\,\mathrm{s}$ (c). The experiments for the rejection algorithm used $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

4.4.2 Accuracy measures

An alternative approach for the rejection algorithm, instead of accepting parameter values based on some tolerance value ε, would be to initially sample a large number of parameter values from the prior and then accept a fraction of these which have the smallest distance measure; that is, the fraction that has the smallest value as in Step 6 of Algorithm 5. This approach would remove the need to find appropriate tolerance values for the different values of T.

In order to compare the original version of the rejection algorithm, which we will call the tolerance rejection approach, with the adaptation described above, which we will call the fraction rejection approach, we introduce two accuracy measures.

Since there can be parameter identifiability issues when using the MSD as a summary statistics, we will first introduce an accuracy measure that is predominately determined by the change in the strongly identifiable parameter. We will call this the harmonic accuracy. To define this accuracy, we will calculate the posterior mean for D and α for the ith tolerance value, denoted by $\overline{D}_i$ and $\overline{\alpha}_i$, respectively, as well as the variance of the accepted values of D and α, denoted by $(\sigma_D^2)_i$ and $(\sigma_\alpha^2)_i$, respectively. The mean squared error (MSE) for D and α for the ith tolerance value is then given by

$$(MSE_D)_i = (D - \overline{D}_i)^2 + (\sigma_D^2)_i, \quad i = 1, \ldots, N_\varepsilon, \tag{4.7}$$

$$(MSE_\alpha)_i = (\alpha - \overline{\alpha}_i)^2 + (\sigma_\alpha^2)_i, \quad i = 1, \ldots, N_\varepsilon, \tag{4.8}$$

where N_ε is the number of tolerance values. The harmonic accuracy for the ith tolerance value is then be given by

$$(\text{Accuracy})_i = \frac{1}{(MSE_D)_i} + \frac{1}{(MSE_\alpha)_i}, \quad i = 1, \ldots, N_\varepsilon. \tag{4.9}$$

We will define the second accuracy measure so that a higher level of accuracy corresponds with estimating both parameters well, rather than only the strongly identifiable parameter. To do this, we need to think about what it means for both parameters to be estimated well. In two dimensions, we could think of this as the shape of the level sets of joint posterior distribution being circular around the true parameter values with a small radius. To implement this idea, we need a measure of circularity of a general shape. One simple measure would be the compactness measure [100], defined as $C = 4\pi A/p^2$, where A is the area enclosed and p is the perimeter of the shape. If the shape is a circle then $C = 1$ and $C < 1$ otherwise. By treating our posteriors as roughly rectangular, then we have that $A = MSE_D \times MSE_\alpha$ and $p = 2(MSE_D + MSE_\alpha)$. This would give an approximate value of $C \approx (\pi MSE_D \times MSE_\alpha)/(MSE_D + MSE_\alpha)^2$. A possible error measure could then be $\sqrt{A}/C \approx (MSE_D + MSE_\alpha)^2/\sqrt{MSE_D \times MSE_\alpha}$. Notice that $\sqrt{A}$ goes to zero as A goes to zero, and since C is maximised when the area is close to a circle then dividing by C puts more emphasis on estimating both parameters well, rather than just one of them. We then define our second accuracy measure, which we call the circular accuracy, as the inverse of the error, giving

$$(\text{Accuracy})_i = \frac{\sqrt{(MSE_D)_i \times (MSE_\alpha)_i}}{((MSE_D)_i + (MSE_\alpha)_i)^2}, \quad i = 1, \dots, N_\varepsilon. \tag{4.10}$$

This measure of accuracy could of course be extended to higher dimensional parameter spaces by using higher-dimensional analogues for the area and perimeter.

4.4.3 Fraction rejection approach

For the fraction rejection approach, we will always accept $N_a = 1000$ parameter values to match the tolerance rejection algorithm. To generate a group of posterior distributions like in Figure 4.2–4.4, we will iteratively sample a larger amount of parameter values from the prior. Since we always accept the best $N_a = 1000$ parameter values, sampling more initial values from the prior will correspond with accepting a smaller fraction. We will initially sample 10^4 parameter values, corresponding with accepting 10% of the best values, and then iteratively increase this value until the final time roughly matches that for the tolerance approach. Note that this will likely result in more iterations of the ABC algorithm for the fraction approach than for the tolerance approach.

We will also need to define a simulation cost. Since both algorithms will be tested on the same computer, we choose to use a measure of the real-life time (in seconds) it takes for the algorithms to run. For this, we will use the in-built MATLAB tic toc function.

The results for the harmonic and circular accuracy against the time to run the algorithms for the tolerance and fraction rejection approaches are given in Figure 4.6. The results show a number of points. Firstly, looking at the plots of the harmonic accuracy, notice that although the highest accuracy is obtained when $T = 500\,\mathrm{s}$, this does not mean that both parameters are best inferred for this value of T. It simply means that the strongly identifiable parameter for the large value of T – α in this case – is inferred more accurately than the strongly identifiable parameter is inferred for the other values of T.

Secondly, looking at the plots of the circular accuracy, we see that the highest accuracy is achieved when $T = 5\,\mathrm{s}$, which means that both parameters are inferred best for this value of T. Looking at Figures 4.2–4.4, the joint posterior distribution when $T = 5\,\mathrm{s}$ is the most isometric and compact, and therefore does infer both parameters best. The circular accuracy when $T = 500\,\mathrm{s}$ decreases as we go through the iterations. This reflects the fact that the joint posterior distributions get thinner in α but remain the same in D, meaning that the joint distribution becomes more rectangular and therefore less circular.

Lastly, we find that neither rejection approach is best since there are values of T where the both the tolerance approach and the fraction approach are better than one another.

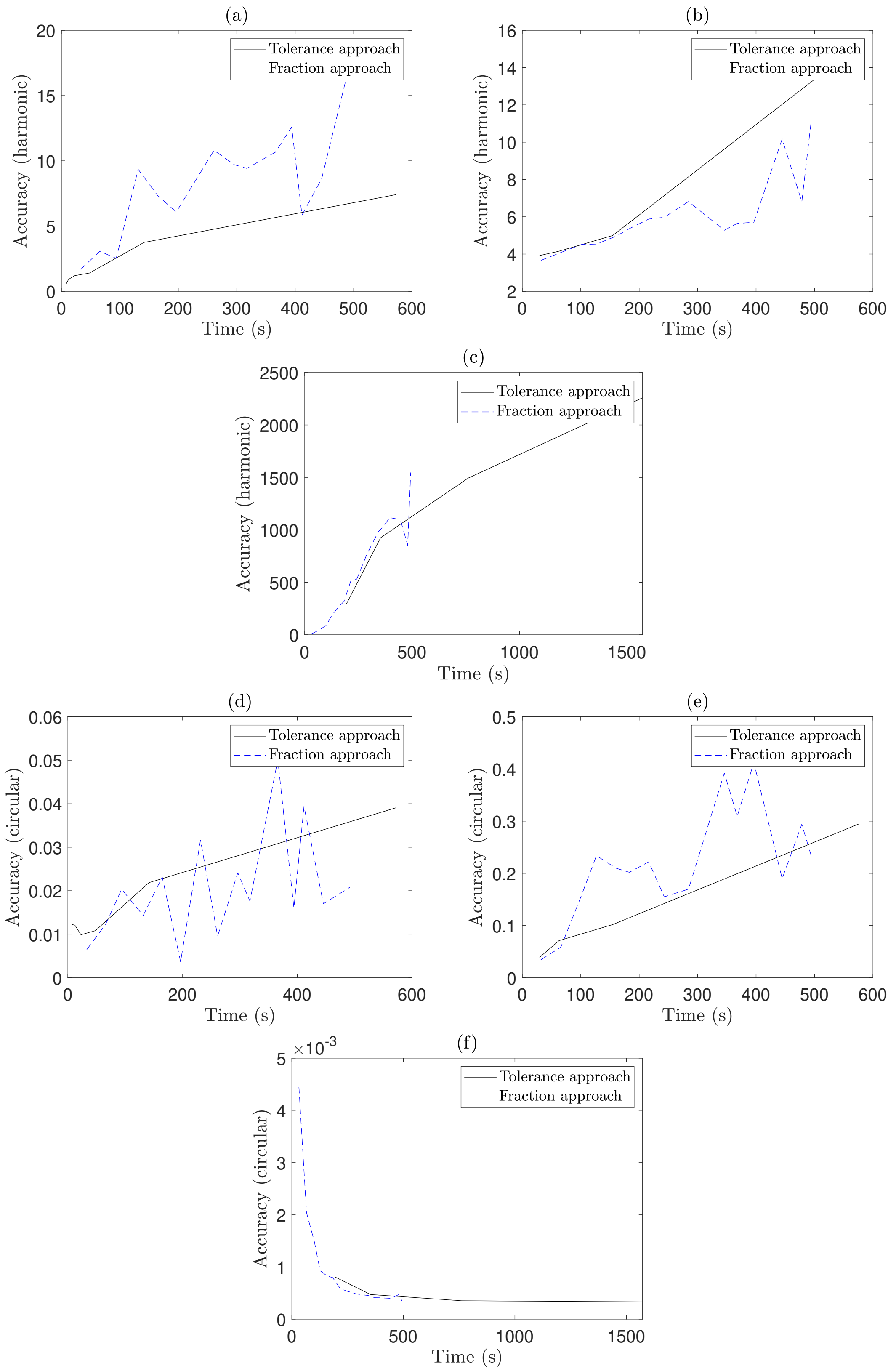

Figure 4.6: The harmonic accuracy [(a)–(c)] and the circular accuracy [(d)–(f)] against the in-built MATLAB time for the tolerance scheme (black solid line) and the fraction scheme (blue dashed line) for the ABC rejection approach for $T = 0.05\,\mathrm{s}$ [(a) and (d)], $T = 5\,\mathrm{s}$ [(b) and (e)], $T = 500\,\mathrm{s}$ [(c) and (f)]. These experiments were for $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

4.5 ABC SMC method

4.5.1 SMC algorithm

It is of interest to see how the SMC algorithm described in Algorithm 6 compares with the rejection algorithm. For the SMC method, we are required to specify a schedule of tolerance values, rather than a single value. Similar to the rejection algorithm, the value of the tolerances will depend on the summary statistics used, and so since the scale and variability of our summary statistics change depending on the value of T, we will need to use different tolerance values for the three different T values. We will initially choose larger values for the tolerances than we did for the rejection algorithm as this is typical for SMC schemes. For $T = 0.05$, we will take $\varepsilon = 8, 4, 2, 1, 0.5$ and 0.25; for $T = 5$, we will take $\varepsilon = 800, 400, 200, 100, 50$ and 25; and for $T = 500$, we will take $\varepsilon = 800000, 400000, 200000, 100000, 50000$ and 25000. Samples from the joint posterior distributions for D and α are shown in Figures 4.7–4.9. The joint posterior distributions are very similar to those from the rejection algorithm, as we would expect. We still find that α is weakly identifiable for $T = 0.05\,\mathrm{s}$, while D is weakly identifiable for $T = 500\,\mathrm{s}$. We choose not to create contour plots from the results of the SMC algorithm as they are very similar to those from the rejection approach.

Before we compare the accuracy of these results with those from the rejection algorithm, we will first introduce a similar fraction approach for the SMC algorithm.

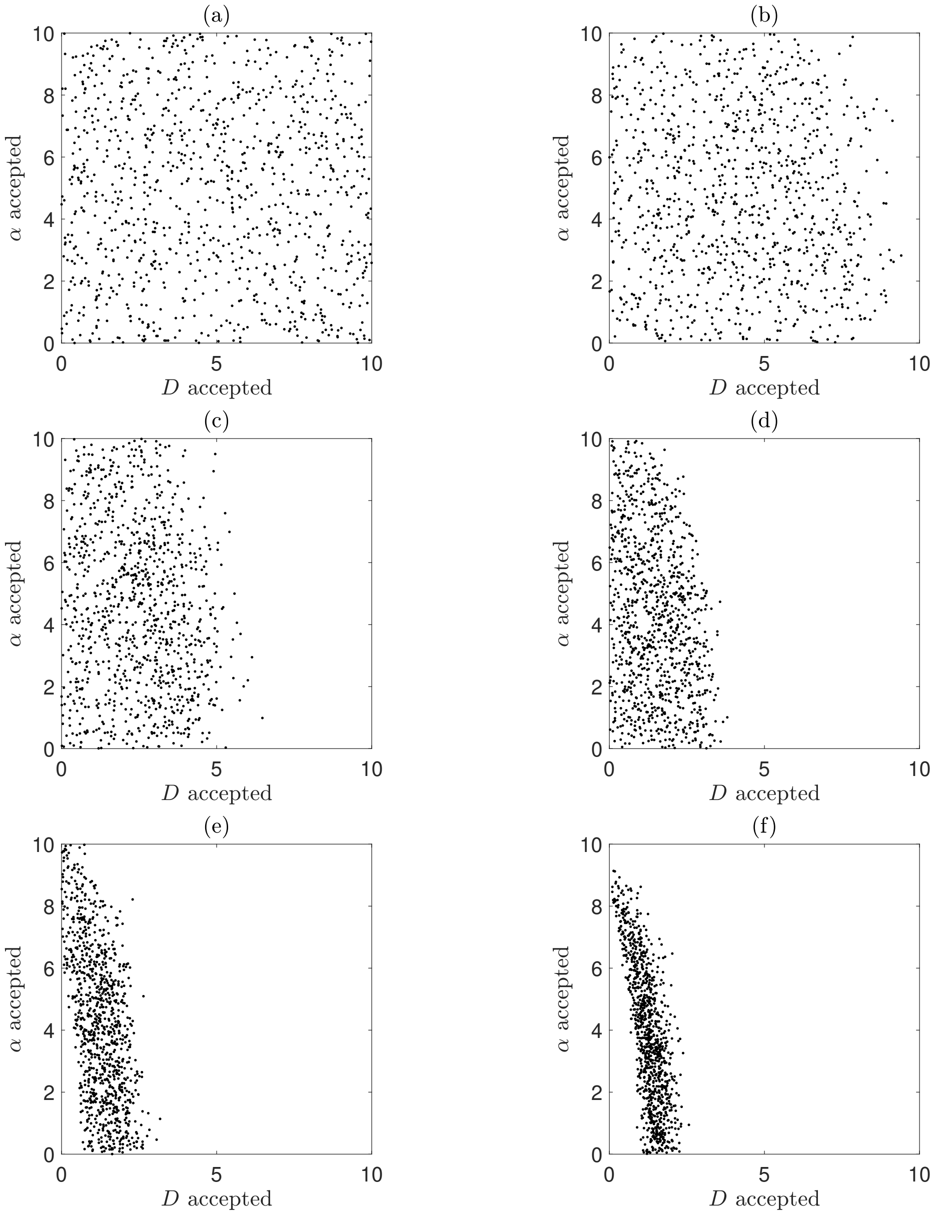

Figure 4.7: Samples from the joint posterior distributions for D and α using the Beaumont SMC algorithm for $T = 0.05\,\mathrm{s}$ and $\varepsilon = 8$ (a), $\varepsilon = 4$ (b), $\varepsilon = 2$ (c), $\varepsilon = 1$ (d), $\varepsilon = 0.5$ (e) and $\varepsilon = 0.25$ (f). These experiments were for $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

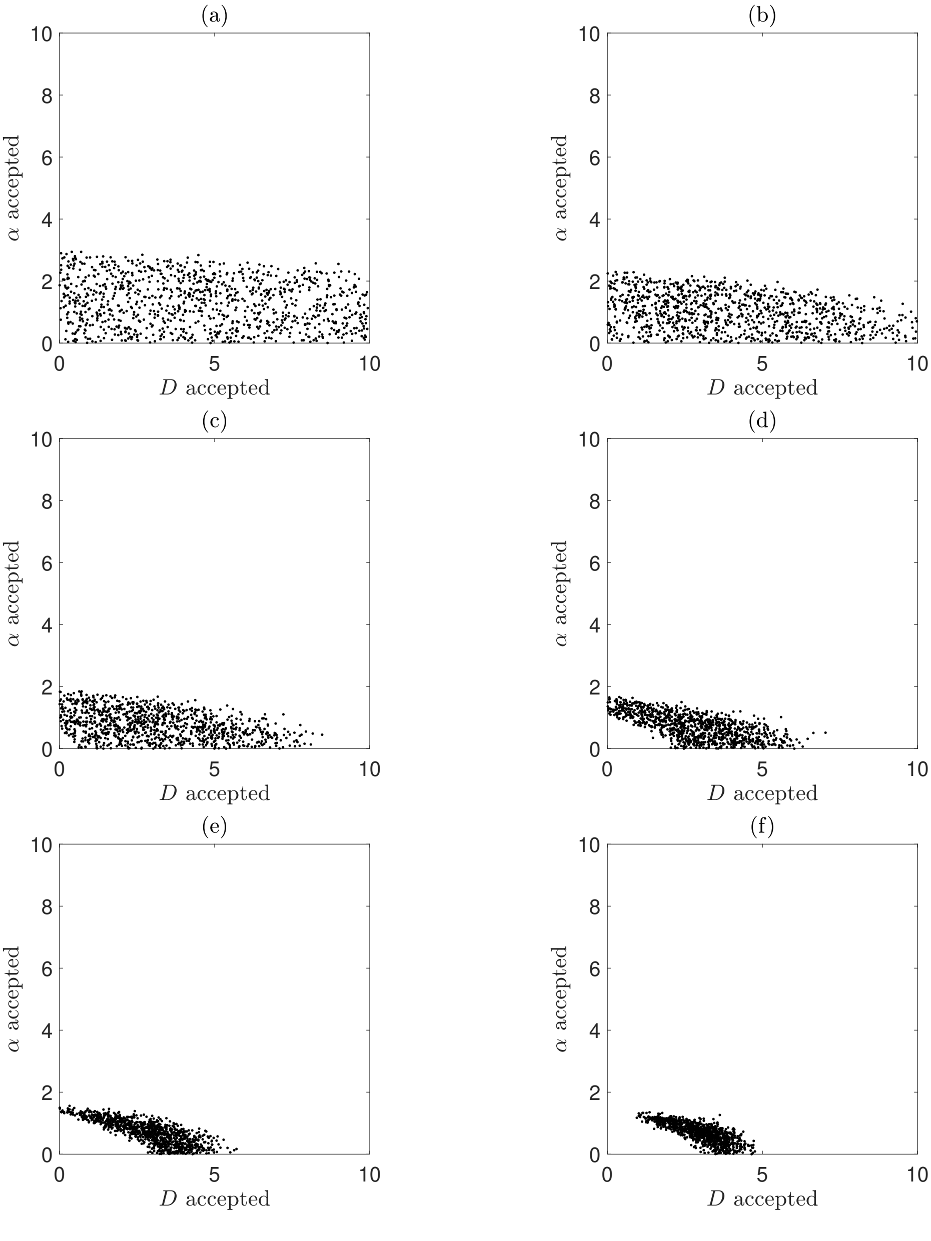

Figure 4.8: Samples from the joint posterior distributions for D and α using the Beaumont SMC algorithm for $T = 5\,\mathrm{s}$ and $\varepsilon = 800$ (a), $\varepsilon = 400$ (b), $\varepsilon = 200$ (c), $\varepsilon = 100$ (d), $\varepsilon = 50$ (e) and $\varepsilon = 25$ (f). These experiments were for $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

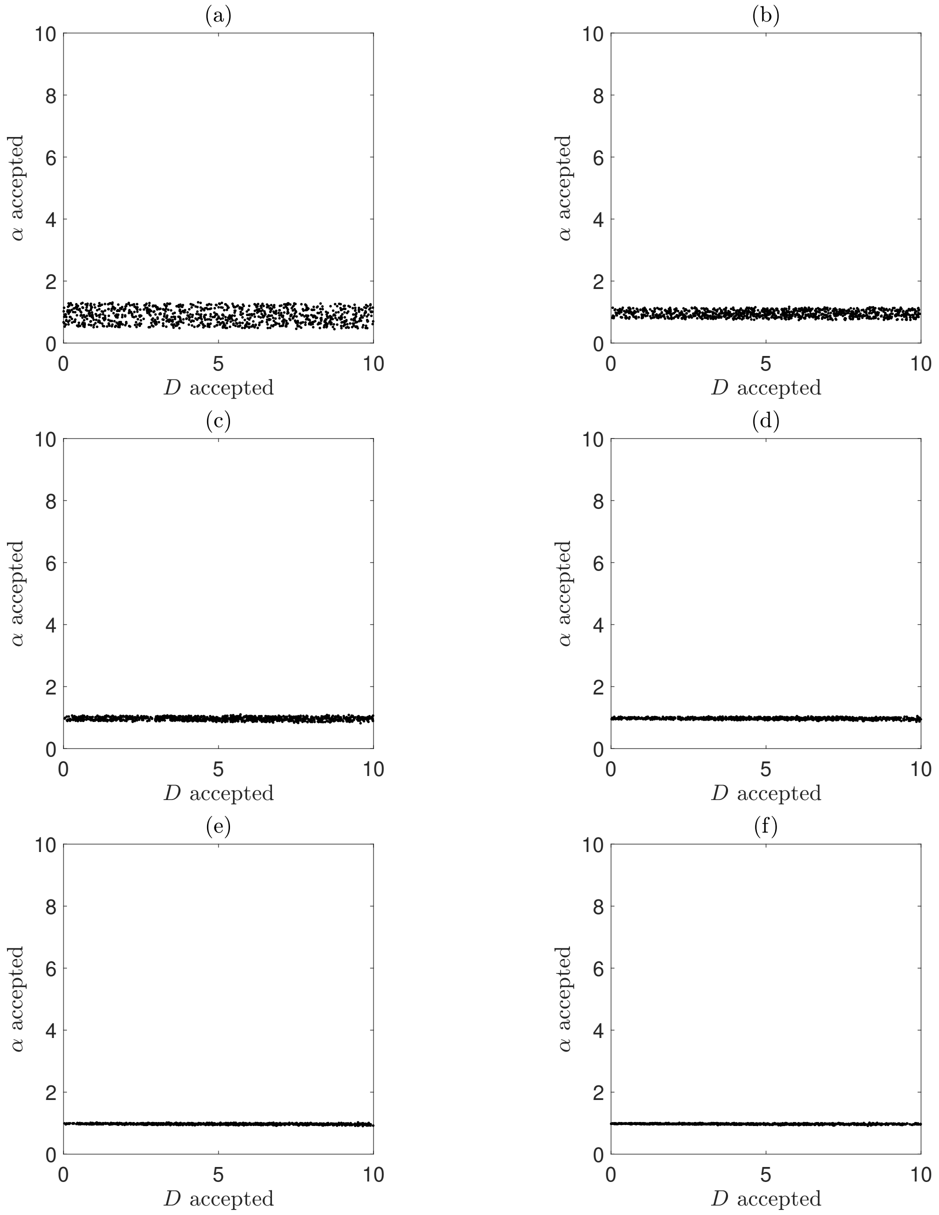

Figure 4.9: Samples from the joint posterior distributions for D and α using the Beaumont SMC algorithm for $T = 500\,\mathrm{s}$ and $\varepsilon = 800000$ (a), $\varepsilon = 400000$ (b), $\varepsilon = 200000$ (c), $\varepsilon = 100000$ (d), $\varepsilon = 50000$ (e) and $\varepsilon = 25000$ (f). These experiments were for $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m}/\mathrm{s}$, $N_S = 100$ and $N = 100$.

4.5.2 Fraction SMC approach

In this case, at each population of the SMC algorithm, we will run the fraction SMC approach so that we initially sample 10^4 parameter values from the prior and then accept the best $N_a = 1000$ values. This corresponds with always accepting the best 10% of the parameter values. As before, we will use the in-built MATLAB tic toc function to measure the real life time (in seconds) it takes for the algorithm to run.

