Challenges in modelling and control of biological systems



Yue Liu
Wolfson College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Hilary 2024

This thesis is dedicated to my parents

Acknowledgements

I would like to thank my supervisors, Prof. Ruth Baker and Prof. Philip Maini, for their patient support and guidance throughout my D.Phil., as well as warm encouragements and advice both technically and professionally.

I would like to thank other members of the Wolfson Centre for Mathematical Biology (WCMB) for creating and providing a welcoming research environment. I would like to especially thank my colleagues at the WCMB, Dr. Andrew Krause, Kara Allum, and Dr. Alex Browning, as well as members of the Baker group, for discussion and inputs on various topics, and Dr. Johannes Borgqvist for his recommendations on presentation.

I would like to thank Kevin Suh and Prof. Daniel Cohen of Princeton University for conducting the experiments and providing the data crucial for Chapter 2, as well as for their valuable inputs on the biological side of the project.

I would like to thank the community of Wolfson College for an enjoyable college experience. I would also like to thank the community of St. Hugh's College, where I had the privilege of teaching as a stipendiary lecturer for much of my D.Phil., for providing me with the opportunity to be a part of it, and for a warm and inclusive environment.

I would like to thank the Oxford University Walking Club, the Board Games Society, and the bell ringers at St. Giles' Church for the friendships and interesting experiences.

I acknowledge the support by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Postgraduate Scholarships – Doctoral program, reference number PGSD3-535584-2019, as well as a generous award by the Canadian Centennial Scholarship Foundation (CCSF). I would especially like to thank the volunteers at CCSF for supporting Canadian students like myself to pursue academic careers.

Abstract

The inherent complexity of biological systems presents numerous challenges in their study through mathematical models. This thesis seeks to address three of these challenges. Firstly, ensuring the practical parameter identifiability of models is crucial for understanding the biological systems being modelled. It can be challenging to determine when a model is practically identifiable and how to design an experiment to collect data to ensure the identifiability of a model. We investigate the impact of various aspects of data, model, and experimental design on practical identifiability in the context of partial differential equation models of cell invasion, using the profile likelihood method. Additionally, we explore ways to optimally design an experiment to ensure the data it provides are informative for parameter estimation. Secondly, Turing's diffusion-driven instability (DDI) is a mechanism believed to be responsible for the formation of stripe patterns in many areas of biology. The basic model tends to produce disorganised stripes, and cannot account for the aligned stripes observed in many instances in biology. We demonstrate that a phenomenon known as a wave of competency, in conjunction with a DDI, can produce aligned patterns, and its speed determines the direction of alignment. Thirdly, agent-based models (ABMs) are widely used in mathematical biology, but applying control to ABMs presents challenges, as traditional control theories are formulated in terms of differential equations. To address this, we propose a method of applying optimal control to ABMs that involves using machine learning to obtain a differential equation-based surrogate model for the ABM. The surrogate model captures key characteristics of the ABM, which can act as a surrogate for applying control theory. We demonstrate the potential of this approach by applying it to control a simple ABM of cell movement.

Contents

1 Introduction **1**

1.1 Parameter identifiability and model selection 2

1.2 Exploring system behaviours 5

1.3 Control 7

1.4 Thesis overview 8

2 Parameter identifiability for PDE models of cell invasion **10**

2.1 Introduction 10

2.1.1 Parameter identifiability 11

2.1.2 Model selection 14

2.1.3 Identifiability in the context of cell invasion 15

2.1.4 Outline 16

2.2 Cell invasion: models and experiments 17

2.2.1 Barrier assay 17

2.2.2 Image processing 19

2.2.3 Fisher–KPP model and its generalisations 19

2.2.4 Numerical methods for PDE simulation 22

2.3 Inference methods for identifiability analysis 23

2.3.1 Notations 23

2.3.2 Maximum likelihood 24

2.3.3 Profile likelihood 27

2.3.4 Bayesian inference 28

2.3.5 Information criterion for model selection 30

2.4 Results 30

2.4.1 Profile likelihoods can recover true parameter values for synthetic data . . 30

2.4.2 All four models are practically identifiable . . . 32
2.4.3 Complicated models are more sensitive to data processing . . . 34
2.4.4 Consistency of parameter estimates across experimental replicates indicates practical identifiability . . . 36
2.4.5 Model complexity is limited by data resolution . . . 42
2.4.6 Computational cost of profile likelihoods is lower than MCMC . . . 45
2.5 Discussion . . . 47
2.5.1 Future work . . . 53

3 Optimal experiment design for improving parameter identifiability 55
3.1 Introduction . . . 56
3.1.1 Design for parameter identifiability . . . 56
3.1.2 Design for model discrimination . . . 58
3.1.3 Aspects of experimental design . . . 59
3.1.4 Overview . . . 61
3.2 Mathematical models . . . 61
3.3 Optimal initial conditions for improving identifiability . . . 64
3.3.1 Logistic model . . . 65
3.3.2 Richards model . . . 68
3.4 Optimal initial conditions for model discrimination . . . 71
3.5 Optimal control for improving identifiability . . . 75
3.5.1 Using u_K as the control variable . . . 77
3.5.2 Using u_δ as the control variable . . . 83
3.5.3 Using u_r as the control variable . . . 83
3.6 Optimal control for model discrimination . . . 85
3.6.1 Applying PMP . . . 86
3.6.2 Numerical methods . . . 87
3.6.3 Logistic model with additive controls . . . 88
3.6.4 Logistic model with multiplicative controls . . . 92
3.6.5 Richards model . . . 93
3.7 Discussion . . . 95

4 Control of diffusion-driven pattern formation behind a wave of competency 99

4.1 Biological motivations of diffusion-driven pattern formation . . . 100

4.2 Models and methods . . . 104

4.2.1 Schnakenberg model . . . 106

4.2.2 CDIMA model . . . 107

4.2.3 Artificial noise . . . 108

4.3 Numerical simulations . . . 109

4.3.1 Simulations in one spatial dimension . . . 110

4.3.2 Simulations in two spatial dimensions . . . 115

4.3.3 Single stripes on a narrow two-dimensional domain . . . 122

4.3.4 Simulations on two-dimensional circular patterning subdomain . . . 125

4.3.5 Simulations with higher noise magnitude . . . 126

4.4 Discussion . . . 127

5 Controlling biological systems through machine learning 132

5.1 Agent-based modelling and control . . . 132

5.1.1 Surrogate model . . . 133

5.1.2 Equation learning . . . 135

5.1.3 Overview . . . 138

5.2 Learning surrogate model using specified summary statistics . . . 139

5.2.1 Notation . . . 139

5.2.2 SINDy with control . . . 140

5.3 Demonstrating the method on a simple ABM . . . 143

5.3.1 Learning the dynamics . . . 145

5.3.2 Applying control . . . 151

5.4 Further automating the equation learning method . . . 154

5.4.1 Learning to compute summary statistics . . . 155

5.4.2 Learning to express the Lagrangian in terms of the summary statistics . . . 157

5.4.3 Automatic feature discovery for selecting summary statistics . . . 158

5.5 Discussion . . . 160

6 Conclusions and future work **164**

6.1 Summary of contributions . . . 164

6.2 Future work . . . 167

A Numerical methods for PDE simulations **170**

A.1 Numerical scheme for Chapter 4 . . . 171

A.1.1 Verification of convergence . . . 171

A.2 Numerical scheme for Chapter 2 . . . 173

A.2.1 Verification of convergence . . . 174

B Methods for parameter inference and identifiability analysis **176**

B.1 Notations . . . 176

B.2 Maximum likelihood . . . 178

B.3 Profile likelihood . . . 179

B.4 Bayesian inference . . . 181

B.5 Conceptual comparison between the methods . . . 182

C Theories of optimal control **185**

C.1 Optimal control theory for ODEs . . . 185

C.2 Numerical optimal control . . . 188

D Technical description of the barrier assay experiments **192**

D.1 Cell culture . . . 192

D.2 Barrier assay . . . 192

D.3 Image analysis . . . 193

E Detailed inference results for cell invasion models **194**

E.1 Inference results for experimental datasets . . . 194

E.2 Inference results for synthetic dataset . . . 198

E.3 Profile likelihoods computed from the experimental datasets . . . 199

E.4 Bivariate profile likelihoods . . . 204

E.5 Profile likelihoods for down-sampled data . . . 206

F Structural identifiability analysis for the logistic and Richard models **207**

F.1 The notion of structural identifiability 207

F.2 Structural identifiability analysis with the Taylor series approach 208

G Computational considerations for optimal control problems **211**

G.1 Comparison of numerical algorithms 211

G.2 Multiple control variables 213

H Linear stability analysis for the Schnakenberg model of pattern formation **215**

Bibliography **217**

List of Figures

1.1 Illustration of identifiable and non-identifiable parameters 3

2.1 Illustration of the barrier assay experiments 18
2.2 Profile likelihoods for the synthetic dataset 31
2.3 Profile likelihoods for Experiment 1, full dataset 33
2.4 Profile likelihoods for Experiment 1, reduced dataset 35
2.5 Final state of the barrier assay experiments 37
2.6 Comparison of confidence intervals for the parameters 38
2.7 Bivariate profile likelihood function for the Porous Fisher model 39
2.8 Comparison of profile likelihood curves calculated using progressively down-sampled data 43
2.9 Widths of the confidence intervals as a function of the amount of available data . 44
2.10 Bivariate profile likelihood and MCMC samples 45
2.11 Residual analysis 49

3.1 Profile likelihoods for the logistic model computed using synthetic data with varying initial conditions 66
3.2 Illustration of different phases of the solution of the logistic model 67
3.3 Widths the the confidence intervals of the parameters in the logistic model as a function of initial condition 68
3.4 Profile likelihoods for the Richards model computed using synthetic data with varying initial conditions 69
3.5 Solutions of the Richards model with varying γ 70
3.6 Widths of confidence intervals of the parameters in the Richards model as a function of initial condition 71
3.7 Comparison between the unperturbed synthetic data, and the fitted logistic and Richards models 73

3.8 Difference between two models as a function of initial condition 74
3.9 Plot of model solutions highlighting the difference between two models 75
3.10 Profile likelihoods for logistic model with additive control with varying control . 79
3.11 Widths of confidence intervals for the logistic model with additive control u_K as univariate function of control parameters 80
3.12 Widths of confidence intervals for the logistic model with additive control u_K as bivariate function of τ_0 and τ 81
3.13 Local sensitivities of Richards model 82
3.14 Widths of confidence intervals for the logistic model with additive control u_r as function of single control parameters 84
3.15 Widths of confidence intervals for the logistic model with additive control u_r as bivariate function of τ_0 and τ 84
3.16 Optimal control for model discrimination, with additive u_K as control variable . 89
3.17 Comparison of continuous and bang-bang optimal controls for model discrimination 91
3.18 Optimal controls model discrimination, with additive u_r as control variable . . . 92
3.19 Optimal controls for model discrimination, with multiplicative control variables . 93
3.20 Optimal controls for model discrimination for Richards models 94

4.1 Example of fish with different pigmentation stripe orientations 102
4.2 Illustration of the effective patterning domain and wave of competency 105
4.3 Typical multi-spike solution of the Schnakenberg model 111
4.4 Solutions of the Schnakenberg and CDIMA models in one spatial dimension, without on-going noise 112
4.5 Solutions of the Schnakenberg and CDIMA models in one spatial dimension, with on-going noise 114
4.6 Steady state pattern of the Schnakenberg model in two spatial dimensions, without on-going noise 117
4.7 Steady state patterns produced by the Schnakenberg and CDIMA models, initialised at homogeneous steady state 118
4.8 Steady state patterns produced by the Schnakenberg and CDIMA models, initialised as horizontal stripes 120

4.9 Steady state pattern produced by the Schnakenberg model with very low wave speed . . . 121
4.10 Snapshot of horizontal stripes in the process of formation . . . 123
4.11 Simulations of the Schnakenberg model on a narrow patterning domain . . . 124
4.12 Steady state patterns on a circular domain . . . 126
4.13 Comparison of pattern formation with respect to varying level of noise . . . 127

5.1 Effects of applying smoothing to data . . . 149
5.2 Dynamics and trajectory predictions of the learned models . . . 151
5.3 Schematic of the proposed machine learning architecture . . . 159

A.1 Convergence verification for finite difference scheme for the Schnakenberg model 172
A.2 Convergence verification for finite difference scheme for the Standard Fisher model 175

E.1 Profile likelihoods for Experiment 2, full dataset . . . 199
E.2 Profile likelihoods for Experiment 2, reduced dataset . . . 199
E.3 Profile likelihoods for Experiment 3, full dataset . . . 200
E.4 Profile likelihoods for Experiment 3, reduced dataset . . . 200
E.5 Profile likelihoods for Experiment 4, full dataset . . . 201
E.6 Profile likelihoods for Experiment 4, reduced dataset . . . 201
E.7 Profile likelihoods for Experiment 5 . . . 202
E.8 Profile likelihoods for Experiment 6 . . . 202
E.9 Profile likelihoods for Experiment 7 . . . 203
E.10 Profile likelihoods for Experiment 8 . . . 203
E.11 Bivariate profile likelihoods for the Standard Fisher model . . . 204
E.12 Bivariate profile likelihoods for the Porous Fisher model . . . 204
E.13 Bivariate profile likelihoods for the Richards model . . . 205
E.14 Bivariate profile likelihoods for the Generalised Fisher model . . . 205
E.15 Profile likelihoods for the down-sampled datasets . . . 206

G.1 Comparison of numerical algorithms for optimal control problem . . . 213
G.2 Controls for model discrimination with two active control variables . . . 214

List of Tables

2.1 Summary of PDE models of cell invasion 21

3.1 Parameter estimates for the logistic model with varying initial conditions 66

4.1 Summary of the classes of behaviours identified for the Schnakenberg model . . . 113

5.1 Summary of synthetic datasets generated from ABM 147

5.2 Performance of the controls 153

E.1 Experiment 1, radially-averaged dataset 195

E.2 Experiment 1, full dataset 195

E.3 Experiment 2, radially-averaged dataset 196

E.4 Experiment 2, full dataset 196

E.5 Experiment 3, radially-averaged dataset 196

E.6 Experiment 3, full dataset 196

E.7 Experiment 4, radially-averaged dataset 197

E.8 Experiment 4, full dataset 197

E.9 Experiment 5, full dataset 197

E.10 Experiment 6, full dataset 197

E.11 Experiment 7, full dataset 197

E.12 Experiment 8, full dataset 197

E.13 Inference results for the synthetic dataset 198

Glossary of abbreviations

ABM	Agent-based model
AIC	Akaike information criterion
BIC	Bayesian information criterion
BINN	Biologically-informed neural networks
cdf	Cumulative distribution function
CDIMA	Chlorine dioxide–iodine–malonic acid
CNN	Convolutional neural networks
DDI	Diffusion-driven instability
FBS	Forward-backward sweep
FIM	Fisher information matrix
GE	Greedy elimination
HJB	Hamilton-Jacobi-Bellman
HSS	Homogeneous steady state
i.i.d.	Independently and identically distributed
IMEX	Implicit-explicit
IVP	Initial value problem
LASSO	Least absolute shrinkage and selection operator
MCMC	Markov Chain Monte Carlo
MDCK	Madin-Darby Canine Kidney
MH	Metropolis-Hastings
MLE	Maximum likelihood estimator
MPC	Model predictive control
ODE	Ordinary differential equation
PDE	Partial differential equation
pdf	Probability density function
PKPD	Pharmacokinetic-pharmacodynamic
PMP	Pontryagin's maximum principle
SDE	Stochastic differential equation
SINDy	Sparse identification of nonlinear dynamics

Chapter 1

Introduction

Broadly speaking, studies on the behaviour of biological systems typically focus on a series of three related aims: understanding, predicting, and controlling. The mechanistic approach towards understanding a system involves identifying and quantifying the underlying mechanisms that govern the behaviour and interactions of the components of the system, and therefore drive the overall behaviour. Once these mechanisms are adequately understood, it is often of interest to predict potential changes in system behaviour when one of these mechanisms is altered, or additional mechanisms or control inputs are introduced. Finally, once the effects of altering these mechanisms on system behaviour are well explored, they can be carefully manipulated to achieve a degree of control over the system, and guide the overall behaviour towards a certain desirable outcome. These three aims naturally form a sequence of dependencies, where achieving each subsequent aim is dependent on the results obtained from the previous investigation.

The main difficulty in achieving these aims arises from the fact that biological systems and processes tend to be highly complex, usually involving a large number of components that interact non-linearly, and in ways that are difficult to directly measure or manipulate in isolation. As a result, studying biological systems using experiments alone is difficult, and so mathematical models have been used in many applications to achieve the three central aims of understanding, predicting, and controlling. However, there are a number of challenges associated with each aim and, in this thesis, we seek to address several of them.

The following sections describe a typical pipeline from model building to control, and discuss the challenges that arise throughout the process. These challenges might not all arise for a given application and the type of modelling utilised. Therefore, for each challenge, we use a different biological application and modelling formalism, each chosen to best demonstrate the way these challenges arise, and how they can be tackled. Through tackling these challenges, the aim of this

thesis is to find ways to ensure that a mathematical model provides an accurate understanding of the underlying biological system, and to make it easier to apply control to a wide range of complex mathematical models for biological systems.

1.1 Parameter identifiability and model selection

A typical mechanistic modelling approach for studying a biological system begins with identifying the components of the system relevant to the behaviour of interest, as well as a set of mechanisms governing their behaviours. The complexity of biological systems necessitates simplification when building a mechanistic model, therefore some biological details are inevitably left out. Mechanistic models use mathematical formalisms to represent the various components and mechanisms in the system. Common types of models include ordinary differential equations (ODE), partial differential equations (PDE), and agent-based models (ABM), as well as stochastic (partial) differential equations (SDE/SPDE), and discrete difference equation models. The first three will be studied in this thesis. Regardless of the mathematical formalism chosen to represent the biological system, the proposed models usually contain a number of parameters, whose values are unknown and so need to be determined using experimental data.

The next step is to explore whether there exist sets of parameter values that enable the model to reproduce the observed experimental data to some reasonable degree of accuracy and, if they exist, whether we can confidently identify a set that "best" represents the true underlying dynamics. We can find candidates for such parameter values by performing model calibration, or model fitting, which involves finding an optimal parameter set that minimises a certain measure of the discrepancy between the model solution and data, for example, the least-squares error. If the optimal parameter set produces a model solution that does not capture important features of the data, then this hints at model mis-specification, which could mean that the descriptions of the mechanisms are wrongly encoded in the model, or that the model neglects some important biological details. This necessitates model refinement before proceeding further.

A different problem arises when we can have multiple parameter sets that lead to model solutions that closely approximate the data, which is a situation known as practical parameter non-identifiability. This issue can manifest in a few different forms, which we illustrate in Fig. 1.1 using univariate profile likelihood curves. Profile likelihoods are functions of the values of a particular parameter, and they are defined in more detail in Sec. 2.3.3. A parameter value with a higher profile likelihood means it is more likely that this value reflects the true parameter

value in the underlying biological system, and the global maximum of the profile likelihood curve coincides with the maximum likelihood estimate of the given parameter. Firstly, non-identifiability can manifest when there are two or more completely distinct parameter sets that produce similar dynamics to the data. Each of these distinct parameter sets would lead to an alternative description of the nature of the relevant mechanisms. Without another means to eliminate the alternative descriptions, we would not be able to decide which description is closer to the truth, therefore hindering our understanding of the system.

Second, the model might be insensitive to changes in the value of a parameter or a combination of parameters. This means that the model solution remains similar to the data as the insensitive parameters are varied within a certain range. This does not necessarily make a model "bad". In fact, it suggests that the model is robust against variability in the biological properties described by the parameter values. Biological systems usually exhibit a high degree of variability from one instance to another, and yet their qualitative behaviours can be surprisingly consistent. This means that robust models, whose behaviours are less sensitive to parameter values, may be more capable of explaining the observed qualitative behaviours, which can be desirable if we are focused on making predictions rather than reaching for a deeper understanding of the biology.

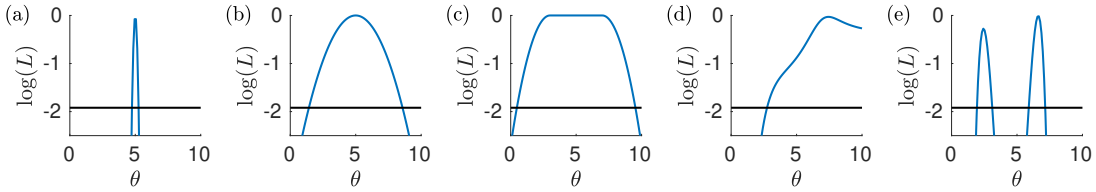


Figure 1.1: Illustration of identifiable parameters and various forms of non-identifiability. Here we present some examples of profile likelihood functions for a parameter θ, which will be defined in Section 2.3, and can be understood as the likelihood that the parameter θ takes a certain value. The 95% confidence region can be approximated as the set of points where the profile likelihood lies above -1.92, marked by the black horizontal line, as discussed in Section 2.3. The range of the x–axis (here $0 \leq \theta \leq 10$) for the profile likelihoods is chosen to display the full confidence interval whenever possible (i.e., when it is not infinite, as it is in case (d)). In this thesis, we consider a parameter to be practically identifiable if the profile likelihood is unimodal, and the confidence interval is finite, as illustrated in case (a). The parameter in (b) is still considered identifiable, but its value is less certain than case (a), since the confidence interval is wider. Cases (c)-(e) all represent non-identifiable parameters. In (c), the flat top of the profile likelihood curve indicates that the model solution is completely insensitive to changes in θ between $3 \leq \theta \leq 7$. (d) illustrates a case where the likelihood function tends to a value above the threshold of -1.92 as $\theta \to \infty$ (the plot is only shown up to $\theta = 10$), resulting in an infinitely wide confidence interval. Case (e) shows a multi-modal profile likelihood, meaning there are two distinct parameter values that can both reproduce the data well. Since we are not able to determine which of the two parameter values better reflects the underlying biology, the model is practically non-identifiable in this case.

However, if the purpose of the model is to understand or measure a particular mechanism, or if we are interested in controlling the system behaviour, then identifiability issues pose a significant challenge. For example, if a wide range of parameter values can reproduce the data equally well, then it is not possible to confidently determine the set of parameter values that best represents the underlying dynamics, and therefore precisely quantify the biological mechanisms driving system behaviour. Further down the line, this uncertainty in understanding poses challenges as we attempt to control the system, since in the presence of a control signal, the variability in parameter values may lead to very different system behaviour even if it does not in the absence of control.

Therefore, parameter non-identifiability poses a challenge to understanding the underlying mechanisms driving the behaviours in a biological system, and hinders our ability to effectively and confidently control the system. The first part of this thesis is focused on finding a way to detect identifiability issues in a model, and strategies to mitigate identifiability issues after they are identified. In Chapter 2, we will use collective cell invasion as an exemplar to illustrate how identifiability issues arise in PDE models, using experimental data in conjunction with several related models of varying complexity. The overall aim of Chapter 2 is to highlight the importance of practical parameter identifiability in understanding the underlying mechanisms driving the behaviours of a biological system, and identify the identifiability-related issues that need to be addressed at each stage of a modelling study, from experimental design and data collection, to model selection, and finally to interpreting the outputs of the model. We will demonstrate that the method of profile likelihoods provides an effective means to investigate practical parameter identifiability, and its performance compares favourably to Bayesian parameter inference, a widely used alternative method.

Our results show that parameter identifiability differs considerably between the four models we consider, which connects identifiability with the problem of model selection. When building a model, it is often not clear which mechanisms should be included in a model, and to what level of detail each mechanism should be modelled. This results in multiple competing models to select from. Traditionally, the primary criteria used for model selection are the complexity of the models, and their ability to recreate the data. Examples of such criteria includes the widely-used Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), as well as their lesser-known variants Deviance Information Criterion (DIC), Takeuchi information criterion (TIC), and Balanced BIC (BBIC) [1, 33, 48, 99]. Here, we argue that, in the case where

quantifying the mechanisms that drive the system behaviour is important, parameter identifiability should be considered as a factor of equal importance, in addition to model complexity and goodness-of-fit.

While the method of profile likelihoods has already been used in several studies to investigate parameter identifiability [175], the novelty of this work lies in the comparison between multiple PDE models, and the implications of parameter identifiability in model selection. Another novel contribution is the investigation into how a reduction in data resolution impacts parameter identifiability.

A notion related to parameter identifiability is model sensitivity and robustness, which refers to the sensitivity in model predictions or outputs with respect to changes in model inputs, or parameter values. Sensitivity analysis is widely employed in modelling studies to understand how uncertainties in parameter values contribute to uncertainties in model output. Some of the methods and applications of sensitivity analysis are reviewed in [77]. Sensitivity analysis is a useful tool for understanding and improving the accuracy of model predictions, but it is less relevant for ensuring that the model provides an accurate description of the underlying biological mechanisms, which is one of the aims of this thesis. For this reason, this thesis focuses on parameter identifiability instead of model sensitivity or prediction uncertainty. The connections between these two notions are discussed in Sec. 2.5.

One of the results in Chapter 2, which is published as [120], shows that practical parameter identifiability not only depends on the resolution of the data, but also on the initial conditions of the experiment that produced the data. This suggests another way to deal with non-identifiability issues, which is to design experiments that provide data in a form that is most useful for parameter identification. This possibility is investigated in Chapter 3, where the aim is to illustrate how experiments can be optimally designed to provide as much information as possible, thereby allowing us to study biological systems more efficiently.

1.2 Exploring system behaviours

Another important step in model analysis is to explore how the model solutions change when one or more parameters governing the mechanisms are varied. With the eventual goal of control in mind, through this exploration, we would like to identify mechanisms that show promise in modifying system behaviour in desirable ways, which are therefore good candidates to act as controls. Controls are variables in a system whose trajectories can be specified by the researcher,

rather than being determined by model dynamics, in contrast to the state variables, whose dynamics depend on the controls. By optimally designing a trajectory for the control variables, one hopes to be able to drive the system behaviour toward a desirable outcome, which will be the goal for the final step of the thesis. This goal is only achievable if the control represents a mechanism that can be easily modulated, and is capable of changing the system behaviour towards a desired outcome if it is modulated correctly. The first property can usually be determined easily by the nature of the application, but the second property requires a thorough exploration of the impact of the control on system behaviours.

The difficulty of the exploration varies greatly, depending on the model. In the simplest cases, the model might yield a closed-form solution. The exploration in this case can be carried out by analysing the solution directly. Unfortunately, this is not the case for most models with sufficient complexity to be of interest in biology. For models that are a little more complex, such as ODE models with a relatively small number of state variables, or simple PDE models in low dimensions, techniques such as bifurcation analysis with numerical continuation, and asymptotic analysis, are effective tools for guiding a systematic exploration of system behaviours. For more complicated systems, however, these techniques can be intractable.

In Chapter 4, we demonstrate the difficulty of exploring a complicated reaction-diffusion PDE model for pattern formation in two spatial dimensions. The candidate for a control mechanism is a phenomenon known as a "wave of competency", where the domain is divided into a region that is capable of supporting a pattern via DDI, and another region that is not, and the former expands behind a propagating wave front. The shape and speed of the wave of competency can be easily modulated, at least in the context of chemical pattern formation. To simplify the problem, we focus on the case where the shape of the wavefront is fixed to be a simple shape, and the speed of the wave is controlled by a single parameter. The aim of the chapter is to investigate the impact of the wave of competency on pattern formation, and whether it can act as a control mechanism for the purpose of inducing certain desirable patterns. At a higher level, the chapter aims to demonstrate the use of numerical analysis to explore a biological phenomenon, and provide qualitative and intuitive insight into the behaviour of a complex system.

The model is difficult to analyse with theoretical techniques. Nonetheless, we show that the exploration of model behaviours can be conducted by using a series of numerical simulations to identify distinct parameter regimes corresponding to different qualitative behaviours. We then investigate how the regimes are delineated with further numerical experimentation on a simplified

model. The results show an interesting connection between the wave of competency and the organisation and alignment of patterns, which paves the way for using a wave of competency as a control to create specific patterns. The results are published as [117].