When calculating the accuracies as in (4.9) and (4.10) for the SMC approach, the ith iteration will correspond with the ith population of the SMC algorithm, meaning that the N_ε term will correspond with the number of tolerance values used.

The results for the harmonic and circular accuracy against the time to run the algorithms for the tolerance and fraction SMC approaches is given in Figure 4.10. We find that the SMC approach gives similar results for the rejection approach. The highest harmonic accuracy is found when $T = 500\,\mathrm{s}$, meaning that α is inferred better for this value of T than the strongly identifiable parameters are inferred for the other values of T. The circular accuracy is highest when $T = 5\,\mathrm{s}$, meaning that both parameters are estimated best for this value of T. The circular accuracy for $T = 500\,\mathrm{s}$ decreases as we go through the iterations since the joint posterior distributions become less circular. Once again, we find that neither the tolerance or the fraction approach are consistently the best, with one approach outdoing the other for different values of T.

Comparing these results with those from Figure 4.6 allows us to compare the SMC approach with the rejection approach. We find that for all values of T, the SMC approach produces roughly equal or higher values for both accuracies in a much smaller amount of time than the rejection approach. This suggests that the SMC approach is more efficient than the rejection approach.

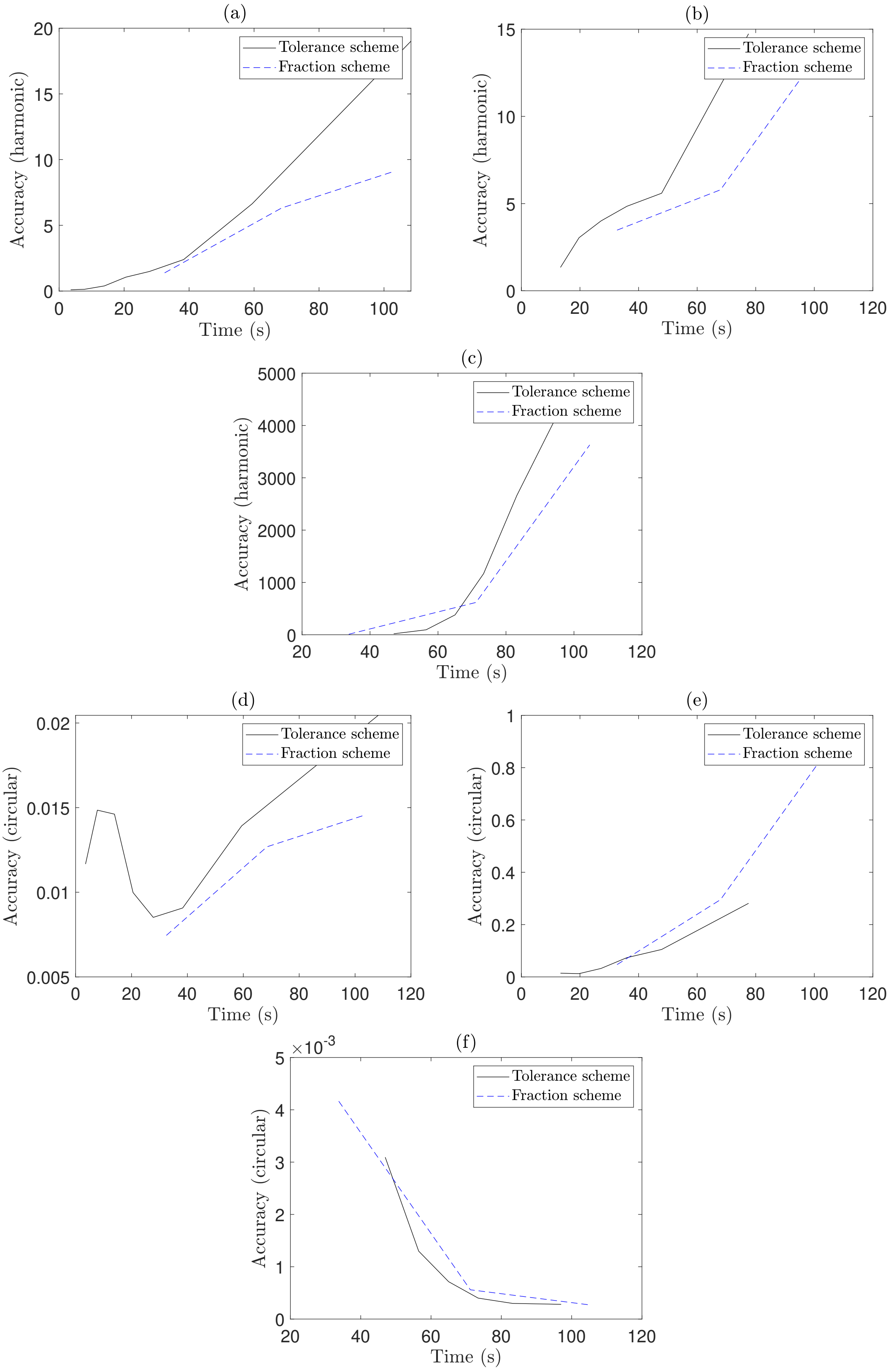

Figure 4.10: The harmonic accuracy [(a)–(c)] and the circular accuracy [(d)–(f)] against the in-built MATLAB time for the tolerance scheme (black solid line) and the fraction scheme (blue dashed line) for the ABC SMC approach for $T = 0.05\,\mathrm{s}$ [(a) and (d)], $T = 5\,\mathrm{s}$ [(b) and (e)], $T = 500\,\mathrm{s}$ [(c) and (f)]. These experiments were for $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

4.6 Prangle approach

From Figure 4.5, we saw that the ABC rejection joint posterior distributions are poor approximations of the exact joint posterior distributions, possibly due to using the MSD as the summary statistics. The Prangle approach, described in Algorithm 7, adapts the summary statistics by scaling by its MAD. This is done to normalise the summary statistics so that they are in scale with one another. We will test whether adapting the MSD in this fashion improves the joint posterior distributions we get from ABC.

For the Prangle algorithm, the initial value of the tolerance will be large enough to accept all parameter samples. All future tolerance values will then be taken to be the ϕ quantile of the previous accepted distance values as in Step 15 of Algorithm 7. Although the algorithm currently stops when the next tolerance value is below a given threshold value, alternatively, the algorithm could be stopped after a fixed number of iterations. We choose to stop the algorithm after 8 iterations. As for deciding the value of ϕ, it was observed that $\phi = 0.5$ was optimal in the examples Prangle tested in his paper [59]. However, the value of ϕ determines how small the tolerance values go, and in our case, we found that taking these values too small can cause issues in the acceptance rate. These issues are caused by using the MSD as the summary statistics. Therefore, we will take the smallest value of ϕ that allows for reasonable run times for the three different values of T. For $T = 0.05\,\text{s}, 5\,\text{s}$ and $500\,\text{s}$, we will take the values $\phi = 0.9, 0.5$ and 0.6, respectively.

Figure 4.11 shows contour plots of the exact joint posterior distributions superimposed with the final Prangle joint posterior distribution samples, along with the contour plots created from these samples. We choose to zoom into the plots for the middle and large value of T as the joint posteriors distributions become small. We find that scaling the MSD by an estimate of its MAD gives similar results as the SMC approach for a small value of T, but improves the joint posterior distributions for a middle and large value of T. However, we see that the final Prangle joint posterior distributions are underdispersed, and so the uncertainty in the parameters are not well captured.

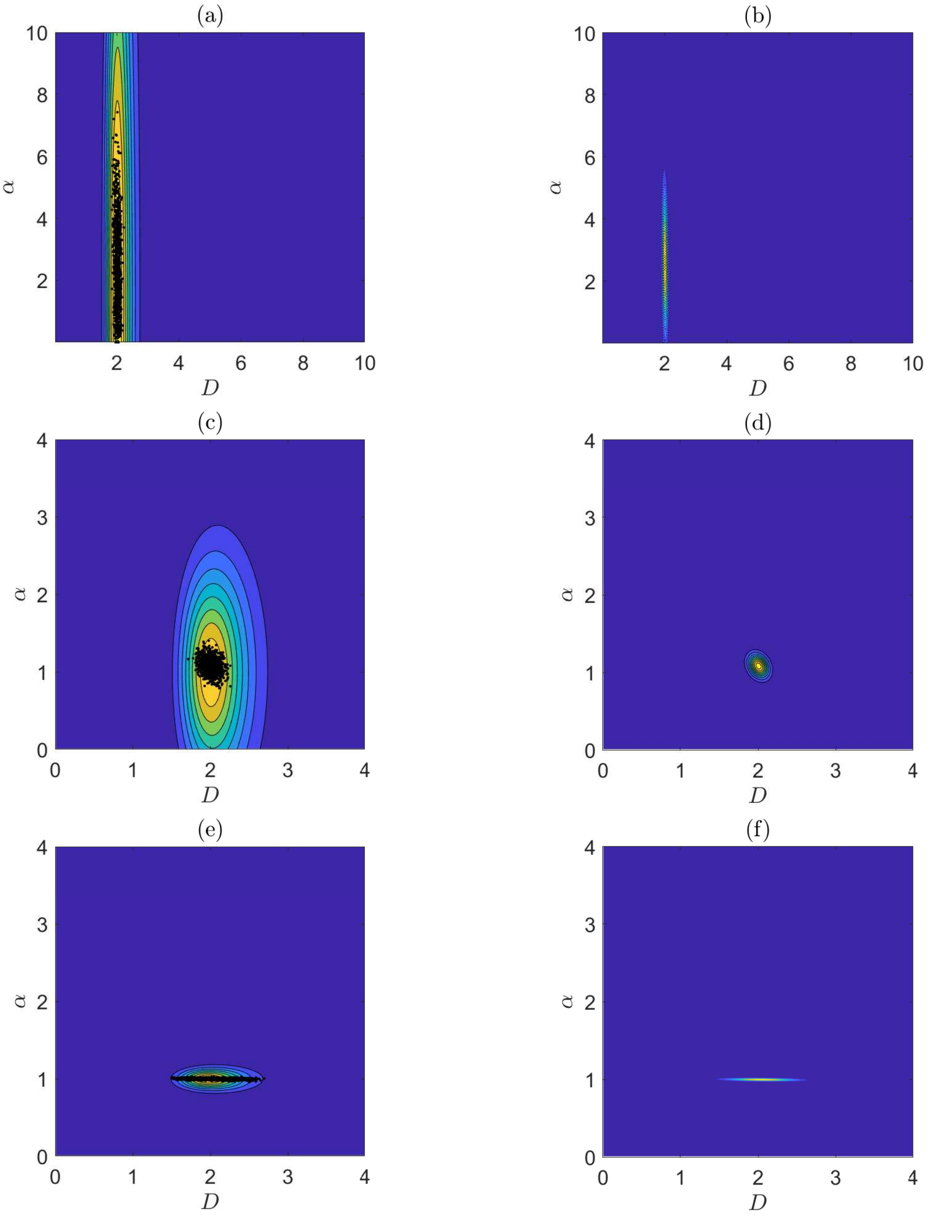

Figure 4.11: Contour plots of the exact joint posterior distributions with the ABC samples [(a), (c) and (e)] and the contour plots created from the ABC samples [(b), (d) and (f)] using the Prangle algorithm for $T = 0.05\,\text{s}$ (a), $T = 5\,\text{s}$ (b) and $T = 500\,\text{s}$ (c). The experiments for the Prangle algorithm used $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $N_S = 100$ and $N = 100$.

When comparing ABC approaches going forward, we will continue to show contour plots of the exact joint posterior distributions superimposed with the ABC samples, along with contour plots created from these samples. We will also plot both the harmonic and circular accuracy, since the joint posterior distributions for future ABC approaches may be similar to that from the Prangle approach. Previously, we defined the simulation count as a measure of the real-life time it took for the algorithm to run. We chose this measure since the fraction approach always accepts the same number of samples. Since the rest of the ABC approaches will accept parameter values based on some tolerance values, meaning that a different number of samples will be accepted at each iteration, we have decided to define a new simulation cost. We define this as the number of times we solve numerically our drift-diffusion SDE throughout the algorithm, which we call the simulation cost. This was chosen as it is a better indicator of cost than the simulation run time which is computer dependent.

Figure 4.12 shows the harmonic and circular accuracy against the simulation count for the three different values of T when using the Prangle algorithm. This is generated for comparison with future ABC approaches.

4.7 Fearnhead & Prangle approach

4.7.1 Fearnhead & Prangle algorithm

An alternative approach that could improve the ABC posterior distributions would be to use different summary statistics, rather than scaling the current summary statistics as in the Prangle approach. The Fearnhead & Prangle (F&P) approach described in Section 3.6 of Chapter 3 replaces the MSD based summary statistics with estimates of the posterior mean for D and α as the summary statistics.

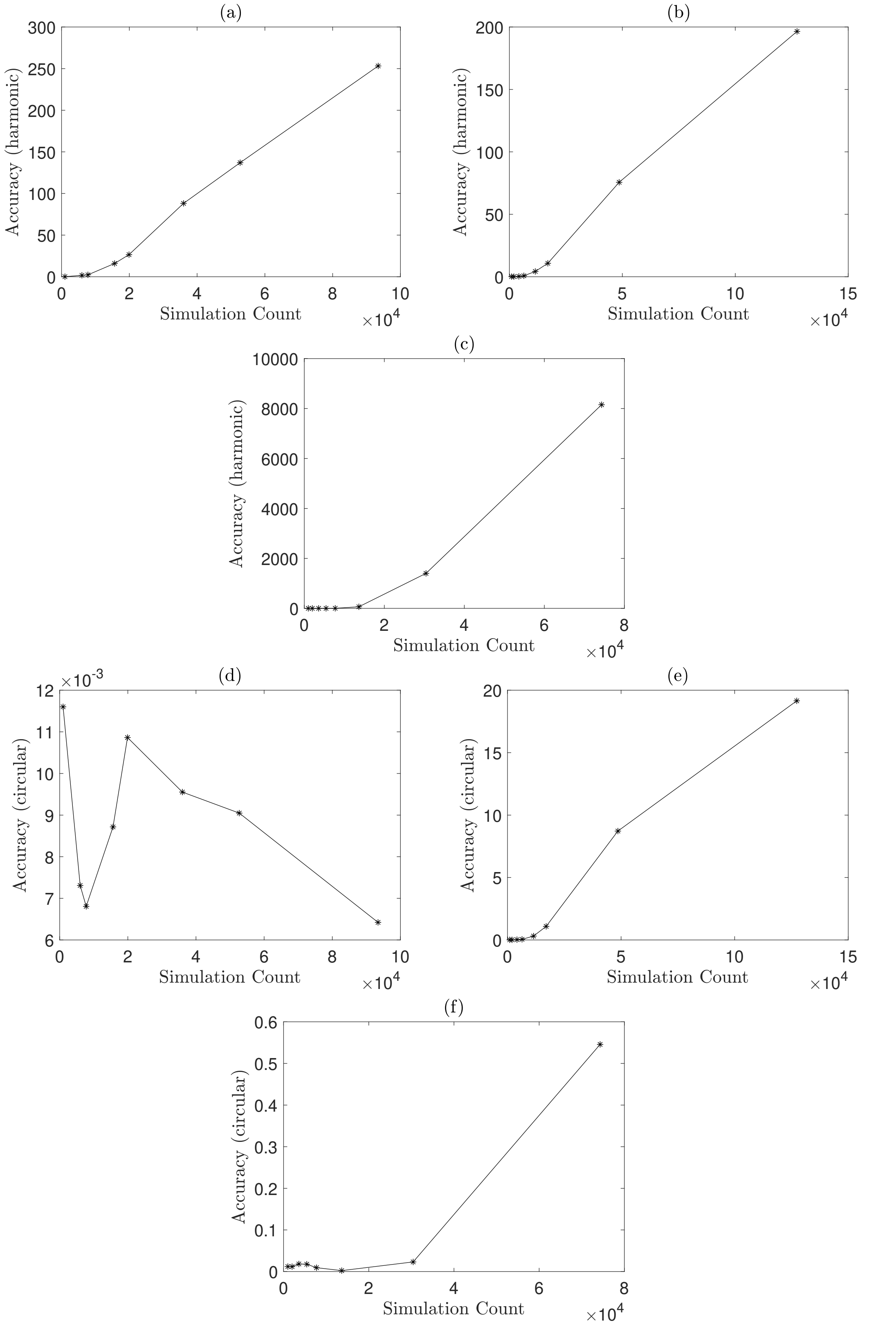

Figure 4.12: The harmonic accuracy [(a)–(c)] and the circular accuracy [(d)–(f)] against the simulation count for the Prangle approach for $T = 0.05\,\mathrm{s}$ [(a) and (d)], $T = 5\,\mathrm{s}$ [(b) and (e)], $T = 500\,\mathrm{s}$ [(c) and (f)]. These experiments were for $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

It is of interest to test whether the F&P approach gives better results than the Prangle approach. We will consider adaptations of the F&P approach later in the chapter, and so we will refer to this method as the Fearnhead & Prangle multivariate linear regression (F&P MVLR) approach. To implement the F&P MVLR approach, we first need to run any ABC approach for a small number of iterations. This step only needs to be done if the prior distributions are uninformative. For this initial ABC step, we shall run the Prangle algorithm for 3 iterations. Next, we simulate M lots of N_S trajectories by sampling M pairs of parameter values, D_i and α_i, $i = 1, \dots, M$, from the third and final population obtained from the Prangle algorithm. We then calculate M estimates of the ensemble time-average MSD, denoted by $\boldsymbol{\rho}^i$, $i = 1, \dots, M$. From this, we use MVLR using the parameter estimates as responses and the MSD as predictors as

$$\begin{aligned} D_i &= \beta_0^D + (\boldsymbol{\beta}^D \cdot \boldsymbol{\rho}^i) + \varepsilon^D, \quad i = 1, \dots, M, \\ \alpha_i &= \beta_0^\alpha + (\boldsymbol{\beta}^\alpha \cdot \boldsymbol{\rho}^i) + \varepsilon^\alpha, \quad i = 1, \dots, M, \end{aligned} \tag{4.11}$$

where $\beta_0^D, \boldsymbol{\beta}^D$ and $\beta_0^\alpha, \boldsymbol{\beta}^\alpha$ are the regression weights for D and α, respectively, and ε^D and ε^α are zero-mean normally distributed error terms for D and α, respectively. Our new observed summary statistics for the second ABC run will then be estimates of the posterior mean for D and α, which can be calculated by applying the regression weights to the observed MSD. During the second ABC run, we continue to calculate the MSD from the sampled D and α values, but then apply the regression weights to calculate the new simulated summary statistics. The distance between the new observed and simulated summary statistics used during the ABC algorithm will be taken to be the l_1 norm. As before, we will weight the summary statistics by an estimate of its MAD. For the initial tolerance value in the second ABC scheme, we will calculate estimates of D and α and their posterior means by applying the regression weights on all M parameter values from the third Prangle population. We can then take the l_1 norm of the difference of all M estimates and their posterior means, and take the ϕ quantile to be the initial tolerance value. All future tolerance values are calculated as the ϕ quantile of the previous accepted distance values. Note that we will adapt the value of ϕ according to the value of T to match that of the Prangle approach.

Figure 4.13 shows contour plots of the exact joint posterior distributions superimposed with the final F&P MVLR joint posterior distribution samples, along with the contour plots created from these samples. We see that replacing the summary statistics with estimates of the posterior means also improves the joint posterior distributions, but continues to unreliably estimate the uncertainty in the parameters. Comparing this with Figure 4.11, we find that the F&P MVLR joint posterior distributions are wider in α but narrower in D. This may be expected as the MSD is linear in D while quadratic in α, and so we would expect using linear regression to extract more reliable values for D than for α.

We now calculate the harmonic and circular accuracy to compare with the Prangle approach. This is shown in Figure 4.14. These results reflect what we found in Figure 4.13. For a small value of T, the harmonic accuracy for the F&P MVLR approach is larger, highlighting that the posterior distribution in the identifiable parameter D is narrower. The circular accuracy for this value of T begins to increase as both parameters are inferred more accurately, but then falls as the joint posterior distributions remain wide in α. For a middle value of T, the harmonic accuracy for the F&P MVLR approach is larger since the posterior distribution in D is narrower, but has a lower circular accuracy due to both parameters not being inferred as well. We find the opposite results for a large value of T – the harmonic accuracy is much lower as the posterior distribution for α is wider, while the circular accuracy is larger as the joint posterior distribution is more isotropic.

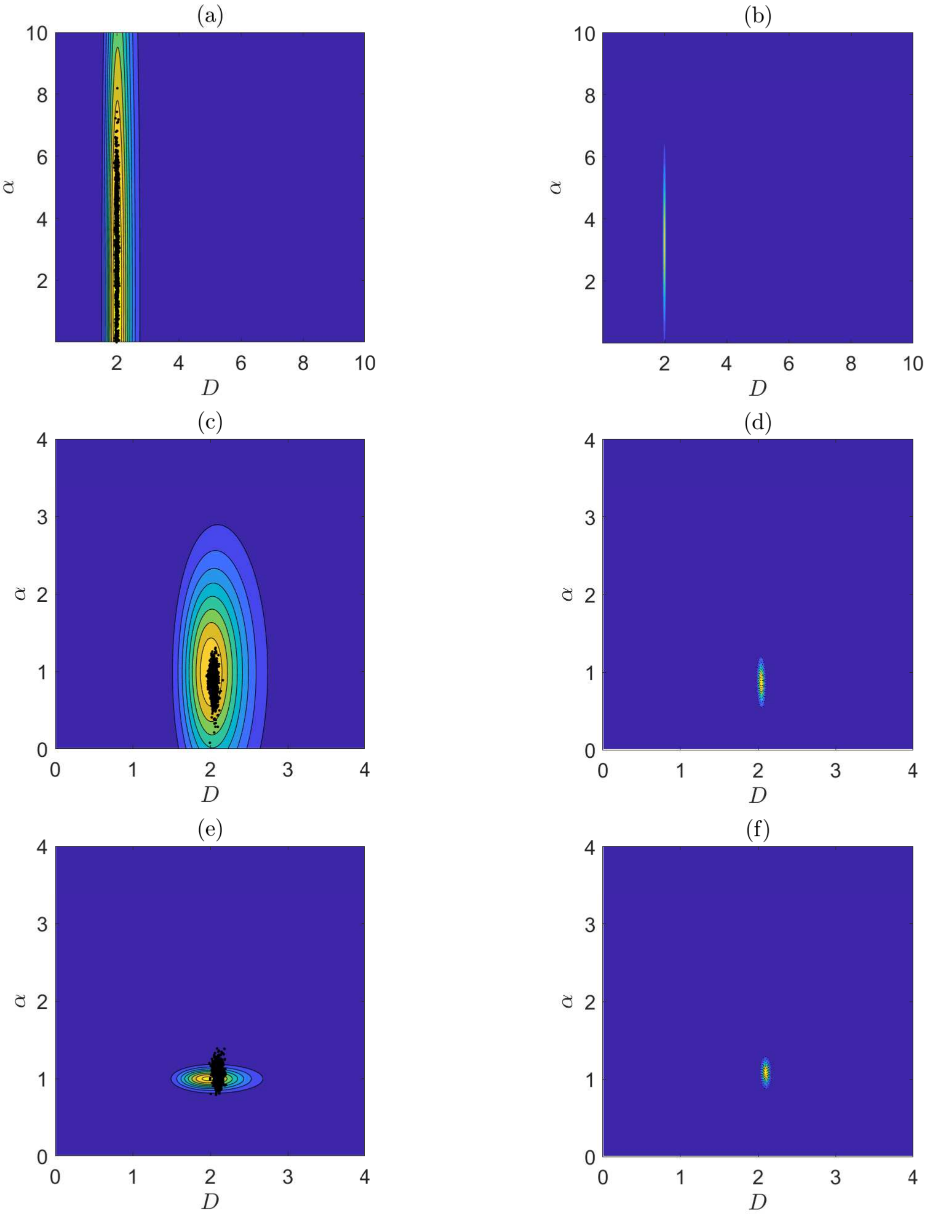

Figure 4.13: Contour plots of the exact joint posterior distributions with the ABC samples [(a), (c) and (e)] and the contour plots created from the ABC samples [(b), (d) and (f)] using the F&P MVLR algorithm for $T = 0.05\,\text{s}$ (a), $T = 5\,\text{s}$ (b) and $T = 500\,\text{s}$ (c). The experiments for the F&P MVLR algorithm used $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $N_S = 100$ and $N = 100$.