1.3 Control

Once a validated and calibrated mathematical model for a biological system is established, along with an appropriate external stimulus that can serve as a control mechanism, we are ready to focus on devising a strategy to control the system. Of particular interest is optimal control, which entails designing a control input in an optimal manner to guide the system towards a desired behaviour. There are many biological systems where a certain behaviour is desirable. For example, consider a pharmacokinetic-pharmacodynamic (PKPD) model that describes the interaction between the body and drugs, with the drugs being the control mechanism. In this context, the aim is to find an optimal schedule and dosage for the drug to effectively treat the illness while minimising the dosage administered [10]. In another ecological example, given a model of population dynamics consisting of native species competing with invasive species, we might want to find the optimal way to intervene in order to manage the population of invasive species [92]. For ODE and PDE models, there are two classical optimal control techniques that can be applied to find the most efficient way to achieve desired system behaviours, which we will introduce below. However, a challenge arises in the case of ABMs, which represent a very large and diverse class of models used widely in mathematical biology, yet there have been few methods developed for controlling them. In Chapter 5, we outline the development of a method that combines optimal control techniques applicable to ODE models, and machine learning, to optimally control ABMs. The aim of the chapter is to pave the way towards a general and efficient method for controlling ABMs, which in turn makes it practical to apply optimal control to a much wider range of biological systems than previously possible. The method is based on using equation learning to automatically obtain an ODE model that approximates the behaviour of the ABM, using data generated from simulating the ABM. We then apply optimal control theory to this ABM, and transfer the resulting optimal control to the original ABM. This approach aims to minimise user effort while effectively capturing the underlying dynamics of the system. While the development of such a method is still in its early stages, we illustrate the capabilities of the preliminary method, and outline the direction for further refinement and development.

While our primary focus lies in applying optimal control to modulate the behaviour of biological systems, it serves as a versatile functional optimisation framework with potential applications beyond its traditional scope. In Chapter 3, we demonstrate how the framework of optimal control can be employed in experimental design. By applying external stimuli to the experimental system and observing its response, we gain valuable insights into the underlying mechanisms driving system behaviour. We can maximise the information we gain from the data collected from the experiment by optimally designing the stimuli.

We introduce the relevant optimal control theories in general terms, as well as numerical methods for applying these theories, in Appendix C. These theories and methods will be applied to specific problems in Chapter 3 and Chapter 5.

1.4 Thesis overview

Mechanistic modelling has enabled great advances in our understanding of biological systems, and our ability to make predictions about their behaviour. The logical next step is to use models to control systems. However, controlling biological systems presents significant challenges due to their inherent complexities. This thesis aims to identify some of these challenges, illustrate each in the context of a specific application, and develop effective techniques to overcome them. By addressing some of these challenges, this work aims to pave the way to make control of a broad range of biological systems possible.

Chapter 2 focuses on the challenge of practical parameter identifiability in PDE models for cell invasion. We investigate how various aspects of the data and model impact parameter identifiability, and how to incorporate parameter identifiability as a criterion in model selection. This work is published as Y. Liu et al., *Parameter identifiability and model selection for partial differential equation models of cell invasion*, in the Journal of The Royal Society Interface [120]. The experimental data are provided by collaborators, while the author of this thesis performed all analysis in the paper. This work contributes toward existing knowledge on practical identifiability by investigating the ways in which the complexity of PDE models can impact parameter identifiability, and quantifying identifiability as a function of data resolution.

The results from Chapter 2 motivate the idea to optimally design experiments to provide the data most useful for increasing identifiability, which we aim to do in Chapter 3. We develop experimental design techniques to optimise for parameter identifiability, as well as for distinguishing between similar models. These techniques could be useful for iteratively refining our

understanding of a biological system through further experiments and model adjustments. In certain cases, the cost of performing additional experiments can be expensive, and the cost of performing a series of them can be prohibitive, which makes it even more important to be able to learn as much as possible from each additional experiment.

Assuming that issues in the model selection and parameter inference stage have been resolved, and a suitable model has been selected along with a parameter set, Chapter 4 provides an example of exploring the behaviours of a complicated model, where we focus on exploring the spatio-temporal behaviour of a complex PDE model of pattern formation under the effect of a wave of competency. We found that the wave of competency can act as a control mechanism for the directionality and alignment of patterns, and is a promising candidate as a control variable for creating specific patterns. The research presented in this chapter is published as Y. Liu et al., *Control of diffusion-driven pattern formation behind a wave of competency* in Physica D: Nonlinear Phenomena [117], where the author of this thesis formulated the problem and carried out the analysis. This work contributes to the growing body of literature on pattern formation by demonstrating that the alignment of stripe patterns is robust against noise, and applies to multiple models. Furthermore, the study links the determination of stripe alignment to the natural rate of pattern propagation, and provides intuitive reasoning for the observations.

With a suitable model and control variable decided, the next chapter focuses on finding an optimal way to modulate the control variable for the purpose of guiding the system behaviour toward a desired behaviour. While mathematical techniques for doing so have been developed for differential equation-based models, they cannot be directly applied to ABMs. In Chapter 5, we outline a method for applying control to ABMs by combining PMP and machine learning. While the full method still requires further development, we demonstrate that a preliminary version of the method is capable of finding effective controls for a simple ABM.

Finally, in Chapter 6, we discuss the implications of the findings in each chapter, the open questions that remain, and the ways in which the work presented here can be extended. We also review the stated aims of each chapter, and discuss the progress made toward achieving these aims.

Chapter 2

Parameter identifiability for PDE models of cell invasion

When employing mechanistic models to study biological phenomena, practical parameter identifiability is important for making accurate predictions across a wide range of unseen scenarios, as well as for understanding the underlying mechanisms. In this chapter, we explore the problem of parameter identification for deterministic PDE models, and the implications of practical parameter identifiability for model selection. We use four extensions of the Fisher–KPP model and experimental data from a cell invasion assay to illustrate how identifiability problems can arise and manifest as a consequence of a combination of model complexity, experimental design, and data quantity. The data consist of high-resolution spatial profiles of cell density over the course of the experiment, while the models describes the dynamics of cell density. We find that more complicated models tend to be less identifiable, with parameter estimates being more sensitive to subtle differences in experimental procedures, and that they require more data to be practically identifiable. These results suggest that parameter identifiability should be considered alongside goodness-of-fit and model complexity as criteria for model selection. The work in this chapter is published as Y. Liu, K. Suh, P. K. Maini, D. J. Cohen and R. E. Baker, *Parameter identifiability and model selection for partial differential equation models of cell invasion* in the Journal of The Royal Society Interface [120]. The experimental data were provided by Suh and Cohen, while the author of this thesis performed all the analysis.

2.1 Introduction

PDE models have been widely employed across many areas of biology as a means to understand mechanisms driving observed behaviours, as well as for making predictions of future behaviours. The complexity of a typical biological system means that often it is not clear which mechanisms

should be incorporated into a PDE model, to what detail should they be described, and what functional form should they take. As a result, there can be multiple models proposed to describe the same system. For example, a variety of growth models have been proposed and adapted to describe the growth of tumours, coral reefs, and microbial proliferation [66, 145, 176], and their advantages and disadvantages have been extensively studied. To be able to both make accurate predictions for unseen scenarios and elucidate underlying mechanisms, it is crucial to be able to conduct model selection. The rival models considered in this chapter can be distinguished by their solutions. In cases where multiple different models are not distinguishable by their solutions, model selection can be more difficult. In such cases, it might be possible to modify the experiment conditions so that the models becomes distinguishable. This possibility is discussed in Ch. 3.

Criteria for model selection have mostly focused on balancing model complexity with the ability of a model to accurately reproduce experimental observations. Typical examples are the AIC and BIC [1, 33]. In this chapter, we argue that attention should also be paid to the issue of parameter identifiability in the process of model selection, which is ignored by information criteria. Parameter identifiability refers to the extent to which the parameters of a model can be accurately estimated from available data. Non-identifiable models can give rise to misleading conclusions about the nature of the underlying mechanisms, as well as inaccurate predictions. As such, it is important to ensure that model selection considers identifiability.

In this chapter, we will explore the connections between model selection and parameter identifiability through the use of *in vitro* experimental models of collective cell invasion, and four related candidate PDE models to describe them. We will use profile likelihoods for identifiability analysis, and investigate the impact of model complexity, data resolution, and experimental design, on parameter identifiability.

2.1.1 Parameter identifiability

Broadly speaking, in the context of PDE models, parameter identifiability refers to the ability to determine the set of parameter values in a model that best fit experimental data. There are two primary notions of identifiability found in the literature. The first notion is structural identifiability, which refers to the ability to uniquely determine the values of the model parameters given noise-free data of arbitrarily high resolution [205]. This is equivalent to ensuring that the PDE model does not yield identical solutions for distinct sets of parameter values. A

formal definition of structural identifiability is given in Appendix F.1. While several methods exist for determining structural identifiability for ODE models [41, 45, 155, 193], methods for PDE models are restricted to specific classes of models, such as age-structured PDEs [158] or systems of linear reaction-advection-diffusion equations [28].

This chapter, however, focuses on the more pragmatic notion of practical identifiability, which is more relevant to our greater-scope aim of controlling the biological system represented by the model. A model is considered practically identifiable if the parameters can be confidently identified using available data [205]. Practical identifiability is a stronger condition than structural identifiability, and its focus on the available data makes it more relevant for real-world applications. The main distinction between the two notions is that structural identifiability can be formulated as a property of the PDE and observation models alone, while practical identifiability is a property of the PDE model, the observation model, the error model, and data [155]. There are several different approaches to defining practical identifiability [205], and there is no universal consensus over the definitions in the literature. In this chapter, we define a parameter to be practically identifiable if the 95% confidence region obtained using profile likelihoods for this parameter is a finite, strict subset of the parameter's domain (the choice of domain is informed by biology, e.g., certain parameters might be restricted to be non-negative), and the profile likelihood function is uni-modal on this domain. We call the overall model practically identifiable if all of its parameters are practically identifiable. Furthermore, for parameters that satisfy the above definition, we call a parameter more (or less) identifiable than another parameter if its confidence interval is narrower (or wider) than the other parameter. Note that identifiability is dependent on the parameterisation of the model, and in some cases it might be possible to eliminate structural or practical non-identifiability by re-parameterising [86, 131]. In this chapter, for simplicity and to ensure interpretability of the biological mechanisms, we consider the models as they are posed, without re-parameterisation.

Parameter identifiability directly affects the reliability of model predictions and the ability of a mechanistic model to precisely pin down biological mechanisms, since non-identifiability may lead the modeller to place confidence in a set of parameter values that, while able to produce model solutions close to experimental data under the conditions in which the experiments were conducted, may lead to model predictions that diverge from the behaviour of the biological system under a different set of conditions. Moreover, for downstream applications involving controlling the biological system, a control strategy derived from a non-identifiable model with

uncertain parameter estimates may perform poorly on the original biological system. One such example is provided later in Sec. 5.3.2.

There are several possible sources of practical identifiability problems, assuming the model is structurally identifiable. First, stronger observational noise tends to lead to higher uncertainty in parameter estimates, leading to wider confidence intervals, and making it more difficult to reject a "wrong" parameter set that does not reflect the true dynamics. Sufficiently strong observation noise can therefore render a parameter non-identifiable. The second source is low data resolution. If the data resolution is not sufficiently high to resolve the finer differences between model solutions produced with different parameter sets, then it will not be possible to confidently identify the right parameter values. The third source is model mis-specification, where there is a discrepancy between the PDE model and the true model of the dynamics that generated the data. If the (noiseless) data cannot be exactly reproduced by the model with any parameter sets, but rather is similar to multiple model solutions produced with different parameter values, then it would not be possible to decide which of these parameter values better represent the underlying system. Fourth, for partially observable systems, where one or more variables in the model are not directly measured, there might be insufficient data to constrain parameter values. There are methods for identifying minimal sets of variables that need to be measured to ensure practical identifiability [5, 85]. These sets are known as minimal observation sets or minimal output sets. Determining the same for practical identifiability is more difficult. All of these sources are common in mathematical biology, where the data are noisy and limited in resolution, and the models are heavily simplified compared to the actual biological system.

As practical parameter identifiability is dependent on both the model and data, the investigation in this chapter will focus on both of these aspects. From a data-oriented perspective, we will investigate how the quality and quantity of data impact parameter identifiability. Specifically, we will explore the extent to which experimental design, data analysis and processing impact parameter identifiability. From a model-oriented perspective, we will explore the effects of model complexity on parameter identifiability, and the connections of these effects to overfitting or mis-specification. The results from these investigations will provide guidance for the design of experiments, as well as for data collection and processing procedures, for the purpose of improving identifiability.

2.1.2 Model selection

A problem closely linked to parameter identifiability is that of model selection. When we build a model for a given biological system, it is often difficult to determine the appropriate level of complexity, more specifically, which mechanisms to include, and to what level of detail, and which mechanisms to simplify or neglect. When the purpose of the model is to investigate a particular mechanism, it is sensible to choose a model that revolves around that mechanism [27], which provides a lower bound on complexity. Guidelines from [66] argue that, while phenomenological models are acceptable for making predictions, a model built for the purpose of understanding the biological system should focus on mechanisms that can be concretely derived from biology. This provides an upper bound on complexity, but in between, there remains room for choice.

Increasing the level of complexity of a model by either including additional mechanisms, or using more parameters to fine-tune the description of the existing mechanisms, will potentially result in a model that is capable of fitting the given data more accurately. The downside of a more complicated model, besides overfitting and making analysis more difficult, is that the model parameters often become less identifiable, since a change in one parameter can be more easily compensated for by changes in other parameters. This trade-off between complexity and identifiability leads us to consider the problem of model selection, where we aim to choose the most appropriate model from a collection of models of varying complexity.

Traditional tools used for addressing the dilemma between goodness-of-fit and model complexity include the AIC [99, Ch. 3] and BIC [99, Ch. 9], as well the lesser-known DIC, TIC, and Balanced BIC (BBIC) [1, 33, 48, 99]. These information criteria assign a score to models that rewards goodness-of-fit, while penalising for model complexity, and are used widely for model selection in mathematical biology, such as in modelling cell invasion [202], HIV infection [25], and ecological modelling [1]. A problem with these information criteria is that they do not involve parameter identifiability which, as we have argued, is important to the utility of a model. In this chapter, we explore the relation between complexity and identifiability in further detail, and argue that parameter identifiability should be used as an explicit criterion in model selection. We show that the model selected using the AIC or BIC might contain parameters that are harder to identify than parameters in competing models that score less well, making the overall model less identifiable. The issue is not simply that the AIC and BIC do not sufficiently penalise complexity, but rather, it is that they do not adequately quantify the relationship between complexity and identifiability. This leads to the proposal that identifiability analysis should be performed

prior to model selection, and that models that are identifiable given the data available should be favoured.

2.1.3 Identifiability in the context of cell invasion

Collective cell invasion plays a central role in development, wound healing, and cancer, and there has been considerable interest in understanding and quantifying the mechanisms that drive, promote, or hinder this phenomenon. In this work, we will use *in vitro* collective cell invasion as a prototypical biological system to explore the ways in which identifiability issues can arise, and how the profile likelihood method can be used to analyse them. The data we will use come from a suite of barrier assay experiments, and we will use the Fisher equation [59] and its generalisations as candidate models.

There is a long line of research on model fitting and parameter identification for cell invasion that leads up to this work. In an early work, Sherratt and Murray used the Fisher model, and its generalisation, the Porous Fisher model, to describe wound healing from a qualitative perspective [172]. These models were fitted to data from cell proliferation assays by Sengers et al. [170]. Later, Treloar and Simpson [184] and Simpson et al. [177] showed that the Fisher model can be fitted to cell invasion data, using only the information on the location of the edge of the cell population. In contrast, Jin et al. [83] fitted the Fisher and Porous Fisher models to cell invasion data consisting of the cell density profile. The paper employed the maximum likelihood method for parameter estimation, and goodness-of-fit as the main criterion for model selection, but it did not address parameter identifiability directly. In a later work, Vittadello et al. [195] used information on cell cycle dynamics, in addition to cell density data, to estimate model parameters. The same data were used by Simpson et al. [175] to explore parameter identifiability by quantifying the uncertainty in parameter estimates using both profile likelihoods and Bayesian inference.

Two classes of methods commonly used for parameter inference and identifiability analysis are profile likelihoods, which are the main tools of this chapter, and Markov Chain Monte Carlo (MCMC)-based methods for Bayesian inference, which aim to construct a posterior distribution for the parameters. MCMC methods have been widely used for inferring parameters in mathematical biology [55, 143, 127, 175]. These methods provide more detailed information on parameter values, but are more expensive to compute. On the other hand, a profile likelihood curve for a certain parameter is the likelihood function, conditional on the given data, with the

parameter of interest varying, and a maximum taken over the other parameters. The profile likelihood curve allows us to construct a confidence interval for that parameter, hence determining its identifiability. The profile likelihood method is used less widely than Bayesian inference, but its potentially lower computational cost makes it attractive. Raue et al. [154] discussed the application of profile likelihoods to detect non-identifiable parameters, illustrated with an example in the context of ODE models. This work continued in Raue et al. [156], which suggests a method that combines MCMC methods and profile likelihoods to inform data collection and iteratively refine parameter estimates. A comparison between the methods of MCMC and profile likelihoods was carried out in [194] for ODE models of varying complexity. Other methods for inference include gradient matching [123, 210] and approximate Bayesian computation [181]. The application of profile likelihoods in model selection was discussed by Simpson et al. [176], who used three different ODE models to describe coral growth, and discussed the importance of parameter identifiability in model selection. We will explain the details of profile likelihoods, maximum likelihood, and Bayesian inference applied to parameter identification in Sec. 2.3, and compare their computational cost in Sec. 2.4.6.

Model selection for cell invasion models has been discussed by Warne et al. [202], who used Bayesian methods for model fitting and information criteria for model selection. The authors emphasised the importance of model complexity as a factor in model selection, and identified the BIC as the best overall criterion. However, the authors realised that information criteria cannot account for all aspects of model comparison. Notably, the issue of parameter identifiability was not given much attention.

2.1.4 Outline

The contribution of this chapter is the investigation of parameter identifiability for multiple PDE models using profile likelihoods, and a discussion as to the implications of practical identifiability in model selection. The chapter also investigates the effects of data resolution and experimental design on parameter identifiability, showing that there is a model-dependent threshold for the amount of data required for the model to be practically identifiable, which tends to be higher for more complicated models. Lastly, we demonstrate a link between parameter identifiability and model robustness in terms of consistency in parameter estimates, which motivates a mixed effects view of the system. Under this view, each experimental replicate can be seen as taking

a sample of the parameter values from a certain distribution, which tends to be wider for less identifiable models.

The remainder of the chapter proceeds as follows. Section 2.2 introduces the biological background of cell invasion, the experimental procedure used to generate the data, the data processing procedures, and the PDE models for cell invasion. Then, Section 2.3 introduces the inference methods for quantifying parameter identifiability. The results are presented in Section 2.4, and we discuss their significance with respect to model selection and experimental design in Section 2.5.

2.2 Cell invasion: models and experiments

This chapter uses cell invasion as a concrete biological application to demonstrate the issues associated with parameter identifiability and model selection. This section introduces the biological background and motivation to study cell invasion, a plain language description of the barrier assay experiments (a technical description is given in Appendix D), the procedure for processing experimental data, and the mathematical models for cell invasion.

Biologically, cell invasion plays a central role in development, wound healing, and cancer invasion, so there has been considerable interest in understanding and quantifying the mechanisms that drive, promote or hinder this phenomenon. Mathematically, cell invasion has served as a prototypical problem for validating modelling techniques, since it can be adequately described with relatively simple models, and *in vitro* data can be readily obtained via experiments such as wound-healing assays or barrier assays.

2.2.1 Barrier assay

Barrier assays, also known as tissue expansion assays, are a way to observe cell invasion *in vitro*. Briefly speaking, a barrier assay involves preparing a plate with a barrier that encloses a central region, which cells cannot move in or out of. Cells are planted within the central region, and then given sufficient time to proliferate to form a monolayer within the central region that they are confined to. The barrier is removed at time $t = 0$, allowing the cells to propagate outward, and invade the rest of the plate, which is initially absent of cells.

The experimental data used in this chapter are kindly provided by Kevin Suh of Princeton University, who conducted the experiments. Eight experiments were done in total, with the first four (Experiment 1, 2, 3, 4) using a circular initial confinement region, and the later four

(Experiment 5, 6, 7, 8) using a triangular initial confinement region. The experiments are otherwise identical, and use Madin-Darby Canine Kidney (MDCK) cells. Ethical approval was not required for experimenting on this cell line. For each experiment, from $t = 0$, the time when the barrier was removed, an image of the plate was taken every $\Delta t = 1/3$ h, producing a series of $n_t = 77$ images for each experiment. Selected images from the experiments are shown in Fig. 2.1, along with corresponding schematic representations of the experiment, and snapshots of model solutions at the corresponding time point.

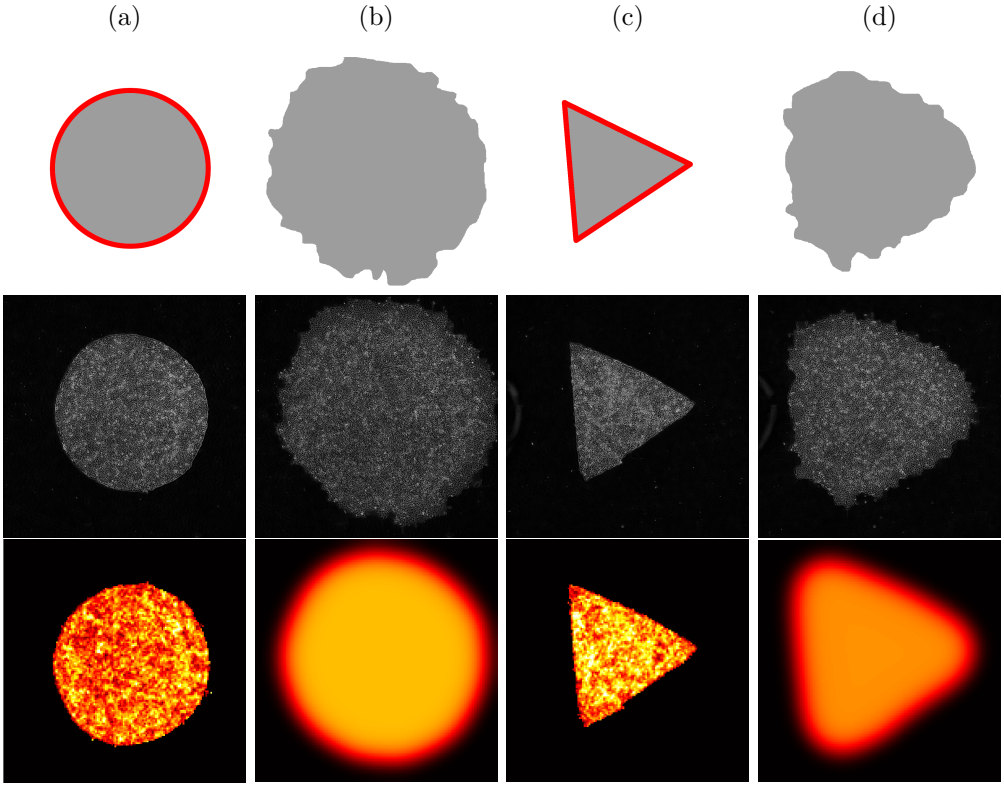


Figure 2.1: Illustration of various stages of the barrier assay experiments. In the first row are simplified representations of the cell culture, where the grey areas represent the regions covered by cells, and the red lines represent the barriers. The second row contains images taken from the experiments, kindly provided by Kevin Suh. The field of view of the camera is limited to a square region of size 4380 µm by 4380 µm in the centre of the plate, while the plate itself extends further beyond. In all experiments, the cells stay within the field of view, so edge effects are not important. The third row represents snapshots from simulations of the Standard Fisher model (see Table 2.1) with the parameters set to the maximum likelihood estimator (MLE) computed from the corresponding experiments. The colours represent cell density, with black indicating a complete absence of cells, while the cell density increases as the colour progresses from red to light yellow. (a) Initial condition of Experiment 1, where the initial confinement region is circular. The barrier is removed at $t = 0$ h. (b) The state of the experiment at $t = 20$ h. The cells have spread out to cover most of the field of view. (c-d) Shows the initial conditions at $t = 0$ h and a later stage $t = 20$ h of Experiment 5. The experiment is similar to Experiment 1, except with a triangular barrier.

2.2.2 Image processing

The images from the barrier assays are processed to obtain the cell density data to be used in calculations. First, the images are analysed with a tool based on convolutional neural networks (CNN) [106] to identify and locate individual cell positions. The field of view of the camera is subdivided into $n_x = 150$ times $n_y = 150$ square boxes, each of size $\Delta x = 58.4$ µm by $\Delta y = 58.4$ µm with 50% overlap between neighbouring boxes. A density profile is produced by counting the number of cells in each box, and dividing by the area. The density is presented as a rank three tensor $C_{\text{data}}(x_i, y_j, t_k)$, with $1 \le i \le n_x,\ 1 \le j \le n_y,\ 1 \le k \le n_t$. This density profile, along with the images, were provided by Kevin Suh, and are available on Zenodo at https://zenodo.org/records/10594283 [119].

In the case where the initial confinement region is circular, the data show that the cell density remains approximately radially symmetric throughout the time span of the experiment. The deviation from radial symmetry caused by stochastic effects is significant only near the end of the experiment. This symmetry provides an opportunity to obtain a reduced dataset of much smaller size, that retains most of the information in the original dataset. To do so, we define the radially averaged cell density, $\tilde{C}_{\text{data}}(\rho_l, t_k)$, as

$$\tilde{C}_{\text{data}}(\rho_l, t_k) = \text{mean}\left\{ C_{\text{data}}(x_i, y_j, t_k) |\ \rho_l - \frac{\Delta\rho}{2} < \sqrt{(x_i - c_x)^2 + (y_j - c_y)^2} \le \rho_l + \frac{\Delta\rho}{2} \right\}, \tag{2.1}$$

where (c_x, c_y) is the centre of the circular initial confinement region, ρ the radial distance from the centre, and $\rho_l = l\Delta\rho$, $l = 1, \ldots, n_\rho = 120$ are the mesh points in radial coordinates, with $\Delta\rho = 30$ µm. Parameter identification requires a large number of simulations of the models to be computed, and it is much faster to numerically simulate a model on the two-dimensional (ρ, t) domain than it is to simulate the equivalent model on a three-dimensional (x, y, t) domain. Therefore, parameter inference using $\tilde{C}_{\text{data}}$ instead of C_{data} will be much less computationally expensive. We will compare the results obtained from using full versus reduced datasets to see if this reduction has any effect on the estimates of parameter values and their uncertainties.

2.2.3 Fisher–KPP model and its generalisations

To describe the cell invasion process observed in the barrier assay experiments, we will use the Fisher–KPP equation, and its extensions, to model the evolution of cell density over time. The original Fisher model was proposed in [59] to describe the spread of mutant alleles in

the context of population dynamics, but can be easily re-interpreted to describe the spatio-temporal dynamics of cell density. The model was later studied in a more generalised context by Kolmogorov, Petrovskii, and Piskunov [96]. More recently in cell biology, this model and its generalisations have been used to model a diverse range of phenomena including wound healing [71, 125] and tumour growth [60, 79, 174]. The original Fisher model can be written as

$$\frac{\partial C}{\partial t} = \nabla \cdot (D(C)\nabla C) + f(C), \quad f(C) = rC\left[1 - \left(\frac{C}{K}\right)\right], \quad D(C) = D_0. \tag{2.2}$$

In the context of our application, $C(x, y, t)$ represents cell density, $D(C) \geq 0$ is the diffusion coefficient (which is assumed to be a constant, $D_0 > 0$). The function $f(C)$ represents net cell proliferation according to the logistic growth law, with $K > 0$ the carrying capacity, and $r > 0$ the linear proliferation rate.

The simplicity of the Fisher model motivates generalisations, which add complexity to the model with the aim to better capture the observed experimental behaviours. We will consider two approaches to generalise the model. First, we will allow the diffusion coefficient to depend on cell density. The intuitive motivation is that as cell density increases, a crowding effect will push the cells to move into less dense areas more quickly, leading to density-dependent diffusion, where the diffusion coefficient is an increasing function of density. The porous media model, so-called because it was originally used to describe the flow of gas through a porous medium, assumes

$$D(C) = D_0(C/K)^\eta,$$

for some $\eta \geq 0$. Combining this generalised diffusion term with the proliferation term in Eq. (2.2) produces the Porous Fisher model, which has been used to model cell invasion and wound healing [55, 83, 130, 170, 172]. Its theoretical properties have been studied in many works, including [51, 209].

The second way to generalise the Fisher model is to modify the logistic growth term, $f(C)$, which was studied by Kolmogorov et al. [96], resulting in the Fisher–KPP equation. We will do so by introducing additional "shape parameters" that adjust the shape of the growth curve, while fixing the zeros of f at $C = 0,\ K$. Richards [159] proposed adding an exponent to the (C/K) term, which allowed for better fitting to data from plant experiments. Tsoularis and Wallace [185] reviewed a number of other generalisations to the logistic growth model, and concluded that they can all be seen as a special case of the generalised logistic growth model:

$$f(C) = rC^\alpha\left[1 - \left(\frac{C}{K}\right)^\gamma\right]^\beta,$$

where $\alpha, \beta, \gamma > 0$ are the shape parameters.

Combining these generalisations gives the following equation as a general form to summarise the models:

$$\begin{aligned} \frac{\partial C}{\partial t} &= \nabla \cdot (D(C)\nabla C) + f(C), \\ f(C) = rC^{\alpha} &\left|1 - \left(\frac{C}{K}\right)^{\gamma}\right|^{\beta} \operatorname{sgn}(K - C), \quad D(C) = D_0 \left(\frac{C}{K}\right)^{\eta}. \end{aligned} \tag{2.3}$$

The absolute value and the sign functions are introduced so that growth is still sensibly defined in the case where β takes a non-integer value, to avoid raising a potentially negative value to a fractional power. For the rest of the chapter, we use the following units for reporting the values of the quantities in this model:

$$[C] = [K] = \text{cells/mm}^2, \ [t] = \text{h}, \ [D_0] = \mu\text{m}^2/\text{h}, \ [r] = \text{h}^{-1}, \ [\alpha] = [\beta] = [\gamma] = [\eta] = 1. \tag{2.4}$$

We also formulate the model in one-dimensional polar coordinates (ρ, t) in terms of $\tilde{C}(\rho, t)$, which allows us to model the reduced dataset. We will drop the tilde in the cases where it is clear that the polar formulation is used, to give

$$\frac{\partial C}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial \rho}\left[rD(C)\frac{\partial C}{\partial \rho}\right] + f(C). \tag{2.5}$$

We will consider four specific models, as summarised in Table 2.1, which are all special cases of Eq. (2.3).

Model name	Free parameters	Fixed parameters	$D(C)$	$f(C)$
Standard Fisher	D_0, r, K	$\alpha = \beta = \gamma = 1, \eta = 0$	D_0	$rC(1 - (C/K))$
Porous Fisher	D_0, r, η, K	$\alpha = \beta = \gamma = 1$	$D_0(C/K)^{\eta}$	$rC(1 - (C/K))$
Richards	D_0, r, γ, K	$\alpha = \beta = 1, \eta = 0$	D_0	$rC(1 - (C/K)^{\gamma})$
Generalised Fisher	D_0, r, α, β, K	$\gamma = 1, \eta = 0$	D_0	$rC^{\alpha}\left\|1 - \left(\frac{C}{K}\right)\right\|^{\beta} \operatorname{sgn}(K - C)$

Table 2.1: Summary of the four PDE models considered, as special cases of Eq. (2.3).

Note that these models are assumed to be fully and directly observable, that is, we have data for C itself, which allows us to avoid considering a measurement model. In many modelling studies, not all state variables of the PDE model are observable, and some variables may be indirectly observed via a measurement function. In such cases, the measurement model is an important component for determining practical identifiability.

All of these models can give rise to travelling wave solutions with constant speed and consistent profiles in one spatial dimension given appropriate initial conditions, and analogous behaviours in two spatial dimensions. These solutions qualitatively agree with the behaviour

of the cell population in experiments. In one spatial dimension, a simple rescaling argument shows that travelling wave solutions in the Standard Fisher model have a speed proportional to $\sqrt{rD_0}$ and a width proportional to $\sqrt{D_0/r}$. This means that r and D_0 can be estimated simply by measuring the speed and width of the wave front, while K can be estimated by observing the long-term cell density behind the wave. The main qualitative difference between the model behaviours is the shape of the wave front, which does not have an analytic solution in general. This means that, while the parameters in the Standard Fisher model can be estimated using quantities directly measurable from the data, we need to look at the shape of the wave front to estimate the parameters in the more complicated models. The shape parameters cannot be expressed in simple terms as a function of summary statistics such the width or speed of the wavefront. Therefore, the estimation of these parameters requires the use of inference methods such as profile likelihoods or MCMC, in conjunction with data on the density profile.

2.2.4 Numerical methods for PDE simulation

The PDE models introduced in Sec. 2.2.3 cannot be solved analytically in general, so they must be solved numerically. We use a finite difference scheme to simulate the general form of the model, Eq.(2.3). We chose to retain the physical dimensions of the quantities, as this makes it easier to interpret the quantities in a biological context, and makes no difference to the computational cost. For simulations in two spatial dimensions, the size of the domain, corresponding to the size of the image, is $L_x = L_y = 4380$ µm. This domain is discretised into $n_x = 150$ by $n_x = 150$ squares, each with side length $\Delta x = \Delta y = 29.2$ µm. This mesh was chosen to coincide with the size of the measuring boxes used during image analysis, so that the matrix representing the model solution at any given time has the same size as the data. Temporal discretisation is done with $\Delta t = 1/30$ h, so that every 10 steps of the simulation corresponds to an experimental image, which was taken 1/3 h apart. We have verified that these choices are sufficient to ensure that the PDE dynamics are resolved accurately, see Appendix A.2.1. The model equations are not stiff for the range of biologically meaning parameter values considered in Eq. (2.14), although numerical instability may arise for parameter values far outside of these ranges.

In approximating the nonlinear diffusion term, we follow the method in [202], where a central difference approximation with step size $\Delta x/2$ or $\Delta y/2$ is used to approximate the two derivatives in the x- and y-directions, respectively. This requires the evaluation of the non-linear diffusion

coefficient $D(C)$ at half mesh points, which is approximated as the average of D at the two adjacent mesh points. We use an implicit-explicit (IMEX) scheme [7, 164] for time-stepping, where the nonlinear diffusion coefficient $D(C)$, and the proliferation term $f(C)$, are treated with the explicit Euler method, and the diffusion term overall is treated with the implicit Crank-Nicolson method, which has second order convergence. The full details of the numerical schemes are provided in Appendix A. The codes for simulations, inference, and analysis, are all written in MATLAB, and provided on Github at https://github.com/liuyue002/woundhealing.

2.3 Inference methods for identifiability analysis

For performing identifiability analysis, we will use three inference methods: maximum likelihoods, profile likelihoods, and Bayesian inference. The general theory behind these methods is introduced in Appendix B, while this section provides the details on the implementations of these methods.

2.3.1 Notations

We will use the following notation. For a given model, let P be the number of free parameters, and let $\boldsymbol{\theta}$ denote the vector of free parameters in the model. Let $\boldsymbol{\theta}_{-i}$ denote the parameter vector with θ_i, the i^{th} parameter, removed. We may also replace the subscript with the name of the replaced parameter for readability. For example, for the Standard Fisher model, $P = 3$, $\boldsymbol{\theta} = (D_0, r, K)$, and $\boldsymbol{\theta}_{-r} = (D_0, K)$. Let $C_{\text{data}}(x, y, t)$ be the cell density measurements resulting from the image analysis in Section 2.2.2, and let $C_{\text{model}}(x, y, t; \boldsymbol{\theta})$ be the solution to Eq. (2.3), obtained by running a model simulation using the methods described in Sec. 2.2.4. We assume that the observed cell density is generated by the model with the given model parameter set $\hat{\boldsymbol{\theta}}$, and perturbed by an independent and identically distributed (i.i.d.) observation noise at each data point that follows a Gaussian distribution with zero mean and unknown variance σ^2. That is,

$$C_{\text{data}}(x_i, y_j, t_k) = C_{\text{model}}(x_i, y_j, t_k; \hat{\boldsymbol{\theta}}) + \epsilon_{ijk}, \quad \epsilon_{ijk} \sim \mathcal{N}(0, \sigma^2). \tag{2.6}$$

It must be acknowledged that this error model is not necessarily realistic. A residual analysis, with results given in Fig. 2.11, shows that the observational noise might depend on cell density, and is likely auto-correlated. Nonetheless, we make this assumption due to its simplicity and wide adoption in the literature. We discuss ways in which the error model could be improved in Sec. 2.5. Note that Eq. (2.6) also forms our observation model, where we assume that the

only variable in the PDE model (Eq. (2.3)) is directly observable. In many cases, some variables might not be directly measurable. In those cases, the likelihood function (defined below) can be formulated in terms of the measured quantities instead of the original variables in the model.

In the case where we use the full dataset $C_{\text{data}}(x, y, t)$ for inference, we let

$$\Delta C_{ijk}(\boldsymbol{\theta}) = |C_{\text{data}}(x_i, y_j, t_k) - C_{\text{model}}(x_i, y_j, t_k; \boldsymbol{\theta})| \tag{2.7}$$

denote the discrepancy between the data and model, and $R(\boldsymbol{\theta}) = \sum_{i,j,k} \Delta C_{ijk}^2(\boldsymbol{\theta})$ be the total squared residual. In the case that the reduced dataset $\tilde{C}_{\text{data}}(\rho, t)$ is used, the residual is instead defined as

$$R(\boldsymbol{\theta}) = \sum_{l,k} |\tilde{C}_{\text{data}}(\rho_l, t_k) - \tilde{C}_{\text{model}}(\rho_l, t_k)|^2 w(\rho_l), \tag{2.8}$$

where $w(\rho_l)$ is the number of data points (x_i, y_j) that fall within the annulus

$$\rho_l - \frac{\Delta\rho}{2} < \sqrt{(x_i - c_x)^2 + (y_j - c_y)^2} \leq \rho_l + \frac{\Delta\rho}{2},$$

i.e., the number of points used to calculate the mean density $\tilde{C}_{\text{model}}(\rho, \cdot)$ in the annulus (Eq. (2.1)). The log-likelihood function $\hat{l}$ is then calculated from $R(\boldsymbol{\theta})$ and Eq. (B.5) as

$$\hat{l}(\boldsymbol{\theta}, \sigma^2) = -N \log(\sigma) - \frac{N}{2} \log(2\pi) - \frac{R(\boldsymbol{\theta})}{2\sigma^2}, \tag{2.9}$$

where $N = n_x n_y n_t$ is the total number of data points in the dataset.

2.3.2 Maximum likelihood

Let $\boldsymbol{\theta}^*$ denote the maximum likelihood estimator (MLE, see Appendix B.2), defined as

$$(\boldsymbol{\theta}^*, \sigma^*) = \underset{\boldsymbol{\theta}, \sigma}{\operatorname{argmax}}\, \hat{l}(\boldsymbol{\theta}, \sigma). \tag{2.10}$$

Given $\boldsymbol{\theta}^*$, we can find σ^* using

$$(\sigma^*)^2 = \frac{R(\boldsymbol{\theta}^*)}{N}, \tag{2.11}$$

which means that the optimisation for σ can be done explicitly after the optimisation for $\boldsymbol{\theta}$. Substituting this expression for σ^* into Eq. (2.9) motivates the definition

$$l(\boldsymbol{\theta}) = \hat{l}\left(\boldsymbol{\theta}, \frac{R(\boldsymbol{\theta})}{N}\right) = \frac{-N}{2}\left[\log\left(\frac{2\pi}{N} R(\boldsymbol{\theta})\right) + 1\right], \tag{2.12}$$

so that

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, l(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, R(\boldsymbol{\theta}).$$

We find $\boldsymbol{\theta}^*$ by minimising $R(\boldsymbol{\theta})$ using Matlab's *globalsearch* routine. The procedure is initialised with the following default parameter values (unless stated otherwise):

$$D_0 = 1300 \text{ μm}^2/\text{h}, \quad r = 0.3 \text{ h}^{-1}, \quad K = 2600 \text{ cell/mm}^2, \quad \alpha = \beta = \gamma = 1, \quad \eta = 0. \tag{2.13}$$

The initial values for D_0 and r are chosen so that they lie in the same order-of-magnitude range as found by [83], and the initial value for K is chosen to be a rough estimate of the cell density in the interior of the domain at the end of the experiments. We also impose the following lower and upper bounds for these parameters to better guide the optimisation algorithm:

$$\begin{aligned} 100 \text{ μm}^2/\text{h} < D_0 < 10000 \text{ μm}^2/\text{h}, \quad 0.01 \text{ h}^{-1} < r < 1 \text{ h}^{-1}, \\ 500 \text{ cell/mm}^2 < K < 5000 \text{ cell/mm}^2, \quad 0 < \alpha, \beta, \eta < 3, \quad 0 < \gamma < 9. \end{aligned} \tag{2.14}$$

The bounds for D_0, r, and K are chosen to be very wide to cover the biologically relevant range of values. The bounds for the shape parameters $(\alpha, \beta, \gamma, \eta)$ are chosen out of numerical stability concerns. A value for these parameters outside these bounds involves raising a base to a high exponent, which can destabilise the numerical scheme used for running the simulations. The commonly given values for these parameters in the literature fall well within these bounds. Since the typical values for these parameters span vastly different orders of magnitude, from $D_0 \approx 10^3$ cell/mm^2 to $r \approx 10^{-1}$ h^{-1}, we rescale them by their initial values when running the optimisation routine. For example, during the computations, we replace D_0 in the model with $1300\tilde{D}_0$, where $\tilde{D}_0$ is of order 1. This choice of scaling is more-or-less arbitrary, as it is sufficient to choose a scale on the same order of magnitude of a typical parameter value to improve the performance of the optimisation routine. This is because the optimisation routine assumes that the impact of a change in any parameter on the objective function is roughly the same.

The *globalsearch* routine works by generating a number of trial points (we use the default of 1000 points) around the given initial point, using a scatter search algorithm [187]. The objective function (for us, $R(\boldsymbol{\theta})$) is evaluated at each trial point, with the points unlikely to be near the global optimum eliminated. For the problems considered here, this tends to leave between five to 40 points. A local search is initialised at each of the remaining points with the *fmincon* routine, which implements the interior point algorithm [36], resulting in a number of local optima. The values of the objective function at these local optima are compared, and the best one is returned as the global optimum.

The multiple start strategy ideally ensures that the optimisation outcome is indeed a global, instead of merely a local, optimum. However, as we will show in the results, under certain

conditions (such as when the dimension of the search space is high, which for the problem considered here, is four or above), *globalsearch* might not find the true global optimum. This is evident if the maximum of the profile likelihood function (introduced in Section B.3) does not agree with the calculated MLE, which means that the purported MLE was, in fact, only a local maximum. This is because the maximum point of the profile likelihood function represents a higher local maximum, which may or may not be the global maximum. When this happens, we restart the optimisation by including the maxima of the profile likelihoods in the set of initial points.

In some cases, when the likelihood landscape is especially rough, *globalsearch* fails to find the global optimum even after five iterations. This is the case for the Generalised Fisher model using the datasets from the experiments with triangular initial conditions. In this case, we instead use the Covariance Matrix Adaptation Evolution Strategy (CM-AES) method [72] to perform the optimisation, using the implementations provided in [73]. The CM-AES method is an instance of an evolutionary algorithm, where the algorithm iteratively samples a population of points in the parameter space according to a multivariate normal distribution, then updates the mean and covariance matrix of the distribution based on the objective function evaluated at these points. The method is known to perform better than derivative-based methods (such as the interior point method used by *fmincon*) in difficult non-linear, non-convex optimisation problems, at the expense of higher computational cost, which is why we do not use it in situations where *globalsearch* or *fmincon* succeeds.

Once the MLE is obtained, we compute the Fisher information matrix (FIM), defined as

$$\mathcal{I}(\boldsymbol{\theta}^*) = -E\left[\left.\frac{\partial^2 l(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}^*}\right], \tag{2.15}$$

by adapting the method outlined in [97]. A common approach for constructing confidence regions for parameter values is to first approximate the distribution of parameters as a multivariate Gaussian distribution with mean at the MLE, $\boldsymbol{\theta}^*$, and the covariance matrix equal to the FIM, $\mathcal{I}$, and then compute confidence regions using known results for Gaussian distributions. This approach, while computationally cheap, as it only requires the MLE and FIM, can produce misleading confidence regions, as found in [205]. In particular, the constructed confidence regions are not valid if the distribution of the parameter values is not approximately Gaussian, which tends to be the case when the parameter is practically non-identifiable, as we have found in

Sec. 2.4, and also found by Simpson et al. in [176]. Therefore, the MLE and FIM alone do not provide reliable information for identifiability analysis.

2.3.3 Profile likelihood

The profile likelihood method is built upon the MLE. The theory for profile likelihoods is introduced in more detail in Appendix B.3. We define the normalised profile log-likelihood function $l_i(\theta_i')$ for the parameter θ_i to be

$$l_i(\theta_i') = \max_{\boldsymbol{\theta}_{-i}} \left[l(\boldsymbol{\theta})|_{\theta_i=\theta_i'} \right] - l(\boldsymbol{\theta}^*), \tag{2.16}$$

where l is given in Eq. (2.12). We will refer to l_i simply as the profile likelihood for brevity. Similarly, a bivariate profile likelihood, $l_{i,j}$, is defined as

$$l_{i,j}(\theta_i', \theta_j') = \max_{\boldsymbol{\theta}_{-i,-j},\sigma} \left[\log L(C_{\text{data}}|\boldsymbol{\theta}, \sigma)|_{\theta_i=\theta_i',\theta_j=\theta_j'} \right] - l(\boldsymbol{\theta}^*). \tag{2.17}$$

The profile likelihoods can be understood as approximations to the marginal posterior distributions of the parameter values in the Bayesian sense, see Appendix. B.5 for details.

Note that the definition of profile likelihood follows the convention where the profile likelihood is defined in terms of the full likelihood $l(\boldsymbol{\theta})$, and the location of the global maximum of the profile likelihood coincides with the MLE. This is the convention adopted in [134, 175, 176], among others. An alternative convention, adopted in [154, 155] and others, is to define the profile likelihood in terms of the fitting residual, $R(\boldsymbol{\theta})$, and the profile likelihood defined following this convention is equivalent to $-l_i(\theta_i)$ defined in Eq. (2.16), with additional shifting and scaling. The consequence is that the MLE corresponds to the global minimum of the profile likelihood instead. As this is a little counter-intuitive, we will follow the convention chosen above.

Following [134] and [176], we use the profile likelihood function to define an approximate 95% confidence interval for a single variable as

$$\{\theta_i | l_i(\theta_i) > -\chi^2(0.95; 1)/2 \approx -1.92\}, \tag{2.18}$$

and a joint confidence region for two variables as

$$\{\theta_i, \theta_j | l_{i,j}(\theta_i, \theta_j) > -\chi^2(0.95; 2)/2 \approx -3.00\}, \tag{2.19}$$

where $\chi^2(\cdot; m)$ denotes the inverse of the cumulative distribution function (cdf) of a χ^2 distribution with m degrees of freedom, see Appendix B.3 for the derivation of this approximation. The shape of the univariate profile likelihood curve indicates whether or not the parameter is

practically identifiable. A profile likelihood curve that is unimodal, smooth, and decays quickly away from its peak (which occurs at the MLE) suggests that the parameter is identifiable. Non-identifiability can be reflected in a profile likelihood in many different ways, such as multi-modality, slow decay away from the MLE, or a flat top (see Fig. 1.1). We compute points on the profile likelihood curve using the same optimisation algorithms as we use for finding the MLEs, as described in Sec. 2.3.2.

The profile likelihood function in Eq. (2.16) is normalised in a way such that l_i has a maximum of zero at $\theta_i = \theta_i^*$. A Taylor expansion of $l_i(\theta_i)$ around θ_i^* gives

$$l_i(\theta_i) = \frac{(\theta_i - \theta_i^*)^2}{2} \left.\frac{\mathrm{d}^2 l_i}{\mathrm{d}\theta_i^2}\right|_{\theta_i^*} + O(|\theta_i - \theta_i^*|^3). \tag{2.20}$$

If this approximation holds for a sufficiently large neighborhood around θ_i^*, then the profile likelihood will have a parabolic shape, and the distribution of parameter values will be approximately Gaussian, see Appendix. B.3. Consequently, the associated parameter will be practically identifiable. This fact will be used in the rest of this chapter.

2.3.4 Bayesian inference

We would like to compare the performance and computational cost of the profile likelihood method with sampling methods based on Bayesian inference for parameter estimation. We take the Metropolis-Hastings (MH) method [161, Ch. 4.5], one of the simplest and most well-known MCMC methods, as a representative of Bayesian methods. A general introduction to MH is given in Appendix B.4. In this chapter, we will implement MH, and use it to construct a posterior distribution for the simplest case, which is the Standard Fisher model outlined in Table 2.1. We will then compare the information on parameter values provided by the resulting posterior distribution density to that of the profile likelihoods, as well as the computational performance of these two methods. The implementation details are as follows.

For simplicity, we use a uniform distribution in the parameter space bounded by Eq. (2.14) for the prior, so we can set $p(\boldsymbol{\theta}) \propto 1$ for $\boldsymbol{\theta}$ within the bounds. To generate a proposal for each subsequent step in the Markov chain, we perturb each parameter independently as follows:

$$g(\boldsymbol{\theta}^{(s+1)}|\boldsymbol{\theta}^{(s)}) \sim \boldsymbol{\theta}^{(s)} + a \cdot \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}),$$

where Σ is a diagonal matrix containing the square of the relative magnitude of the parameters. We use 1 for D_0 and K, and 10^{-4} for the other parameters. It is known that MH is more

efficient when the distribution g is similar to the posterior distribution. We assume that the changes in likelihood caused by the relative changes in a parameter value are similar for each parameter, so we scale the components of the perturbations by the magnitude of their corresponding parameters. This assumption turns out to be sensible, as we will see in Sec. 2.4, the width of the confidence interval for each parameter, as an indirect indicator of the impact of the parameter on model solution, is roughly proportional to their magnitude. As with scaling the parameters for computing the profile likelihoods, this choice of scaling is partially arbitrary, and does not need to be precise. A well-chosen scaling speeds up the convergence of the Markov chains (meaning that the samples in the chain constitute a good representation of the posterior distribution). If the scaling was chosen poorly, or no scaling was used at all, which might be the case in scenarios with little prior knowledge on the magnitude of the parameters, MH will still converge if it is run for a sufficiently long time, albeit more slowly. This choice of transition kernel is known as a random walk MH. This is the simplest and most common choice for the transition kernel.

The step size, a, is initially set to $a = 1$. We implement a simple scheme for adaptive step size based on [4]. For every 100 accepted samples, we calculate the acceptance rate for the previous 100 samples, and increase a by 10% if the acceptance rate is higher than 0.6, decrease it by 10% if the acceptance rate is lower than 0.2, and leave it unchanged otherwise. Results from [64, 160] suggest that, under some mild assumptions, and depending on the dimension of our parameter space, the optimal acceptance rate is between 0.234 and 0.44. This adaptive scheme aims to keep the acceptance rate near the optimal values.

We implement the multiple-sequence strategy from [65] to obtain a termination criterion for the sampling process. We compute four independent Markov chains, with the four starting points chosen such that they are within the biologically plausible range for the parameters, but over-dispersed, as recommended by [65]. For each parameter θ, we calculate R_θ, which is, roughly speaking, the ratio between the variance of within-chain sample means, and the mean of within-chain sample variances, with some additional minor correcting factors. R_θ measures how much the current estimates for the variance of θ could be improved if we continue the chains in the limit of infinitely many samples, and it can be shown that $R_\theta \geq 1$, and $\lim_{n\to\infty} R_\theta = 1$, where n is the number of samples. We choose to terminate the chains when R_θ falls below 1.01 for all parameters, or after 10^5 steps of the chains, whichever happens first. We will report the

number of steps required for convergence, or R_θ at the final step if the termination criterion is not met.

2.3.5 Information criterion for model selection

Out of the many information criteria mentioned in Sec. 2.1.2, we consider two most commonly used criteria for model selection: AIC and BIC. They can both be computed from the MLE [99, Ch. 3, 9]:

$$AIC = -2l(\boldsymbol{\theta}^*) + 2P, \tag{2.21}$$

$$BIC = -2l(\boldsymbol{\theta}^*) + \log(N)P. \tag{2.22}$$

Recall that P is the number of free parameters in the model, and N is the number of data points. For both the AIC and BIC, a lower score is considered better. Both criteria can be thought of as balancing the quality of data fitting (favouring higher likelihood) and the complexity of the model (favouring fewer free parameters).

2.4 Results

This section presents the profile likelihoods of the parameters in the four models, calculated under a variety of conditions that differ in the dataset used and data resolution. While in some cases the difference in data leads to differences in the profile likelihoods in expected ways, in other cases the effects are unexpected and interesting to explain.

2.4.1 Profile likelihoods can recover true parameter values for synthetic data

Before presenting results for the experimental data, we first calculate the profile likelihoods for each model for two sets of synthetic data. The main purpose of this exercise is to verify that profile likelihoods can recover the correct parameter values under ideal conditions. The synthetic data are generated by simulating the model Eq. (2.3) in one spatial dimension, using the parameter values in Eq. (2.13), and perturbed by adding Gaussian noise to the model solution as in Eq. (2.6). The time span of the simulations is $t = 77/3$ h, and the temporal resolution of the synthetic data is $\Delta t = 1/3$ h, both the same as for the real data, in order to make the results here comparable with the results for real data. The "low noise" dataset uses $\sigma = 20$, while the "high noise" dataset uses $\sigma = 400$. In comparison, the σ^* estimated from real data via Eq. (2.11) ranges between 380 and 460, depending on the dataset and the model.