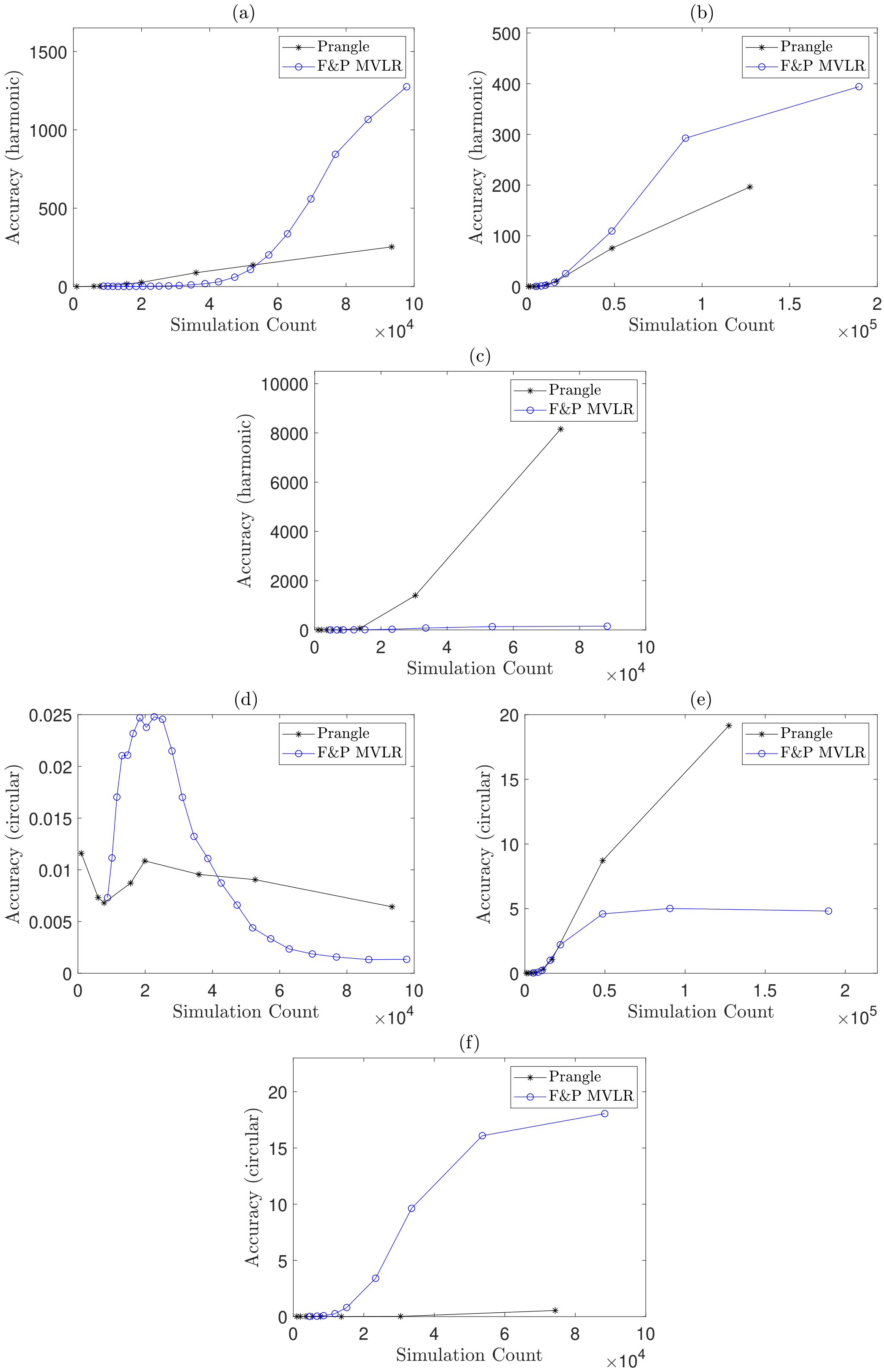

Figure 4.14: A comparison of the harmonic accuracy [(a)–(c)] and the circular accuracy [(d)–(f)] against the simulation count for the Prangle approach (black line with asterisks) and the F&P MVLR approach (blue line with circles) for $T = 0.05\,\text{s}$ [(a) and (d)], $T = 5\,\text{s}$ [(b) and (e)], $T = 500\,\text{s}$ [(c) and (f)]. These experiments were for $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $N_S = 100$ and $N = 100$.

4.7.2 Gaussian Process approach

Fearnhead & Prangle use MVLR on the initial summary statistics to estimate the posterior means of the parameters to be used as the new summary statistics. However, if the parameters are not linear in terms of the initial summary statistics, then this may not be an effective method. For example, in our case, the MSD is of the form $\alpha^2 t^2 + 2Dt$ and hence is linear in D, meaning that MVLR should work well to infer D accurately. However, the MSD is quadratic in α and so MVLR may not be a good method to infer α accurately. We saw this effect in Figures 4.13 and 4.14. In general, it may not be known a priori whether an initial summary statistic is linear in terms of its parameters.

A possible better alternative to using MVLR is Gaussian Processes (GP). GPs are a stochastic process whose distributions is the distribution over continuous functions. Since this does not assume linearity, GPs could give better results compared with MVLR.

For linear regression models, we consider a linear combination of M basis functions $\boldsymbol{\phi}(\boldsymbol{x})$ given by

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}), \tag{4.12}$$

where $\boldsymbol{x}$ is the vector of the predictors and $\boldsymbol{w}$ is a weight vector. For $\boldsymbol{w}$, we assume the Gaussian distributional form

$$p(\boldsymbol{w}) = \mathcal{N}\left(\boldsymbol{0}, \frac{1}{\sigma^2}\boldsymbol{I}\right), \tag{4.13}$$

where σ^2 is the precision. We would like to evaluate y at N training points, $x_1, x_2, \ldots, x_N$. To do so, we need to consider the joint distribution of $\boldsymbol{y} = [y(x_1), y(x_2), \ldots, y(x_N)]$. From (4.12), we have that

$$\boldsymbol{y} = \Phi \boldsymbol{w}, \tag{4.14}$$

where Φ is the design matrix with elements given by $\Phi_{ij} = \phi_i(x_j)$. Since the elements of $\boldsymbol{w}$ are Gaussian, $\boldsymbol{y}$ will also be Gaussian. Its mean and covariance will therefore be [75]

$$\mathbb{E}(\boldsymbol{y}) = 0, \tag{4.15}$$

$$\mathrm{Cov}(\boldsymbol{y}) = \frac{1}{\sigma^2}\Phi\Phi^T = K, \tag{4.16}$$

where K is called the Gram matrix whose elements are

$$K_{ij} = \frac{1}{\sigma^2}\phi(x_i)^T\phi(x_j) = k(x_i, x_j), \tag{4.17}$$

where $k(\cdot,\cdot)$ is the kernel function.

For Gaussian Processes, we can define the kernel function ourselves, rather than having it defined by the basis functions. A widely used kernel function is the squared exponential (SE) kernel function given by

$$k(x_i, x_j) = \theta_1 \exp\left(-\frac{||x_i - x_j||^2}{2\theta_2}\right), \tag{4.18}$$

where θ_1 and θ_2 are hyperparameters of the GP model. The hyperparameter θ_1 controls the amplitude of the kernel function and θ_2 controls the length scale. Obtaining values for the hyperparameters is usually done by maximising the log-likelihood by efficient gradient-based optimisation methods such as conjugate gradients [101]. If we allow the hyperparameter θ_2 to have a different value for each predictor, then this gives the ARD squared exponential (ARDSE) kernel function. It is common practice to standardise the predictors when using GPs. To do this, for each predictor, we shall subtract their mean and divide by their standard deviation. We will refer to the approach which uses the SE kernel as the F&P GP SE approach, and that which uses the ARDSE kernel as the F&P

GP ARDSE approach. For both approaches, we use a zero mean GP. We will use different GPs for D and α and calculate the posterior means by applying these to the MSD. We would expect the ARDSE method to take longer to train as a different value needs to be found for each length scale hyperparameter, but due to the different time scales associated with the MSD, we would expect this extra flexibility to give a higher accuracy compared with the SE kernel.

Contour plots of the exact joint posterior distributions superimposed with the final F&P GP joint posterior distribution samples, along with the contour plots created from these samples are given in Figures 4.15 and 4.16 for the F&P GP SE and F&P GP ARDSE approaches, respectively. We do not find much of a difference between the two F&P GP approaches. However, for a middle value of T, the F&P GP approaches have identifiability issues, causing the ABC joint posterior distributions to become bimodal. On the positive side however, for a large value of T, the ABC joint posterior distributions for the F&P GP approaches appear better than for the F&P MVLR approach.

A comparison of the harmonic and circular accuracy for the Prangle, F&P MVLR and F&P GP approaches are given in Figure 4.17. For a small value of T, the harmonic accuracy for the F&P GP approaches are roughly equal to the F&P MVLR approach. The circular accuracy for both F&P GP approaches are higher than for the F&P MVLR as the posterior distributions are slightly narrower in α. For a middle value of T, we find that the harmonic accuracy for both F&P GP approaches are either equal or larger than the F&P MVLR approach. This is surprising since the posterior distributions for both are bimodal. The circular accuracy, however, for both F&P GP approaches are much lower than for the Prangle and F&P MVLR approaches. This is due to the bimodality of the posterior distributions. For a large value of T, while the posterior distributions are narrower, they are less circular and so have a smaller circular accuracy.

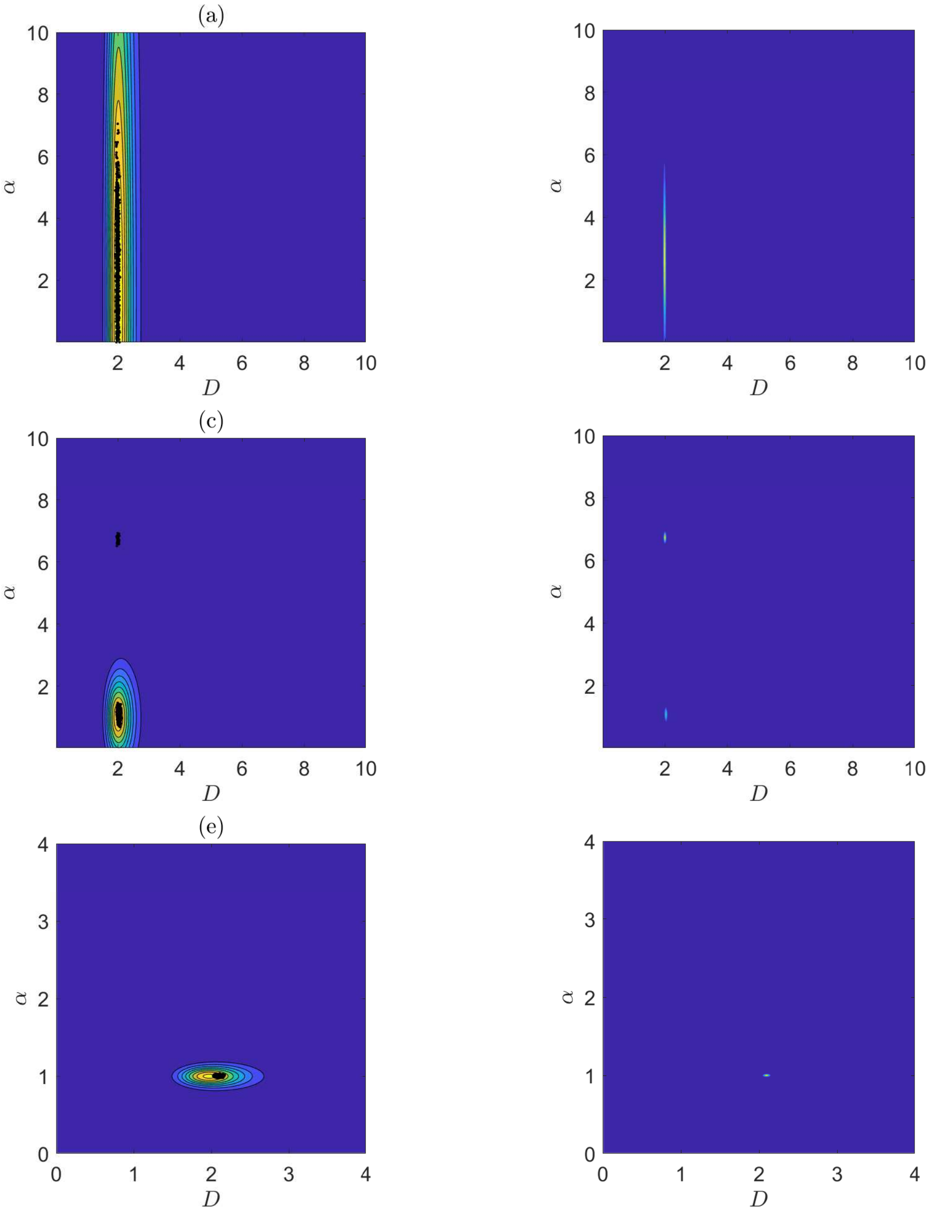

Figure 4.15: Contour plots of the exact joint posterior distributions with the ABC samples [(a), (c) and (e)] and the contour plots created from the ABC samples [(b), (d) and (f)] using the F&P GP SE algorithm for $T = 0.05\,\mathrm{s}$ (a), $T = 5\,\mathrm{s}$ (b) and $T = 500\,\mathrm{s}$ (c). The experiments for the F&P GP SE algorithm used $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

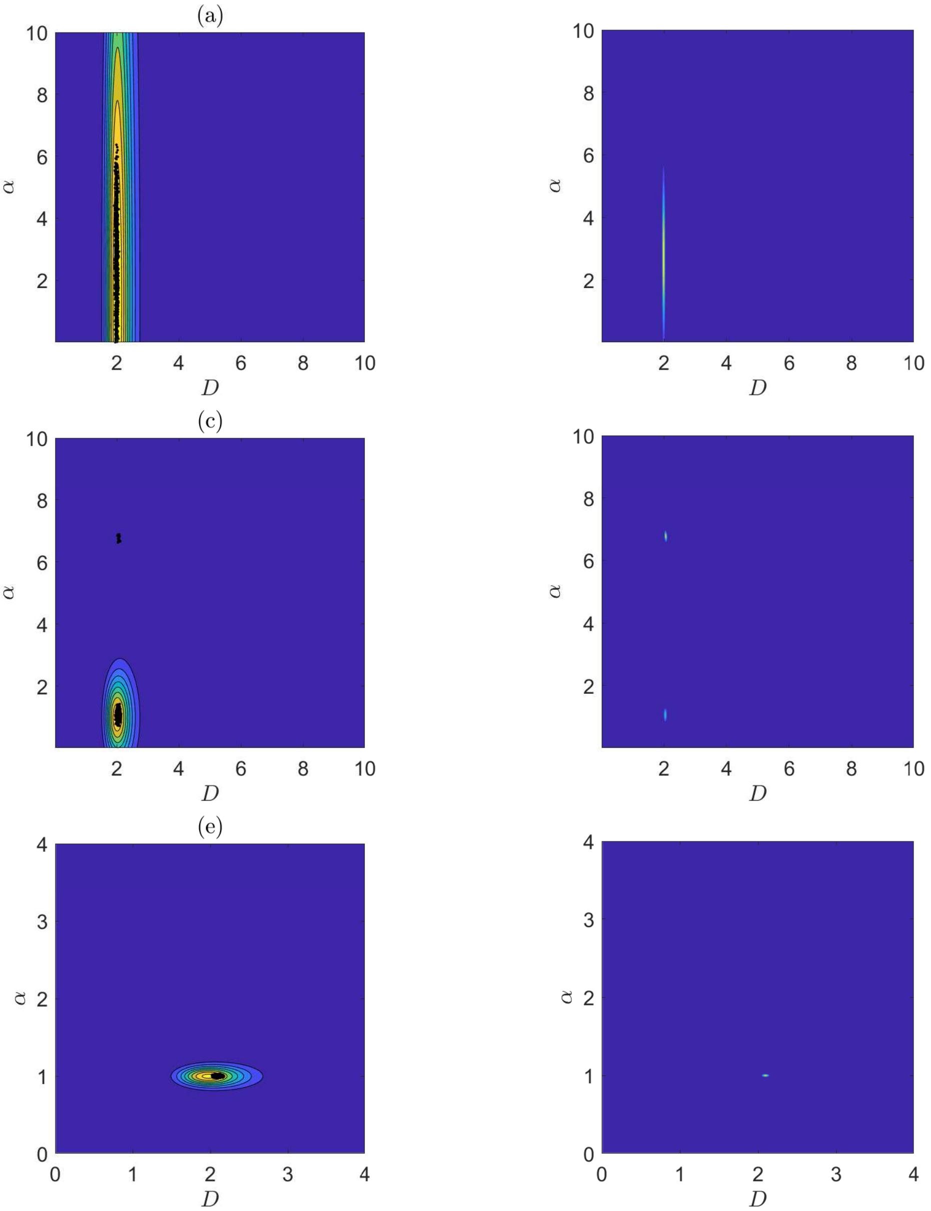

Figure 4.16: Contour plots of the exact joint posterior distributions with the ABC samples [(a), (c) and (e)] and the contour plots created from the ABC samples [(b), (d) and (f)] using the F&P GP ARDSE algorithm for $T = 0.05\,\text{s}$ (a), $T = 5\,\text{s}$ (b) and $T = 500\,\text{s}$ (c). The experiments for the F&P GP ARDSE algorithm used $D = 2\,\mu\text{m}^2/\text{s}$, $\alpha = 1\,\mu\text{m/s}$, $N_S = 100$ and $N = 100$.

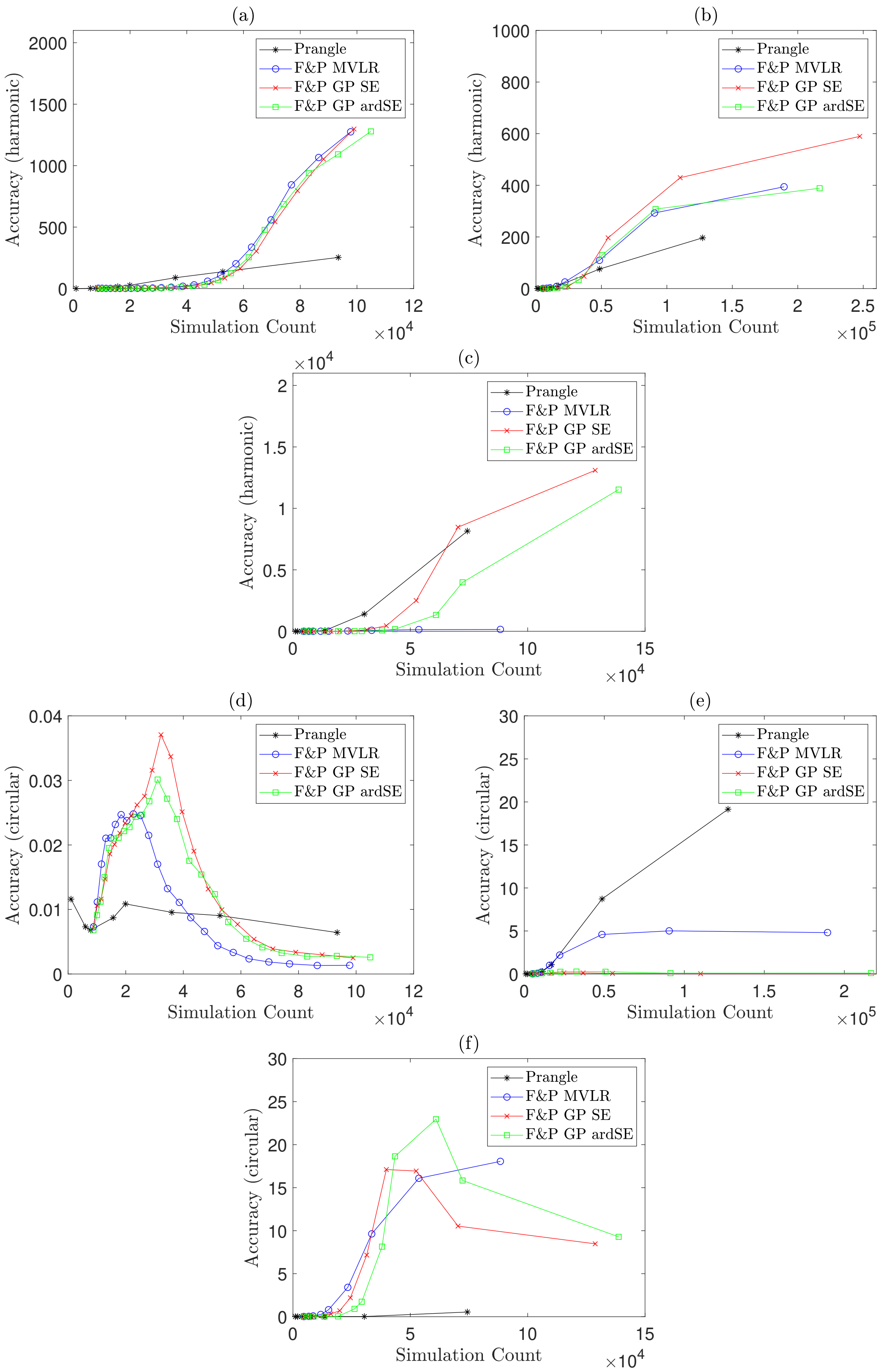

Figure 4.17: A comparison of the harmonic accuracy [(a)–(c)] and the circular accuracy [(d)–(f)] against the simulation count for the Prangle approach (black line with asterisks), the F&P MVLR approach (blue line with circles), the F&P GP SE approach (red line with crosses) and the F&P GP ARDSE approach (green line with squares) for $T = 0.05\,\mathrm{s}$ [(a) and (d)], $T = 5\,\mathrm{s}$ [(b) and (e)], $T = 500\,\mathrm{s}$ [(c) and (f)]. These experiments were for $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

Gaussian Processes are designed for interpolation and do not extrapolate well [102]. SMC algorithms perturb parameter samples, meaning that values could be sampled from outside the GP training region, requiring the GPs to extrapolate. This extrapolation could be the explanation why the approaches involving GPs have parameter identifiability issues, by giving rise to bimodality. As a way to test if this is the issue, we will try restricting the sampling region for the SMC algorithm to the convex hull of the GP training region to ensure the samples stay within the training region.

4.7.3 GPs with convex hulls

To implement this approach, we will create a convex hull during the training step, then when we perturb the parameter samples during ABC, if the perturbation takes the samples outside of the training region, they will be rejected and re-sampled.

Contour plots of the exact joint posterior distributions superimposed with the final F&P GP joint posterior distribution samples, along with the contour plots created from these samples are given in Figures 4.18 and 4.19 for the F&P GP SE and F&P GP ARDSE approaches using convex hulls, respectively. We find that restricting the sampled parameter values using convex hulls removes the bimodality of the ABC joint posterior distributions. This suggests that extrapolation from the GP training region was the reason for the parameter identifiability issues. Again, we do not find much of a difference between the two GP approaches.