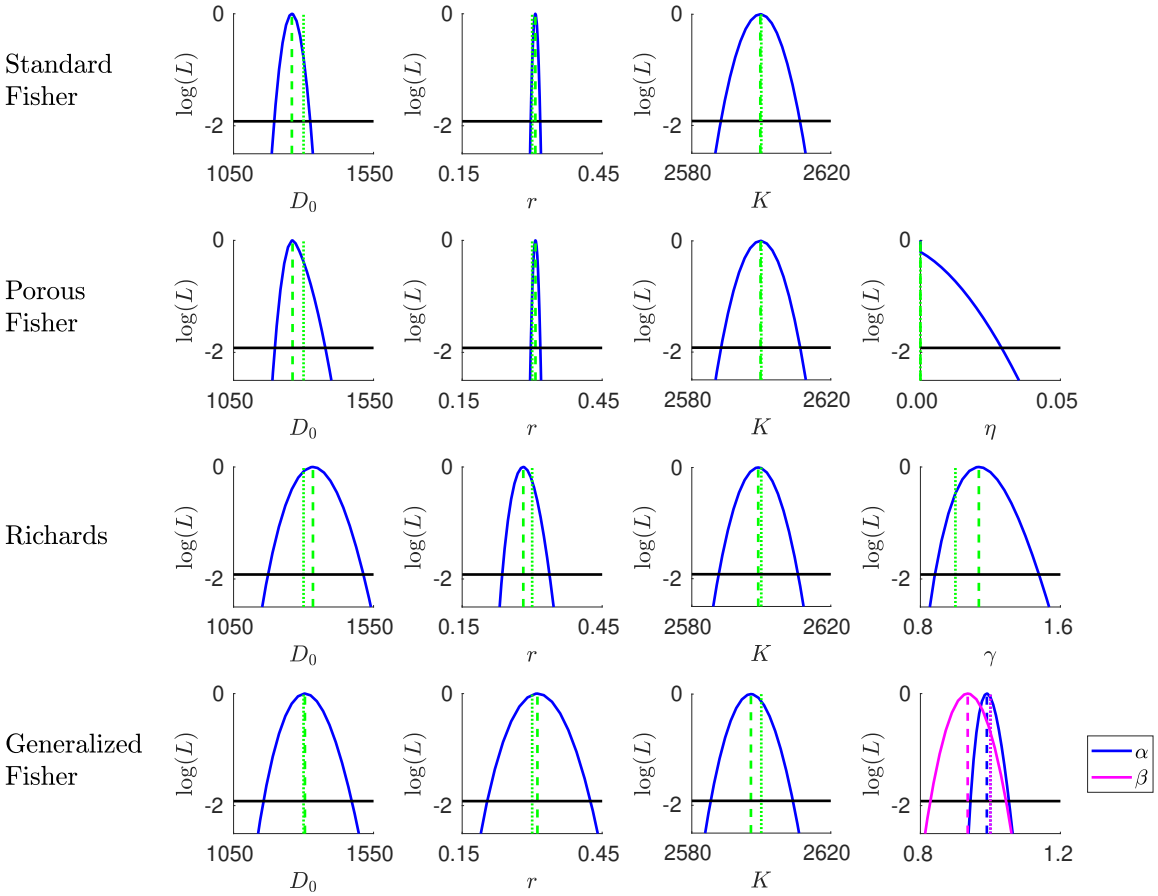


Figure 2.2: Profile likelihoods for the four models as described in Eq. (2.3) and Table 2.1, for a synthetic dataset generated with Eq. (2.3) and parameter values in Eq. (2.13), perturbed as in Eq. (2.6) with $\sigma = 400$ (the "high noise" dataset). The dotted vertical lines mark the location of the true parameter values, while the dashed vertical lines mark the MLE for each parameter. The black horizontal lines at -1.92 mark the threshold for the 95% confidence interval. The axis scale for the parameters shared between the models (D_0, r, K) is kept consistent. The units of the quantities in the plots are given in Eq. (2.4). The detailed inference results are given in Table E.13.

Since the synthetic data are generated using the basic Standard Fisher model, all four models are capable of reproducing the (unperturbed) data exactly, therefore the inference is not affected by model mis-specification. For the synthetic model, the true model parameters are known to us, the assumption on observation error in Eq. (2.6) is known to be satisfied, and the data resolution is high. In this case, structural non-identifiability and observational noise are the only sources of non-identifiability, which allows us to evaluate the performance of profile likelihoods in an ideal situation. Since we have control over the magnitude of the observational noise, this also allows us to observe the effect of such noise on parameter identifiability, which we would not be able to do with real data.

The profile likelihoods for the high noise dataset are presented in Fig. 2.2, which shows that

all profile likelihood curves are uni-modal and have a roughly parabolic shape, which is expected given the asymptotic normality result (see Appendix. B.3). The MLEs and confidence intervals for the model parameters obtained with this dataset are presented in Sec. E.2. The MLEs for all parameters are close to the true parameter values, and located within the 95% confidence interval given by the profile likelihoods, as defined in Eq. (2.18). Moreover, all of these confidence intervals are finite. For the low noise dataset, the profile likelihood curves are very narrow, and centred almost exactly at the true parameter values. (The figure is not shown, since the profile likelihoods are so narrow that they would be barely visible on the same scale as Fig. 2.2.) These results verify that the profile likelihoods can recover the parameter values that best describe the underlying system, at least in a highly idealised case, as the theories suggest.

Notice that in Fig. 2.2, for the parameters D_0 and r, their confidence intervals are much wider in the Richards and Generalised Fisher models compared to the Standard Fisher and Porous Fisher models. This indicates that the former two models are more capable of compensating for a change in these parameters by shifting other parameters, making them less identifiable. In contrast, the confidence intervals for K remain consistent across all models, and the relative widths of the confidence intervals are much tighter for K than for D_0 and r. Given that the carrying capacity K has a clear physical interpretation with a direct long term impact on the model solution (it is the value to which $C(x,t)$ tends as $t \to \infty$), it is not surprising that this parameter is easier to estimate. Despite this, estimating K is non-trivial, since the stochastic nature of the underlying biological system means that the cell density continues to fluctuate, rather than settling down to a constant carrying capacity behind the travelling front.

While the estimates for the parameters are accurate overall (in the sense that the widths of the confidence intervals are small compared to the magnitude of the parameters), they are not as accurate for certain parameters compared to others, most notably the γ parameter in the Richards model, where the MLE and the profile likelihood skew to the right of the true value. However, this discrepancy is reduced as σ, the magnitude of the noise, decreases. This means that the parameter estimates in the Richards model is more sensitive to observational noise compared to other models.

2.4.2 All four models are practically identifiable

The results for the synthetic dataset showed that the models are identifiable under ideal settings. This remains the case for the experimental datasets. This section is concerned with the full

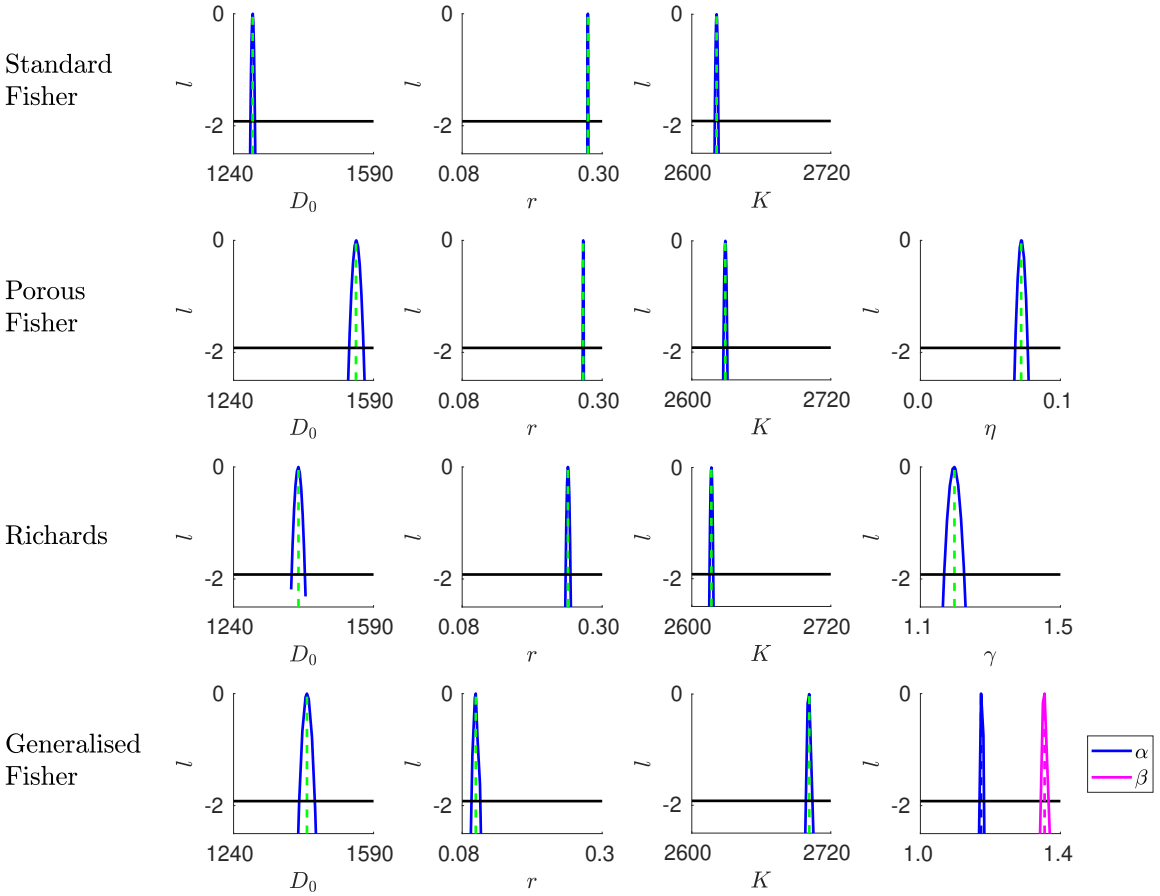


Figure 2.3: Profile likelihoods using data from Experiment 1. The vertical dashed lines mark the MLE for each parameter, and the black horizontal line at -1.92 marks the threshold for the 95% confidence interval. The units of the quantities in the plots are given in Eq. (2.4). The detailed inference results are given in Table E.2.

datasets ($C_{\text{data}}(x, y, t)$), whereas the results for the reduced datasets ($C_{\text{data}}(\rho, t)$) are given in the next section. We present the profile likelihoods for Experiment 1 in Fig. 2.3, which is arbitrarily chosen as an illustrative representative for the purpose of this section. The results for the other experiments are qualitatively similar. We present the profile likelihoods for the other experiments, and the MLEs, the 95% confidence intervals, and the AIC and BIC for each model for each experiment, in Appendix E. Visual inspection indicates that the MLEs for all four models give rise to model predictions that match the data very well, and the residuals are small compared to the magnitude of the solution. As such, it is difficult to select between models by such visual inspection alone. A likelihood ratio test for nested models ([169], details in Appendix E) shows that the predictions of all three more complicated models are significantly better than the Standard Fisher model.

Observe that in Fig. 2.3, all profile likelihood curves are uni-modal, and roughly parabolic in shape, and the confidence intervals are narrow. These observations, and similar results from the

other experiments (shown in Appendix E), suggest that all models are practically identifiable given the available data from any one of the eight experiments, analysed separately. Comparing the Standard Fisher model with the more complicated models we notice that, as model complexity increases, the profile likelihood curves broaden, and therefore the confidence intervals widen, making the parameters less identifiable. This suggests that increases in model complexity tend to lead to decreases in parameter identifiability.

Intuitively, in the absence of model mis-specification, for nested models, the more complex model (with more free parameters) will be less identifiable since there is a greater possibility that a change in the value of one parameter can be compensated for by changes in other parameters. However, in the presence of model mis-specification, identifiability may improve as a result of introducing additional parameters, for example, if the additional parameter rectifies a flaw in a mis-specified model. Such cases should happen rarely in practice, since ideally such a flaw should be identified prior to the model selection stage of a study. Both residual analysis and posterior predictive checks provide effective tools for detecting model mis-specification. Residual analysis is computationally cheaper, does not rely on assumptions on a prior distribution for parameter values, and complements likelihood-based methods well, whereas a posterior predictive check is natural to implement after performing Bayesian inference.

2.4.3 Complicated models are more sensitive to data processing

We now turn to the question of whether the way we process the data impacts parameter identifiability. Fig. 2.4 shows the profile likelihoods computed using the radially averaged dataset from Experiment 1. Comparing Fig. 2.4 and Fig. 2.3, we notice that, in general, the profile likelihood curves are broader for the radially averaged dataset compared to the full dataset. This is expected, since the full dataset has many more data points. A more surprising observation is that, while the MLEs for the parameters in the Standard Fisher model remain consistent between the two datasets, the MLEs for the other three models are much less consistent. These observations show that the more complicated models are more sensitive to the way the data are processed or represented.

In the absence of noise or model mis-specification, and an abundance of data, consistent MLEs will be obtained regardless of data representation, since applying the same transformation (in this case, radial averaging) to the data and the model will not change the fit. This is what we have observed for the synthetic data in Fig. 2.2. However, when the data are noisy and/or

the model is mis-specified, this will generally not be the case, as we see here. There are several reasons for this. First, we have implicitly assumed within our model that the transform is applied to the (noiseless) data, then additive noise is applied to the transformed data. However, in reality, the transform is applied to the noisy data, and so the radially-averaged dataset is not equivalent to the original full dataset. Second, radial averaging smoothes the cell density. In particular, near the edge of the tissue where there are deviations from radial symmetry clearly observable in the data (see Fig. 2.5) we see a smoothing of the leading edge density profile. The more complicated models have increased flexibility to fit the leading edge data, which drives changes in the MLEs. As a result, we suggest that in cases where process noise is significant we should favour a simpler model, which is more robust to overfitting.

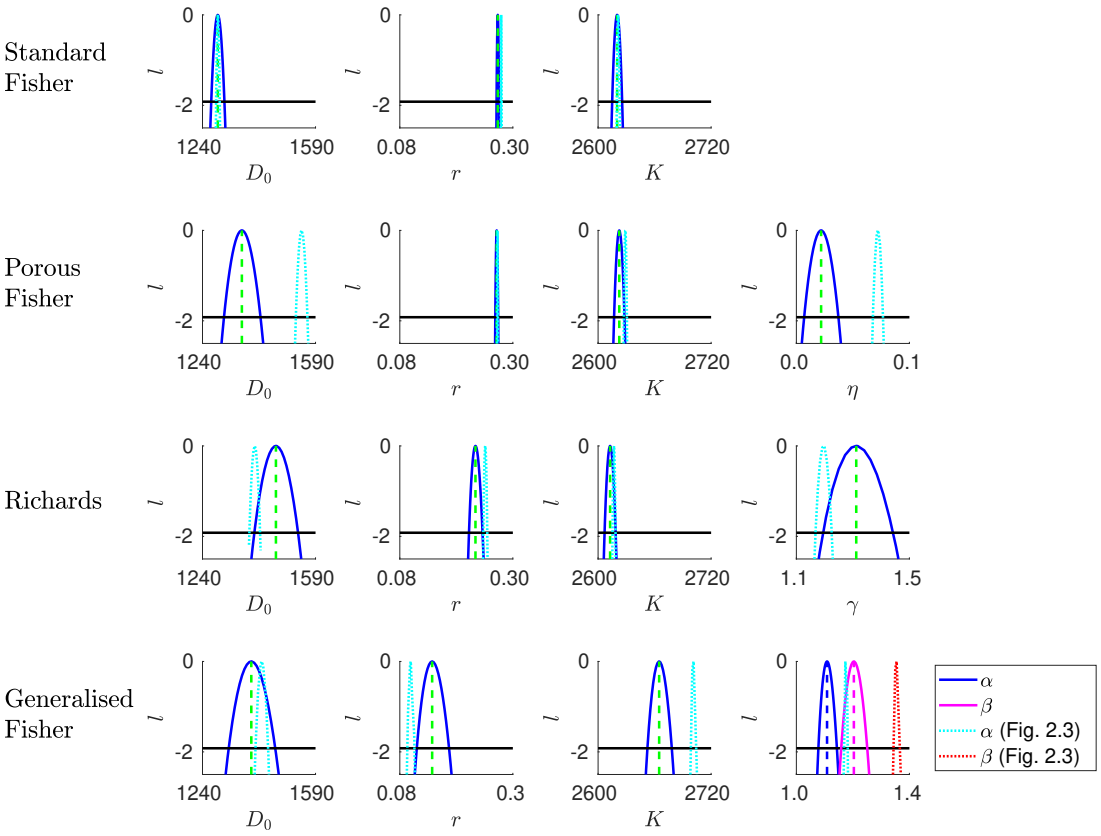


Figure 2.4: As per Fig. 2.3, except using the radially averaged dataset $\tilde{C}_{\text{data}}(\rho, t)$. The axes are the same as in Fig. 2.3. The profile likelihood curves from Fig. 2.3 are included here as dotted lines for ease of comparison. Notice that the profile likelihood curves are broader compared to Fig. 2.3, and the location of the MLE for the Porous Fisher, Richards, and Generalised Fisher models are shifted, but remain virtually unchanged for the Standard Fisher model. The units of the quantities in the plots are given in Eq. (2.4).The detailed inference results are given in Table E.1.

2.4.4 Consistency of parameter estimates across experimental replicates indicates practical identifiability

In this section, we will examine the profile likelihood results from the other experimental replicates, and compare them to the results for Experiment 1 presented in Sec. 2.4.2. The qualitative properties of the profile likelihood curves for the other experiments are largely similar to Experiment 1, in the sense that most model parameters have uni-modal, parabolic profile likelihoods, with confidence intervals having similar widths compared to Experiment 1. The detailed results are given in Appendix E, and we summarise the notable observations here.

For the experiments with circular initial conditions, the profile likelihoods calculated from the full dataset versus the reduced dataset are related in ways similar to Experiment 1, that is, the MLEs for parameters in the Standard Fisher model are consistent across the two datasets, while the MLEs for parameters in the other models display less consistency. Furthermore, the confidence intervals calculated from the full datasets are narrower compared to the ones calculated from the reduced datasets.

A comparison of the profile likelihood results across all experimental replicates shows that the parameter estimates are less consistent for the more complicated models compared to the simpler Standard Fisher model, in the sense that the variances of the eight MLEs of analogous parameters are higher for the more complicated models. In Fig. 2.6, we present the confidence intervals for all parameters in the four models found across the eight experimental replicates, except for the Generalised Fisher model, where we only present the results from the first four experiments.[1]

For all parameters, the widths of the confidence intervals for individual experiments are much narrower compared to the range of estimates across all experiments. For example, the widths of the confidence intervals for D_0 in the Standard Fisher model are on the order of 10 $\mu m^2/h$, whereas the MLEs for D_0 for Experiment 3 and 7 differ by around 700 $\mu m^2/h$. The confidence intervals for the same parameter estimated from different experiments rarely overlap; these variations in parameter estimates reflect the high variability commonly observed in biological processes, see Fig. 2.5. Visual inspection indicates no clear difference in the ability of the models to predict the data across the experimental replicates. Furthermore, the residuals for fitting the four models using MLE parameter values, as measured by σ (Eq. (2.11)), are

[1]The profile likelihoods for the four experiments with triangular initial conditions could not be computed accurately, due to difficulties in performing the optimisations during the computations for the profile likelihoods. These computational difficulties are discussed at the end of this subsection.

consistent across the experimental replicates. These results suggest that it may be appropriate to model the system using a fixed effects or a mixed effects framework so that parameters can vary across biological replicates. In addition, we note that the discrepancies could, in part, arise from spatial and temporal auto-correlations in the noise (Fig. 2.5 and [109]) – see the discussion (Sec. 2.5) for more details on this point.

We also find that the parameter estimates for the more complicated models seem to depend on the initial conditions, with the experiments with circular initial conditions yielding significantly different parameter estimates compared to the ones with triangular initial conditions. For the rest of the section, we will review these differences in detail, and discuss possible biological explanations.

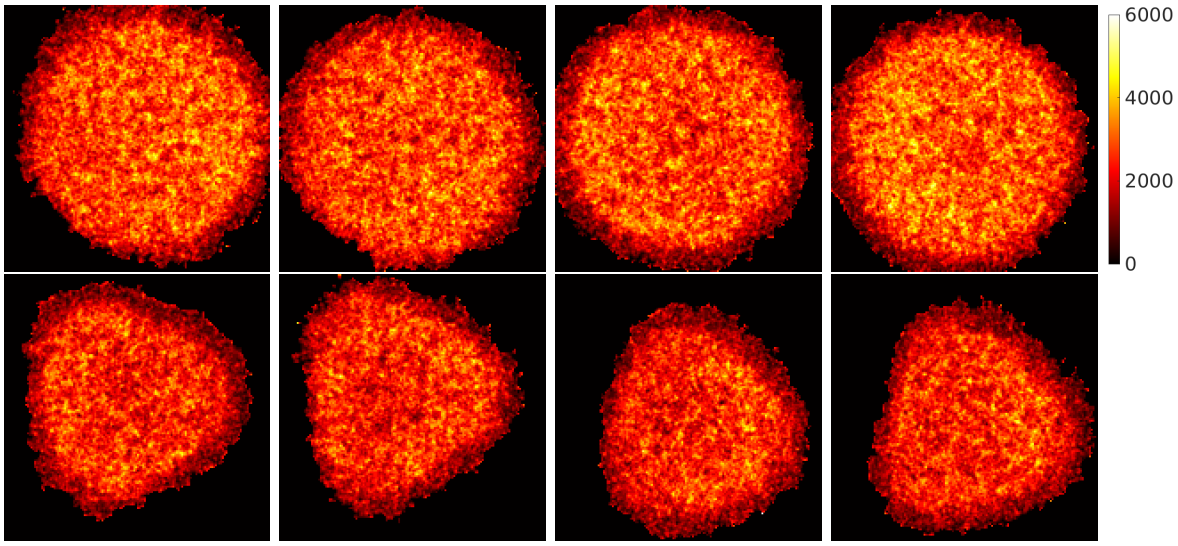


Figure 2.5: The cell density in the eight barrier assay experiments at the final time point. Top row: Experiments 1–4, Bottom row: Experiments 5–8. Notice that although the final states are qualitatively similar across experiments with the same initial conditions, there are quantitative differences. For example, the mean cell density for Experiments 1–4 lies between 2413-2552 cells/mm^2, which is higher than the 2064-2138 cells/mm^2 observed in Experiments 5–8.

For the Standard Fisher model, the MLEs for the parameters show only a mild degree of variability between the experiments. The difference in initial conditions does not lead to significant differences in the estimates of D_0 and r[2], while the estimate for K tends to be slightly lower for the experiment with triangular initial conditions as opposed to circular initial conditions[3].

While the variability of parameter estimates in the Standard Fisher model across experiments

[2]A two-sample Kolmogorov–Smirnov test (a statistical test for determining whether two distributions are the same [95]) could not reject the null hypothesis that the MLEs for the experiments with different initial conditions come from the same distribution, with a p-value of 0.1074 for D_0 and 0.5344 for r, respectively.

[3]With a Kolmogorov–Smirnov test p-value of 0.0110, therefore not significant.

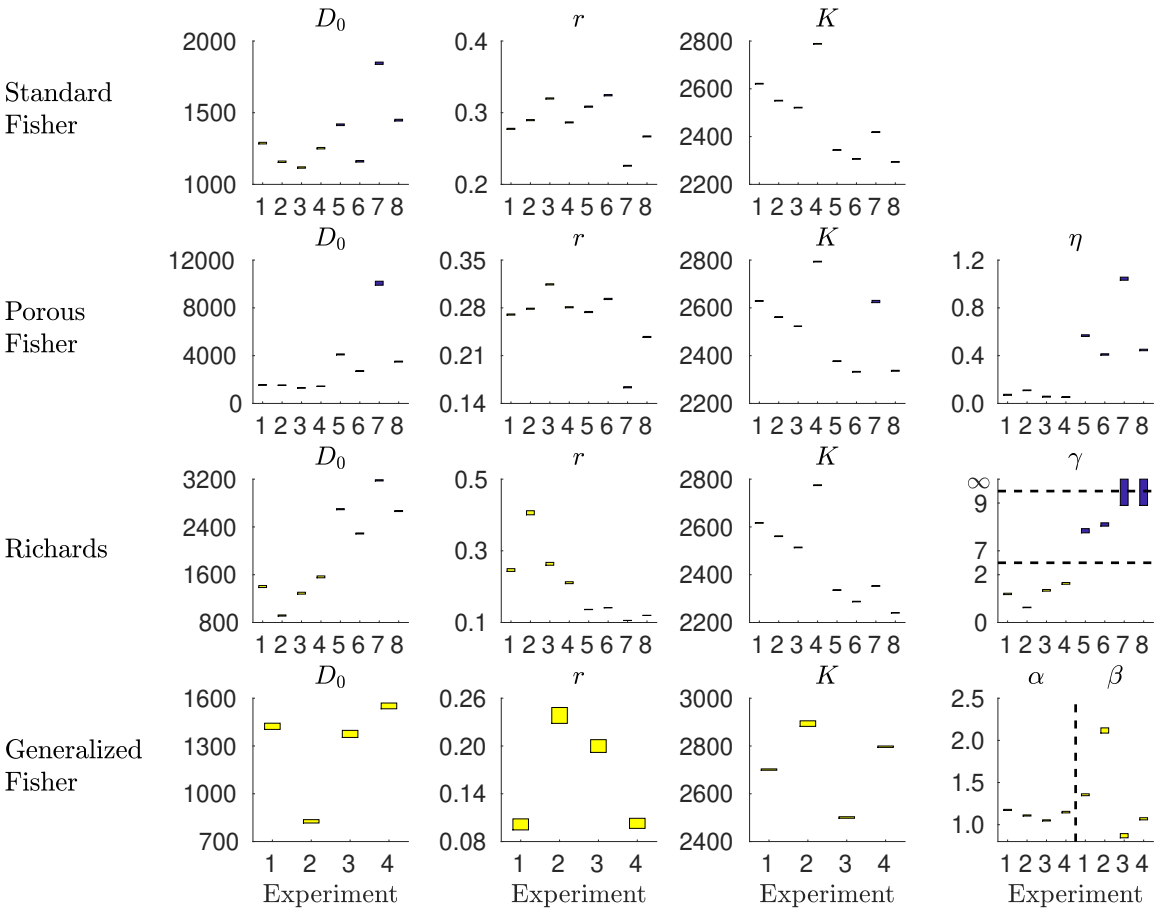


Figure 2.6: Comparison of the 95% confidence intervals for the parameters in the four models obtained using profile likelihoods on the eight experimental datasets. The vertical range of the bars represents the range of the confidence intervals. Note that some confidence intervals are very narrow. The experiments with circular initial conditions (Experiments 1–4) are labelled in yellow, while the ones with triangular initial conditions (Experiments 5–8) are labelled in blue. The horizontal dashed lines in the figure for γ in the Richards model indicate breaks in the $y-$axis. Note that some vertical axes do not start from zero. The units of the quantities in the plots are given in Eq. (2.4). See Appendix E for the full results on the confidence intervals.

is modest, it is much larger in the other models: for example, the standard deviations for the MLEs for D_0 across the experiments are 2937.8 $\mu m^2/h$ for Porous Fisher, 812.9 $\mu m^2/h$ for Richards, and 320.4 $\mu m^2/h$ for the Generalised Fisher models, compared to 238.3 $\mu m^2/h$ for Standard Fisher, although this comparison might not be entirely fair for the Porous Fisher model, since the diffusion coefficient is density-dependent, hence the interpretation of D_0 is different. Similar results hold for r, while the variability for K is approximately the same across the four models.

For the Porous Fisher model, the experiments with circular initial conditions (Experiments 1–4) estimate a very small η, well below 0.2, suggesting that there is little evidence for nonlinear diffusion in the dynamics, and the estimates for the other parameters (D_0, r, K) are relatively

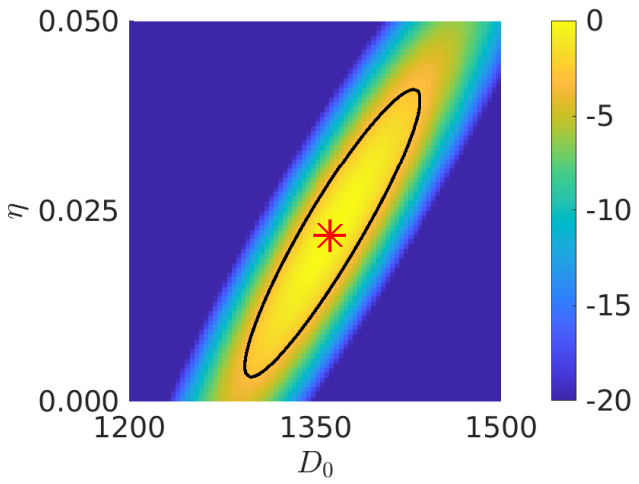


Figure 2.7: Bivariate profile likelihood function for D_0 and η in the Porous Fisher model represented as a heat map, calculated using the reduced dataset from Experiment 1. The location of the MLE is marked with the red star, and the black curve marks the boundary of the 95% confidence region, using $l_{D_0,\eta} > -3.00$ as the threshold (Eq. (2.19)). The slant of the bivariate likelihood contours indicates that the two parameters can compensate for each other. The units are $[\eta] = 1, [D_0] = \mu\text{m}^2/\text{h}$. The bivariate profile likelihood plots for the other parameters are given in Appendix E.4.

similar to their estimates in the Standard Fisher model. However, the experiments with triangular initial conditions (Experiments 5–8) estimate a much larger η, along with a higher D_0, suggesting that nonlinear diffusion effects are important. [4] Since the speed of the propagation of the cell colony (which can be identified as the speed of the travelling wave in the model solutions) does not vary much between the experiments, and an increase in η corresponds to a decrease in wave speed, the estimated values for D_0 and r are much higher for these experiments to compensate for a higher η. This compensation is reflected by the slanted ridge seen in the bivariate likelihood function for D_0 and η, shown in Fig. 2.7. Since the confidence region (the roughly elliptical region encircled by the black curve in Fig. 2.7) is bounded, and the bivariate likelihood function is uni-modal, we can call both parameters identifiable. However, correlations in parameter values are a typical source of non-identifiability. When parameters in a model exhibit such correlations, the model is referred to as "sloppy" [70], which is a common occurrence in mathematical biology. See Appendix E.4 for more bivariate profile likelihood plots illustrating correlation between parameter estimates, and Appendix E.1 for FIMs calculated to quantify these correlations.