A comparison of the harmonic and circular accuracy for the Prangle, F&P MVLR and F&P GP approaches using convex hulls are given in Figure 4.20. Comparing this with Figure 4.17, we find a slight increase in the harmonic accuracies for the F&P GP approaches using convex hulls for all values of T. The main difference we find for the circular accuracies is for a middle value of T, where the F&P GP approaches using convex hulls now outperform all other approaches.

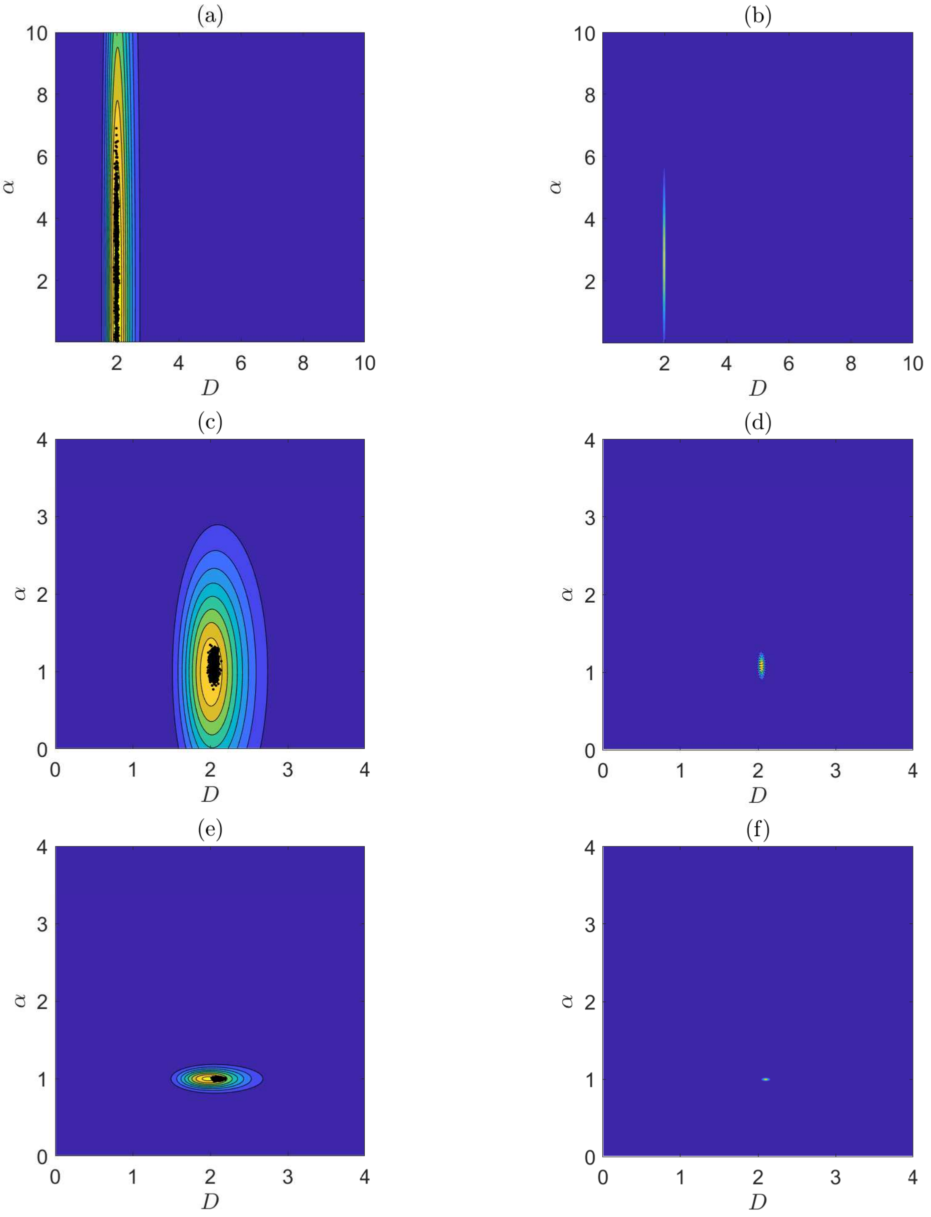

Figure 4.18: Contour plots of the exact joint posterior distributions with the ABC samples [(a), (c) and (e)] and the contour plots created from the ABC samples [(b), (d) and (f)] using the F&P GP SE algorithm using convex hulls for $T = 0.05\,\mathrm{s}$ (a), $T = 5\,\mathrm{s}$ (b) and $T = 500\,\mathrm{s}$ (c). The experiments for the F&P GP SE algorithm using convex hulls used $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

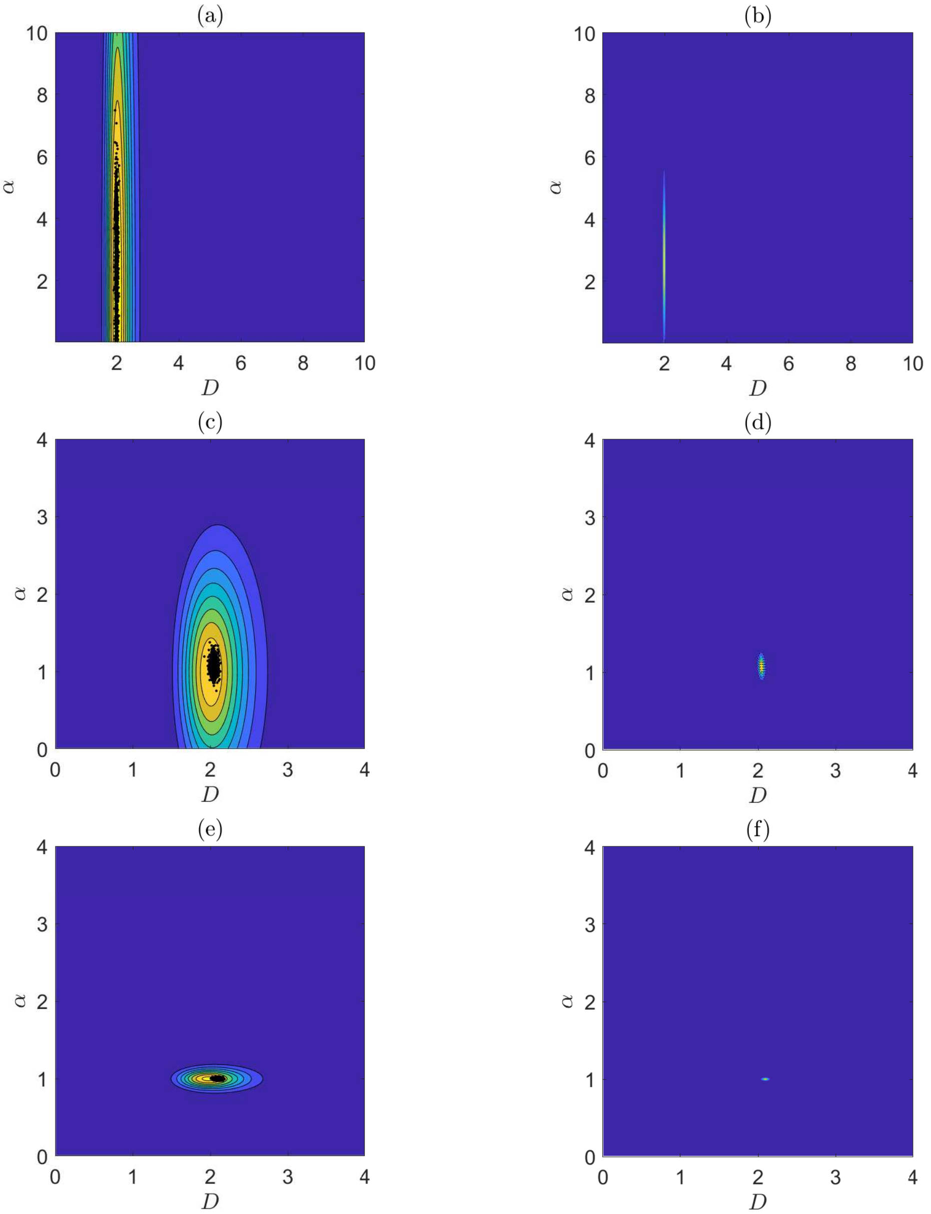

Figure 4.19: Contour plots of the exact joint posterior distributions with the ABC samples [(a), (c) and (e)] and the contour plots created from the ABC samples [(b), (d) and (f)] using the F&P GP ARDSE algorithm using convex hulls for $T = 0.05\,\mathrm{s}$ (a), $T = 5\,\mathrm{s}$ (b) and $T = 500\,\mathrm{s}$ (c). The experiments for the F&P GP ARDSE algorithm using convex hulls used $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

One possible issue with using convex hulls is that restricting the sampling region of the SMC algorithm to the GP training region could introduce a bias to the ABC results. An alternative approach that does not require convex hulls is to first use MVLR to obtain the residuals for the parameter values, which can then be fit with a zero-mean GP. This approach should hopefully circumvent the possible issues that arise from using convex hulls, while still benefiting from the advantages of GPs.

4.7.4 The residual approach

To impement the F&P residual (Res) approach, after we run the three Prangle iterations, we will use MVLR on the parameter samples. Using the regression weights from the MVLR, we will estimate the parameter values from the third Prangle posterior and subtract these from the true values to retrieve the residuals. A GP will be fit to the residuals using the SE kernel. This last step is done to try and correct any errors when using MVLR. The posterior means are then calculated by first applying the regression weights from the MVLR to the MSD, then adding the prediction of the GP applied to the MSD.

Contour plots of the exact joint posterior distributions superimposed with the final F&P Res joint posterior distribution samples, along with the contour plots created from these samples is given in Figure 4.21. The joint posterior distributions for a small and middle value of T seems very similar to the F&P GP approaches using convex hulls, while for a large value of T, they are slightly wider in both D and α.

A comparison of the harmonic and circular accuracy for the Prangle, F&P MVLR, F&P GP approaches using convex hulls, and the F&P Res approaches are given in Figure 4.22. While the harmonic accuracy for the F&P Res approach underperforms for all values of T, meaning that the identifiable parameters in each case are not inferred as accurately, we find that the F&P Res approach gives the highest circular accuracies for all values of T, meaning that both parameters are inferred best using this approach.

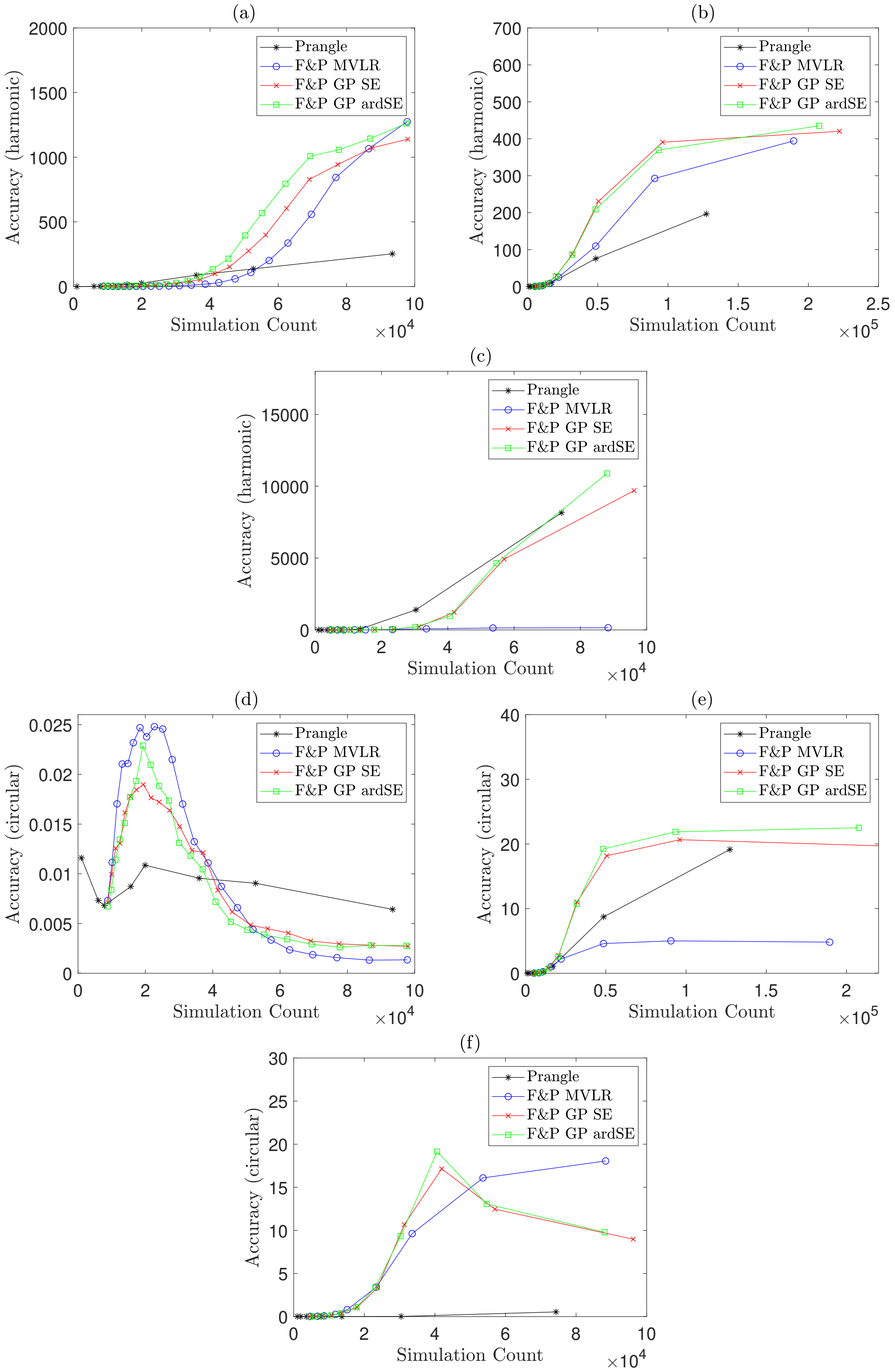

Figure 4.20: A comparison of the harmonic accuracy [(a)–(c)] and the circular accuracy [(d)–(f)] against the simulation count for the Prangle approach (black line with asterisks), the F&P MVLR approach (blue line with circles), the F&P GP SE approach using convex hulls (red line with crosses) and the F&P GP ARDSE approach using convex hulls (green line with squares) for $T = 0.05\,\mathrm{s}$ [(a) and (d)], $T = 5\,\mathrm{s}$ [(b) and (e)], $T = 500\,\mathrm{s}$ [(c) and (f)]. These experiments were for $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

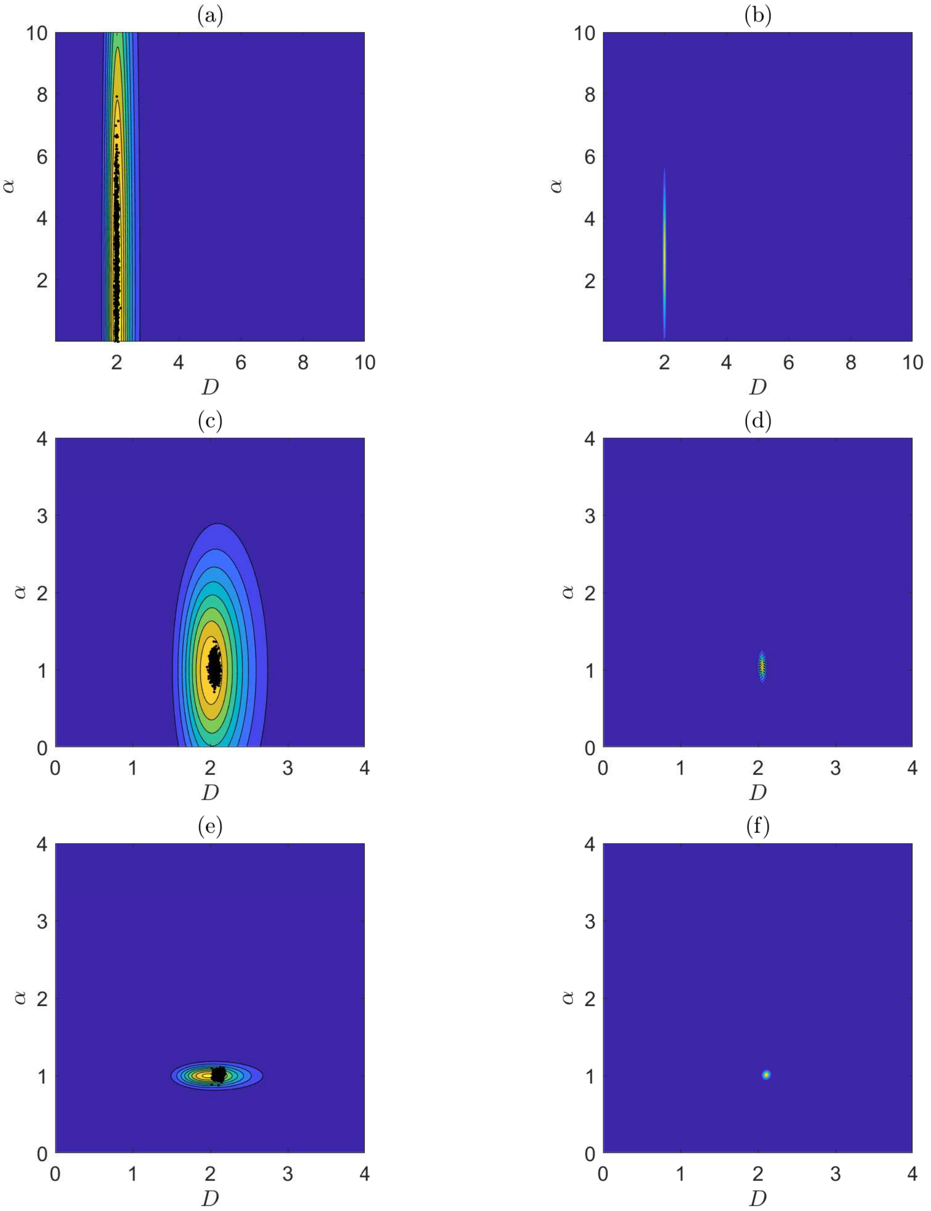

Figure 4.21: Contour plots of the exact joint posterior distributions with the ABC samples [(a), (c) and (e)] and the contour plots created from the ABC samples [(b), (d) and (f)] using the F&P Res algorithm for $T = 0.05\,\mathrm{s}$ (a), $T = 5\,\mathrm{s}$ (b) and $T = 500\,\mathrm{s}$ (c). The experiments for the F&P Res algorithm used $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

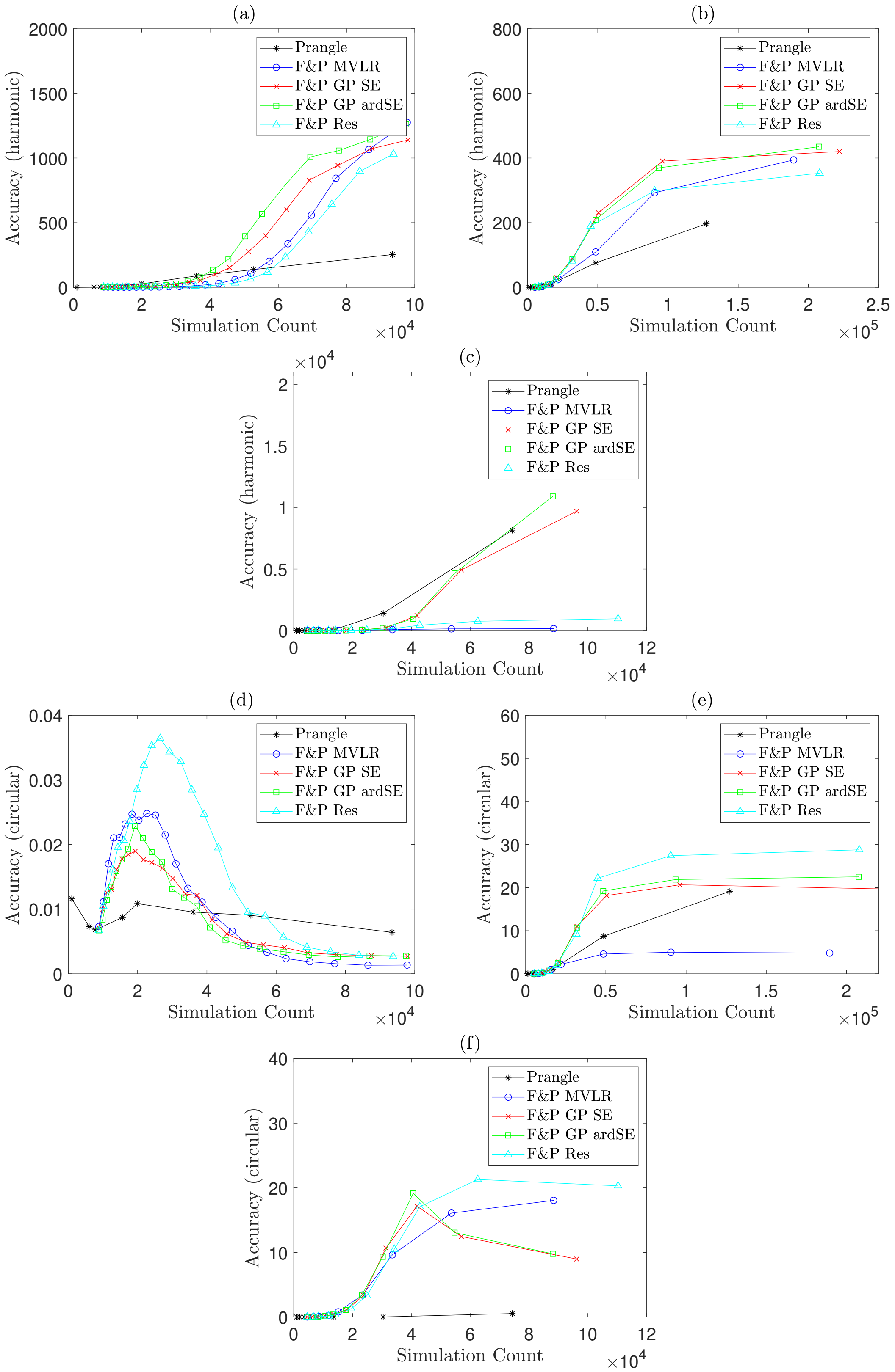

Figure 4.22: A comparison of the harmonic accuracy [(a)–(c)] and the circular accuracy [(d)–(f)] against the simulation count for the Prangle approach (black line with asterisks), the F&P MVLR approach (blue line with circles), the F&P GP SE approach (red line with crosses), the F&P GP ARDSE approach (green line with squares) and the F&P Res approach (cyan line with triangles) for $T = 0.05\,\mathrm{s}$ [(a) and (d)], $T = 5\,\mathrm{s}$ [(b) and (e)], $T = 500\,\mathrm{s}$ [(c) and (f)]. These experiments were for $D = 2\,\mu\mathrm{m}^2/\mathrm{s}$, $\alpha = 1\,\mu\mathrm{m/s}$, $N_S = 100$ and $N = 100$.

4.8 Conclusions

In this chapter, we compared different ABC approaches to inferring the diffusion coefficient D and the drift velocity α from the drift-diffusion SDE (4.1).

Our main conclusion is, regardless of the algorithm used, the ABC posterior distributions depend crucially on the value of the measurement time interval T, as shown, for example, in Figures 4.7–4.9. For a small value of T, the posterior distribution has a larger spread in α compared with D, meaning that α is inferred less accurately. In contrast, for a large value of T, the posterior distribution has a larger spread in D compared with α, meaning that D is inferred less accurately. However, there is an intermediate value of T for which the posterior distribution is more isotropic, leading to good inference for both D and α. These findings mirror the results found in Chapter 2, where too small or large a value of T results in poor inference in either D or α but an intermediate value of T was shown to balance the accuracy of the inference of both parameter values.

We began by calculating the exact joint posterior distributions by making use of the likelihood function. This was then used as a basis for comparison of all the ABC results. Starting with the ABC rejection algorithm described in Algorithm 5, we found that the resultant ABC joint posterior distributions were poor approximations of the exact joint posterior distributions, as seen in Figure 4.5. The harmonic and circular accuracy measures were introduced as an easier comparison between the different ABC approaches. An alternative approach, named the fraction approach, which generates a large number of samples and accepts the best fraction, was implemented with the ABC rejection algorithm. We found that neither the original ABC approach or the fraction approach performs best. The ABC SMC algorithm was then implemented and was found to outperform the rejection algorithm, giving higher harmonic and circular accuracies in a shorter time. The fraction approach was implemented with the SMC approach and again we found that neither the original or the fraction approach produces better results for all values of T. Going forward, we chose to compare the future ABC approaches with the original SMC results.

The main focus of the chapter was to compare the approaches presented by Prangle [59] and Fearnhead & Prangle [60]. The key difference between the two approaches is the choice of summary statistics. For the Prangle approach, the MSD is weighted by the MAD during the ABC acceptance step. For the F&P approach, we use regression on the MSD to estimate the posterior mean of the model parameters. These estimates are then used as the new summary statistics within the ABC framework. The regression method that Fearnhead & Prangle used was multivariate linear regression. We found that the Prangle and F&P approaches resulted in better inference of the parameters in that the bias was smaller, but the posterior variance was substantially underestimated. We found that this was an intrinsic aspect of all ABC approaches going forward. A comparison of the harmonic and circular accuracies showed that the two different approaches performed better depending on the value of T. For a small and middle value of T, the F&P MVLR approach has higher harmonic accuracies but lower circular accuracies. This is due to the identifiable parameters being inferred more accurately using the F&P MVLR approach, with both parameters being inferred best together simultaneously using the Prangle approach. For a large value of T, the Prangle approach now has the higher harmonic accuracy but lower circular accuracy.

Since the MSD is quadratic in α and linear in D, we believed that using MVLR may not be an effective method to estimate the posterior means accurately. Therefore, we adapted the F&P approach to use GPs instead of MVLR. For this, we fit GPs to the sample data with the squared exponential and the ARD squared exponential kernels. The hope was that the GP approach would improve the accuracy of α while maintaining the high accuracy of D, resulting in a higher overall accuracy than the MVLR approach. Looking at Figures 4.15 and 4.16, we found that both approaches produced bimodal distributions for a middle value of T. We suspected that this parameter identifiability issue was due to the GPs having to extrapolate when the SMC algorithm perturbed the parameter samples outside the GP training region. To test if this was the reason, we implemented an approach which restricted the ABC samples to the convex hull of the GP training region. This approach ensured that the perturbed samples from the SMC algorithm remained in the GP training region. We found that the GP approach using convex hulls removed the bimodality of the resultant joint posterior distributions, as seen in Figures 4.18 and 4.19.

A possible issue with using convex hulls is, since the perturbed parameter samples during the SMC algorithm are forced to remain in the GP training region, this could result in a bias to the ABC joint posterior distributions. To prevent this, we implemented the F&P Res approach. For this, we initially used MVLR to obtain the residuals of the parameter values, which were then fit with a zero-mean GP. We found that the F&P Res approach ensured that the joint posterior distributions remained unimodal for all values of T, as seen in Figure 4.21. A comparison of the harmonic and circular accuracies, as seen in Figure 4.22, showed that the F&P Res approach did not perform as well as the other approaches in terms of the harmonic accuracy, meaning that the indentifiable parameters were not inferred as accurately. However, the F&P Res approach produced the highest circular accuracy for all values of T. Therefore, if the aim is to accurately infer both parameters simultaneously then the F&P Res approach would be best.

Chapter 5

Model for self-generated gradient cell movement

5.1 Introduction

To our knowledge, no one has used a drift-diffusion stochastic model to describe self-generated gradient chemotaxis. In this chapter, we present details of a hybrid discrete-continuum model of a population of cells moving in response to a self-generated chemotactic gradient. The experimental set-up is motivated by that in the paper by Tweedy *et al.* [6]. After introducing the theory and numerical solutions of the model, we will then perform a sensitivity analysis to see how the identifiability of certain parameters depends on the time over which the MSD is measured. Lastly, ABC will be used to try and infer the parameters of interest.

5.2 Experimental set-up

In the experiment by Tweedy *et al.* [6], *Dictyostelium discoideum* cells move within a two dimensional chamber of length L and height H. Initially, a uniform amount of saturating level of folic acid is uniformly dissolved in an agarose gel. The cells start in a small well at the left hand side of the chamber and gradually migrate away from the well by creating self-generated gradients of the chemoattractant. A rough illustration of the set-up is given in Figure 5.1.

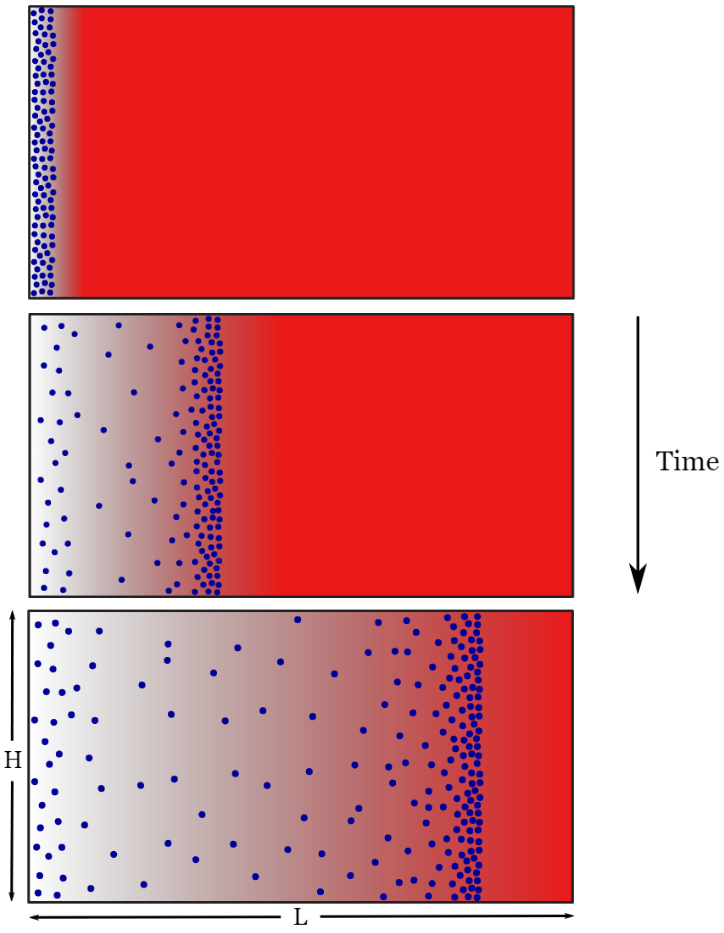

Figure 5.1: A rough illustration of the experimental set-up. The blue circles represent cells and the red areas represent the chemical attractant.

5.3 Modelling cell movement

In terms of the mathematical model, we will use a one-dimensional version of our drift-diffusion SDE. The reasoning for this is discussed in Section 4.1. To briefly reiterate, Ferguson *et al.* [40] used a Kolmogorov-Smirnov test to confirm that the y-coordinate data were not significantly different from uniform distributions, along with a test that confirmed that the x- and y- coordinate data are independent from each other. Both of these give evidence that a one-dimensional model is appropriate to study the self-generated gradient data. Therefore, we assume the data has been reduced in the y-direction, leaving the one-dimensional x-coordinate data.

We will model the movement of each individual cell by the one-dimensional drift-diffusion SDE

$$\mathrm{d}X_t = \nu \, \mathrm{d}l \, \frac{K_d}{(K_d + c)^2} \frac{\partial c}{\partial x} \, \mathrm{d}t + \sqrt{2D} \, \mathrm{d}W_t, \tag{5.1}$$

where X_t is the location of the cell at time t, ν is a chemotactic velocity parameter, $\mathrm{d}l$ is the diameter of the cell, K_d is the disassociation constant, $c \equiv c(x,t)$ is the concentration of the chemical at position x at time t, D is a measure of the random motility of the cells, assumed to be equal and constant for all cells, and $\mathrm{d}W_t$ is a Wiener processes. Here, the domain of X_t is $[0, L]$, with the initial condition $X_0 = \upsilon$, where υ follows the uniform distribution $U(0, L/20)$. Note that this uniform distributions ensures that the cells begin in a small well of length $L/20\,\mu$m. To ensure that the cells remain in the chamber, we have the boundary conditions $X_t = -X_t$ if $X_t < 0$, and $X_t = L - X_t \,(\mathrm{mod\,L})$ if $X_t > L$.

The chemotaxis velocity term in (5.1) is motivated by looking at receptor-ligand kinetics. First, imagine cells interacting with a chemical attractant. Over time, ligands begin to bind on and off the cell receptors. The rate at which ligands bind on to the receptors depends on the number of free receptors and the concentration of the chemical, while the rate at which they bind off the receptors depends on the number of bound receptors. From this, if we let ψ denote the number of bound receptors, then we have

$$\frac{\partial \psi}{\partial t} = k_1 c (R_{tot} - \psi) - k_{-1} \psi, \tag{5.2}$$

where k_1, k_{-1} are the rates at which the ligand binds on and off the receptors, respectively, and R_{tot} is the total receptor number. For simplicity, we will assume that R_{tot} is constant. If we denote $R = \psi/R_{tot}$ as the fractional receptor occupancy, then we can rewrite (5.2) so that

$$\frac{\partial R}{\partial t} = k_1c(1-R) - k_{-1}R \tag{5.3}$$

$$= k_1c - (k_1c + k_{-1})R \tag{5.4}$$

$$= (k_1c + k_{-1})\left(\frac{k_1c}{k_1c + k_{-1}} - R\right). \tag{5.5}$$

If the chemical concentration remains constant over a long time scale, then we can assume that the receptor occupancy reaches an equilibrium value where $\partial R/\partial t = 0$. Therefore, we get

$$R = \frac{k_1c}{k_1c + k_{-1}} \tag{5.6}$$

$$= \frac{c}{K_d + c}, \tag{5.7}$$

where $K_d = k_{-1}/k_1$ denotes the disassociation constant. Analysis by [103] showed that the rate of change of the number of bound receptors given by (5.2) has solution

$$\psi = \frac{k_1R_{tot}c}{k_1c + k_{-1}} + \left(\psi_0 - \frac{k_1R_{tot}c}{k_1c + k_{-1}}\right)\exp(-(k_1c + k_{-1})t), \tag{5.8}$$

where ψ_0 is the initial number of bound receptors. We can see therefore that the rate to reach equilibrium is determined by $k_1c + k_{-1}$. We will assume that the initial background concentration $c \gg K_d$, and so

$$k_1c + k_{-1} = (k_1c - k_{-1}) + 2k_{-1} > k_1c - k_{-1} \gg 0. \tag{5.9}$$

Therefore, the exponential term in (5.8) will decay rapidly, and so the timescale to reach equilibrium will be small compared to the other processes taking place.

Denoting the difference in fractional receptor occupancy from the front to the back of the cell by ΔR, we can approximate this by

$$\Delta R \approx \mathrm{d}l \, \frac{\partial R}{\partial x} \tag{5.10}$$

$$= \mathrm{d}l \, \frac{\partial R}{\partial c} \, \frac{\partial c}{\partial x} \tag{5.11}$$

$$= \mathrm{d}l \, \frac{K_d}{(K_d + c)^2} \, \frac{\partial c}{\partial x}. \tag{5.12}$$

If we assume that the chemotactic velocity is proportional to ΔR with velocity ν then we arrive at (5.1). The chemotactic term in (5.12) is similar to that used in Hillen and Painter [104] and others [105, 106]. They looked at PDE chemotaxis models of advection-diffusion type, where the advection models the cell density movement.

It is instructive to consider the behaviour of this chemotatic term under different scenarios. For example, if we have a fixed relative concentration gradient, then

$$\frac{\Delta c}{c_0} \approx \frac{\mathrm{d}l}{c_0} \frac{\partial c}{\partial x} = \text{constant},$$

where Δc and c_0 denotes the difference and average concentration across the cell, respectively. In this situation we find that

$$\Delta R \propto \frac{c K_d}{(K_d + c)^2}. \tag{5.13}$$

We can see that the chemotactic term therefore decays to zero as the absolute concentration level tends to zero as expected. We also see that $\Delta R \to 0$ when $c \gg K_d$ as in this situation almost all of the cell's receptors are occupied and hence it is difficult for the cell to determine the gradient of the chemoattractant.

It is easy to show that in fact ΔR is maximised when $c \approx K_d$. At this level of chemoattractant roughly half of the cell's receptors are occupied at the front and the back of the cell.

5.4 Modelling the chemical concentration

It is assumed that the chemical concentration evolves according to the diffusion equation with constant diffusion coefficient and moving point sinks which model the degradation of the chemical by membrane-bound enzymes on each cell. The governing equation is therefore

$$\frac{\partial c}{\partial t} = D_c \frac{\partial^2 c}{\partial x^2} - \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{j=1}^{N_S} \gamma(c(x^{(j)}, t)) \exp\left(\frac{-(x - x^{(j)})^2}{2\sigma^2}\right), \tag{5.14}$$

$$c(x, 0) = c_0, \quad t > 0, \tag{5.15}$$

where D_c is the diffusion coefficient of the chemical, $x^{(j)}$ is the location of the jth cell, σ^2 is variance of the Gaussian degradation term, c_0 is the initial concentration and $\gamma(c(x^{(j)}, t))$ denotes the rate of decay of the chemical at the jth cell. We will assume that the strength of the cell degradation is given by the Michaelis-Menten form

$$\gamma(c(x^{(j)}, t)) = \frac{V_{max}\, c(x^{(j)}, t)}{K_m + c(x^{(j)}, t)}, \tag{5.16}$$

where V_{max} is the maximum rate of decay and K_m is the Michaelis-Menten constant.

5.5 Experimental quantities and model parameters

From the data collected by Tweedy *et al.* [6] we can obtain estimates for the experimental quantities. The dataset contains the coordinates of a group of *Dictyostelium discoideum* cells moving by self-generated gradients under-agarose of length $L = 2500\,\mu\text{m}$ for a time of $T = 5.5\,\text{h} = 19800\,\text{s}$. Initially, there was a uniform amount of folate of concentration $c_0 = 10\,\mu\text{M}$ that covered the entire chamber. Many of the parameter values can be retrieved from the literature. The chemotactic velocity parameter ν does not have an equivalent literature value. This value controls how far along the domain the cells will travel. Therefore, we choose a value which allows the cells to move a similar distance as those from Tweedy *et al.* [6]. Although the diffusion coefficient for folic acid has been estimated in [107], this value is likely calculated for the diffusion of folic acid in solution, whereas we have folic acid diffusion in agarose gel. This would likely lead to a reduced effective diffusion rate which we use here. The parameter values that we will use for our numerical simulations are given in Table 5.1.

Parameter	Dimensional
D	$3\,\mu\mathrm{m}^2/\mathrm{s}$
K_d	$150\,\mathrm{nM}$ [108]
dl	$10\mu m$ [109]
ν	$31.57\,\mu\mathrm{m/s}$
D_c	$11.05\,\mu\mathrm{m}^2/\mathrm{s}$
V_{max}	$3\times10^{-2}\,\mathrm{nM/s}$
K_m	$5\,\mu\mathrm{M}$ [110]

Table 5.1: Nominal model parameter values for the simulation of *Dictyostelium discoideum* cells moving in response to a self-generated gradient in the chemoattractant folic acid.

5.6 Numerical discretisation

We will assume that there are $N_S = 100$ cells calculated at $N = 500$ time points, $t_n = (n-1)T/(N-1) = (n-1)\Delta t$, $n = 1,\ldots,N$. This will give a value for the time step of $\Delta t = 19800/499 = 39.68\,\mathrm{s}$. The position of the jth cell at the nth time point is given by $x_n^{(j)}$, $1 \le n \le N$, $1 \le j \le N_S$. Initially, the cells are given the position $x_1^{(j)} = 125v$, where v follows a standard uniform distribution $U(0,1)$. Note that this condition ensures that the cells begin in the small well.

The cells are moved by solving numerically the SDE (5.1) by the Euler-Maruyama method. This gives

$$x_{n+1}^{(j)} = x_n^{(j)} + \nu\,\mathrm{d}l \frac{K_d}{(K_d + c_n^{(j)})^2}\frac{\partial c_n}{\partial x}\Delta t + \sqrt{2D}\,\Delta W_n, \quad 1 \le n \le N, \quad 1 \le j \le N_S, \tag{5.17}$$

where $c_n^{(j)}$ is the chemical concentration evaluated at the location of the jth cell at the nth time point, $\partial c_n/\partial x$ is the chemical gradient evaluated at the location of the jth cell at the nth time point, and $\Delta W_n = W_{t_{n+1}} - W_{t_n}$ follows a normal distribution of the form $\mathcal{N}(0, \Delta t)$.

Notice that equation (5.17) depends on the concentration and gradient of the concentration for each cell over all time. To estimate these quantities, we will use an implicit-explicit finite difference scheme to numerically solve (5.14). To do this, we split the spatial domain into $N_X + 1$ points, $x^i = (i-1)L/N_X = (i-1)h$, for $i = 1, \dots, N_X + 1$. Then, denoting the approximation of the concentration at the point x^i at time point t_k by c_k^i, we look to solve

$$\begin{aligned}\frac{c_{k+1}^i - c_k^i}{\Delta t} &= D_c\left(\frac{c_{k+1}^{i+1} - 2c_{k+1}^i + c_{k+1}^{i-1}}{h^2}\right)\\ &\quad - \frac{1}{\sqrt{2\pi\sigma^2}}\sum_{j=1}^{N_S}\frac{V_{max}\, c(x_k^{(j)})}{K_m + c(x_k^{(j)})}\exp\left(\frac{-(x^i - x_k^{(j)})^2}{2\sigma^2}\right), \qquad (5.18)\end{aligned}$$

for c_{k+1}^i, along with an approximation of the boundary conditions that $\partial c/\partial x = 0$ at $x = 0$ and $x = L$ which gives $c_{k+1}^0 = c_{k+1}^2$ and $c_{k+1}^{N_X-1} = c_{k+1}^{N_X+1}$. The updated concentration c_{k+1}^i, $i = 1, \dots, N_X + 1$ can be obtained by solving a tri-diagonal system of equations. Once we have calculated the concentration at the $N_X + 1$ spatial points, we use linear interpolation to estimate the concentration at the location of the cells. Similarly, we use a linear approximation of the gradient of the concentration so that $\partial c/\partial x \approx (c_{k+1}^{i+1} - c_{k+1}^{i-1})/2h$ when $x = x^i$, and again use linear interpolation to estimate its value at the location of the cells. The same size of time step is used to solve (5.18) as is used to moved the cells in (5.17).

Once we have solved numerically equation (5.17), we must ensure that the cells remain in the simulated chamber by imposing appropriate boundary conditions. This is done by assuming that if $x_{n+1}^{(j)} < 0$ then $x_{n+1}^{(j)} = -x_{n+1}^{(j)}$, and if $x_{n+1}^{(j)} > L$ then $x_{n+1}^{(j)} = L - x_{n+1}^{(j)} \pmod{\mathrm{L}}$.

Figure 5.2 shows a progression of the location of the cells, chemical concentration profile and the cell location PDF at six equally spaced time points. The PDF for the cell density is obtained using Kernel Density Estimation with a normal kernel and optimal bandwidth for estimating normal distributions. Note that this is done using the default settings of the MATLAB function fitdist. We calculate the location of the cells by (5.17) and the chemical concentration by (5.18). We take $N_X = 1000$, giving a spatial grid size of $h = 2.5\,\mu$m for the implicit-explicit finite difference scheme. The parameters values for the simulations are taken from Table 5.1. We can see that the cells move from the small well to the right as expected. We see a leading wave of cells, a key property of self-generated gradient chemotaxis. In the paper from Tweedy *et al.* [6], they are able to measure the chemical concentration profile at a single time point corresponding to the end of the experiment. They find that the chemical concentration is high in front of the cell wave and quickly drops off to near zero concentration at the location of the wave. We see very similar results with our simulated concentration profiles. Finally, we find a single mode in the cell location PDF corresponding with the cell wave, whereas the experiments done by Tweedy *et al.* [6] find a bimodal distribution for the PDF. In their experiments, new cells continue to move into the chamber during the experiment, while in our simulated experiments, the number of cells in the chamber is constant from the start. We believe this is why we do not find a bimodal cell location PDF.

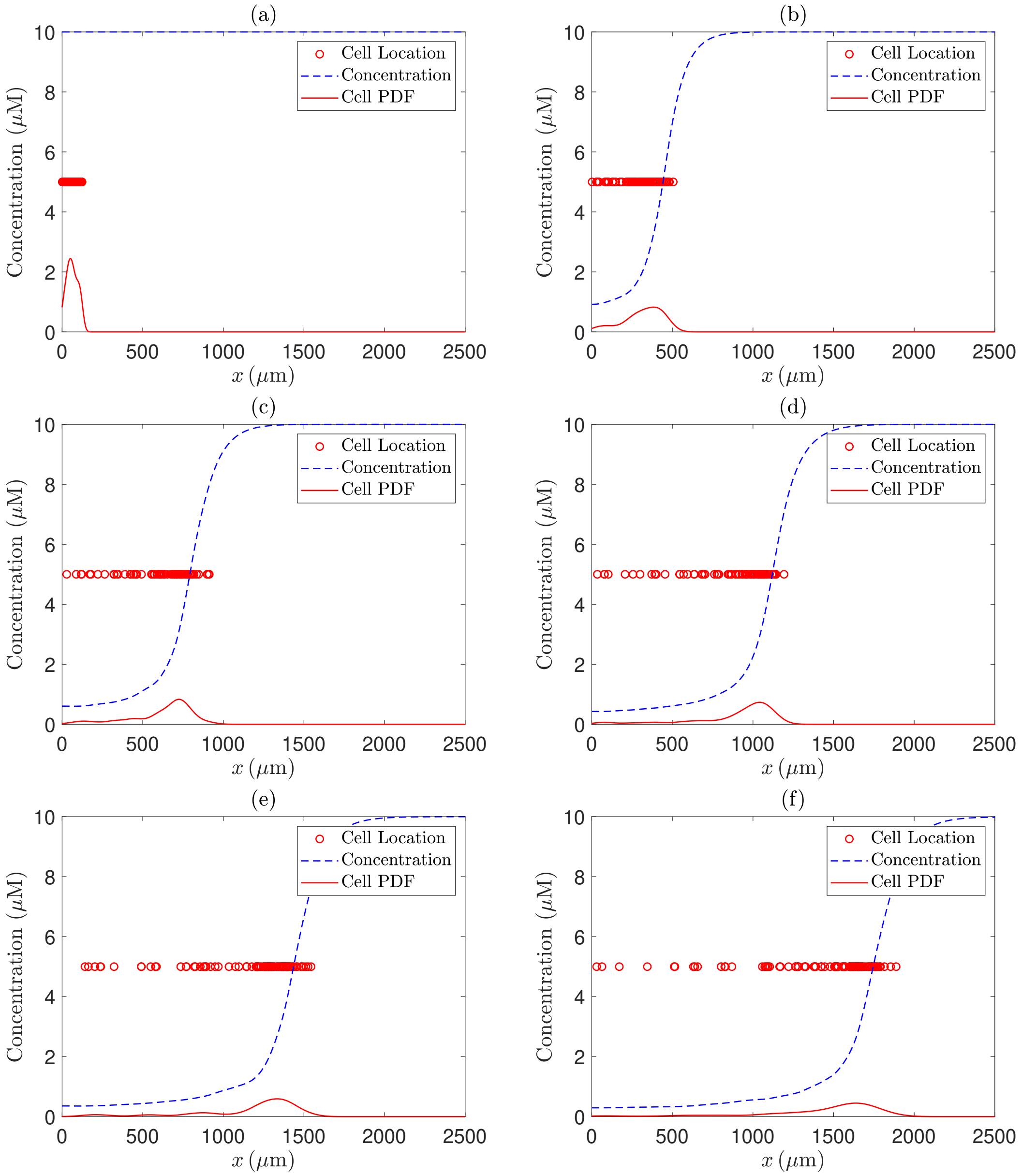

Figure 5.2: A plot of the cell locations (red circles), the cell concentration (dashed blue line) and the cell location PDF (solid red line) over time, where time progresses from (a) to (f). The parameter values which produced these plots are those from Table 5.1.