We see a similar story for the Richards model. Observe that in the third row of Fig. 2.6, there

[4] The Kolmogorov–Smirnov test shows that the differences in D_0 and η are significant, while the differences in r and K are not.

seems to be a clear dependence of the estimated parameter values on the experimental initial conditions. Experiments 1–4 (circular initial conditions) in general give lower values for D_0, r, and γ, and higher values for K, compared to Experiments 5–8 (triangular initial conditions). Especially striking is the case of γ, where Experiments 1–4 give estimates in the range of 0.6–1.7, whereas Experiments 5–8 gave estimates either around eight to nine, or above nine (we use $\gamma = 9$ as an upper bound for calculations when searching for the MLE to ensure numerical stability). The Richards model is also the only model that has profile likelihoods which do not have a roughly parabolic shape. For Experiments 5 and 6, the profile likelihoods for γ are bi-modal (but the peaks are close together, and the valley between them lies above the -1.92 threshold, so the confidence region remains a connected interval). This bi-modality can also be seen in the profile likelihoods for the other parameters, but is much less prominent. For Experiments 7 and 8, the profile likelihood functions for γ appear to be monotonically increasing for large γ until the imposed upper bound at $\gamma = 9$, suggesting that the MLE is either very large or tends to infinity.

In the limit of $\gamma \to \infty$, the proliferation function $f(C)$ in Eq. (2.3) is a function that is linear for $0 \leq C < K$, and abruptly drops to zero at $C = K$. In fact, at a moderately high value of γ (say, above seven), the shape of $f(C)$ is already very close to the limiting behaviour, such that an even higher γ makes little difference to either the proliferation function or to the model solution. For this reason, the Richards model is known to be locally structurally non-identifiable in the high γ regime [175]. This explains the divergence to infinity of the confidence intervals for γ for Experiments 7–8.

It is more difficult to explain why the experiments with triangular initial conditions suggest a very large γ, hence nearly linear proliferation, whereas the experiments with circular initial conditions suggest γ is relatively close to one, reflecting dynamics closer to logistic growth. However, there are prior studies making similar observations. In [83], Jin et al. showed that parameter estimates for similar experiments and models depend strongly on the initial cell density, when the shape of the initial cell population is kept constant. In contrast, however, Jin et al. [82] found that the shape of the initial cell density profile does not seem to have a major effect on the estimates for the parameters in the Standard Fisher model, which we have also found to have consistent parameter estimates for circular and triangular initial conditions.

The variability of the parameter estimates in the Porous Fisher and Richards models, and their dependence on initial conditions, is a concern, since we expect the experimental replicates

to represent the same biological processes regardless of the initial conditions. The observed variability could be due to the fact that the more complicated models have a greater propensity to overfit to noise, as we highlighted earlier. Furthermore, note that the high inherent variability in biological processes means that the dynamics driving cell behaviour in each replicate are slightly different. Tissue mechanics offer another biological explanation for the dependence of dynamics on initial conditions. In an empirical study, Ravasio et al. [157] studied wound closure with epithelial cells. During the wound healing process, cells invade inward into a void, which is not unlike in a barrier assay, where cells expand outwards. Both processes involve the same mechanisms of cell proliferation and movement, so it is reasonable to assume that similar biological mechanisms are at play in both scenarios. The study demonstrated that edge geometry has a significant influence on the rate of wound closure. This effect arises from the exertion of force on the cells by supra-cellular acto-myosin cables around the wound edge which, in turn, is regulated by the underlying geometry. The Standard Fisher model might be too simplistic to capture the effects on cell behaviour due to geometry, and therefore is mis-specified. The more complicated models are able to reflect this effect to some extent and therefore partially mitigate against model mis-specification, although without explicitly incorporating tissue mechanics and geometry into the model. More specifically, the curvature of the tissue edge changes more dramatically in the experiments with triangular initial conditions (the straight edges evolve to rounded, convex edges, while the sharp corners are rounded out) compared to the experiments with circular initial conditions (where the curvature only slightly decreases as the circle enlarges). Since cell density is also increasing throughout the experiment, the curvature effect might be reflected in the Porous Fisher model as an apparent stronger dependence of the diffusion coefficient on cell density for the experiments with triangular initial conditions, and therefore a higher estimated η.

For the Generalised Fisher model, we were not able to compute the profile likelihoods for Experiments 5–8, so we only present the parameter estimates for Experiments 1–4 in the last row in Fig. 2.6. For Experiments 5–8, the optimisation procedures struggled to find the correct global maximum compared to Experiments 1–4. The likelihood landscape in this case is much more complex, and the *fmincon*, *globalsearch* and CM-AES algorithms all repeatedly converged to "wrong" local minima, despite many iterations of restarts with improved initialisations to the algorithms. This is because, for reasons that we do not yet fully understand, there are many more (deeper) local minima in the likelihood landscape when the initial condition is a triangle,

perhaps due to a lack of symmetry in the data. This is a difficult computational problem for MCMC methods as well, such as MH, where we found that Markov Chains failed to converge after a very large number of iterations, compared to the number of iterations required for the simpler models. We will discuss computational performance more in Sec 2.4.6.

2.4.5 Model complexity is limited by data resolution

In order to investigate the relationship between parameter identifiability and data resolution, we repeat the profile likelihood calculation on the reduced dataset from Experiment 1, but progressively down-sample the data to reduce temporal resolution by utilising only an equally-spaced subset of the $n_t = 77$ time slices in our data. It should be noted that the timing of the observations, in addition to the number, can affect parameter identifiability. This section considers only equally-spaced observation times for simplicity. By making observations at strategically chosen time points, we might be able to achieve better parameter identifiability compared to using observations chosen at the same number of naively chosen time points. This is further discussed in Sec. 2.5. We will compute the profile likelihoods with $n_t = 39, 26, 20, 13, 7, 3$, which correspond to $\Delta t = 2/3, 1, 4/3, 2, 4, 12$ h, respectively. Related studies on parameter identifiability of cell invasion often had access only to more coarsely-resolved data. For example, [175] used four images taken 16 h apart, and [208] only used three images taken 24 h apart. This means even the lowest data resolution we considered remains within realistic bounds. As data resolution decreases, we expect the uncertainty in parameter estimates to increase, as we will no longer be able to distinguish the model solutions using the transient behaviours in between the snapshots.

We observed three different ways in which the profile likelihood curves change qualitatively as data resolution decreases. We present the profile likelihood for one parameter to illustrate each case, and leave the rest of the parameters to Appendix E.5. In the first case, the peak of the profile likelihood curve (i.e., the MLE) remains mostly in the same location, while the curve itself broadens, but nonetheless remains uni-modal. This is illustrated by D_0 in the Standard Fisher model, shown in Fig. 2.8(a). In this case, despite increases in the uncertainty in parameter estimates, the parameter remains practically identifiable even at the lowest data resolution considered. This is the case for all parameters in the Standard Fisher model, as well as K in all models, and D_0 and r, except in the Generalised Fisher model.

In the second case, as illustrated by γ, a shape parameter in the Richards model in Fig. 2.8(b), the location of the MLE dramatically changes, and the shape of the profile likelihood curve changes in such a way that the confidence interval becomes infinite, making γ non-identifiable in the case where data resolution is low. Interestingly, γ is the only parameter that exhibits this behaviour. The final case is shown in Fig. 2.8(c), where the profile likelihood curve for β in the Generalised Fisher model becomes bimodal when the data resolution is sufficiently low. D_0, r and α in the Generalised Fisher model also exhibit this behaviour.

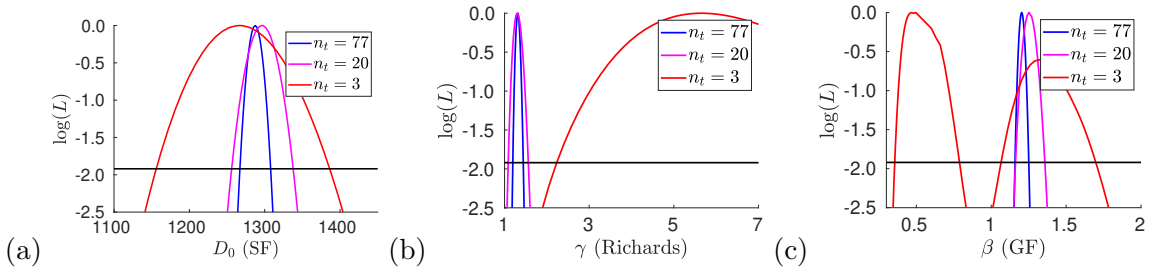


Figure 2.8: Comparison of profile likelihood curves calculated using progressively down-sampled data from Experiment 1, for (a) D_0 in the Standard Fisher model, (b) γ in the Richards model, and (c) β in the Generalised Fisher model. n_t denotes the number of time points used. The blue curve corresponds to $n_t = 77$, i.e. the profile likelihood calculated with all available data, with a temporal resolution of $\Delta t = 20$ min. The pink curve corresponds to $n_t = 20$ and $\Delta t = 80$ min, and the red curve corresponds to $n_t = 3$ and $\Delta t = 12$ h. The black horizontal line at -1.92 is the threshold for the confidence interval. Note that in (b), the red curve remains above the threshold as $\gamma \to \infty$, and we chose to truncate the plot at $\gamma = 7$ for ease of visualisation. The unit of D_0 is $\mu m^2/h$, while γ and η are dimensionless.

Finally, we would like to quantify the dependence of uncertainty in parameter estimation on data resolution. Let $\Delta\theta_i$ denote the width of the 95% confidence interval for the parameter θ_i. From Eqs. (2.16) and (2.12), the profile likelihood function can be written as

$$l_i(\theta_i) = N \log\left(\frac{R(\theta_i^*)}{R(\theta_i)}\right). \tag{2.23}$$

The sum of squared residuals, R, should depend linearly on N, given our assumption that the observational noise at each data point is i.i.d, so overall the profile likelihood function is linearly proportional to N. Furthermore, we can use the approximation in Eq. (2.20), which approximates the profile likelihood function to be a quadratic function in $(\theta_i - \theta_i^*)$. At the edges of the confidence intervals, we have

$$-1.92 = l_i(\theta_i^* \pm \Delta\theta_i) \propto N\Delta\theta_i^2,$$

hence $\Delta\theta_i$ should be proportional to $1/\sqrt{N}$, as long as the approximation used above holds.

Next, we compare the $\Delta\theta_i$ obtained using the recalculated profile likelihoods with the theoretical prediction above by plotting both against the fraction of the total amount of data used in Fig. 2.9, presenting the results for the same three parameters as in Fig. 2.8. Observe that for D_0 in the Standard Fisher model, which remains identifiable at low data resolution, the calculated ΔD_0 remains close to the inverse-square-root predictions as $N \to 0$, while $\Delta\gamma$ and $\Delta\beta$ deviate significantly from the theoretical prediction as $N \to 0$.

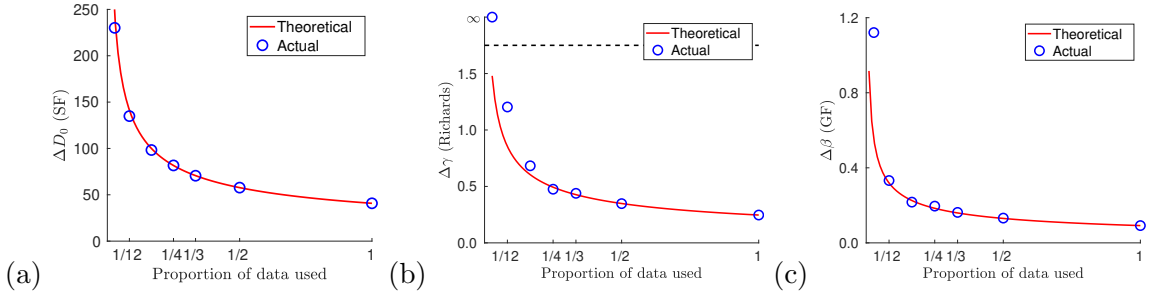


Figure 2.9: The widths of the confidence intervals of three selected parameters plotted against the proportion of data used, for (a) D_0 in the Standard Fisher model, (b) γ in the Richards model, and (c) β in the Generalised Fisher model. Here, using a proportion of $1/m$ of the data means that only one in every m images was used, so $n_t = \lceil 77/m \rceil$. The blue circles represent the widths of the confidence intervals calculated by finding the intersection between the profile likelihood curves and the threshold -1.92, while the red line is proportional to $1/\sqrt{n_t}$ and normalised so that it goes through the point representing the case where all data were used. This red line is the theoretical prediction for how the widths of confidence intervals relate to the data resolution. The unit of D_0 is $\mu m^2/h$, while γ and η are dimensionless.

Similar analysis on the effects of data resolution was also carried out with the synthetic dataset from Section 2.4.1. We found that all parameters in the four models remain identifiable as data resolution is decreased, all the way to the lowest resolution of $n_t = 3$. The profile likelihoods for the parameters in the Standard Fisher and Porous Fisher models remain parabolic, specifically, they are symmetric about the MLEs, and the widths of the confidence intervals are almost exactly proportional to $\sqrt{n_t}$. In contrast, the profile likelihoods for the parameters in the Richards and Generalised Fisher models become skewed when data resolution is sufficiently low (when the number of time points is reduced to $n_t = 3$), that is, the curve is no longer symmetric about the MLEs. In this case, the widths of the confidence intervals are also larger than the inverse square root prediction.

These observations show that parameter identifiability is limited by data resolution. There is a model-dependent minimum amount of data required for the model to be practically identifiable, and this minimum increases as model complexity increases. Therefore, model selection should be tied to the amount of data available and, as a criterion for choosing a model, we should only

use models for which we have sufficient data to make it identifiable, where this sufficiency can be determined using a profile likelihood analysis similar to what was performed in this section.

2.4.6 Computational cost of profile likelihoods is lower than MCMC

Identifiability analysis with either profile likelihoods or MCMC methods is computationally expensive. In this section, we implement the most basic version of MH, described in Sec. 2.3.4, as a benchmark for evaluating the computational cost of profile likelihoods. There are more sophisticated and efficient MCMC methods than MH, some of which are reviewed in [198]. We do not aim to evaluate the performance of MCMC methods in general, but only to compare profile likelihoods to a widely used method, so the choice of MH suffices for this aim. For this comparison, we only consider the Standard Fisher model with the reduced dataset from Experiment 1. In Fig. 2.10, we plot the bivariate profile likelihood function for the pair of parameters (D_0, r) as a heat map, together with the samples from MH projected onto the D_0–r plane. Note that most MH samples are located in regions of high profile likelihood. Furthermore, the sample mean from MH ($D_0 = 1304$ cells/mm^2, $r = 0.2672$ h^{-1}) is close to the MLE ($D_0 = 1302$ cells/mm^2, $r = 0.2675$ h^{-1}), which means the two methods provide similar information on the parameter values.

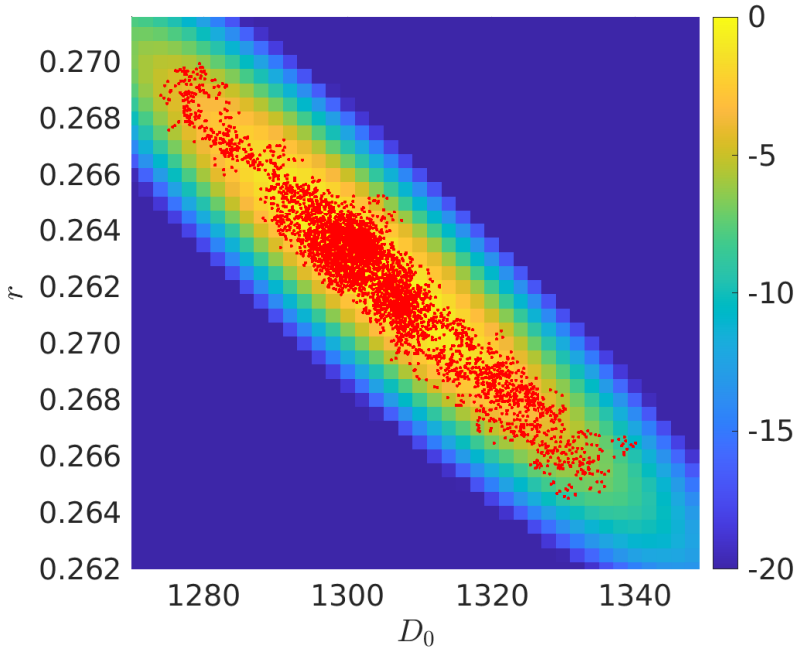


Figure 2.10: Comparison between the bivariate profile likelihood function for (D_0, r) in the Standard Fisher model as a heat map, using the reduced dataset from Experiment 1, and the last 5000 samples from MH before convergence, represented as red dots. The units are $[D_0] = \mu\text{m}^2/\text{h}$, $[r] = \text{h}^{-1}$.

We will use the number of model simulations as a metric for computational performance, since the vast majority of computational time is spent on simulations. The main factor impacting the computational cost of both methods is the dimension d of the search space. The calculation for the MLE, as well as the profile likelihood function at each point, involves an optimisation process in $d = P - m$ space, where P is the number of parameters in the model, $m = 0$ for MLEs, $m = 1$ for single-variable profile likelihood curves, and $m = 2$ for bivariate profile likelihoods. We found that, in the cases where the $fmincon$ algorithm succeeds, which only reliably happens if $d \leq 3$, we typically need 5–30, 40–65, 60–100, 140–250 simulations for $d = 1, 2, 3, 4$, respectively. When $fmincon$ fails to find the global optimum, which often happens for this problem when $d \geq 4$, we have to resort to either the *globalsearch* or CM-AES algorithms, which tend to require of the order of 2000–6000 simulations for $d = 4$. We have found that $fmincon$ is much more likely to get stuck at local optima for datasets from experiments with triangular initial conditions.

In comparison, MH performs a random walk in $d = P$-dimensional space. For the Standard Fisher model, $P = 3$, and MH required 35200 iterations (for each chain) for the R statistics (Sec. 2.3.4) to fall below $R = 1.05$ for all three parameters. It did not fall below the more stringent termination criterion of $R = 1.01$ after 100000 iterations, at which point we stopped the computations. This is much more costly compared to calculating the profile likelihood curves for the Standard Fisher model. We should be careful about interpreting this comparison, however, since there are many factors impacting the performance of both methods that we did not fully test. For example, a well-chosen initial point for the optimisation can substantially speed up the optimisation, while the choice of the transition kernel has a large impact on the convergence of MH.

Prior studies such as [194] have carried out similar comparisons between the performance of MCMC and profile likelihoods with ODE models, using the adaptive parallel tempering algorithm, a more sophisticated algorithm compared to MH, as a representative of MCMC methods. The study found that in low-dimensional, identifiable settings (a system of five ODEs with $P = 5$), there was good agreement between the two methods, and the profile likelihood method is computationally much cheaper, in the sense that it requires only about 1/4 of the CPU time compared to MCMC. The study also found that, for even moderately large models (such as a system of 25 ODEs with $P = 27$), MCMC required less computational time than profile likelihoods, but the quality of the outcome is poorer, in the sense that the parameter values sampled

by MCMC produced worse fits to the data compare to the parameters inferred using profile likelihoods, and the samples are further away from the ground truth. The authors believed that this was due to the slow convergence of MCMC methods for more complex models. The termination criterion used in the study might also be too lenient, resulting in premature termination of the Markov chain before it has converged, which means the MCMC method likely would require similar, if not more time than profile likelihoods, if the termination criterion was suitably strict. In another study, [156] found that MH struggles to converge to the correct posterior distribution when the likelihood landscape is rough, and especially when non-identifiability is present.

The observations for the Standard Fisher model (a relatively simple PDE model) presented above, which found good agreement between the results from profile likelihoods and MCMC, with profile likelihoods requiring less computation, is in agreement with the conclusions from [194]. While we did not try MH for the more complicated models, the likelihood landscapes for those models are likely rough, judging from the way the optimisation routines for calculating profile likelihoods behaved. Therefore, we expect MH to take much longer to converge for those models than for the Standard Fisher model. In this case, we believe that profile likelihoods will still be computationally cheaper, even if more expensive algorithms such as CM-AES have to be used to perform the optimisations, in contrast to the conclusion from [194], which states that MCMC is cheaper for more complicated models. This is likely due to the fact that the authors of [194] chose a lenient termination criterion for the Markov chains, which makes the MCMC method appear to require less computation, but produced inaccurate parameter estimates.

2.5 Discussion

In this chapter, we have carried out an in-depth study of practical parameter identifiability for four related PDE models of cell invasion using the profile likelihood method. By computing the profile likelihoods of the model parameters using synthetic data, we have demonstrated that the profile likelihood method is capable of recovering the true parameter values in the absence of model mis-specification, even in the presence of a realistic level of noise. The univariate profile likelihoods of all model parameters computed from experimental datasets are uni-modal, with the corresponding confidence intervals relatively narrow (except for the Richards model for datasets with triangular initial conditions), suggesting that the parameters are practically identifiable.

We explored the effects of several aspects of data quantity and quality on parameter identifiability. First, the widths of the profile likelihood curves was found to be greater for the reduced dataset compared to the full dataset. This means that the confidence intervals for the parameters are wider, hence we are less certain about the parameter estimates. This is expected since the reduced dataset has fewer data points available for inference. More interestingly, a comparison of the profile likelihoods obtained using the full density profile with the ones obtained using the radially-averaged density profile shows that the parameter estimates obtained for the Standard Fisher model are approximately the same for the two cases, but differ considerably for the more complicated models. This result suggests that simpler models are more robust against subtle changes in data representation. For such models, it is therefore safe to perform inference using the radially averaged dataset to reduce computational costs, whereas the full dataset should be used for parameter inference for the more complicated models, which are more sensitive to changes introduced by data processing. If process noise is believed to be significant, then simpler models should be preferred, since they tend to be more robust against overfitting to the process noise.

We have shown that, for any parameter, the confidence intervals obtained for a single experiment are narrow relative to the spread of the estimated values for the same parameter across experiments. There are two plausible explanations for this. First, due to the high inherent variability in biological systems, the true dynamics behind each experiment replicate may be slightly different. This result suggests that a nonlinear mixed effects framework may be appropriate. In this view, each experimental replicate corresponds to a parameter set sampled from a specified distribution. However, significantly more experimental replicates will be required to determine a suitable form for these random effects. Furthermore, since the initial conditions have a consistent effect on some parameters (e.g., estimates for the value of η in the Porous Fisher model are greater when the initial condition is triangular as opposed to circular), it is reasonable to include the shape as a fixed effect in the models.

Second, the narrowness of the confidence intervals likely stems from an underestimate of the uncertainty in parameter values, due to the fact that the observational errors are spatially and temporally correlated. This correlation is evident from the localised bursts of proliferation activity present in the experiments, which can be seen by visually inspecting the animated data density, provided with alongside the data on Github, also from the clusters of high cell density visible in Fig. 2.5, and from inspecting the fitting residuals directly in Fig. 2.11, which highlights

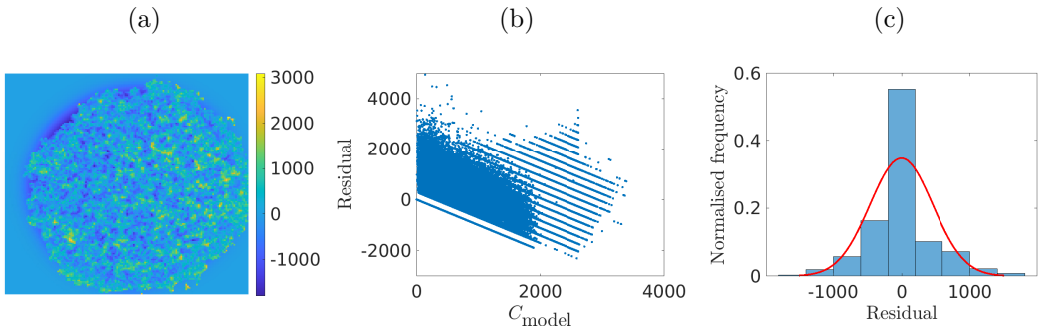


Figure 2.11: Residual analysis for the maximum likelihood fit of the Standard Fisher model, highlighting a few known issues with the noise model. In these plots, C_{data} is the dataset from Experiment 1, and C_{model} is produced using the MLE parameter values obtained for the same dataset.(a) Snapshot of Residual ($C_{\text{data}} - C_{\text{model}}$) at $t = 20$ h. The bright and dark spots indicate the places where the model under/over estimates the cell density, respectively. These bright or dark spots are localised, implying spatial auto-correlation. Moran's I test [132] gives an auto-correlation score of 0.2208 between the residual at adjacent points, and finds that the positive spatial auto-correlation is statistically significant. (b) The residuals across all times, plotted as a function of cell density predicted by the model. In regions where the cell density is very low, or zero, the model can give rise to negative cell densities (specifically, when the model predicts a cell density near zero, the error model still allows a negative observational noise, which is un-physical), and these will not be observed experimentally. Therefore, the residual can depend on cell density. The quantification of cell density is carried out by counting cell nuclei centroids inside square boxes that match the spatial discretisation of the PDE model. As a result, the cell density can only take a discrete set of values, which explains the diagonal stripes. This means that the residuals are not normally distributed. (c) Distribution of the residuals. The residuals are significantly more concentrated near zero compared to a fitted normal distribution, represented by the red curve. A Lilliefors test [115], a statistical test based on the Kolmogov-Smirnov test for determining whether an empirical distribution is normal, shows that the residuals are not normally distributed. This can be explained by the fact that since the cell population covers a small part of the domain for most of the experiment, there are large number of points where the cell density is zero, and the model tends to correctly predict a very low density at these points.

a number of known issues in the noise model. Auto-correlated noise means we have fewer (effective) data points than we naively believe, leading to over-confident parameter estimations. This possibility is discussed in [109], which discusses the fact that auto-correlated measurements, combined with an assumption of i.i.d. observational errors, can lead to an underestimation in parameter uncertainty in an ODE context. A potential solution to this problem is to use a more realistic noise model where the observational noise is allowed to have spatial and temporal auto-correlation, by prescribing a correlation kernel with additional parameters to be fitted. This approach has also been discussed in [68] in the context of generalised linear models. In [152], the authors considered a variety of noise models for the Gompertz model, a well-known ODE model for tumour growth. The study showed that allowing the observational errors to be dependent on the size of the tumour may improve the performance of model fitting, as measured by the

BIC. Another approach that could be taken is to avoid making assumptions on the noise by calibrating a generalised profile likelihood function with bootstrapping, which mitigates the effect of inaccurate assumptions on the noise model. The details of this approach are discussed in [203]. Alternatively, stochastic models might be more suitable in cases where process noise is significant. However, inference for stochastic models introduces computational challenges, making them unwieldy for many situations.

For the more complicated models, we observed a dependency of the parameter estimates on the experimental initial conditions. Specifically, for the Porous Fisher and Richards models, the parameter estimates obtained using datasets with circular initial conditions are remarkably different from those estimated using datasets with triangular initial conditions. For the Generalised Fisher model, the optimisation process for computing profile likelihoods is much more difficult if we use a dataset with triangular initial conditions rather than circular initial conditions, which suggests that the profile likelihood landscape is more rough. For the Standard Fisher model, we observe less dependency of parameter estimates on initial conditions, with the variations between MLEs from the different datasets roughly in line with the observations of [82], where the authors fitted the Standard Fisher model to data from a cell invasion assay done using several different initial conditions.