To test whether the time and space steps used in the Euler-Maruyama method and the implicit-explicit finite difference scheme give rise to accurate numerical approximations, we will repeat the simulations which led to Figure 5.2, but this time we will double N and N_x, which results in a halving of both the time step and the spatial grid size. The results are shown in Figure 5.3. To plotting accuracy, we find very similar results to Figure 5.2, suggesting that the original values give rise to accurate numerical approximations.

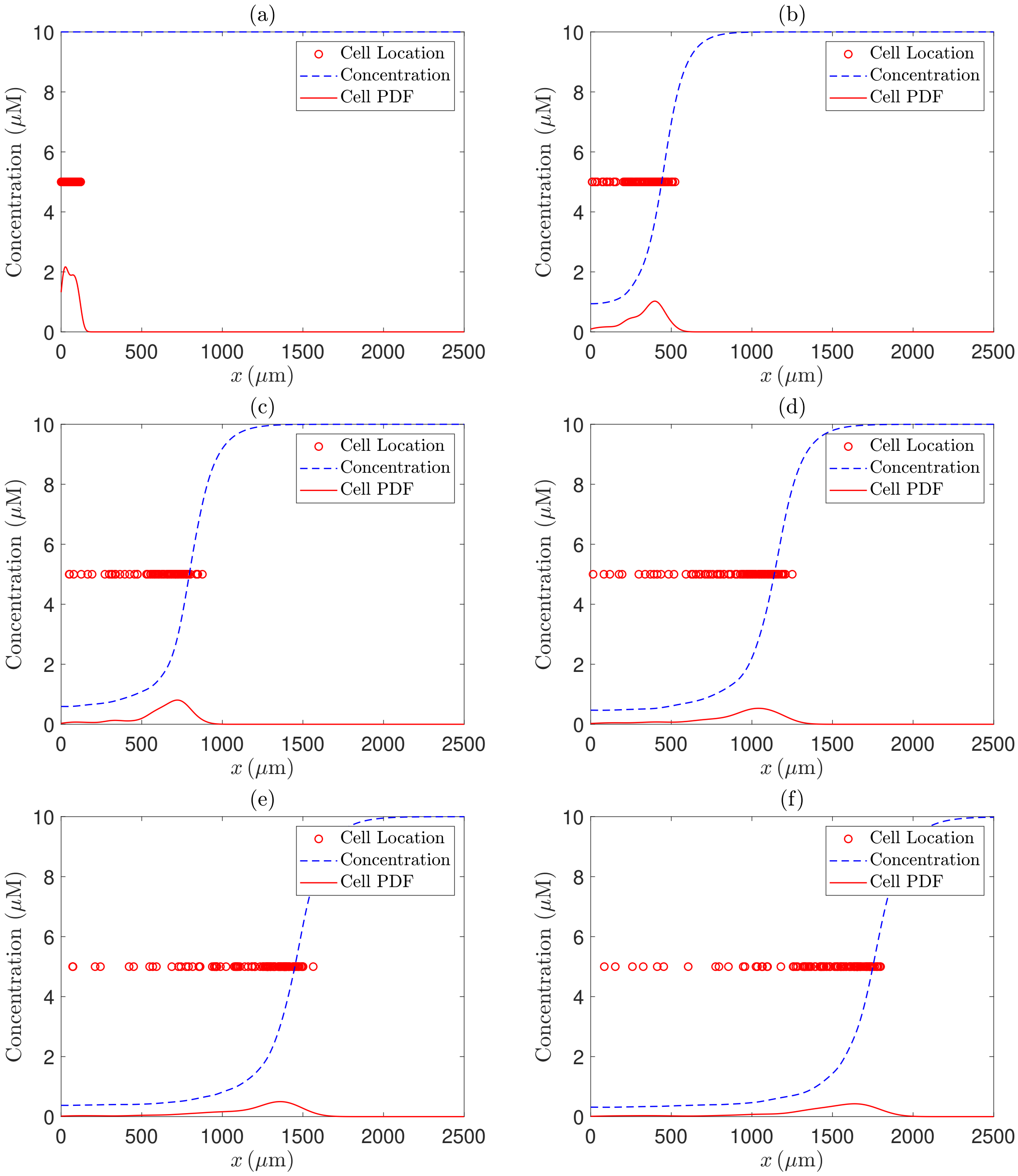

Figure 5.3: A plot of the cell locations (red circles), the cell concentration (dashed blue line) and the cell location PDF (solid red line) over time, where time progresses from (a) to (f). These plots use the same parameter values as those which produced Figure 5.2, but instead we doubled the values of N and N_x, which results in a halving of both the time step and the spatial grid size.

5.7 Sensitivity analysis

We now look to perform a sensitivity analysis to see how the output from our hybrid discrete-continuum model is affected by the input parameters, here called input factors. For our model, we have six input factors: ν, K_d, D, D_c, V_{max} and K_m. We initially considered the use of Monte Carlo techniques to estimate the main and total Sobol indices for each of the model parameters. The number of Monte Carlo samples required to estimate the Sobol indices accurately is often of the order of 10^3 and with each sample requiring a full solution of the numerical model, the cost of this approach was prohibitively high. Therefore, we chose to use a screening method instead. These have the advantage of determining the input factors which cause the biggest variation in the output from the model by using a small number of model evaluations. Screening methods are qualitative methods in that they ranks the input factors by their importance.

We will use the Morris screening method [111]. This method calculates so-called elementary effects for each input factor. It does this by numerically calculating the output at a number of points from each parameter's assumed distributions. To do this, points are sampled from the p-dimensional unit hypercube, where p is the number of input factors, and an inverse cumulative function is used to calculate the physical values for each parameter given the samples from the unit hypercube. Note that the output from the model used in the Morris method needs to be a scalar value. The outcome of the Morris method is a distribution of elementary effects for each parameter which can then be plotted. A larger mean and standard deviation in the distribution of elementary effects indicates that the input factor has a higher importance.

We will test the Morris method by using the ensemble time-average overlapping MSD as the output from our model. To do this, we calculate the time-average overlapping MSD at the $N - 1$ non-zero time points for each cell and then av-

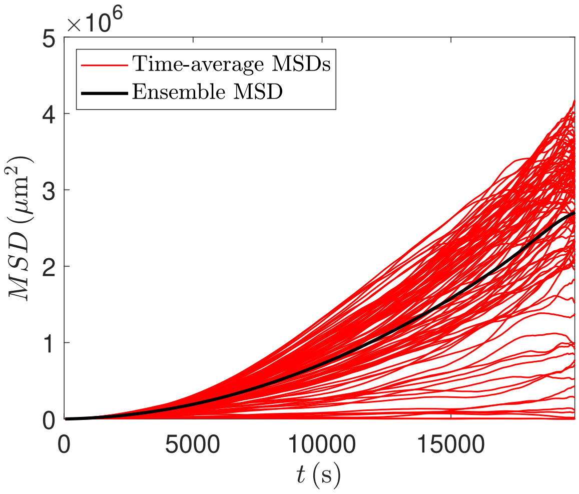

Figure 5.4: A plot of the time-average overlapping MSDs for each individual cell (red lines) and the ensemble MSD (dashed black line) using the parameter values from Table 5.1.

erage over all N_S cells to obtain the ensemble MSD. A plot of the time-average MSDs for each individual cell along with the ensemble MSD is shown in Figure 5.4. It is important to note that there is variability in the individual MSD curves for this particular problem. This is due to some cells staying at the front of the propagating population, while others lose their chemotactic response as they get left behind the travelling wave. Since the output from the model for the Morris method needs to be a scalar value, we will take the value of the ensemble MSD at a single time point as our output. From the previous chapters, we learned that the time over which the MSD has been calculated has an effect on the quality of inference of the drift and diffusion coefficients from our SDE model. Therefore, we will see what effect using the MSD at different time points as output of the model has on the identifiability of the random motility of the cells D and the chemotactic velocity of the cells ν, which is a similar term to the drift velocity from previous chapters. We will assume that each parameter is sampled from a triangular distribution, as described in [112]. This distribution is characterised by three values (a, b, c), where b is the nominal value of the parameter, which we will take to be the values provided in Table 5.1, and a and c are the maximum and minimum values, respectively, which we take to be half and double the nominal value. The inverse of the cumulative distribution function for the triangular distribution was therefore implemented within the Morris method, also given in [112].

Figure 5.5 shows the distributions of the elementary effects for all six input factors using the ensemble MSD evaluated at the first, 10th, 15th and final time point (out of $N - 1 = 499$ time points) as the output from the model. We see that using the first time point results in D being the most identifiable parameter. As we increase the value of the time point, D become less identifiable, eventually becoming less identifiable than ν when using the 15th time point. When we use the final time point, D then becomes the least identifiable parameter. Although we only have a scalar output here, these results suggest that the time over which the MSD is calculated, or what fraction of MSD points are used, could have an effect on the identifiability of D and ν when using ABC.

5.8 ABC for self-generated gradient model

We will now consider the use ABC to try and infer the parameters D and ν. We have chosen to use the F&P Res approach, detailed in Section 4.7.4, since it was found to be the best approach of those tested in Chapter 4 to accurately infer the diffusion and drift coefficients simultaneously. Since we are trying to infer D and ν, we will keep the other model parameters fixed at their values which produced the results from Figure 5.2. We will take the prior distributions for D and ν to be the triangular distributions described in Section 5.7. For this, the nominal values for D and ν will be $3\,\mu\text{m}^2/\text{s}$ and $31.57\,\mu\text{m/s}$, with maximum and minimum values equal to double and half the nominal values, respectively. For the vector of summary statistics used during the initial Prangle ABC run, we

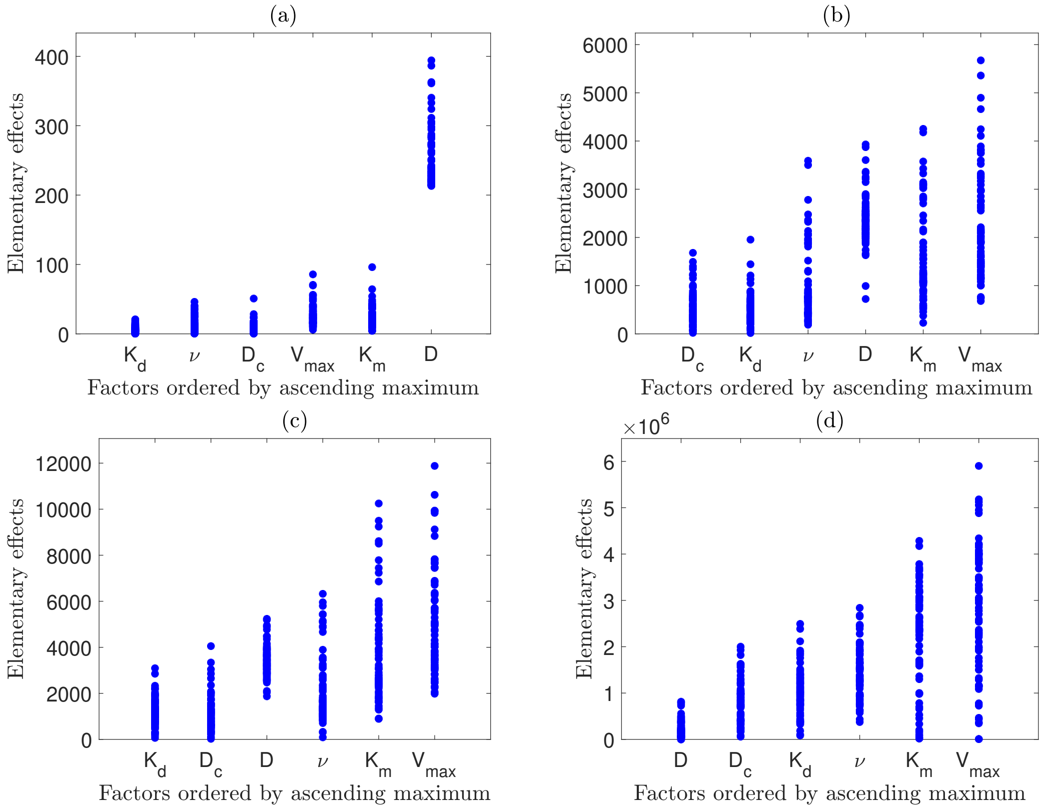

Figure 5.5: Distributions of the elementary effects from the Morris method for the input factors ν, K_d, D, D_c, V_{max} and K_m when using the MSD calculated at the first (a), 10th (b), 15th (c) and final (d) time point as the output from the model.

will use the ensemble MSD calculated at the $N - 1$ non-zero time points. The observed vector of summary statistics will be the ensemble MSD calculated using the nominal values from Table 5.1.

Samples from the joint posterior distributions for D and ν for 8 iterations of the F&P Res approach are shown in Figure 5.6. The dashed lines correspond with the true values of D and ν. Although we do not have access to estimates of the exact joint posterior distributions, we find in the final joint posterior distribution that both D and ν are inferred relatively well. However, there appears to be a slight bias in the inference of D.

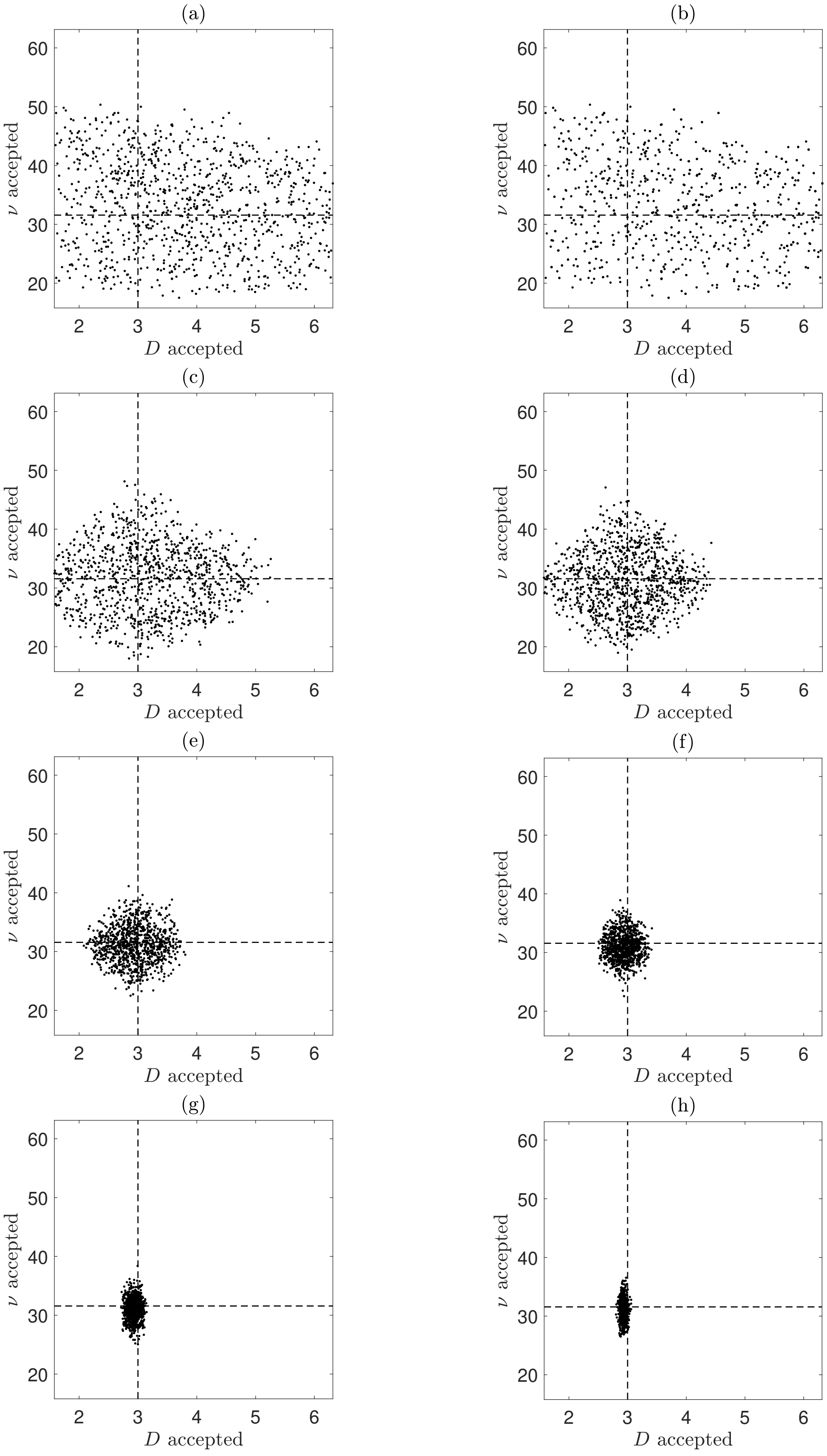

Figure 5.6: Samples from the joint posterior distributions for D and ν using the F&P Res algorithm. The value of ε decreases as we progress through (a) to (g). The dashed lines correspond with the true values of D and ν. These experiments were for $N_S = 100$, $N = 500$ and $T = 19800\,\mathrm{s}$, with the parameter values given in Table 5.1.

In the previous chapters we found that taking a smaller value of T generally improved the inference of the diffusion coefficient, while taking a larger value of T generally improves the inference of the drift velocity. Here, we will test re-running the ABC F&P Res approach where we calculate the ensemble MSD as before, but the summary statistics will be taken to be the first and final ten MSD points. We would expect using the first ten MSD points to improve the inference of D and the final ten MSD points to improve the inference of ν. Figure 5.7 shows a plot of the ABC F&P Res results when we use the first ten MSD points, all the MSD points (note that this is the same plot as Figure 5.6(h)), and the last ten MSD points as the summary statistics. We find that taking the first ten MSD points corrects the bias in the inference of D. When we use the final ten MSD points, D becomes unidentifiable as we would expect. For ν, however, the posterior distributions appear similar using the three different summary statistics and we do not find an improvement in the inference of ν when we use the final ten MSD points. This is likely due to the chemotactic term of our chemotaxis model being much more complex than the simple drift term considered in previous chapters.

5.9 Conclusions

In this chapter, we presented a hybrid discrete-continuum model describing self-generated cell chemotaxis and the diffusion of a local chemoattractant. The cells were moved according to the drift-diffusion SDE (5.1) and the evolution of the chemical was modelled using the PDE (5.14). Sensitivity analysis was performed using the Morris screening method to see how the identifiability of D and ν changes as we change the output function. Finally, the F&P Res ABC approach was used to try and infer the parameters D and ν.

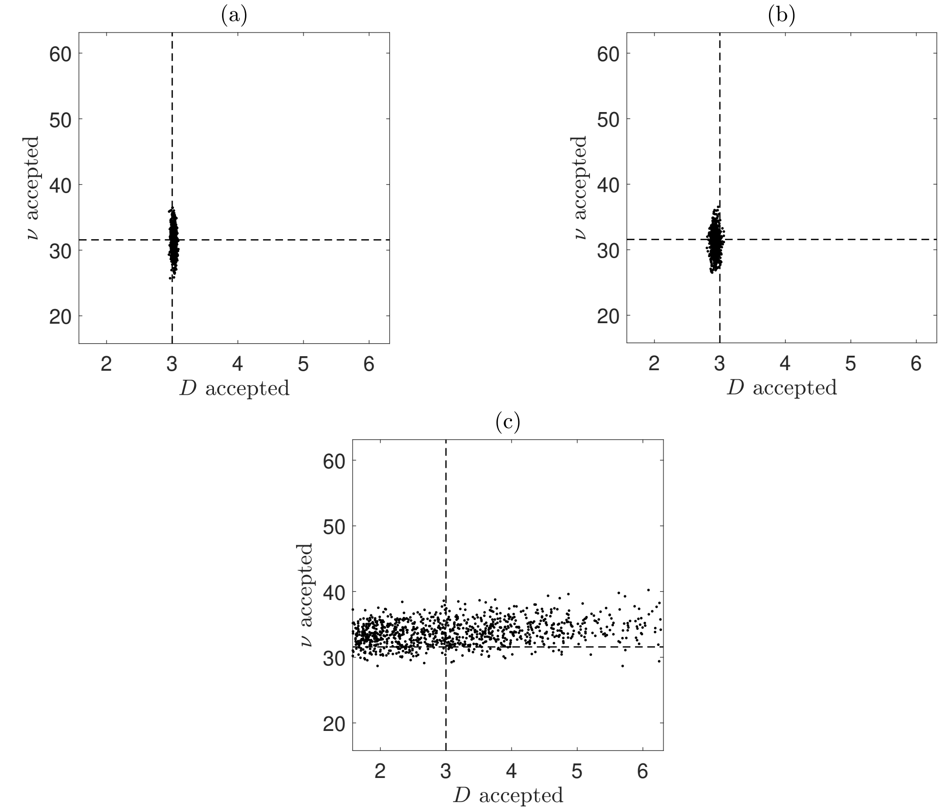

Figure 5.7: Samples from the joint posterior distributions for D and ν using the F&P Res algorithm where the summary statistics is taken to be the first ten MSD points (a), all the MSD points (b), and the final ten MSD points. The dashed lines correspond with the true values of D and ν. These experiments were for $N_S = 100$, $N = 500$ and $T = 19800\,\text{s}$, with the parameter values given in Table 5.1.

The drift term in the SDE was derived by considering receptor-ligand kinetics. By looking at the rate at which ligands bind on and off the cell receptors, assuming a constant total receptor concentration, we arrive at the chemotactic term given by (5.12). This term allowed cells to chemotax proportional to the chemical gradient when the concentration is small relative to the disassociation constant K_d and induced random cell movement when the concentration is large relative to the disassociation constant. The evolution of the chemoattractant was described using the diffusion equation with constant diffusion coefficient, along with a Gaussian-like term which modelled the degradation of the chemical by the cells. The strength of the cell degradation was assumed to have a Michaelis-Menten form.

To numerically simulate the movement of the cells and the evolution of the chemoattractant, experimental quantities for the self-generated gradient data were retrieved from Tweedy *et al.* [6] and values for the model parameters from the literature. We assumed that there were $N_S = 100$ cells calculated at $N = 500$ time points within a chamber of length $L = 2500\,\mu\text{m}$ for a time of $T = 5.5\,\text{h} = 19800\,\text{s}$. Initially, the cells were placed randomly in a small well of length $125\,\mu\text{m}$ at the left side of the chamber. The initial chemical concentration was $c_0 = 10\,\mu\text{M}$. To simulate the movement of the cells, we solved our drift-diffusion SDE numerically by the Euler-Maruyama method, as given by (5.17). The evolution of the chemoattractant was simulated on a uniform background grid using an implicit-explicit finite difference scheme to numerically solve (5.14). The updated concentration was then found by solving a tri-diagonal system of equations. Linear interpolation was used to estimate the chemical concentration at the location of the cells, as well as using a linear approximation for the chemical gradient at the location of the cells. The location of the cells, chemical concentration profile and the cell location PDF at six equally spaced time points was shown in Figure 5.2. Our simulations showed a leading wave of cells, a key property of self-generated gradient, and concentration profiles which matched that found in Tweedy *et al.* [6]. This showed that our drift-diffusion model could give rise to self-generated gradient chemotaxis.

We then performed a sensitivity analysis of our hybrid discrete-continuum model. Our model consisted of the six input factors ν, K_d, D, D_c, V_{max} and K_m. We initially investigated using the Sobol method but found that it was computationally expensive. Therefore, we chose to use the Morris screening method instead. This method calculates a distribution of elementary effects for each input factor. A larger mean and standard deviation in the distribution of the elementary effects indicates that the input factor has a higher importance. The output from the model used in the Morris method needs to be a scalar value. We tried using the value of the MSD at a single time point as the output from the model. Since we have previously observed that the time over which the MSD is calculated has an effect on the identifiability of the diffusion and drift coefficients of the cells, we investigated how choosing the MSD calculated at different time points affects the identifiability of the random motility D and the drift velocity ν of the cells. We found that taking the first time point results in D being the most identifiable parameter with ν being the second least identifiable. As we took the value of the MSD at later time points, the identifiablility of D decreases, until it becomes the least identifiable parameter when using the final time point. These results suggested that the time over which we calculate the MSD could have an impact on the inference of D and ν.

Finally, we used ABC to try and infer the parameters D and ν. We used the F&P Res approach as this was found to be the best approach of those tested in Chapter 4 for inferring both the diffusion and drift coefficients simultaneously. The prior distributions for D and ν were taken to be triangular distributions, with nominal values taken to be those that produced Figure 5.2, and maximum and minimum values taken to be double and half the nominal values, respectively. The summary statistic for the initial Prangle ABC run was taken to be the ensemble time-average overlapping MSD calculated at the $N-1$ non-zero time points. The observed summary statistic was also simulated from our model. We found that D and ν are inferred fairly well using this approach, as seen in Figure 5.6, but that there appears to be a bias in D. We also tried running ABC using the first and final ten MSD points as the summary statistics. We find taking the first ten MSD points corrects the bias and improves the inference in D, but all summary statistics infer ν with roughly the same accuracy. We believe this is due to the chemotactic term of our hybrid model being more complex than the drift term looked at in previous chapters.

Chapter 6

Conclusions and future work

6.1 Conclusions

In this thesis, we demonstrated that optimal estimation of the diffusion coefficient D and the drift velocity α from a drift-diffusion SDE model depends crucially on the time interval over which experiments are measured. Choosing too small a time interval results in poor inference of the drift velocity, while choosing too large a time interval results in poor inference of the diffusion coefficient. Additionally, we found that choosing a small time interval can result in the data being dominated by static localisation error, while choosing a large time interval results in poor estimates of the MSD as the variance of the MSD scales cubically in time. An intermediate time interval was found that balances the inference of both parameters when using regression on the MSD curve and when applying ABC procedures.

In Chapter 2, we used weighted least squares regression on the MSD to estimate the diffusion coefficient and the magnitude of the drift velocity. This required the calculation of an analytical form for the variance and covariance of the MSD. We showed that there exists an optimal number of regression points which minimises the error in the parameter estimates when experiments can be carried out only once. An optimal measurement time interval was also shown to exist when experiments were able to be repeated. Iterative procedures to calculate the optimal number of regression points and optimal measurement time interval have been presented and been shown to work well. Simulations show that the error in the parameter estimates can be reduced when we use the optimal number of regression points, and we find a dramatic reduction in the error when we use the optimal measurement time interval.

In Chapter 3, we give a review of different ABC procedures used for parameter inference. Two of these procedures are then used for parameter inference in the next chapter.

In Chapter 4, we compared the quality of the inference of the diffusion coefficient and the drift velocity. Exact joint posterior distributions were calulated using the likelihood function which were then compared with all ABC joint posterior distibutions calculated throughout the chapter. We initially tested the rejection and SMC algorithms described in Chapter 3. Again we found that the quality of the inference depended crucially on the time interval over which experiments are measured. Similar to the results in Chapter 2, we showed that too small a time interval results in wide posterior distributions in D, while too large a time interval gives wide posterior distributions in α. An intermediate time interval gave rise to more isotropic posterior distributions which balanced the uncertainty in both parameters. We showed that the SMC algorithm was more efficient than the rejection algorithm. The tested the Prangle approach [59], which scales the MSD by its MAD to normalise the summary statistics, and the Fearnhead & Prangle approach [60] (referred to as the F&P MVLR approach), which attempts to calculate more informative summary statistics, to see whether they could improve the joint posterior distributions. Two different accuracies – the harmonic and circular accuracy – were used to compare different ABC approaches. The harmonic accuracy is predominately determined by the change in the strongly identifiable parameter, while a higher value of the circular accuracy corresponds with better inferring both parameters simultaneously. We found that both approaches improved the inference of both parameters, but resulted in the uncertainty in the parameters being underestimated. Comparing both accuracies, the best approach depends on the value of T. The Prangle approach had a higher harmonic accuracy for a small and middle value of T, while for the same values of T the F&P MVLR approach was better. For the rest of the chapter, we focused on comparing the two approaches with adaptations of the Fearnhead & Prangle approach. Two adaptations used GPs with the squared exponential and the ARD squared exponential kernels in place of multivariate linear regression, referred to as the F&P GP SE and F&P GP ARDSE approaches, respectively. The F&P GP approaches gave rise to bimodal joint posterior distributions which were caused by the perturbed parameter samples from ABC going outside the GP training region. Restricting the sampling region to the complex hull of the GP training region fixed this bimodality issue. However, it is not clear whether using complex hulls would result in a bias in the joint posterior distributions. Therefore, we considered a final approach which attempted to circumvent the issues with using convex hulls but still allow us to make use of GPs. For this, we used GPs with the squared exponential kernel on the residuals retrieved from the multivariate linear regression method, referred to as the F&P Res approach. We found that the F&P Res approach gave similar results to the F&P GP approaches using convex hulls. Comparing all approaches, we found that the F&P Res approach had the highest circular accuracy for all values of T, suggesting that it is the best approach to infer both parameters simultaneously.

In Chapter 5, we presented a hybrid discrete-continuum model of a population of cells moving in response to a self-generated gradient. The movement of the cells was assumed to follow a drift-diffusion SDE, while the evolution of the chemoattractant was calculated using a PDE model. The drift term in the cell movement SDE was motivated by receptor-ligands kinetics, giving a drift velocity proportional to the chemical gradient when the chemical concentration is small and no chemotaxis when the chemical concentration is large. We assumed that the evolution of the chemical concentration follows the diffusion equation with the inclusion of a term which models the degradation of the chemical by the cells. The strength of the cell degradation term was assumed to have a Michaelis-Menten form. Experimental quantities were retrieved from *et al.* [6] and some model parameters were found by using literature values. We solved numerically the cell movement SDE by the Euler-Maruyama method and the chemical PDE by an implicit-explicit finite difference scheme. From this, we were able to simulate cells movement which showed characteristic properties of self-generated gradient movement. Further simulations provided evidence that the space and time steps used in the numerical schemes gave rise to accurate numerical approximations. We performed a sensitivity analysis of our hybrid discrete-continuum model. For this, we used the Morris screening method which ranks the input parameters, here called input factors, by their importance. The output from the model used within the Morris method needs to be a scalar value. Therefore, we chose to use the value of the ensemble MSD at a single time point. Since we saw in previous chapters that changing the time over which the MSD is calculated and using a subset of MSD points both affect the quality of the inference of the diffusion and drift coefficient, we tested how using the ensemble MSD at different time points affected the inference of the random motility of the cells D and the chemotactic velocity ν. We found that using the first time point resulted in D being the most identifiable parameter of the model, and that taking later time points resulted in D becoming less identifiable, eventually becoming the least identifiable when using the final time point. This suggested that the time over which the MSD is calculated or taking a subset of the MSD points used would both affect the quality of the inference of D and ν when using ABC. Finally, we performed the

F&P Res ABC approach to try and infer the parameters D and ν. We used the ensemble MSD calculated at all non-zero time points as the summary statistics. Both D and ν were inferred fairly well, although we found a slight bias in D. We also ran ABC using the first and final ten MSD points as our summary statistics. We found that taking the first ten MSD points corrects the bias in D, but there was no improvement in ν when using the final ten MSD points. We believe this is due to the chemotactic term of our hybrid model being more complex than the drift term looked at in previous chapters.

6.2 Future Work

We have shown that using the F&P Res approach with the MSD as the summary statistics can sufficiently infer the random motility of the cells D and the chemotactic velocity ν. An interesting further investigation would be to see how well we can infer the other model parameters using ABC. For example, from the sensitivity analysis performed in Chapter 5, we found that, excluding the first time point, the maximum rate of decay of the chemoattractant by the cells V_{max} was the most identifiable parameter for all outputs tested. Therefore, it would be interesting to see whether we can reasonably infer V_{max} along with D and ν.

It would also be interesting to investigate whether there are more informative summary statistics for the self-generated gradient data. From Tweedy *et al.* [6], we have access to experimental cell location data at given time intervals. To our knowledge, there is no way of obtaining experimental data for the spatio-temporal evolution of the concentration of the chemoattractant. However, the single chemical profile found by Tweedy *et al.* appears to be roughly sigmoidal in shape. Ferguson *et al.* [38, 40] performed inference on PDE equations modelling self-generated gradient movement using the data from Tweedy *et al.* [6] and found that assuming a sigmoid function form for the chemical concentration gave results that fit the data well. From the cell movement data, while we could calculate the ensemble MSD as was done in Chapter 5, we could also calculate the cell location PDF. The summary statistic for the ABC procedure could then be some appropriate norm of the difference between the experimental PDF and the PDF obtained from the numerical model. It is not known whether all the parameters can be accurately inferred using the PDF for the cell density or the MSD or a combination of both, for example, as in [97], as the summary statistic. There could be identifiability issues which could require either a different summary statistic or a combination of several statistics.

Daly *et al.* [113] investigated the identifiability of parameters from inference-based approaches on non-linear biological models. One of their investigations looked at using a Bayesian MCMC algorithm to infer the two parameters of a logistic growth model. They found that when time is small, one of the parameters becomes unidentifiable, while using a large time makes the other parameter unidentifiable. They found that an intermediate time exists where the joint posterior distribution of the parameters was curved but much more informative about both parameters. This reflects the results we found in Chapter 4. Interestingly, they looked at the effect of including a time point from the three different time scales and found that this gave the best results, with a more isotropic and narrower joint posterior distribution than any of the previous results. Another of their investigations looked at using an ABC SMC algorithm to infer the parameters of the Hodgkin-Huxley model. Using what they found to be the best three summary statistics, they showed that trying to infer the parameters from real data resulted in very wide posteriors, meaning that no parameters could be identified reasonably. They comment that in this case, the experimental data would need to collected in a more clever way, or a simplified model would need to be used.

Harrison and Baker [56] also considered the identifiability of parameters from a velocity jump process (VJP) model using particle Markov Chain Monte Carlo (pMCMC) and ABC methods. They examined the effect of changing the time step and the measurement noise (static localisation error) on the identifiability of a parameter of the VJP model and the measurement noise itself. When using the pMCMC algorithm, they found that increasing the time step worsens the inference of both parameters, with posteriors getting wider until they eventually become uniform over the whole prior. Increasing the measurement noise was also shown to create wider posteriors for the VJP model parameter, but didn't have much of an effect on the posteriors for the measurement noise. When using the ABC algorithm, they considered the effect of changing the time step and measurement noise, as well as using different summary statistics, on the identifiability of the same parameters. Regardless of the choice of summary statistics, they found that increasing the time step widened the posteriors to become uniform over the prior as before. The posteriors for the measurement noise also widen when increasing the time step, but even at the optimal time step are still poor. The choice of summary statistic was shown to have a dramatic effect on the inference of the VJP model parameter, but does not help with the poor inference of the measurement noise.

The papers by Daly *et al.* [113] and Harrison and Baker [56] highlight important issues on the identifiability of parameters. They reflect the importance of the experimental set-up to the inference of the parameters. When the experimental data is collected at inappropriate time measurements, the inference of certain parameters can become challenging. Notice that the sensitivity analysis from Chapter 5 suggests a similar finding. If the ensemble MSD is used as the summary statistic, we found that taking the measurement time interval to be small resulted in D being the most identifiable parameter, while a large measurement time interval resulted in D becoming the least identifiable parameter. A further investigation could be done for our self-generated gradient model to determine the best experimental set-up to infer all the parameters accurately.

Additionally, the hybrid discrete-continuum model in Chapter 5 could be used to investigate the effect of cell heterogeneity. Although we assume throughout Chapter 5 that all the cell model parameters are constant, simulations could be performed where we consider the parameters as random variables which can be drawn from distributions covering physiologically relevant ranges. It may be important to consider cell heterogeneity in order to model certain experimental data. For example, in the papers by McLennan *et al.* [35, 36], they found that different cell populations were necessary to model the NC cell migration in a developing embryo. The inclusion of this type of modelling complexity could easily be accommodated in our hybrid discrete-continuum model to see the effect on the results.

Appendix A

MATLAB routine to calculate the optimal number of fitting points

Notice that the value of p_{opt} depends on the model parameters D, α and η, as well as the size of the time interval between frames Δt and the total number of time points N. Therefore, we provide a MATLAB routine which determines $p_{opt}(D, \alpha, \eta, \Delta t, N)$ given these input parameters.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                  %
% Code to calculate the optimal number of fitting points for the   %
% overlapping time-averaged mean-square displacement (MSD).        %
%                                                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% Input:    D - Estimate of the diffusion coefficient.
%          al - Estimate of the drift magnitude.
%         eta - Estimate of the static error.
%          dt - The time interval between frames.
%           N - The number of time points.
%
%
% Output:  popt - The optimal number of fitting points for the given
%                 parameter values.
%
%
% Notes -  This code works for the overlapping time-averaged MSD
%          calculated both from a single particle and an ensemble of
%          particles as they have the same optimal number of fitting
%          points.
%

function [popt] = optimal_preg(D,al,eta,dt,N)

T=N*dt;
alp=4*D*dt;
ep=4*eta^2;

%
% Calculate the variance of the MSD
%

for n=1:N
    t(n)=n*dt;
    K=N+1-n;
    if n<=K
        varmsd(n)=(n*(4*n^2*K+2*K-n^3+n)/(6*K^2))*(4*D*dt)^2+...
            8*al^2*D*(dt)^3*(n^3*(3*K*n+1-n^2)/(3*K^2));
        varmsd(n)=varmsd(n)+((8*eta^2)/(K^2))*((K-n)*(eta^2-...
            al^2*(n*dt)^2)+K*(al^2*(n*dt)^2+4*D*n*dt+2*eta^2));
    else
        varmsd(n)=((6*n^2*K-4*n*K^2+4*n+K^3-K)/(6*K))*((4*D*...
            dt)^2)+8*al^2*D*(dt)^3*(n^2*(3*n*K+1-K^2))/(3*K);
        varmsd(n)=varmsd(n)+8*eta^2*(al^2*(n*dt)^2+4*D*n*dt+...
            2*eta^2)/K;
    end
end

%
% Calculate the covariance of the MSD
%

for n=1:N
    for m=1:N
        K=N-n+1;
        P=N-m+1;
        if m+n<=N
            covarmsd(n,m)=((alp^2*n)/(6*K*P))*(-n^3-2*P*n^2+(1-...
                6*m^2+(6*N+6)*m)*n+2*P)+2*ep*n*alp/K+((8*al^2*...
                dt^3*m*n^2*D)/(3*K*P))*(-n^2+(-3m^2+3*m*(N+1)+...
                1))+((-n+2*P)*ep^2)/(2*K*P)+(2*al^2*dt^2*m*n^2*...
                ep)/(K*P);
        else
            covarmsd(n,m)=((alp^2)/(6*K))*(-m^3+(3+3*N-4*n)*m^2+...
                ((8+8*N)*n-2-3*N^2-6*N)*m-6*n^3+(6+6*N)*n^2-...
                (4*N^2+8*N)*n+N*(N+2)*(N+1))-...
                ((8*al^2*dt^3*m*n*D)/(3*K))*(m^2-2*(N+1)*m+...
                (3*n^2-3*(N+1)*n+N^2+2*N))+...
                (2*(al^2*dt^2*m*n+n*alp)*ep+ep^2/2)/K;
        end
    end
end
```

```
for n=1:N
    for m=1:n
        covarmsd(n,m)=0;
    end
end

%
% Calculate the variance of the regression coefficients
%

for p=3:N
    s0over=0;s1over=0;s2over=0;s3over=0;s4over=0;
    for n=1:p
        tn=(n*T/N);
        variover=varmsd(n);
        s0over=s0over+1/variover;
        s1over=s1over+tn/variover;
        s2over=s2over+tn^2/variover;
        s3over=s3over+tn^3/variover;
        s4over=s4over+tn^4/variover;
    end

    del_over=s0over*s2over*s4over-s0over*(s3over)^2-...
        (s1over)^2*s4over+2*s1over*s2over*s3over-(s2over)^3;

    vara_over(p)=0;
    varb_over(p)=0;
    varc_over(p)=0;

    for n=1:p
        tn=(n*T/N);
        variover=varmsd(n);
        da_over(n)=(s2over*s4over-s3over^2-s1over*s4over*tn+...
            s1over*s3over*tn^2+s2over*s3over*tn-...
            s2over^2*tn^2)/(del_over*variover);
        db_over(n)=(s0over*s4over*tn-s0over*s3over*tn^2-...
            s1over*s4over+s2over*s3over+s1over*s2over*tn^2-...
            s2over^2*tn)/(del_over*variover);
        dc_over(n)=(s0over*s2over*tn^2-s0over*s3over*tn-...
            s1over^2*tn^2+s1over*s2over*tn+s1over*s3over-...
            s2over^2)/(del_over*variover);
        vara_over(p)=vara_over(p)+variover*da_over(n)^2;
        varb_over(p)=varb_over(p)+variover*db_over(n)^2;
        varc_over(p)=varc_over(p)+variover*dc_over(n)^2;
        if n>1
            for m=1:n-1
                vara_over(p)=vara_over(p)+2*covarmsd(m,n)*...
                da_over(n)*da_over(m);
                varb_over(p)=varb_over(p)+2*covarmsd(m,n)*...
                db_over(n)*db_over(m);
                varc_over(p)=varc_over(p)+2*covarmsd(m,n)*...
                dc_over(n)*dc_over(m);
            end
        end
    end
end

%
% Calculate the optimal number of fitting points
%

[~,I]=min(sqrt(varb_over(3:N))/(4*D)+sqrt(varc_over(3:N))/(al*al));
popt=I+2;

end
```

Appendix B

MATLAB routine to calculate the optimal measurement time interval

Similar to the p_{opt} case, the value of T_{opt} depends on the model parameters D, α and η, as well the total number of time points N. Therefore, we again provide a MATLAB routine which determines $T_{opt}(D, \alpha, \eta, N)$ given these input parameters.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                  %
% Code to calculate the optimal measurement time interval for the  %
% overlapping time-averaged mean-square displacement (MSD).        %
%                                                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% Input:    D - Estimate of the diffusion coefficient.
%          al - Estimate of the drift magnitude.
%         eta - Estimate of the static error.
%          N  - The number of MSD points.
%
% Output:  Topt - An estimate of the optimal measurement time
%                 interval.
%
%
% Notes -  This code works for the overlapping time-averaged MSD
%          calculated both from a single particle and an ensemble
%          of particles as they have the same optimal measurement
%          time interval.
%

function [Topt] = optimal_over(D,al,eta,N)

NT=251;
Topt_init=100;

for i=1:NT
    T=Topt_init*(1.03)^(i-(NT+2)/2);
    time(i)=T;
    dt=T/N;
    alp=4*D*dt;
    ep=4*eta^2;

%
% Calculate the variance of the MSD
%

    for n=1:N
        K=N+1-n;
        if n<=K
            varmsd(n)=(n*(4*n^2*K+2*K-n^3+n)/(6*K^2))*(4*D*dt)^2+...
```

```
                8*al^2*D*(dt)^3*(n^3*(3*K*n+1-n^2)/(3*K^2));
            varmsd(n)=varmsd(n)+((8*eta^2)/(K^2))*((K-n)*(eta^2-...
                al^2*(n*dt)^2)+K*(al^2*(n*dt)^2+4*D*n*dt+2*eta^2));
        else
            varmsd(n)=((6*n^2*K-4*n*K^2+4*n+K^3-K)/(6*K))*((4*D*...
                dt)^2)++al^2*D*(dt)^3*(n^2*(3*n*K+1-K^2))/(3*K);
            varmsd(n)=varmsd(n)+8*eta^2*(al^2*(n*dt)^2+4*D*n*dt+...
                2*eta^2)/K;
        end
    end

%
% Calculate the covariance of the MSD
%

    for n=1:N
        for m=1:N
            K=N+1-n;
            P=N+1-m;
            if m+n<=N+1
                covarmsd(n,m)=((alp^2*n)/(6*K*P))*(-n^3-2*P*n^2+...
                    (1-6*m^2+(6*N+6)*m)*n+2*P)+2*ep*n*alp/K+...
                    ((8*al^2*dt^3*m*n^2*D)/(3*K*P))*(-n^2+3*...
                    (-m^2+(N+1)*m+1/3))+((-n+2*P)*ep^2)/(2*K*P)+...
                    (2*al^2*dt^2*m*n^2*ep)/(K*P);
            else
                covarmsd(n,m)=((alp^2)/(6*K))*(-m^3+(3+3*N-...
                    4*n)*m^2+((8+8*N)*n-2-3*N^2-6*N)*m-6*n^3+(6+...
                    6*N)*n^2-(4*N^2+8*N)*n+N*(N+2)*(N+1))-...
                    ((8*al^2*dt^3*m*n*D)/(3*K))*(m^2-2*(N+1)*m+...
                    (3*n^2-3*(N+1)*n+N^2+2*N))+...
                    (2*(al^2*dt^2*m*n+n*alp)*ep+ep^2/2)/K;
            end
        end
    end

    for n=1:N
        for m=1:n
            covarmsd(n,m)=0;
        end
    end

%
% Calculate the variance of the regression coefficients
%

    s0=0;s1=0;s2=0;s3=0;s4=0;
    for n=1:N
        tn=(n*T/N);
        vari=varmsd(n);
        s0=s0+1/vari;
        s1=s1+tn/vari;
        s2=s2+tn^2/vari;
        s3=s3+tn^3/vari;
        s4=s4+tn^4/vari;
    end

    del=s0*s2*s4-s0*(s3)^2-...
        (s1)^2*s4+2*s1*s2*s3-(s2)^3;

    var_a(i)=0;
    var_b(i)=0;
    var_c(i)=0;

    for n=1:N
        tn=(n*T/N);
        vari=varmsd(n);
        da(n)=(s2*s4-s3^2-s1*s4*tn+...
            s1*s3*tn^2+s2*s3*tn-s2^2*tn^2)/...
            (del*vari);
        db(n)=(s0*s4*tn-s0*s3*tn^2-s1*s4+...
            s2*s3+s1*s2*tn^2-s2^2*tn)/...
            (del*vari);
        dc(n)=(s0*s2*tn^2-s0*s3*tn-...
            s1^2*tn^2+s1*s2*tn+s1*s3-s2^2)/...
            (del*vari);
        var_a(i)=var_a(i)+vari*da(n)^2;
        var_b(i)=var_b(i)+vari*db(n)^2;
        var_c(i)=var_c(i)+vari*dc(n)^2;
        for m=1:n-1
            var_a(i)=var_a(i)+2*covarmsd(m,n)*da(n)*da(m);
            var_b(i)=var_b(i)+2*covarmsd(m,n)*db(n)*db(m);
            var_c(i)=var_c(i)+2*covarmsd(m,n)*dc(n)*dc(m);
        end
    end
end

%
% Calculate the optimal measurement time interval
%

[~,I]=min(sqrt(var_b)/(4*D)+sqrt(var_c)/(al^2));
Topt=time(I);

end
```

References

[1] L. Li, Y. He, M. Zhao, and J. Jiang, “Collective cell migration: Implications for wound healing and cancer invasion,” *Burns & Trauma*, vol. 1, no. 1, pp. 2321–3868, 2013.