This observed dependency is at odds with the assumption that the only mechanisms driving tissue expansion are cell motility and proliferation, and that the only factor modulating these mechanisms is cell density. A likely explanation for this dependency is that mechanical properties of the tissue may play a significant role in cell invasion. As such, an interesting avenue for future research entails including mechanical effects at the leading edge in the models. Furthermore, this observation shows that experimental initial conditions can impact parameter identifiability. Originally we expected the datasets with triangular initial conditions will lead to narrower confidence intervals compared to those with circular initial conditions, since the relative lack of symmetry in the geometry allows us to observe cells in more distinct situations (for example, at the corner versus the straight edges), therefore providing more information on cell behaviour. While the results are more complicated than the expectation, it nonetheless raises the possibility of designing optimal initial conditions that provide the most informative data for parameter inference. This idea is explored in Chapter 3.

Our findings reveal the pivotal impact of data resolution on parameter identifiability. As the amount of available data decreases, the confidence intervals for all parameters widen. Some

parameters remain practically identifiable even at low data resolution, and the uncertainty in their estimates, measured by the width of the confidence interval, increases proportional to the inverse square root of data quantity. However, other parameters can become non-identifiable if the data resolution is sufficiently low, and the size of their confidence region (which might no longer be a connected interval) increases dramatically. This observation indicates that each model has its own minimum data requirement for practical identifiability, which tends to be higher for more complicated models.

We have also shown that the profile likelihood method provides much more information about identifiability compared to methods based on the FIM. A more traditional, and less computationally intensive, method for quantifying uncertainty in parameter estimates is to approximate the parameter distribution as a joint Gaussian distribution centred at the MLE, with the covariance matrix derived from the FIM. However, this method essentially assumes a quadratic approximation to the likelihood function which, as we have seen, can be far from valid in non-identifiable cases. The performance of such methods for uncertainty quantification is compared with profile likelihoods in [194].

A caveat of the profile likelihood is that it relies on the user to specify the range of parameter values on which the profile likelihood is computed and visualised. In cases where the true profile likelihood is multi-modal, which tends to be the case for models that are structurally locally identifiable but not globally identifiable, one or more peaks of the profile likelihood might lie outside of the specified range. In this case, there might only be a single peak within the specified interval, so a profile likelihood-based analysis can fail to reveal non-identifiability. This issue is not unique to profile likelihoods, as MCMC-based sampling methods can get stuck near a peak of the likelihood function for a long time before exploring other peaks. This can mislead the user into terminating the sampling process early before the other peaks are found. This problem can be mitigated by using existing knowledge of the biological system to inform the likely range of values a parameter might take, and ensure that this range is explored in its entirety.

Finally, we return to the problem of model selection. For most of our datasets, the AIC and BIC select the same model as each other, which is usually the Richards or the Generalised Fisher models (see Appendix E for details). However, as we have seen, these models require more data to be identifiable, and they are also more sensitive to subtle changes in the data introduced by data processing (e.g., full density profile versus radial averaging). Simply increasing the penalty for model complexity in the AIC or BIC does not solve the problem, since identifiability cannot be

properly accounted for by counting free parameters. Further, as we have seen, the Porous Fisher and Richards models have the same number of parameters, but the Porous Fisher has better practical identifiability. Therefore, we argue that model selection methods should also take the of availability data into account. For the problem considered in this work, more complicated models can be adopted if there are sufficient data to render them practically identifiable, whereas the Standard Fisher model should be favoured if the data resolution is low, or if we have only radially averaged data.

We would also like to address the distinction between parameter identifiability, which can be thought of as uncertainty quantification in parameter values, and the related notion of model sensitivity and prediction uncertainty, and the implications of each notion on model selection. Parameter uncertainty is not always mirrored in prediction uncertainty, which is, in fact, what drives practical identifiability issues. On the one hand, practical identifiability means the predictions of the model are sensitive to changes in parameter values, which, in turn, entails that the resulting posterior distributions and posterior predictive distributions are narrow. On the other hand, a practically non-identifiable model is insensitive to changes in parameter values, which implies that the predictions are robust to changes in parameter values. Therefore, it is difficult to determine, *a priori*, the relationship between identifiability and prediction uncertainty.

As such, both identifiability and prediction uncertainty are important in model selection, but the relative weight placed on each in making decisions depends on the particular focus of the modelling study. For example, if the goal of a modelling study is to provide quantitative insights into a particular biological mechanism, then that mechanism must be included in any candidate models considered. Identifiability, especially that of the parameters associated with the mechanism in focus, is hugely important in this case. If all models incorporating the necessary mechanisms are non-identifiable, then a practitioner should aim to use the models to design experiments that, when implemented, resolve the identifiability issues and thereby provide mechanistic insights. Furthermore, in cases where model parameters have concrete physical or biological meaning, it might be relevant to use these parameters in other studies. Precise parameter estimates are crucial for the validity of such approaches. However, for the purpose of investigating possible qualitative mechanisms to explain a biological phenomenon, a certain degree of non-identifiability is acceptable, and arguably desirable in certain biological applications, since it reflects robustness. Finally, while confidence in model predictions is clearly central for making predictions, identifiability remains important, since non-identifiability can lead the

modeller to misplace their confidence in the reliability of model predictions in alternative scenarios.

2.5.1 Future work

There are several ways to extend the investigation into the effects of data on parameter identifiability. One question relevant to experimentalists is how long should an experiment be run. Running the experiment for longer provides more data, which is in theory beneficial for parameter inference. However, for cell invasion assays, running the experiment for too long causes cells to form structures with multiple layers, instead of an orderly monolayer, which changes the behaviour of the system, making modelling difficult, as well as complicating the assessment of cell density, since some cells will no longer be visible to the camera. Therefore, it would be interesting to evaluate the dependence of practical identifiability on the time frame over which data is available. Such analysis was done for the Standard Fisher model in [201].

One weakness of the models used in this chapter is the over-simplistic noise model, which could be improved upon in future works. Given that the data suggest structural noise present in the system, a reasonable approach is to use stochastic models, such as agent-based models or SPDEs. Such models allow us to deal with process noise, at the cost of making inference much more computationally expensive. Instead of being able to evaluate the likelihood function with a single model simulation, a stochastic model must be simulated many times to give an accurate estimate for the likelihood. Methods such as approximate Bayesian computation and rejection sampling are commonly employed in the field to circumvent this difficulty. It would be interesting to see how data resolution and experimental initial conditions affect parameter inference and identifiability in this context.

Another direction for future research is that in addition to cell density, we additionally have access to information on the stage of the cell cycle for each cell in the experiment obtained using the FUCCI reporter technique. This would allow us to modify the model by adding a more detailed description of the proliferation mechanism involving the cell cycle, at the cost of increasing model complexity and the number of unknown parameters. A model for cell movement and proliferation incorporating the cell cycle has been proposed in [56]. It would be interesting to see whether introducing cell cycle mechanisms, along with additional data, allows us to better understand the cell invasion process, or if the additional complexity causes non-identifiability and hence prevents us from gaining any additional insight.

While we have investigated the effects of data resolution on parameter identifiability with equally-spaced measurement times, questions remain on how the timing of the measurements affects parameter identifiability. The sensitivity of the model solution to a particular parameter varies both over the time-span and the spatial domain of the experiment. Intuitively, for the purpose of inferring the value of a particular parameter, it would be more effective to make additional measurements at the points where the value of that parameter has the greatest impact on the measured outputs. This will be demonstrated in the context of an ODE model in Sec. 3.3. It would be interesting to investigate this effect in a PDE model.

While we have compared the computational cost of profile likelihoods with a basic MCMC method, there are more sophisticated MCMC methods that may perform better in certain situations. In the future, it is important to determined when profile likelihoods can be feasibly deployed, and when might we prefer MCMC methods instead. The wider adoption of profile likelihoods in the scientific community depends on the development of more efficient optimization algorithms, which is another direction for future research. It may be possible to develop more efficient and specialised algorithms to compute profile likelihoods by leveraging the structure of the likelihood function and the PDE model.

In the next chapter, we maintain the focus on parameter identifiability, but pivot to explore it through the lens of experimental design rather than model selection. Specifically, we investigate ways of optimally designing an experiment to maximise the informativeness of the data it provides for the purpose of parameter inference.

Chapter 3

Optimal experiment design for improving parameter identifiability

In Chapter 2, we emphasised the critical role of parameter identifiability in understanding the underlying mechanisms that govern the behaviour of biological systems. It was demonstrated that practical parameter identifiability is a property of the combination of the model and available data. In certain scenarios, however, the data might be insufficient to uniquely determine a set of parameter values that mirrors the observed behaviour. While the previous chapter addressed this challenge by arguing for the selection of an identifiable model, this chapter shifts the focus towards the data side of the problem. Specifically, we aim to find ways to optimally design an experiment, so that the data it generates are most effective for pinning down the parameter values and selecting the most appropriate model.

This chapter explores two objectives for experimental design. The first objective, improving the identifiability of a single model, revolves around a specific model with unknown parameters, where the objective is to minimise the width of the confidence intervals for parameter estimates obtained through profile likelihoods. The second objective, model discrimination, is concerned with two candidate models, each with a fixed set of parameters obtained through model fitting. Here, the aim is to maximise the disparity between the solutions of these two models, which would enable the user to select between them. For each objective, we consider designing two aspects of an experiment. The first centres on the strategic selection of an optimal initial condition for the experiment in order to maximise identifiability. The second aspect involves augmenting the experiment with a control variable, designed to allow for the observation of the system's behaviour across a broader range of conditions, consequently providing more information.

We employ growth models in the form of ODEs for our study, and illustrate how a judicious choice of initial conditions and control can greatly improve the identifiability of a model and aid

model selection. We also demonstrate how to formulate and solve the associated optimisation problems.

3.1 Introduction

In Chapter 2, we highlighted the importance of practical identifiability of a model and its relation to the problem of model selection. We also quantified the effects of various aspects of the model and the data on practical identifiability. In particular, we found that the initial conditions and the frequency of data collection significantly influence the identifiability of model parameters. This naturally prompts the question: how can we enhance identifiability, and aid model selection? While Chapter 2 explored the option of using a less complex model with better identifiability, this chapter focuses on the data aspect: can we design an experiment to yield the most informative data? This concept dates back over a century, as discussed in [104], with [76] serving as an early example. Earlier studies primarily focused on statistical models (such as regression), while the transition to dynamical models (like ODE models) is a relatively recent development. In mathematical biology, the ideas of experimental design have been applied to models of signalling pathways [13, 46], gene regulatory networks [179], and yeast fermentation [54], for PKPD models to study the effects of drugs [178], and also in physics for models of crystallisation [42] and polymerisation [32].

There are multiple ways to interpret the question of experimental design. In particular, how to define the "most informative" experiment, and how to measure the performance of a certain experimental design? The work in this chapter distinguishes itself from existing work by designing more complex aspects of an experiment, such as modulating an external stimuli with a more general control to allowing for greater flexibility and potentially improved performance, as well as using parameter identifiability calculated from profile likelihoods to evaluate an experimental design, which could lead to more informative experiments in practice.

3.1.1 Design for parameter identifiability

The meaning of "most informative" depends on the objective. Parameter estimation and model selection, inherently intertwined, can both benefit from careful experimental design. We can tailor the experiment to enhance parameter identifiability, allowing for more precise parameter estimation. Alternatively, we can optimise the experiment for model discrimination, aiming to maximise the discrepancy between the outputs of multiple models. This discrepancy makes it

easier to distinguish between the models given data, and to select the model that more accurately represents the biological system. Although these two objectives share a connection, they are nonetheless distinct. Previous authors recognised this connection, as reflected in numerous papers that discuss both simultaneously [54, 104]. However, it is worthwhile to elaborate on the similarities and distinctions between them.

Optimising for identifiability revolves around a single model, where parameter values are initially unknown. The objective is to find an experimental design in which these parameters have the greatest impact on the model's output, where a slight perturbation to parameter values causes significant changes in the model's output. Consequently, this sensitivity in model output allows us to more narrowly constrain regions of plausible parameter values, therefore reducing uncertainty in parameter estimates.

In this case, the performance of an experimental design is measured by the uncertainty in parameter estimates. Each framework of parameter estimation gives different ways to measure uncertainty. First, the frequentist Maximum Likelihood approach gives rise to the FIM $\mathcal{I}$ (Eq. (2.15)) which, as discussed in Sec. 2.3.2 and B.2, reflects the local curvature of the likelihood function at the MLE, with larger eigenvalues implying a sharper likelihood function, hence less uncertainty in parameter values. Based on this, some reasonable design objectives are to maximise the trace of $\mathcal{I}$, or the determinant, or the smallest eigenvalue. These are referred to as the A-, D-, and E-optimal criterion, respectively [57, 104], although in the literature these criteria are often formulated in terms of the covariance matrix $\mathcal{C} = \mathcal{I}^{-1}$. Second, in the framework of Bayesian inference, the variance and entropy (and its derived variants) of the posterior distribution of parameter values are all reasonable measures of uncertainty, as used in [190].

Lastly, for profile likelihoods, we can use the widths of the confidence intervals for parameter values (with a fixed confidence threshold) as a measure of uncertainty, as we have done in Sec. 2.3.3. More complicated measures, such as the one defined in [116], avoid the need to choose a confidence threshold, albeit having a much higher computational cost. The use of profile likelihoods for experimental design has been examined in [179], but its use is less widespread than the other two approaches.

In this chapter, we will employ the profile likelihood approach, with the width of the 95% confidence interval serving as the measure of uncertainty to be minimised. The reason behind selecting this approach is the same as that of the previous chapter: profile likelihoods strike a good balance between providing information about the parameters and computational cost.

The FIM-based measures, by far the most commonly seen approach in literature, rely on point estimates and implicitly presume that the likelihood function approximates a parabolic shape (or, in higher dimensions, an elliptical oval). This assumption does not hold when the parameters are non-identifiable, making the FIM an inaccurate measure of parameter uncertainty. In the structurally non-identifiable case, the FIM is singular, which means the method cannot be directly used without adaptation. Bayesian inference provides a much more accurate representation of the distribution of parameter values, but its computational cost is too high. The profile likelihood approach compromises between the two.

3.1.2 Design for model discrimination

In contrast to the problem of improving parameter identifiability, model discrimination involves two or more models. We assume that these models have fixed parameter values, possibly estimated from earlier experimental data. The models may share the same equations but with different parameter values. Hence, it is more accurate to refer to this objective as discriminating between model-parameter combinations. For this chapter, we will focus on the case where we seek to discriminate between a pair of two model-parameter combinations with shared equations. This approach is general, since two models that do not share equations can be viewed as special cases of a single, more general model with different parameter values (for example, both the Richards and Porous Fisher models in Chapter 2 can be viewed as special cases of the most general model given in Eq. (2.3)). Here, our focus lies less on the uncertainty in parameter estimates or the models' behaviour across a continuous parameter space, but focuses instead on discerning the differences in model behaviours at two fixed points within the parameter space. Using a well-designed experiment, we would be able to decide which one better reflects the underlying biology, which we would not be able to do if the behaviours of the two models are indistinguishable under a poorly designed experiment.

The simplest measure of difference between the two models is the integral of the squared difference between model solutions over time, which we would like to maximise in order to distinguish between the models. This idea was proposed in [76] for statistical models, which we adapt to dynamical models. Later, more sophisticated measures were proposed to take the variance from the observational noise into account, with three such measures reviewed in [32], but it was found that these more sophisticated measures do not result in significant differences in

experimental design [8]. In this chapter, therefore, we use the integral of the squared difference for simplicity.

Conceptually, the problem of eliminating certain kinds of practical non-identifiability can be reframed as a model discrimination problem. For instance, consider a scenario where identifiability analysis reveals a model to be practically non-identifiable, where the profile likelihood is multi-modal. Each parameter set associated with a local maximum of the profile likelihood function represents a plausible hypothesis concerning the underlying biological mechanism. We can take the same model, with parameter values corresponding to the highest local maxima of the profile likelihood as the model-parameter combinations to be discriminated, and design an additional experiment to eliminate one of these parameter sets, which could potentially make the overall model identifiable.

Alternatively, we may wish to reduce the uncertainty of parameter estimates for a model that is already found to be practically identifiable, given data from an initial experiment. We can take the same model with the parameter set corresponding to the MLE, and a parameter set at the edge of the confidence interval, as two model-parameter combinations to be discriminated. An experiment designed to maximally discriminate between these two model-parameter combinations allows us to focus on the parameter space near the MLE, providing a more detailed understanding of how the model behaviour depends on parameter values in that specific region. Consequently, this leads to more precise parameter estimates and improved identifiability when combined with data from the earlier experiment.

In biological research, it is common practice to iteratively design an experiment, estimate the parameters, perform identifiability analysis, then design subsequent experiments [57, 63, 104]. This cyclic process aims to achieve more accurate model parameter estimates than the previous iteration. The problem of improving identifiability, as formulated here, marks the first iteration of such a process, which should be done before the first experiment. The problem of model discrimination, on the other hand, relies on prior experiments to provide fixed sets of parameter values, so it is an approach suitable for the later iterations. Overall, these two problems can be viewed as sequential steps within a larger pipeline of procedures.

3.1.3 Aspects of experimental design

Now that we are clear about the objectives, we ask the question of what aspects of the experiment can be changed for optimisation? For a dynamical model in mathematical biology, these aspects

include the initial conditions of the experiment, the time span, the measurement or observation times and frequency, the observation procedure and what to observe, any external stimuli to the experiment, and more. Each of these has received some attention in the literature. For example, Kreutz and Timmer [104] and Wieland et al [205] discussed choosing the optimal timing of observations for ODE models in mathematical biology in general. Raue et al [153] discuss the most useful quantity to measure in an experiment when there are a large number of quantities that can potentially be measured, in the context of protein binding. Deciding on what quantities to measure is also an important part of experimental design. In many cases, it can be difficult and costly to measure certain variables in a model, therefore not all variables are measured. For example, in a model for glucose-insulin interaction [44], insulin excitable tissue is usually not measured during experiments due to technical constraints. Such "missing" measurements can lead to identifiability issues, since we are not then able to use the trajectories of the unobservable quantities to constrain parameter values.

In the works that discuss the design of external stimuli on the system, which are formulated as control variables in the model, the control input is often restricted to take simple forms. For example, Bandara et al. [13] restricted the control to be step functions, and Faller et al. [57] parameterised the control as a low degree polynomial, where only a small number of coefficients in the polynomial need to be determined. While these restrictions can be reasonable in experiments where the ability to precisely manipulate the stimulation is limited, the availability of modern instruments that allow for much more precise control over elements of the experiment means it is appropriate to lift some of these restrictions and consider a more general class of controls. This chapter improves upon existing results by considering both continuous control, where the control can be arbitrary piecewise continuous functions of time, and "bang-bang" control, where the control is restricted to a small number of discrete values, usually two. The term "bang-bang" is used in [19] to emphasise the discontinuous nature of such control.

Given the many possible aspects of experimental design, we are not be able to discuss all of them in this chapter. We choose to focus on designing the initial conditions of the experiment, and the external stimuli to the experiment. Previously, we discussed measurement frequency, and touched on measurement procedure, in Chapter 2. The problems of deciding what to measure, the timing of the observations, and designing the time span of the experiment are left to future research. Consequently, the models used in this chapter are fully observable, with

the observations made at n_t evenly spaced points in time Δt apart, within a fixed time span $T = n_t \Delta t$, and both n_t and Δt are fixed.

3.1.4 Overview

In this chapter, we seek to achieve two objectives (improving identifiability of a single model, and discriminating between a pair of models) by optimally designing two aspects of the experiment (initial conditions and control), resulting in four problems to solve. We use models of cell growth, specifically for MDCK cells, as a case study to illustrate the objectives. This choice is motivated by the fact that MDCK cells are the same cell type studied in the experimental data used in Chapter 2. This allows us to apply what we have learned so far about these cells to guide experimental design efforts. These models are also used to describe the growth of tumours, for which determining the most appropriate model remains ongoing research [66]. We hope the results in this chapter will make it possible to design better experiments for investigating the proliferation of tumours. The models used in this chapter are explained in Sec. 3.2. Sec. 3.3 discusses finding the optimal initial condition to improve identifiability, while Sec. 3.4 does the same for model discrimination. These two sections use the Richards growth law as the model in focus, which is the non-spatial version of the Richards model in Chapter 2.

Sec. 3.5 and 3.6 discuss designing the optimal control for improving identifiability and model discrimination, respectively. For these two sections, the Richards growth law is modified with the addition of an explicit death term and control. The addition of the death term makes the model structurally non-identifiable in the absence of control, which creates an opportunity to illustrate how the control can be useful in eliminating this non-identifiability. The death term is not included in the model for the earlier sections because changing the initial conditions will not help in structurally non-identifiable cases. Finally, the results are summarised, and their significance is addressed in Sec. 3.7.

3.2 Mathematical models

We use the Richards growth law with an additional term for cell death to illustrate experimental design principles. This model can be written as

$$\frac{\mathrm{d}C}{\mathrm{d}t} = rC\left[1 - \left(\frac{C}{K}\right)^{\gamma}\right] - \delta C = f(C; (r, \delta, \gamma, K)), \quad C(0) = C_0, \tag{3.1}$$

where $C(t)$ represents cell density, $r > 0$ the growth rate, $K > 0$ the carrying capacity, $\gamma > 0$ the shape parameter, and $\delta \geq 0$ the death rate. We also stipulate $r > \delta$, so that the net

growth rate is positive. Throughout this chapter, the following units will be used for reporting the values of the variables and parameters in this model:

$$[C] = [K] = \text{cells/mm}^2, \quad [t] = \text{h}, \quad [r] = [\delta] = \text{h}^{-1}, \quad [\gamma] = 1. \tag{3.2}$$

The original Richards growth law [159] is the special case with $\delta = 0$:

$$\frac{\mathrm{d}C}{\mathrm{d}t} = rC\left[1 - \left(\frac{C}{K}\right)^{\gamma}\right] = f(C; (r, 0, \gamma, K)), \quad C(0) = C_0. \tag{3.3}$$

The addition of the death term makes the model structurally non-identifiable, since if we let

$$r_{\text{eff}} = r - \delta, \quad K_{\text{eff}} = K\left(1 - \frac{\delta}{r}\right)^{1/\gamma} = K\left(\frac{r_{\text{eff}}}{r}\right)^{1/\gamma}, \tag{3.4}$$

then the model with the death term is equivalent to the model without the death term, with a change of parameter values:

$$f(C; (r, \delta, \gamma, K)) = f(C; (r_{\text{eff}}, 0, \gamma, K_{\text{eff}})).$$

This can be used to show that, given any parameter set, there is another parameter set, obtained by perturbing the original, so that both parameter sets produce the exact same model solution, making the model structurally non-identifiable. As in Chapter 2, we will use $\boldsymbol{\theta} = (r, \delta, \gamma, K)$ to denote the vector of parameters in this model. It can be shown via the Taylor series approach that the Richards model without an explicit death term (i.e., $\delta = 0$ fixed) is at least locally identifiable. With the explicit death term, the parameter γ, and the parameter combinations $r - \delta$ and r/K^{γ} are at least locally identifiable, but the parameters r, δ, K individually are structurally non-identifiable. The details of the analysis are presented in Appendix F.

There are biologically relevant reasons why we might introduce an explicit death term. For example, it is plausible that the proliferation rate is density-dependent while the death rate is density-independent, or we might be interested in estimating both parameters separately, such as in [26], where the authors used a linear decay term to represent the necrosis of tumour cells, separate from the logistic growth term. In this chapter, we aim to see if the addition of control can eliminate the non-identifiability in this model.

This model (Eq. (3.1)) has a closed-form solution:

$$C_{\text{model}}(t; \boldsymbol{\theta}, C_0) = \frac{K_{\text{eff}} C_0}{[C_0^{\gamma} + (K_{\text{eff}}^{\gamma} - C_0^{\gamma})\exp(-\gamma r_{\text{eff}} t)]^{1/\gamma}}, \tag{3.5}$$

where K_{eff} and r_{eff} are as given in Eq. (3.4). Setting $\gamma = 1$ reduces the model to the logistic model. We will use the exact solution in our calculations whenever possible.

We further expanded the model to incorporate three possible controls: $u_r(t)$, $u_\delta(t)$, and $u_K(t)$. They represent a change in the corresponding parameter as the result of an experimental intervention. For example, u_r can represent the addition of certain signalling proteins that accelerate the cell cycle, therefore speeding up growth; u_δ can represent the addition of a toxin that kills cells, therefore increasing the death rate; and u_K can represent a change in the amount of nutrient provided to the cells, therefore causing a corresponding change in the carrying capacity. We choose to let positive u_K represent a decrease in carrying capacity, since in the cases we considered, decreasing the carrying capacity is more effective for model discrimination or improving identifiability than increasing it. Note that this form of control problem, where the model parameters are perturbed by the control, is somewhat unusual compared to the more commonly seen form where the control perturbs the ODE variables directly (e.g., $\mathrm{d}C/\mathrm{d}t = f(C;\boldsymbol{\theta}) + u(t)$). In the context of the cell growth models considered here, it is more practical to modify the model parameters than to modify the cell density. The latter would involve adding or removing cells during the course of the experiment, which is difficult to do precisely. In contrast, it is comparatively easier, although not trivial, to add a precise amount of nutrient or toxin to the cell culture to modify the model parameters.

We consider two possible formulations of incorporating the control variables into the model, where the effects of the control variables on the associated parameters can be either additive or multiplicative. In the first formulation, the control variable modifies the corresponding parameters additively. The model can be written as

$$\begin{aligned}\frac{\mathrm{d}C}{\mathrm{d}t} &= (r + u_r(t))C\left[1 - \left(\frac{C}{K - u_K(t)}\right)^\gamma\right] - (\delta + u_\delta(t))C \\ &= g_a(C;\boldsymbol{\theta}, u_r(t), u_\delta(t), u_K(t)).\end{aligned} \tag{3.6}$$

The units of the controls are the same as their corresponding parameters. In the second formulation, the control variable modifies the corresponding parameters multiplicatively, giving the model

$$\begin{aligned}\frac{\mathrm{d}C}{\mathrm{d}t} &= r(1 + u_r(t))C\left[1 - \left(\frac{C}{K(1 - u_K(t))}\right)^\gamma\right] - \delta(1 + u_\delta(t))C \\ &= g_m(C;\boldsymbol{\theta}, u_r(t), u_\delta(t), u_K(t)).\end{aligned} \tag{3.7}$$

The controls are dimensionless in this formulation. In practice, which formulation is appropriate depends on knowledge of the way the controls act on the parameters. Despite a small conceptual difference, these two formulations differ significantly in their mathematical structure. In the

multiplicative formulation, the effect of the control input on model dynamics is tied to the base value of the corresponding parameter, and the unit of the control input is $[u] = 1$. In contrast, in the additive formulation, the control modifies the corresponding model parameter by a constant amount, and has the same unit as the corresponding parameter. As we will see, the effects of these controls, and therefore the optimal control, differ significantly between these two formulations.