[2] E. Scarpa and R. Mayor, “Collective cell migration in development,” *Journal of Cell Biology*, vol. 212, no. 2, pp. 143–155, 2016.

[3] S. De Oliveira, E. Rosowski, and A. Huttenlocher, “Neutrophil migration in infection and wound repair: going forward in reverse,” *Nature Reviews Immunology*, vol. 16, no. 6, p. 378, 2016.

[4] C. Stuelten, C. Parent, and D. Montell, “Cell motility in cancer invasion and metastasis: insights from simple model organisms,” *Nature Reviews Cancer*, vol. 18, no. 5, pp. 296–312, 2018.

[5] E. Roussos, J. Condeelis, and A. Patsialou, “Chemotaxis in cancer,” *Nature Reviews Cancer*, vol. 11, no. 8, pp. 573–587, 2011.

[6] L. Tweedy, D. Knecht, G. Mckay, and R. Insall, “Self-Generated Chemoattractant Gradients: Attractant Depletion Extends the Range and Robustness of Chemotaxis,” *PLoS Biology*, vol. 14, no. 3, p. e1002404, 2016.

[7] D. Bray, *Cell movements: from molecules to motility.* Garland Science, 2000.

[8] N. Andrew and R. Insall, "Chemotaxis in shallow gradients is mediated independently of PtdIns 3-kinase by biased choices between random protrusions," *Nature Cell Biology*, vol. 9, no. 2, pp. 193–200, 2007.

[9] R. Insall, "Understanding eukaryotic chemotaxis: a pseudopod-centred view," *Nature Reviews Molecular Cell Biology*, vol. 11, no. 6, pp. 453–458, 2010.

[10] P. van Haastert, "Chemotaxis: insights from the extending pseudopod," *Journal of Cell Science*, vol. 123, no. 18, pp. 3031–3037, 2010.

[11] C. Weijer, "Collective cell migration in development," *Journal of Cell Science*, vol. 122, no. 18, pp. 3215–3223, 2009.

[12] E. Donà, J. Barry, G. Valentin, C. Quirin, A. Khmelinskii, A. Kunze, S. Durdu, L. Newton, A. Fernandez-Minan, W. Huber, M. Knop, and D. Gilmour, "Directional tissue migration through a self-generated chemokine gradient," *Nature*, vol. 503, no. 7475, pp. 285–289, 2013.

[13] A. Muinonen-Martin, D. Veltman, G. Kalna, and R. Insall, "An Improved Chamber for Direct Visualisation of Chemotaxis," *PLoS One*, vol. 5, no. 12, p. e15309, 2010.

[14] A. Muinonen-Martin, O. Susanto, Q. Zhang, E. Smethurst, W. Faller, D. Veltman, G. Kalna, C. Lindsay, D. Bennett, O. Sansom, R. Herd, R. Jones, L. Machesky, M. Wakelam, D. Knecht, and R. Insall, "Melanoma Cells Break Down LPA to Establish Local Gradients That Drive Chemotactic Dispersal," *PLoS Biology*, vol. 12, no. 10, p. e1001966, 2014.

[15] L. Tweedy, P. Thomason, P. Paschke, K. Martin, L. Machesky, M. Zagnoni, and R. Insall, “Seeing around corners: Cells solve mazes and respond at a distance using attractant breakdown,” *Science*, vol. 369, no. 6507, 2020.

[16] L. Tweedy and R. Insall, “Self-Generated Gradients Yield Exceptionally Robust Steering Cues,” *Frontiers in Cell and Developmental Biology*, vol. 8, p. 133, 2020.

[17] C. Tomlin and J. Axelrod, “Biology by numbers: mathematical modelling in developmental biology,” *Nature Reviews Genetics*, vol. 8, no. 5, pp. 331–340, 2007.

[18] S. Motta and F. Pappalardo, “Mathematical modeling of biological systems,” *Briefings in Bioinformatics*, vol. 14, no. 4, pp. 411–422, 2013.

[19] S. Hori, L. Tong, S. Swaminathan, M. Liebersbach, J. Wang, S. Gambhir, and D. Felsher, “A mathematical model of tumor regression and recurrence after therapeutic oncogene inactivation,” *Scientific Reports*, vol. 11, no. 1, pp. 1–14, 2021.

[20] J. Osborne, A. Walter, S. Kershaw, G. Mirams, A. Fletcher, P. Pathmanathan, D. Gavaghan, O. Jensen, P. Maini, and H. Byrne, “A hybrid approach to multi-scale modelling of cancer,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1930, pp. 5013–5028, 2010.

[21] F. Spill, P. Guerrero, T. Alarcon, P. Maini, and H. Byrne, “Hybrid approaches for multiple-species stochastic reaction–diffusion models,” *Journal of Computational Physics*, vol. 299, pp. 429–445, 2015.

[22] J. Harrison and C. Yates, “A hybrid algorithm for coupling partial differential equation and compartment-based dynamics,” *Journal of The Royal Society Interface*, vol. 13, no. 122, p. 20160335, 2016.

[23] R. Bardini, G. Politano, A. Benso, and S. Di Carlo, “Multi-level and hybrid modelling approaches for systems biology,” *Computational and Structural Biotechnology Journal*, vol. 15, pp. 396–402, 2017.

[24] O. Purcell, B. Jain, J. Karr, M. Covert, and T. Lu, “Towards a whole-cell modeling approach for synthetic biology,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 23, no. 2, p. 025112, 2013.

[25] A. Babtie and M. Stumpf, “How to deal with parameters for whole-cell modelling,” *Journal of The Royal Society Interface*, vol. 14, no. 133, p. 20170237, 2017.

[26] N. Bhat and S. Balaji, “Whole-cell modeling and simulation: A brief survey,” *New Generation Computing*, pp. 1–23, 2019.

[27] M. Neilson, J. Mackenzie, S. Webb, and R. Insall, “Modeling Cell Movement and Chemotaxis Using Pseudopod-Based Feedback,” *SIAM Journal on Scientific Computing*, vol. 33, no. 3, pp. 1035–1057, 2011.

[28] H. Meinhardt, “Orientation of chemotactic cells and growth cones: models and mechanisms,” *Journal of Cell Science*, vol. 112, no. 17, pp. 2867–2874, 1999.

[29] M. Neilson, J. Mackenzie, S. Webb, and R. Insall, “Use of the parameterised finite element method to robustly and efficiently evolve the edge of a moving cell,” *Integrative Biology*, vol. 2, no. 11-12, pp. 687–695, 2010.

[30] M. Neilson, D. Veltman, P. van Haastert, S. Webb, J. Mackenzie, and R. Insall, "Chemotaxis: A Feedback-Based Computational Model Robustly Predicts Multiple Aspects of Real Cell Behaviour," *PLoS Biology*, vol. 9, no. 5, p. e1000618, 2011.

[31] G. MacDonald, J. Mackenzie, M. Nolan, and R. Insall, "A Computational Method for the Coupled Solution of Reaction–Diffusion Equations on Evolving Domains and Manifolds: Application to a Model of Cell Migration and Chemotaxis," *Journal of Computational Physics*, vol. 309, pp. 207–226, 2016.

[32] J. Mackenzie, C. Rowlatt, and R. Insall, "A Conservative Finite Element ALE Scheme for Mass-Conservative Reaction-Diffusion Equations on Evolving Two-Dimensional Domains," *SIAM Journal on Scientific Computing*, vol. 43, no. 1, pp. B132–B166, 2021.

[33] A. Buttenschön and L. Edelstein-Keshet, "Bridging from single to collective cell migration: A review of models and links to experiments," *PLoS Computational Biology*, vol. 16, no. 12, p. e1008411, 2020.

[34] R. Mayor and C. Carmona-Fontaine, "Keeping in touch with contact inhibition of locomotion," *Trends in Cell Biology*, vol. 20, no. 6, pp. 319–328, 2010.

[35] R. McLennan, L. Dyson, K. Prather, J. Morrison, R. Baker, P. Maini, and P. Kulesa, "Multiscale mechanisms of cell migration during development: theory and experiment," *Development*, vol. 139, no. 16, pp. 2935–2944, 2012.

[36] R. McLennan, L. Schumacher, J. Morrison, J. Teddy, D. Ridenour, A. Box, C. Semerad, H. Li, W. McDowell, D. Kay, P. Maini, R. Baker, and P. Kulesa, "Neural crest migration is driven by a few trailblazer cells with a unique molecular signature narrowly confined to the invasive front," *Development*, vol. 142, no. 11, pp. 2014–2025, 2015.

[37] H. Berg and E. Purcell, "Physics of chemoreception," *Biophysical Journal*, vol. 20, no. 2, pp. 193–219, 1977.

[38] 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*, vol. 13, no. 123, 2016.

[39] J. Nardini, R. Baker, M. Simpson, and K. Flores, "Learning differential equation models from stochastic agent-based model simulations," *Journal of the Royal Society Interface*, vol. 18, no. 176, p. 20200987, 2021.

[40] E. Ferguson, J. Matthiopoulos, R. Insall, and D. Husmeier, "Statistical inference of the mechanisms driving collective cell movement," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 66, no. 4, pp. 869–890, 2017.

[41] B. Hu, D. Fuller, W. Loomis, H. Levine, and W. Rappel, "Phenomenological approach to eukaryotic chemotactic efficiency," *Physical Review E*, vol. 81, no. 3, p. 031906, 2010.

[42] C. Shi, C. Huang, P. Devreotes, and P. Iglesias, "Interaction of Motility, Directional Sensing, and Polarity Modules Recreates the Behaviors of Chemotaxing Cells," *PLoS Computational Biology*, vol. 9, no. 7, p. e1003122, 2013.

[43] M. Tang, M. Wang, C. Shi, P. Iglesias, P. Devreotes, and C. Huang, "Evolutionarily conserved coupling of adaptive and excitable networks mediates eukaryotic chemotaxis," *Nature Communications*, vol. 5, no. 1, pp. 1–13, 2014.

[44] D. Giurghita and D. Husmeier, "Statistical modelling of cell movement," *Statistica Neerlandica*, vol. 72, no. 3, pp. 265–280, 2018.

[45] D. Wilkinson, "Bayesian methods in bioinformatics and computational systems biology," *Briefings in Bioinformatics*, vol. 8, no. 2, pp. 109–116, 2007.

[46] M. Secrier, T. Toni, and M. Stumpf, "The ABC of reverse engineering biological signalling systems," *Molecular BioSystems*, vol. 5, no. 12, pp. 1925–1935, 2009.

[47] G. Lillacci and M. Khammash, "Parameter Estimation and Model Selection in Computational Biology," *PLoS Computational Biology*, vol. 6, no. 3, p. e1000696, 2010.

[48] N. Pullen and R. Morris, "Bayesian Model Comparison and Parameter Inference in Systems Biology Using Nested Sampling," *PloS one*, vol. 9, no. 2, p. e88419, 2014.

[49] J. Pritchard, M. Seielstad, A. Perez-Lezaun, and M. Feldman, "Population growth of human Y chromosomes: a study of Y chromosome microsatellites," *Molecular Biology and Evolution*, vol. 16, no. 12, pp. 1791–1798, 1999.

[50] M. Beaumont, W. Zhang, and D. Balding, "Approximate Bayesian computation in population genetics," *Genetics*, vol. 162, no. 4, pp. 2025–2035, 2002.

[51] J. Lintusaari, M. Gutmann, R. Dutta, S. Kaski, and J. Corander, "Fundamentals and Recent Developments in Approximate Bayesian Computation," *Systematic Biology*, vol. 66, no. 1, pp. e66–e82, 2017.

[52] B. Lambert, A. MacLean, A. Fletcher, A. Combes, M. Little, and H. Byrne, "Bayesian inference of agent-based models: a tool for studying kidney branching morphogenesis," *Journal of Mathematical Biology*, vol. 76, no. 7, pp. 1673–1697, 2018.

[53] P. Jones, A. Sim, H. Taylor, L. Bugeon, M. Dallman, B. Pereira, M. Stumpf, and J. Liepe, "Inference of random walk models to describe leukocyte migration," *Physical Biology*, vol. 12, no. 6, p. 066001, 2015.

[54] R. Ross, R. Baker, A. Parker, M. Ford, R. Mort, and C. Yates, "Using approximate Bayesian computation to quantify cell–cell adhesion parameters in a cell migratory process," *NPJ Systems Biology and Applications*, vol. 3, no. 1, pp. 1–10, 2017.

[55] J. Kursawe, R. Baker, and A. Fletcher, "Approximate Bayesian computation reveals the importance of repeated measurements for parameterising cell-based models of growing tissues," *Journal of Theoretical Biology*, vol. 443, pp. 66–81, 2018.

[56] J. Harrison and R. Baker, "The impact of temporal sampling resolution on parameter inference for biological transport models," *PLoS Computational Biology*, vol. 14, no. 6, p. e1006235, 2018.

[57] A. Borowska, D. Giurghita, and D. Husmeier, "Gaussian process enhanced semi-automatic approximate Bayesian computation: parameter inference in a stochastic differential equation system for chemotaxis," *Journal of Computational Physics*, vol. 429, p. 109999, 2021.

[58] J. Devlin, D. Husmeier, and J. Mackenzie, “Optimal estimation of drift and diffusion coefficients in the presence of static localization error,” *Physical Review E*, vol. 100, no. 2, p. 022134, 2019.

[59] D. Prangle, “Adapting the ABC Distance Function,” *Bayesian Analysis*, vol. 12, no. 1, pp. 289–309, 2017.

[60] P. Fearnhead and D. Prangle, “Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 74, no. 3, pp. 419–474, 2012.

[61] H. Taylor, J. Liepe, C. Barthen, L. Bugeon, M. Huvet, P. Kirk, S. Brown, J. Lamb, M. Stumpf, and M. Dallman, “P38 and JNK have opposing effects on persistence of *in vivo* leukocyte migration in zebrafish,” *Immunology & Cell Biology*, vol. 91, no. 1, pp. 60–69, 2013.

[62] E. Banigan, T. Harris, D. Christian, C. Hunter, and A. Liu, “Heterogeneous CD8+ T Cell Migration in the Lymph Node in the Absence of Inflammation Revealed by Quantitative Migration Analysis,” *PLOS Computational Biology*, vol. 11, no. 2, p. e1004058, 2015.

[63] M. Hooten, D. Johnson, B. McClintock, and J. Morales, *Animal Movement: Statistical Models for Telemetry Data.* CRC Press, 2017.

[64] D. Helbing, A. Johansson, and H. Al-Abideen, “Dynamics of crowd disasters: An empirical study,” *Physical Review E*, vol. 75, no. 4, p. 046109, 2007.

[65] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, "Simulation of pedestrian dynamics using a two-dimensional cellular automaton," *Physica A: Statistical Mechanics and its Applications*, vol. 295, no. 3–4, pp. 507–525, 2001.

[66] M. Thai, W. Wu, and H. Xiong, *Big Data in Complex and Social Networks*. CRC Press, 2017.

[67] X. Michalet, "Mean square displacement analysis of single-particle trajectories with localization error: Brownian motion in an isotropic medium," *Physical Review E*, vol. 82, no. 4, p. 041914, 2010.

[68] T. Savin and P. Doyle, "Static and dynamic errors in particle tracking microrheology," *Biophysical Journal*, vol. 88, no. 1, pp. 623–638, 2005.

[69] C. Vestergaard, "Optimizing experimental parameters for tracking of diffusing particles," *Physical Review E*, vol. 94, no. 2, p. 022401, 2016.

[70] H. Qian, M. Sheetz, and E. Elson, "Single particle tracking. Analysis of diffusion and flow in two-dimensional systems," *Biophysical Journal*, vol. 60, no. 4, pp. 910–921, 1991.

[71] M. Saxton and K. Jacobson, "Single-particle tracking: applications to membrane dynamics," *Annual Review of Biophysics and Biomolecular Structure*, vol. 26, no. 1, pp. 373–399, 1997.

[72] M. Saxton, "Single-particle tracking: the distribution of diffusion coefficients," *Biophysical Journal*, vol. 72, no. 4, pp. 1744–1753, 1997.

[73] A. Berglund, "Statistics of camera-based single-particle tracking," *Physical Review E*, vol. 82, no. 1, p. 011917, 2010.

[74] E. Codling, M. Plank, and S. Benhamou, “Random walk models in biology,” *Journal of the Royal Society Interface*, vol. 5, no. 25, pp. 813–834, 2008.

[75] C. Bishop, *Pattern recognition and machine learning.* Springer, 2006.

[76] P. Bevington and D. Robinson, *Data Reduction and Error Analysis for the Physical Sciences.* McGraw-Hill, 2003.

[77] M. Goulian and S. Simon, “Tracking single proteins within cells,” *Biophysical Journal*, vol. 79, no. 4, pp. 2188–2198, 2000.

[78] N. Fisher, *Statistical analysis of circular data.* Cambridge University Press, 1993.

[79] K. Mardia and P. Jupp, *Directional statistics.* John Wiley & Sons, 2000.

[80] K. Murphy, *Machine Learning: A Probabilistic Perspective.* MIT press, 2012.

[81] S. Wood, “Statistical inference for noisy nonlinear ecological dynamic systems,” *Nature*, vol. 466, no. 7310, pp. 1102–1104, 2010.

[82] K. Cranmer, J. Brehmer, and G. Louppe, “The frontier of simulation-based inference,” *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30055–30062, 2020.

[83] D. Rubin, “Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician,” *The Annals of Statistics*, vol. 12, no. 4, pp. 1151–1172, 1984.

[84] S. Tavaré, D. Balding, R. Griffiths, and P. Donnelly, “Inferring Coalescence Times From DNA Sequence Data,” *Genetics*, vol. 145, no. 2, pp. 505–518, 1997.

[85] P. Joyce and P. Marjoram, "Approximately sufficient statistics and Bayesian computation," *Statistical Applications in Genetics and Molecular Biology*, vol. 7, no. 1, 2008.

[86] S. Kay, *Fundamentals of Statistical Processing, Volume I: Estimation Theory.* Prentice-Hall Inc, 1993.

[87] S. Sisson, Y. Fan, and M. Tanaka, "Sequential Monte Carlo without likelihoods," *Proceedings of the National Academy of Sciences*, vol. 104, no. 6, pp. 1760–1765, 2007.

[88] M. Beaumont, J. Cornuet, J. Marin, and C. Robert, "Adaptive approximate Bayesian computation," *Biometrika*, vol. 96, no. 4, pp. 983–990, 2009.

[89] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. Stumpf, "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems," *Journal of the Royal Society Interface*, vol. 6, no. 31, pp. 187–202, 2009.

[90] M. Beaumont, "Approximate Bayesian Computation in Evolution and Ecology," *Annual Review of Ecology, Evolution, and Systematics*, vol. 41, pp. 379–406, 2010.

[91] K. Csilléry, O. François, and M. Blum, "abc: an R package for approximate Bayesian computation (ABC)," *Methods in Ecology and Evolution*, vol. 3, no. 3, pp. 475–479, 2012.

[92] C. Drovandi and A. Pettitt, "Estimation of parameters for macroparasite population evolution using approximate Bayesian computation," *Biometrics*, vol. 67, no. 1, pp. 225–233, 2011.

[93] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré, "Markov chain Monte Carlo without likelihoods," *Proceedings of the National Academy of Sciences*, vol. 100, no. 26, pp. 15324–15328, 2003.

[94] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.

[95] J. Owen, D. Wilkinson, and C. Gillespie, "Likelihood free inference for Markov processes: a comparison," *Statistical Applications in Genetics and Molecular Biology*, vol. 14, no. 2, pp. 189–209, 2015.

[96] M. Blum, M. Nunes, D. Prangle, and S. Sisson, "A comparative review of dimension reduction methods in approximate Bayesian computation," *Statistical Science*, vol. 28, no. 2, pp. 189–208, 2013.

[97] J. Harrison and R. Baker, "An automatic adaptive method to combine summary statistics in approximate Bayesian computation," *PLoS One*, vol. 15, no. 8, p. e0236954, 2020.

[98] P. Green, K. Łatuszyński, M. Pereyra, and C. Robert, "Bayesian computation: a summary of the current state, and samples backwards and forwards," *Statistics and Computing*, vol. 25, no. 4, pp. 835–862, 2015.

[99] R. Izbicki, A. Lee, and T. Pospisil, "ABC–CDE: Toward approximate Bayesian computation with complex high-dimensional data and limited simulations," *Journal of Computational and Graphical Statistics*, vol. 28, no. 3, pp. 481–492, 2019.

[100] J. Žunić and K. Hirota, "Measuring shape circularity," in *Iberoamerican Congress on Pattern Recognition*, pp. 94–101, Springer, 2008.

[101] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain, “Gaussian processes for time-series modelling,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, p. 20110550, 2013.

[102] J. Brynjarsdóttir and A. O’Hagan, “Learning about physical parameters: the importance of model discrepancy,” *Inverse Problems*, vol. 30, no. 11, p. 114007, 2014.

[103] H. Fogler, *Elements of Chemical Reaction Engineering (5th ed.)*. Prentice Hall, 2016.

[104] T. Hillen and K. Painter, “A users guide to PDE models for chemotaxis,” *Journal of Mathematical Biology*, vol. 58, no. 1, pp. 183–217, 2009.

[105] L. Segel, “A theoretical study of receptor mechanisms in bacterial chemotaxis,” *SIAM Journal on Applied Mathematics*, vol. 32, no. 3, pp. 653–665, 1977.

[106] R. Tyson, S. Lubkin, and J. Murray, “A minimal mechanism for bacterial pattern formation,” *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 266, no. 1416, pp. 299–304, 1999.

[107] P. Kalimuthu and S. John, “Selective electrochemical sensor for folic acid at physiological pH using ultrathin electropolymerized film of functionalized thiadiazole modified glassy carbon electrode,” *Biosensors and Bioelectronics*, vol. 24, no. 12, pp. 3575–3580, 2009.

[108] B. Wurster and U. Butz, “Reversible binding of the chemoattractant folic acid to cells of *Dictyostelium discoideum*,” *European Journal of Biochemistry*, vol. 109, no. 2, pp. 613–618, 1980.

[109] F. Rivero, B. Köppel, B. Peracino, S. Bozzaro, F. Siegert, C. Weijer, M. Schleicher, R. Albrecht, and A. Noegel, “The role of the cortical cytoskeleton: F-actin crosslinking proteins protect against osmotic stress, ensure cell size, cell shape and motility, and contribute to phagocytosis and development,” *Journal of Cell Science*, vol. 109, no. 11, pp. 2679–2691, 1996.

[110] P. Kakebeeke, R. de Wit, and T. Konijn, “Folic Acid Deaminase Activity During Development in *Dictyostelium discoideum*,” *Journal of Bacteriology*, vol. 143, no. 1, pp. 307–312, 1980.

[111] H. Sohier, H. Piet-Lahanier, and J. Farges, “Analysis and optimization of an air-launch-to-orbit separation,” *Acta Astronautica*, vol. 108, pp. 18–29, 2015.

[112] P. Ekström, “Eikos: A Simulation Toolbox for Sensitivity Analysis,” *Masters Degree Project, Faculty of Science and Technology, Uppsala University*, 2005.

[113] A. Daly, D. Gavaghan, J. Cooper, and S. Tavener, “Inference-based assessment of parameter identifiability in nonlinear biological models,” *Journal of The Royal Society Interface*, vol. 15, no. 144, p. 20180318, 2018.