3.3 Optimal initial conditions for improving identifiability

In this section, we explore how to choose experimental initial conditions to enhance the parameter identifiability of a single model, in the absence of controls or the death term (i.e., we fix $\delta = 0$). All other aspects of the experiment, e.g., the time span T and the number of data points n_t, are fixed, and no external stimulus is present. We use synthetic datasets to evaluate parameter identifiability. These datasets are generated by perturbing the exact solution of the model (Eq. (3.5)) with a Gaussian noise, independently and identically distributed at every time point:

$$C_{\text{data}}(t_i) = C_{\text{model}}(t_i; (\hat{r}, 0, \hat{\gamma}, \hat{K}), C_0) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2), \tag{3.8}$$

where $t_i = i\Delta t$ are the discretised time points, $i = 0, \ldots, n_t$, and the hat over a parameter denotes the parameter value used to generate the data, which will be given later. Note that, to perform experimental design in this way, we must make certain assumptions on the parameter values to generate the synthetic data. These assumptions can be informed by parameter estimates from earlier experiments, which we would like to improve, or by similar biological processes that have been modelled or studied before. If none of the above are available, as often is the case for a novel biological system, it then might be necessary to perform a naively designed experiment first, to gain some limited insight into the parameter values before designing the subsequent experiments. After generating the datasets, we then compute the univariate profile likelihoods $l(\theta)$ for each parameter θ to obtain a 95% confidence interval for this parameter using the method described in Sec. 2.3.3, with $\Delta\theta$ denoting the width of the confidence interval. We use MATLAB's *fzero* routine, which is a root-finding routine implementing a combination of bisection search and secant algorithms, to find the boundaries of the confidence interval, where the profile likelihood function intersects $l(\theta) = -1.92$, the likelihood threshold for univariate confidence intervals. We initialise the root-finding algorithm at $\theta = 0$, and $\theta = \theta^*$ the MLE, to find the lower bound of the confidence interval. This is because we expect the MLE to be within the confidence interval,

and all parameters in the model are restricted to be non-negative, so the lower bound of the confidence interval must lie between these two values. We search between $\theta = \theta^*$ and $\theta = \theta_{\max}$ to find the upper bound, where we choose $\theta_{\max}$ to be larger than any biologically realistic value. If the profile likelihood function lies above the threshold at $\theta_{\max}$, then we consider the upper bound of the confidence interval to be infinite, consequently the width of the confidence interval is also taken to be infinite. We take $r_{\max} = 1$ h^{-1}, $\gamma_{\max} = 9$, $K_{\max} = 10000$ cell/mm^2, coinciding with Eq. (2.14), which are comfortably above the biologically realistic range of these parameters for MDCK cells, especially compared to values given in [83, 175] and found in Chapter 2.

Finally, given a parameter θ, we can visualise the optimal initial condition for estimating this parameter, that is the one resulting in the narrowest confidence interval, by plotting $\Delta\theta$ against C_0. As we will show, the optimal initial condition can vary depending on the parameter.

3.3.1 Logistic model

For the logistic model (Eq. (3.3) with $\gamma = 1$), we use the following parameter values to generate the synthetic data:

$$\hat{r} = 0.3 \text{ h}^{-1}, \quad \hat{K} = 2600 \text{ cells/mm}^2, \quad \sigma = 400, \quad \Delta t = 0.5 \text{ h}, \quad n_t = 50, \quad T = 25 \text{ h}. \tag{3.9}$$

These values are chosen to approximate the parameter values estimated from real data in Ch. 2. While the quantitative results might change, the qualitative results are not sensitive with respect to the parameter values chosen in Eq. (3.9). Recall that the parameters with hats denote the true values of the corresponding parameters used to generate the synthetic data, and n_t is the number of data points. These parameter values are chosen to be similar to the net growth rate and carrying capacity estimated from the data from Experiment 1 in Chapter 2 (see Table 2.3). Assuming the parameters in the non-spatial model (Eq. (3.3)) have similar values to the corresponding parameters in the spatial model (Eq. (2.3)), these values chosen in Eq. (3.9) are biologically realistic. Here only the parameters r and K need to be inferred from data, whereas $\delta = 0$ and $\gamma = 1$ are fixed, and the nuisance parameter σ can be estimated separately, as explained in Sec. 2.3.2. Note that the model with both δ and γ fixed is structurally globally identifiable, as found by the analysis in Appendix F.

In Fig. 3.1, we present the profile likelihoods for both r and K, given a range of initial conditions. The corresponding MLEs and correlation between parameter estimates are given in Table 3.1. Observe that, for an initial condition of $C_0 = 100$ cells/mm^2, both r and K are identifiable with relatively narrow confidence intervals compared to other cases. With a much

lower initial condition of $C_0 = 1$ cells/mm^2, r remains identifiable, although the confidence interval widens, but K becomes non-identifiable. The opposite is true for $C_0 = 2300$ cells/mm^2, when K remains identifiable with a wider confidence interval, but r becomes non-identifiable.

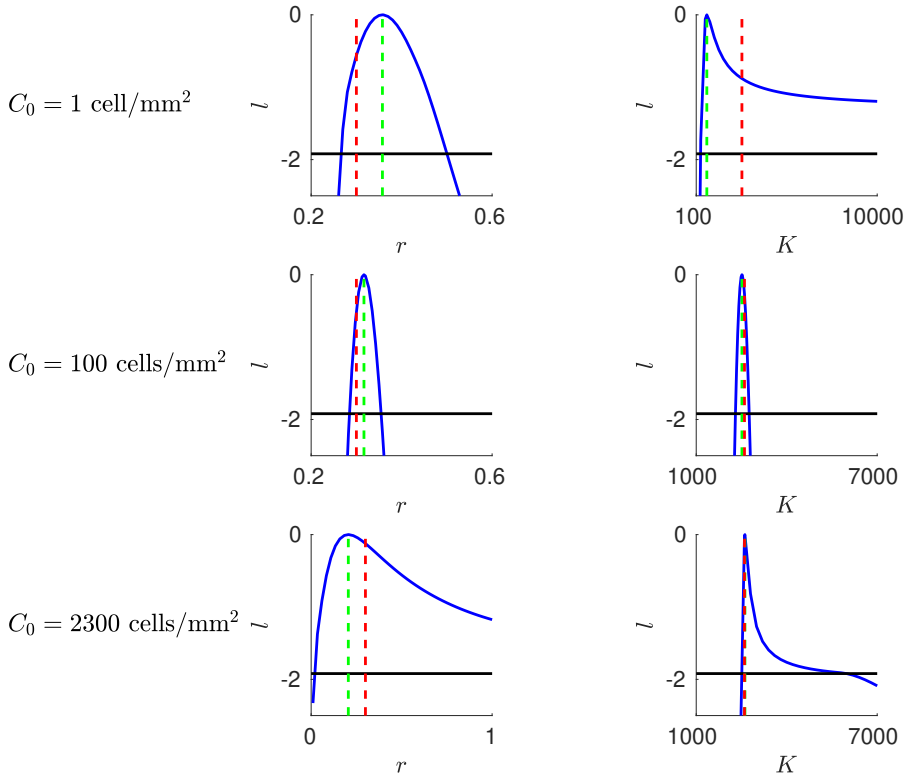


Figure 3.1: Profile likelihoods for the logistic model computed using synthetic data generated with the parameter values in Eq. (3.9). Each row corresponds to a different initial condition, C_0. As before, the black horizontal line is the threshold for the 95% confidence interval, at $l = -1.92$. The green dashed lines are the locations of MLEs, and the red dashed lines mark the location of the parameter values used to generate the synthetic dataset. Notice that the scales of the horizontal axis may be different across rows for better visualisation. The units are $r = \text{h}^{-1}$, $[K] = \text{cells/mm}^2$.

C_0 (cells/mm^2)	r (h^{-1})	K (cells/mm^2)	Correlation
1	0.3575 [0.2665, 0.5002]	680 [339, ∞)	-1050
100	0.3167 [0.2848, 0.3546]	2515 [2300, 2755]	-8873
2300	0.2054 [0.0205, ∞)	2614 [2511, 5980]	-5002

Table 3.1: The MLE and 95% confidence interval for the parameters in the logistic model, estimated with a range of initial conditions. The last column shows the off-diagonal entry of the FIM, which is an indicator for the correlation between the estimates of the two parameters. The negative sign means that the estimates of r and K are negatively correlated.

We can explain this observation intuitively. The solution of the logistic equation is a sigmoid curve, which can be broken down into three phases: the early linear growth phase, the middle inflection phase, and the late saturation phase. This is visualised in Fig. 3.2. The linear growth

phase occurs when the cell density is much lower than the carrying capacity, hence $C/K \approx 0$, and the model behaviour is effectively determined by the parameter r alone. The saturation phase occurs as the cell density approaches the carrying capacity, and the model behaviour is mostly determined by K. The inflection phase is the transition between the other two, where both parameters have a significant impact on system behaviour. Therefore, to accurately identify both parameters, the data should ideally span all three phases, which is the case for $C_0 = 100$ cells/mm^2, as shown in Fig. 3.2(b). This is the case where both parameters are identifiable (Fig. 3.1, middle row). If the initial density is too low, e.g., $C_0 = 1$ cells/mm^2, then we would not be able to observe the saturation phase (Fig. 3.2(a)), which means K will not be identifiable, as shown in Fig. 3.1, top row. On the other hand, if the initial density is too high (near K), then the system is already in the saturation phase at the start of the experiment (Fig. 3.2(c)) , hence r will not be identifiable as shown in Fig. 3.1, bottom row. These observations also point to the importance of T. If we are allowed to choose T rather than considering it fixed, then choosing T to be as large as possible while choosing a small C_0 would be a good strategy to maximise the identifiability of both parameters.

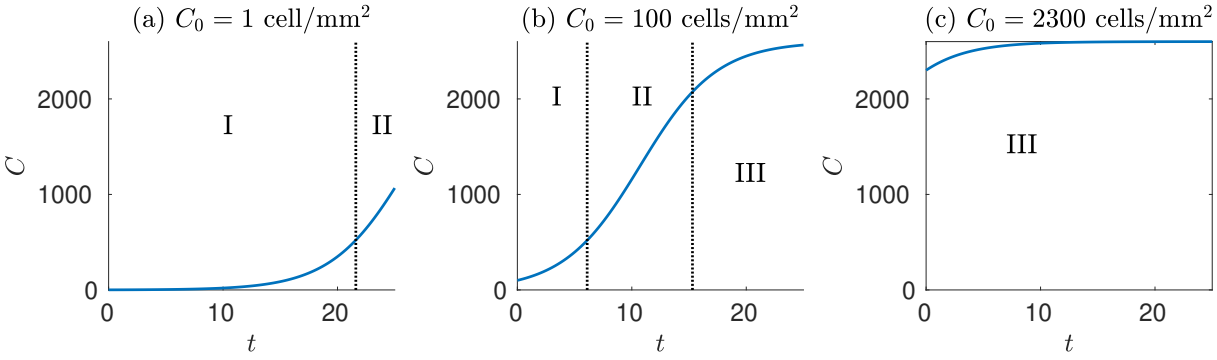


Figure 3.2: An illustration of the different phases of the solutions of the logistic model, with the same parameter values (Eq. (3.9)) but different initial conditions. Units are given in Eq. (3.2).

Next, we plot the width of the confidence intervals of the two parameters, Δr and ΔK, as functions of C_0, in Fig. 3.3. Notice that the optimal initial density for identifying r is much lower than that of the carrying capacity, $C_0 \ll \hat{K}$. In contrast, for estimating K, the optimal initial density is around $C_0 \approx \hat{K}/2$. This reinforces the point that the identifiability of each parameter is maximised by a different experimental design. If we can only do one experiment, with time-span T fixed, then choosing a C_0 between 100 cells/mm^2 and 1000 cells/mm^2 would allow us to estimate both parameters to a reasonable degree of accuracy. If we could do two experiments, again with T fixed, then it would be a good strategy to use $C_0 \approx 20$ cells/mm^2

for one experiment, which is optimal for identifying r, and $C_0 \approx 1300$ cells/mm^2 for the other, which is optimal for identifying K. This is an example of a multi-experiment approach, where multiple experiments with different experimental conditions are used to infer model parameters. There have been studies demonstrating how such approaches can resolve parameter identifiability issues [61]. Note that, in practice, it might not be practical to follow the designed experiment procedure precisely, and in fact, it is not necessary. The experimental design relies on assumptions about model parameters, which are necessarily imprecise since their values are exactly what we would like to determine. Therefore, the conclusions of the optimal experimental design should be treated as an approximate suggestion.

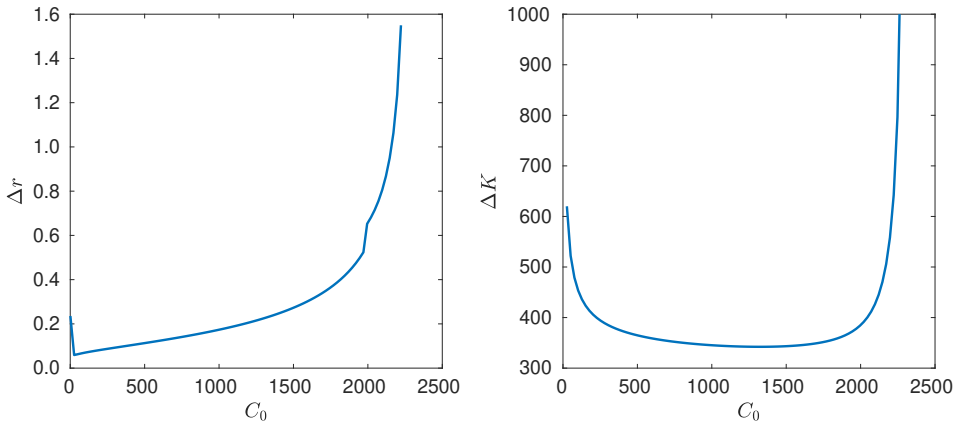


Figure 3.3: The widths the confidence intervals of the parameters r and K in the logistic model, versus the initial condition C_0. The minimum of Δr occurs at $C_0 = 20.85$ cell/mm^2, while the minimum of ΔK occurs at $C_0 = 1323$ cell/mm^2. While Δr is mostly an increasing function of C_0, this is not the case for extremely small values of C_0, where Δr increases as C_0 approaches zero. This is likely because with such small C_0, the model solution is approximately identically zero for all t, and noise dominates the data, hence it becomes more difficult to estimate r compared to the case with a slightly larger C_0.

3.3.2 Richards model

Next, we repeat the same problem of finding an appropriate initial condition for identifying parameter values, but for the Richards model (Eq. (3.3) with $\gamma > 0$ free), to see how the addition of the γ parameter impacts the experimental design. In [176], the authors showed that, given a set of real-world data, the logistic model is identifiable, whereas the Richards model is not. In particular, the γ parameter tends to be one-sided unidentifiable, where a lower bound can be established, but not an upper bound. This is also reflected in our results in Chapter 2, where the Richards PDE model shows non-identifiability under conditions where the Standard Fisher model (which is the logistic model with simple diffusion) is identifiable. We will use the

following parameter values to generate the synthetic datasets:

$$\hat{r} = 0.225 \text{ h}^{-1}, \quad \hat{\gamma} = 8, \quad \hat{K} = 2380 \text{ cell/mm}^2,$$
$$\sigma = 400, \quad \Delta t = 0.5 \text{ h}, \quad n_t = 50, \quad T = 25 \text{ h}. \tag{3.10}$$

These are chosen so that the model solution is similar to the model solution produced by the logistic model using the parameter values in Eq. (3.9). Here there are three parameters to be inferred: r, γ, and K.

In Fig. 3.4, we show the profile likelihoods for all three model parameters as the initial condition changes. Observe that when $C_0 = 100$ cells/mm^2, r and K are identifiable, while γ displays the expected one-sided non-identifiability. However, for $C_0 = 1$ cells/mm^2 and $C_0 = 2300$ cells/mm^2, the non-identifiability is more severe compared to the logistic model, with either r or K completely non-identifiable, respectively, and the profile likelihood curve being nearly flat for a large range of parameter values.

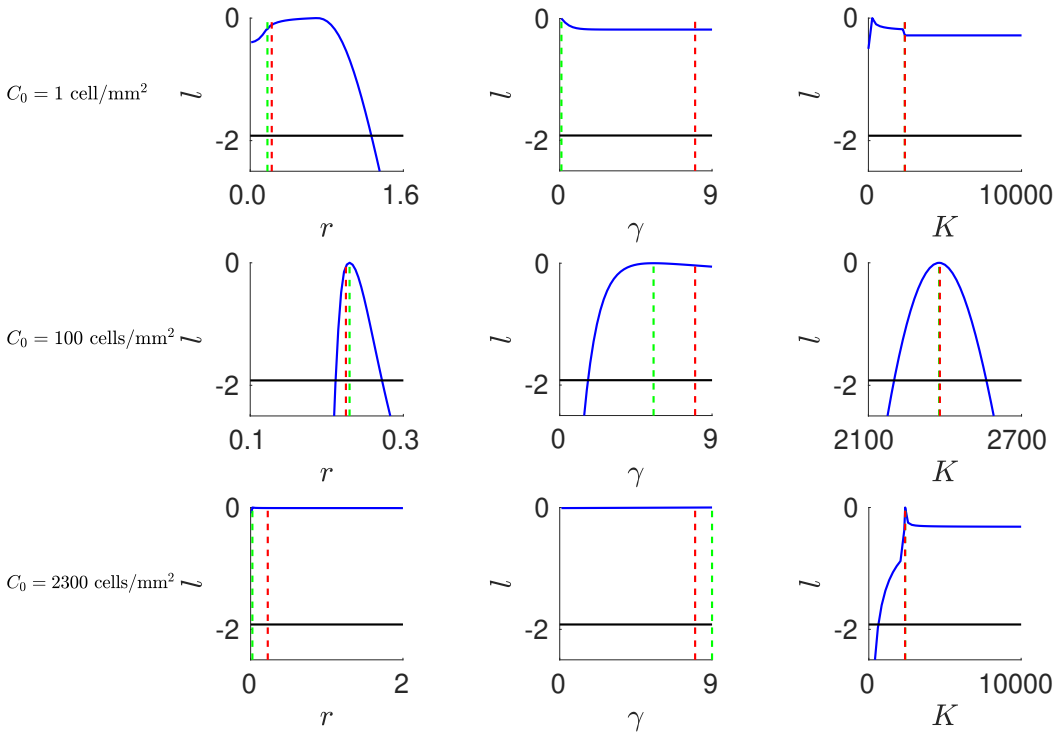


Figure 3.4: Profile likelihoods for the Richards model computed using synthetic data generated with the parameter values in Eq. (3.10). Note that in the plots where the green dashed lines denoting the MLEs appear to be invisible, they are in fact covered by the red dashed lines denoting the parameter value used to generate the dataset.

The reason for this is that γ, as a shape parameter, mostly controls the shape of the solution curve as it approaches K. In Fig. 3.5, we plot a series of solutions to the Richards model, each using the same K, but with a range of representative values for γ. The $\gamma = 8$ curve corresponds

to the synthetic dataset minus the added noise, while the values for r for the other curves are chosen such that the solution is a best fit to the $\gamma = 8$ curve. Observe that the differences between these curves are at their greatest just before the cell density reaches carrying capacity K. Therefore, if there are not enough data to show how the cell density approaches K, then identifying the value of γ will be difficult. This is the case for $C_0 = 1$ cell/mm^2, where the simulation ends before the cell density is close to K, and also for $C_0 = 2300$ cell/mm^2, where the cell density is already too close to K for γ to make any difference.

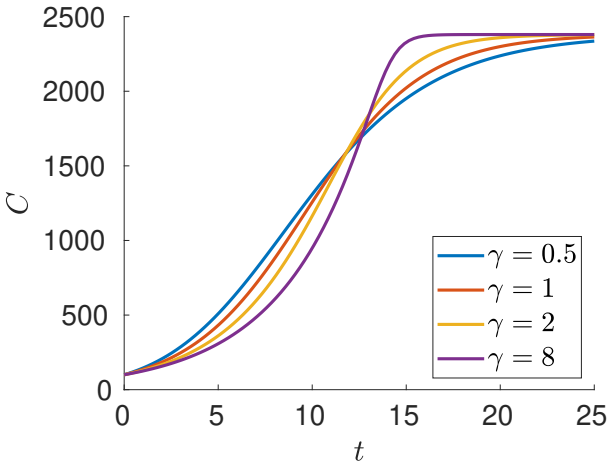


Figure 3.5: Plots of several solutions of the Richards model (Eq. (3.1)) with $K = 2380$ cell/mm^2, $C_0 = 100$ cell/mm^2, and varying r and γ, with $(r, \gamma) = (0.482\ \text{h}^{-1}, 0.5), (0.324\ \text{h}^{-1}, 1), (0.259\ \text{h}^{-1}, 2)$, and finally, the synthetic dataset, with $(\hat{r}, \hat{\gamma}) = (0.225\ \text{h}^{-1}, 8)$.

In Fig. 3.6, we plot Δr and ΔK, the widths of confidence intervals for the respective parameters, against C_0. The plot for $\Delta\gamma$ is omitted since $\Delta\gamma$ is always infinite for any C_0. We can see that the qualitative observations from the logistic model still hold, but the range of C_0 for either r or K to be identifiable is now narrower than in Fig. 3.3. In this case, an informed choice of the experimental initial conditions can still improve the identifiability of a subset of the model parameters, despite the overall non-identifiability of the model.

The results in this section show that an intelligent choice of initial conditions can result in much lower uncertainty in parameter estimates, allowing us to learn much more about the system from the data. In the case where there is more than one unknown parameter, each parameter might have different optimal initial conditions for estimating them, and we might have to compromise between those and choose an initial condition that can lead to experimental data that can make all parameters identifiable, even if it is not the optimal initial condition for any one parameter. In the pursuit of finding the optimal initial condition to improve parameter

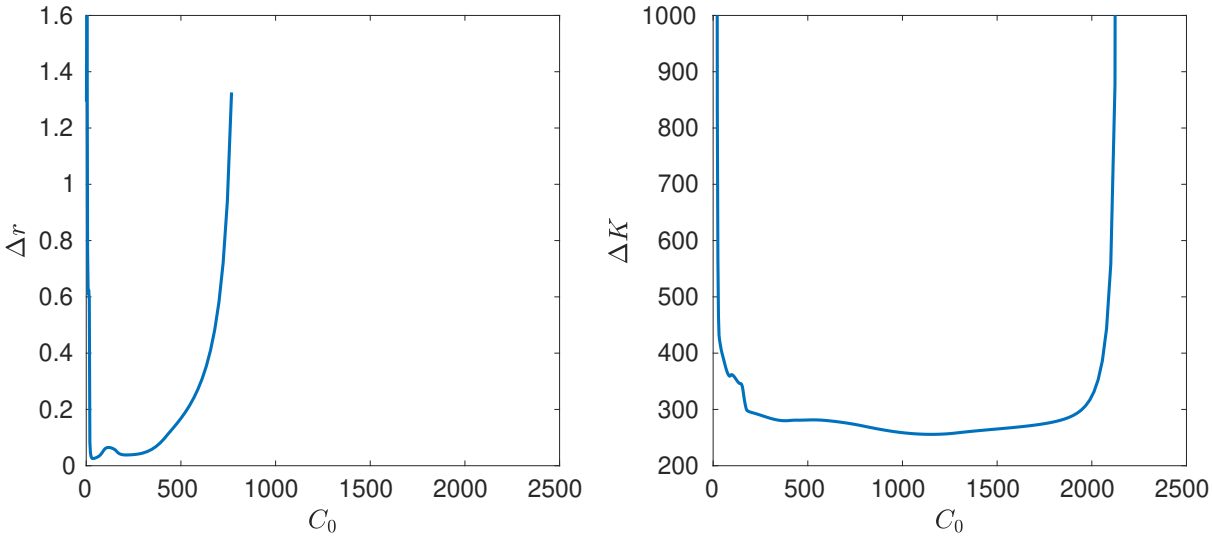


Figure 3.6: The widths of the confidence intervals of the parameters r and K, versus the initial condition C_0. The minimum of Δr occurs at $C_0 = 35.54$ cell/mm^2, while the minimum of ΔK occurs at $C_0 = 1152$ cell/mm^2. Δr is infinite for $C_0 > 786$ cells/mm^2. Overall both Δr and ΔK as a function of C_0 are U-shaped, which decreases for small C_0 up until the global minimum, and increases past that point. The exception to this is in an interval, roughly $0 \leq C_0 \leq 200$ cells/mm^2, where the dependence is complicated, with many shallow local extrema. This phenomenon has been verified through careful numerical computation, and is not an artifact.

identifiability, we are assisted by our understanding of the biological system. Specifically, the analysis made use of the true parameter values $\hat{\boldsymbol{\theta}}$ to compute the profile likelihoods. This is not realistic, as $\hat{\boldsymbol{\theta}}$ are exactly what we wish to identify through parameter inference. Nonetheless, in many cases, we have partial knowledge on parameter values, where the values are known within a certain (possibly very wide) range. In this scenario, a Bayesian approach can be adopted. We can define a prior distribution on the parameter values, then calculate the expectation of the confidence interval width as a function of the initial conditions, with the expectation taken over the prior distribution. We can then select an initial condition with narrow expected confidence interval width for all parameters. In a scenario where nothing at all is known about the parameter values, it may be necessary to perform a naively designed initial experiment, and use the resulting information on parameter values to design subsequent experiments.

3.4 Optimal initial conditions for model discrimination

We now turn to the problem of model discrimination, where we try to maximally differentiate between two models that may have similar solutions for naively chosen initial conditions. Briefly speaking, for this problem, we fix two sets of parameter values, which, together with the model equation (Eq. (3.3)), represent two models that we would like to distinguish. The goal of this

section is to maximise the difference between the two model solutions, by making careful choice of initial condition.

To proceed, let the final time of the experiment, $T = 25$ h, be fixed, and fix two parameter sets, $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, for the model being considered. Let

$$\frac{\mathrm{d}C_i}{\mathrm{d}t} = f(C_i; \boldsymbol{\theta}_i, K, C_0), \quad i = 1, 2,$$

where f is as in Eq. (3.3). The solution $C_{\text{model}}(t; \boldsymbol{\theta}_i, C_0)$ is given in Eq. (3.5). We wish to maximise the model difference:

$$\max_{C_0>0} \mathcal{D}(C_0), \quad \mathcal{D}(C_0) \coloneqq \int_0^T [C_{\text{model}}(t; \boldsymbol{\theta}_1, C_0) - C_{\text{model}}(t; \boldsymbol{\theta}_2, C_0)]^2 \, \mathrm{d}t. \tag{3.11}$$

This integral is evaluated by applying the trapezoid rule with 100 points, which is sufficient to ensure that the integration error is much smaller than the lowest possible value of $\mathcal{D}$, therefore sufficiently accurate for the purpose of this section.

We consider the problem of optimally discriminating between the Richards model, which has γ free in Eq. (3.3), and the logistic model, which share the same equation but has $\gamma = 1$ fixed. We will use the version of the model without the parameter δ (or equivalently, fixing $\delta = 0$). We choose the following two parameter sets to be discriminated:

$$(r, \delta, \gamma, K)_1 = (0.291, 0, 1, 2605), \quad (r, \delta, \gamma, K)_2 = (0.222, 0, 3.709, 2435), \tag{3.12}$$

where the unit of r is h^{-1}, and the unit of K is cells/mm^2. These two parameter sets represent the best fit of the logistic and Richards models to the synthetic dataset generated with parameter values in Eq. (3.10), respectively. Notice that the second parameter set $\boldsymbol{\theta}_2$ corresponding to the Richards model deviates noticeably from the parameters used to generate the data, especially the γ parameter. This is not surprising, given that we know γ is practically non-identifiable. Fig. (3.7) compares the model solutions corresponding to these parameters and the unperturbed synthetic dataset, and a visual inspection shows that both models fit the dataset quite well, although the Richards model is slightly better.

In Fig. 3.8, we plot the model difference $\mathcal{D}(C_0)$ defined in Eq. (3.11) against C_0, to visualise the effect of the initial condition on model difference. First, $\mathcal{D}(C_0)$ tends to infinity as C_0 tends to infinity. The reason is that for $C_0 \gg \max(K_1, K_2)$, both model solutions will decay exponentially to their carrying capacity from C_0, with the one with higher carrying capacity decaying slower, resulting in a large difference between the model solutions. However, in experiments, it is difficult

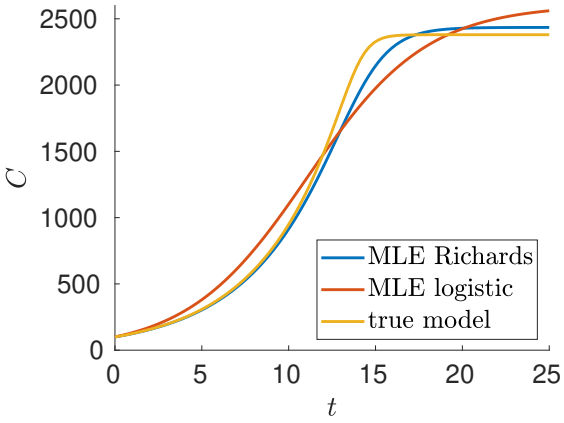


Figure 3.7: A comparison between the results of fitting the Richards and logistic models to the synthetic data. Yellow: the unperturbed synthetic data (solution to the Richards model (Eq. (3.1)) with parameters from Eq. (3.10)). Blue: model solution corresponding to the MLE of the parameter values of the Richards model, $(r, \delta, \gamma, K) = (0.222, 0, 3.709, 2435)$. Red: model solution corresponding to the MLE of the parameter values of the Richards model, $(r, \delta, \gamma, K) = (0.291, 0, 1, 2605)$. The unit of r is h^{-1}, and the unit of K is cells/mm^2.

to achieve a cell density much higher than the carrying capacity, so it might not be practical to choose a very high C_0. Furthermore, simple models such as the Richards model could cease to be valid when applied beyond their typical scope, which assumes that the cell density does not exceed the carrying capacity significantly. Therefore, the limit $C_0 \to \infty$ is not biologically relevant or interesting. Furthermore, a local maximum occurs at $C_0 \approx 4$ cells/mm^2, which is much smaller than the carrying capacity in either model, and $\mathcal{D}(C_0)$ decreases sharply away from this optimal initial value. Solutions to the Richards model with a high γ have a sharper sigmoid shape, that is, the solutions remain lower for longer before the inflection point, and approach K quicker after the inflection point. The optimal initial condition $C_0 \approx 4$ cells/mm^2 exploits the fact that for such a low initial density, the Richards model "takes off" behind the logistic model, and by the final time T the Richards model has just begun to "catch up" with the Logistic model, leaving a large gap between the predictions of both models. However, this local maximum also might not be biologically relevant, since at such a low cell density, the continuum hypothesis implicit in the model breaks down, and the model ceases to be valid. If we restrict C_0 to within a biologically realistic range, say $[100, \max(K_1, K_2)]$, there is one additional local maximum at $C_0 \approx 720$ cells/mm^2, which would be a good choice for an experiment. We have plotted the model solutions and highlighted the areas between them in Fig. 3.9, using values of C_0 where local extrema of $\mathcal{D}(C_0)$ occur, to explain the shape of $\mathcal{D}(C_0)$.

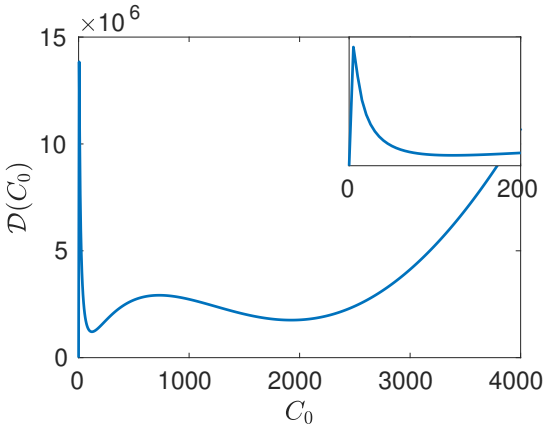


Figure 3.8: Plot of the model difference, $\mathcal{D}(C_0)$, as defined in Eq. (3.11), as a function of the initial condition, C_0. Local maxima occur at $C_0 = 4.7$ and 723 cells/mm^2 (as well as at $C_0 \to \infty$). Two local minima occur at $C_0 = 117$ and 1925 cells/mm^2. The inset is zoomed in to provide a closer look in the vicinity surrounding the first maximum.

Note that in this problem, we are trying to distinguish between two models where the parameter sets are obtained by fitting the models to a synthetic dataset, which could represent the data coming from a first, preliminary experiment. If we use an initial condition C_0 similar to the one used to generate the synthetic data, then we expect both models to behave very similarly, which explains why $\mathcal{D}(C_0)$ is small at $C_0 = 100$ cells/mm^2, which is the initial condition used to generate the synthetic data. The local minimum at $C_0 = 117$ cells/mm^2 is not far away. In fact, if the two parameter sets being discriminated were obtained by fitting to the synthetic data without added noise, the minimum would occur exactly at $C_0 = 100$ cells/mm^2.

Fig. 3.8 also shows that the problem of optimising for model discrimination is not quite the same as optimising for parameter identifiability in Sec. 3.3, since the optimal initial condition for model discrimination (ignoring biological realism), $C_0 = 4.718$ cells/mm^2, is a poor choice for narrowing down the values of either r or K in the Richards model, as seen in Fig. 3.6, while the optimal initial condition for identifying r or K ($C_0 = 35.54$ cells/mm^2 and 1152 cells/mm^2, respectively) are less than optimal for model discrimination. The reason is that, for the model discrimination problem, we have knowledge gained from the synthetic dataset, representing a first, preliminary experiment, built into our choice of the two parameter sets to be discriminated, and we are designing a second, follow-up experiment to further distinguish between those two chosen parameter sets, so the chosen initial condition for the second experiment should be significantly different from the first, to avoid simply gathering redundant information. In contrast, in the scenario we faced for the problem of optimising parameter identifiability, we have no information on the parameters from earlier experiments, and nothing will be redundant. This

explains why the optimal initial conditions for the two problems are different.

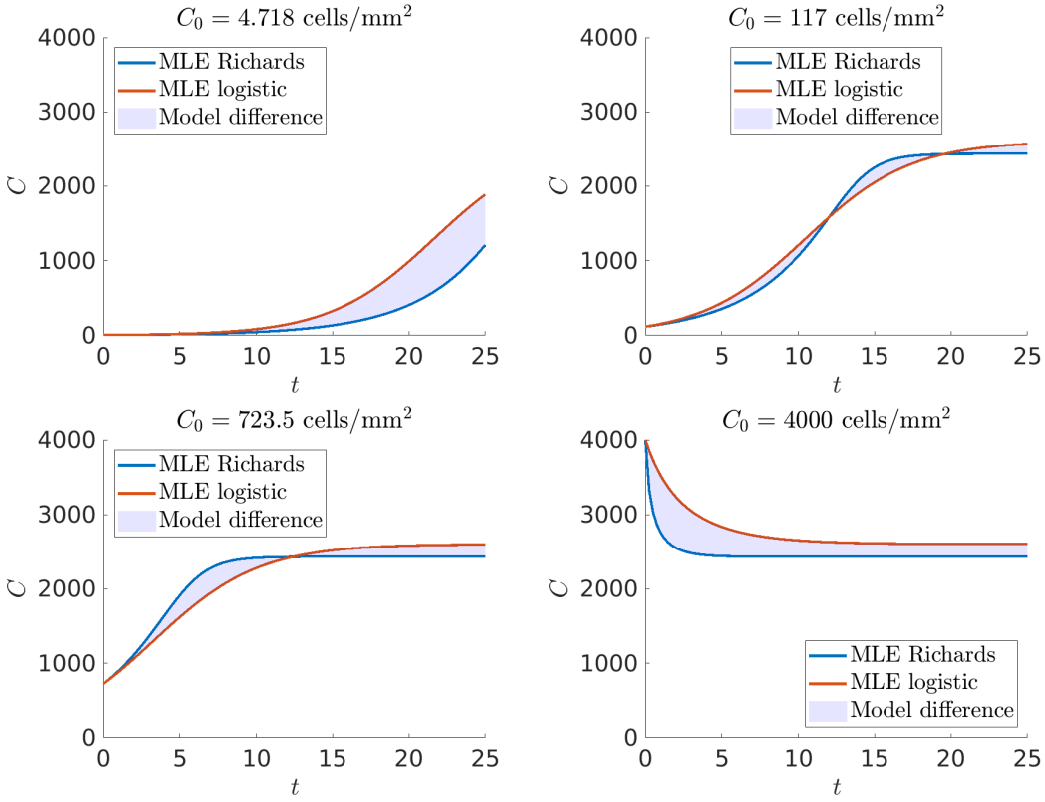


Figure 3.9: Plot of the model solutions produced with the MLEs of parameter values (Eq. (3.12)) of the Richards and logistic models with three different initial conditions: $C_0 = 4.718$ cells/mm^2, which is a local maximum of $\mathcal{D}(C_0)$, $C_0 = 117$ cells/mm^2, a local minimum, and $C_0 = 4000$ cells/mm^2, to illustrate the effect of a very large initial condition. Units are given in Eq. (3.2).

3.5 Optimal control for improving identifiability

In this section, we consider using a control to improve the identifiability of a model. The goal, as it was in Sec. 3.3, is to minimise the widths of the confidence intervals of model parameters, $\Delta\theta$, by optimally choosing a control function, with the initial condition C_0 fixed. This problem is rather difficult to formulate as a traditional optimal control problem, as the objective, $\Delta\theta$, cannot be easily expressed as a function of the state and control variables. To make the problem easier, we will restrict our attention to the case of applying a single control at a time, with the control taking the form of a window function,

$$u(t) = \begin{cases} u_{\max}, & \tau_0 < t < \tau_0 + \tau, \\ 0, & \text{otherwise}, \end{cases} \tag{3.13}$$

where $u_{\max} \geq 0$ is the height of the window function, $0 \leq \tau_0 \leq T$ is the time when the control is turned on, and $\tau \leq T - \tau_0$ is the duration for which the control is turned on. For this section, we use the logistic model with the additive version of the control in Eq. (3.6).

We use the following parameters to generate a synthetic dataset:

$$\hat{r} = 0.45 \text{ h}^{-1}, \quad \hat{\delta} = 0.15 \text{ h}^{-1}, \quad \hat{\gamma} = 1, \quad \hat{K} = 3900 \text{ cell/mm}^2, \quad \sigma = 20,$$
$$\Delta t = 0.25 \text{ h}, \quad n_t = 100, \quad T = 25 \text{ h}, \quad C_0 = 100 \text{ cell/mm}^2. \tag{3.14}$$

These values are chosen so that the effective parameter values, r_{eff} and K_{eff} as defined in Eq. (3.4), are the same as the parameter sets used earlier in Eq. (3.9), which were in turn chosen to approximate the parameter values inferred using real data in Ch. 2. Without the addition of control inputs, the model solution produced by the parameters in Eq. (3.14) is identical to that produced by the parameter values in Eq. (3.9). In contrast to the optimal initial value problem, we add the death term by using a positive δ, and use a lower noise level σ and more time points n_t, so that the main source of non-identifiability comes from the structural non-identifiability of the model, rather than from having insufficient or noisy data. In this way, we can illustrate the effects of the control more clearly. Using this synthetic dataset, we will investigate how the control parameters $u_{\max}$, τ_0, and τ impact the confidence intervals of the model parameters, as a measure of practical identifiability.

Before performing any calculations, let us consider whether the inclusion of the control can at least provide structural identifiability. Assuming $u(t)$ is not identically zero, that is $u_{\max} > 0$ and $\tau > 0$, a set of parameter values (r, δ, K) can exactly reproduce the unperturbed data if both of the following conditions hold:

$$g_a(C; r, \delta, 1, K; 0) = g_a(C; \hat{r}, \hat{\delta}, 1, \hat{K}; 0), \ \forall C \geq 0, \tag{3.15a}$$
$$g_a(C; r, \delta, 1, K; u_{\max}) = g_a(C; \hat{r}, \hat{\delta}, 1, \hat{K}; u_{\max}), \ \forall C \geq 0. \tag{3.15b}$$

Recall that g_a, given in Eq. (3.6), can be written as

$$g_a(C; \boldsymbol{\theta}, u_r(t), u_\delta(t), u_K(t)) = (r + u_r(t))C \left[1 - \left(\frac{C}{K - u_K(t)}\right)^\gamma\right] - (\delta + u_\delta(t))C.$$

Trivially, $(r, \delta, K) = (\hat{r}, \hat{\delta}, \hat{K})$ satisfy the conditions in Eq. (3.15). If this is the only parameter set that does so, then the model will be structurally globally identifiable. If there exists another distinct parameter set that also satisfies the same conditions, then the model can be shown to be structurally non-identifiable.

Each of the conditions can be expanded as a condition in terms of $\hat{r}$, and another condition in terms of $\hat{K}$, for a maximum of four conditions in total (examples are given in each of the subsections below). These conditions state that the effective parameters for both models, r_{eff}

and K_{eff} as in Eq. (3.4), must match those of the effective parameter values used to generate the synthetic dataset, both when the control is on, and when it is off. However, depending on which control variable u is being applied, some of the four conditions might be repeated, so Eq. (3.15) amounts to a system of four or fewer equations in terms of three unknowns, r, δ and K. If there are at least three distinct equations (which is the case if we use u_K or u_r as control variable, see Sec. 3.5.1 and Sec. 3.5.3), it can be shown that $(\hat{r}, \hat{\delta}, \hat{K})$ will be the unique parameter set satisfying the conditions, hence no other set of parameter values is able to perfectly reproduce the unperturbed data, and the model will be structurally globally identifiable. On the other hand, if there are only two or fewer distinct equations (which is the case if we use u_δ as control variable, see Sec. 3.5.2), then there exist multiple parameter sets able to perfectly reproduce the data, so the model will not be structurally globally identifiable. In fact, for the model considered here, in such cases we would always be able to find arbitrarily similar parameter sets that leads to the same model solution, which means that the model is structurally non-identifiable.

3.5.1 Using u_K as the control variable

First, we consider u_K to be the sole control variable to be applied. Note that if u_K is non-zero for any positive length of time, then the model is structurally globally identifiable, since Eq. (3.15) can be written as three conditions on the parameter values:

$$r_{\text{eff}} = r - \delta = \hat{r} - \hat{\delta}, \tag{3.16a}$$

$$K_{\text{eff}} = K(1 - \delta/r) = \hat{K}(1 - \hat{\delta}/\hat{r}), \tag{3.16b}$$

$$(K - u_{\max})(1 - \delta/r) = (\hat{K} - u_{\max})(1 - \hat{\delta}/\hat{r}). \tag{3.16c}$$

These three conditions are sufficient to ensure that $(\hat{r}, \hat{\delta}, \hat{K})$ is the only set of parameters able to exactly recreate the unperturbed data. Suppose we have (r, δ, K) satisfying Eq. (3.16). Then Eq. (3.16a) gives us $\delta = r - r_{\text{eff}}$. Substituting this into Eq. (3.16b) gives $r_{\text{eff}}/r = K_{\text{eff}}/K$, while substituting the same into Eq. (3.16c) gives $r_{\text{eff}}/r = (\hat{K} - u_{\max})(1 - \hat{\delta}/\hat{r})/(K - u_{\max})$. Equating the right hand side of these two equations gives a linear equation in K, which can be solved to show that $K = \hat{K}$, which can then be substituted back into Eqs. (3.16a), (3.16b) to show that $r = \hat{r}, \delta = \hat{\delta}$.

In Fig. 3.10, we demonstrate how the profile likelihood curves change as the control parameters ($u_{\max}, \tau_0$, and τ) are varied. First, in Fig. 3.10(a), we show the profile likelihoods with no control applied. Since the model is known to be structurally non-identifiable, so that a change

in one parameter can be perfectly compensated by changes in other parameters, we obtain a flat top for all three profile likelihood functions, as expected. Nonetheless, we can identify lower bounds for r and K. This is because we are restricting all parameters to be non-negative, based on the biological interpretations of the model. A value of r below r_{eff}, or a value of K below K_{eff}, would require a negative δ to compensate, which is not allowed, hence the profile likelihood functions for r and K drop sharply at these values.

Fig. 3.10(b) shows the results of applying a very weak control, of magnitude $u_{\max} = 50$ cells/mm^2 (compared to $\hat{K} = 3900$ cells/mm^2). This results in only a marginal improvement in parameter identifiability, as the flat top in the profile likelihoods in Fig. 3.10(a) changes to a curved one, yet the parameters remain non-identifiable. Increasing the magnitude of the control to $u_{\max} = 200$ cells/mm^2 (Fig. 3.10(c)) makes all parameters identifiable, and increasing it to $u_{\max} = 400$ cells/mm^2 (Fig. 3.10(d)) further improves the accuracy of the parameter estimates. In Fig. 3.10(e), the control is applied at the beginning of the time span of the simulation, rather than in the middle. Since K has relatively little impact on the dynamics in the early phase of the simulation, this choice for control is less effective for improving parameter identifiability than applying the control at a later time, and the parameters remain non-identifiable. Finally, Fig. 3.10(f) shows that non-identifiability will also persist if the control is applied too briefly, since there are too little data within the control window to observe the dynamics of the system. On the other hand, applying the control for too long (Fig. 3.10(g)) also results in non-identifiability, but for the opposite reason. In this case, we have too little data outside the window.

Furthermore, observe that the shapes and widths of the profile likelihood curves of the three parameters are highly correlated, as can be seen in the bivariate likelihood plots. In each of those plots, the confidence region, represented by the yellow pixels, forms a very narrow ridge, indicating that the combination of any two parameters is identifiable, even if individually they are not. This is because fixing the value of any one parameter allows us to determine the best value of the other two parameters by using Eq. (3.16a) and Eq. (3.16b). Eq. (3.16a) gives hence the profile likelihood curve for δ, l_δ, is very close to l_r shifted to the left by r_{eff}. Similarly, Eq. (3.16b) gives $K = K_{\text{eff}} r / r_{\text{eff}}$, hence l_K is close to a shifted and rescaled version of l_r. For this reason, it is sufficient to look at the identifiability of r to understand the overall identifiability of the whole model. Therefore, the rest of this section will focus on r.

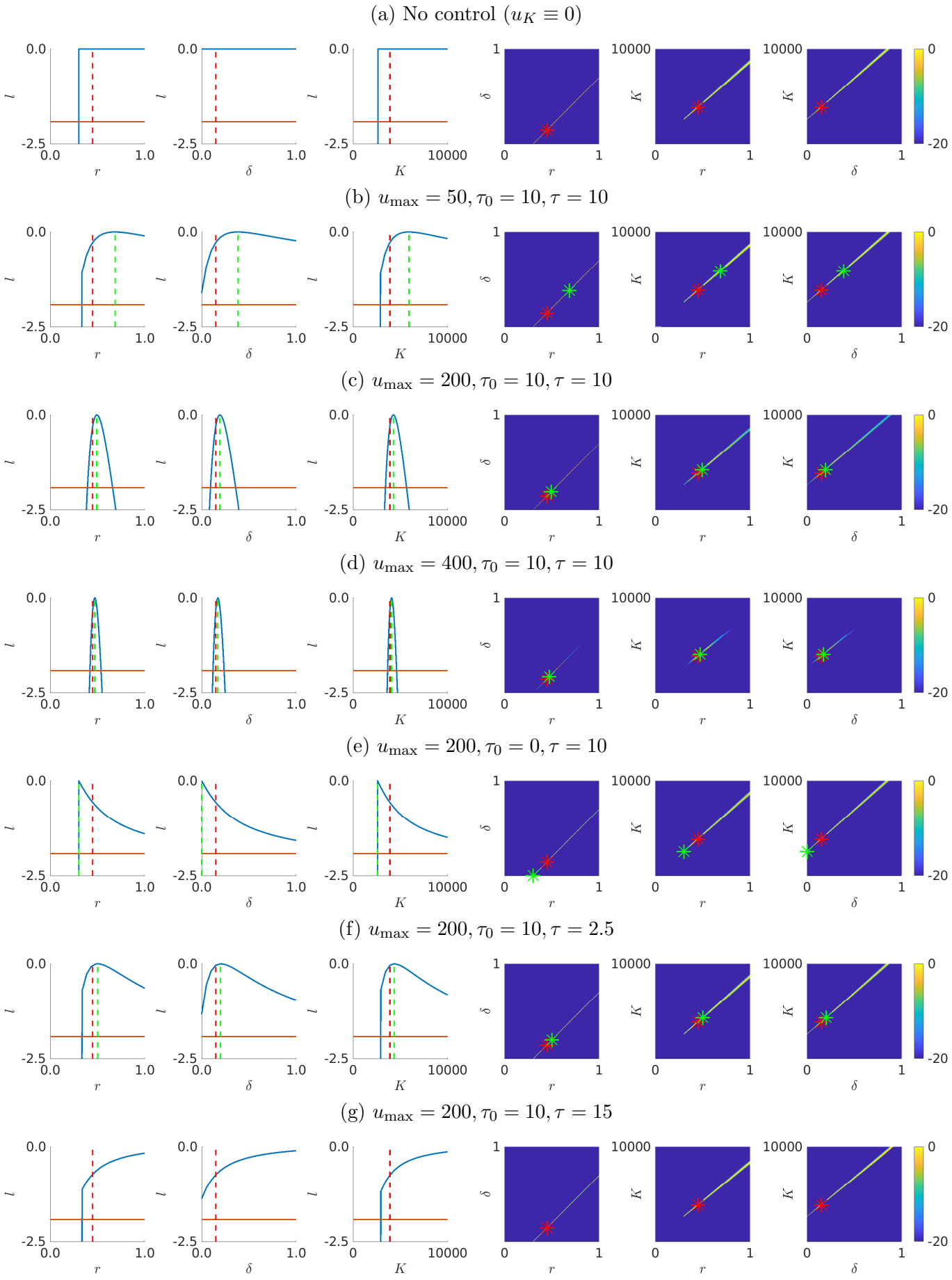


Figure 3.10: The profile likelihood functions for the parameters in the logistic model with additive control (Eq. (3.6)), using the synthetic dataset generated using parameter values in Eq. (3.14), with $u_K(t)$ being a window function as in Eq. (3.13) with height $u_{\max}$, turning on at $t = \tau_0$, with a width of τ, and u_δ and u_r set to zero. Starting from the left, the first three columns show the univariate profile likelihoods for r, δ and K respectively, and the next three columns show the bivariate profile likelihoods for r–δ, r–K, and δ–K, respectively. The green dashed lines in the univariate likelihood plots denote the MLEs, and the red dashed lines denote the parameter values used to generate the data. These are denoted with green and red stars in the bivariate plots. Units are given in Eq. (3.2).

Next, we explore the effects of the control on identifiability more systematically by plotting Δr, the width of the confidence interval of r, as a function of the control variables. First, in Fig. 3.11, we take $u_{\max} = 200$ cells/mm^2, $\tau_0 = 10$ h, $\tau = 10$ h which, as shown in Fig. 3.10, makes the model identifiable. Then, we vary one control parameter at a time while leaving the other two fixed, and plot Δr as a function of the varying parameter. Observe that Δr decreases monotonically as $u_{\max}$ increases, confirming our earlier observations, while the effects of τ_0 and τ on Δr are more nuanced. There seem to be two good choices for τ_0: one around $\tau_0 = 9$ h and another around $\tau_0 = 15$ h, which are the local minima of Δr as a function of τ_0.

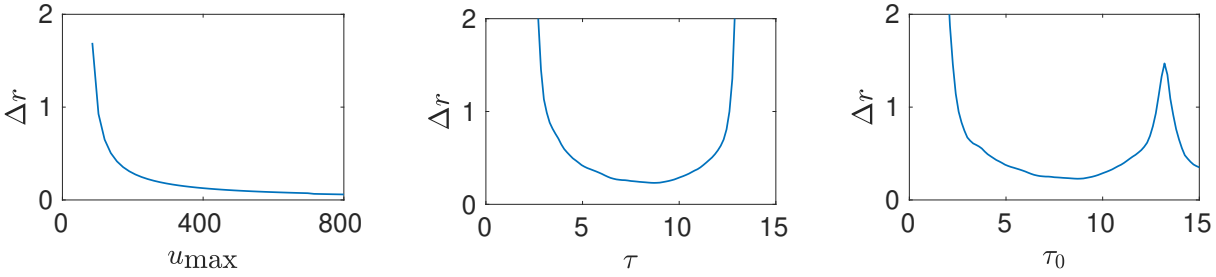


Figure 3.11: The widths of the confidence intervals for r, Δr, in the logistic model with additive control (Eq. (3.6)), using the synthetic dataset generated using parameter values in Eq. (3.14), as a function of the control parameters for u_K defined in Eq. (3.13). The control parameter being varied is labelled on the x-axis in each plot, while the other control parameters are held fixed at $u_{\max} = 200$ cells/mm^2, $\tau_0 = 10$ h, $\tau = 10$ h. Units for quantities in the plots are given in Eq. (3.2).

In order to further look into the effects of the location of the control window on parameter identifiability, we fix $u_{\max}$ and plot Δr as both τ_0 and τ vary in Fig. 3.12, which contains Fig. 3.11(b,c) as cross-sections. We can ignore the part of the plot above the red diagonal dashed line as it represents the region where $\tau_0 + \tau > T$. Observe that there are two diagonal valleys (darkest blue) representing the combinations of (τ_0, τ) that give the highest degree of identifiability. One of these two valleys represents $\tau_0 < 20$ h and $\tau_0 + \tau \approx 19$ h, and the other represents $\tau_0 > 13$ h and $\tau_0 + \tau \approx 25$ h.

Altogether, the results show that in order to achieve identifiability, the magnitude of the control must be high enough, and the timing of the control window $[\tau_0, \tau_0 + \tau]$ must be chosen carefully, so that the span of time where the system behaviour depends strongly on the parameter being acted upon by the control (K in this case), overlaps significantly with both the time span inside the window when the control is turned on, which is $[\tau_0, \tau_0 + \tau]$, and the time span outside of it.

We can understand this intuitively by thinking about the source of the non-identifiability.

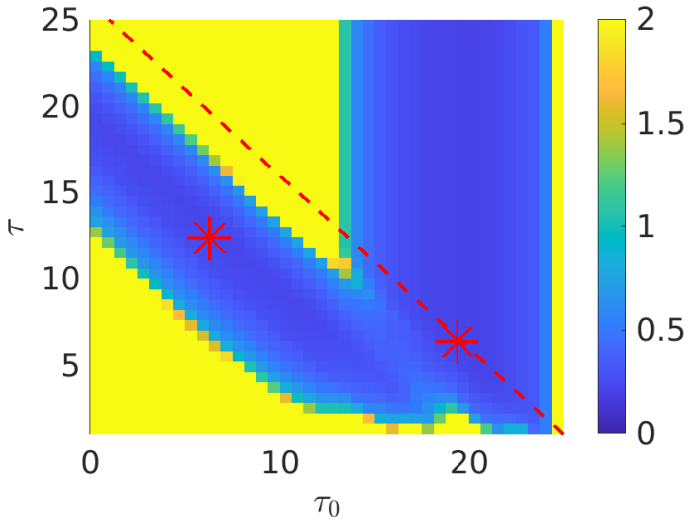


Figure 3.12: Similar to Fig. 3.11, but with Δr shown as a function of τ_0 and τ, represented as a heat-map, with $u_{\max} = 1200$ cells/mm^2 fixed. The red dashed line marks the line $\tau_0 + \tau = 25$ h. The two red stars denote local minima of Δr at $(\tau_0, \tau) = (6.3 \text{ h}, 12.4 \text{ h}), (19.5 \text{ h}, 5.5 \text{ h})$. Units are $[\tau] = [\tau_0] = \text{h}$, $[\Delta r] = \text{h}^{-1}$.

For any set of parameter values, we need sufficient data on the violation of at least one condition in Eq. (3.16) in order to confidently conclude that it does not reflect the data. If $u_{\max}$ is not sufficiently high, then any set of parameters that satisfy Eq. (3.16b) would also approximately satisfy Eq. (3.16c), leaving us with effectively only two conditions. If the control window is too short, then we will not have enough data on the violation of Eq. (3.16c). Similarly, if the control window is too large, leaving us little data on system behaviour outside of the window, then we would not be able to conclude a violation of Eq. (3.16b). In all of these cases, we have two conditions and three unknown parameters, preventing unique identification of parameter values.

This idea provides a heuristic approach to approximate the optimal controls in terms of parameter sensitivity. Let $\phi_\theta(t)$ be the local sensitivity of the solution with respect to parameter θ, defined as

$$\phi_\theta(t) = \frac{\partial C_{\text{model}}(t; \boldsymbol{\theta}, C_0)}{\partial \theta}. \tag{3.17}$$

The sensitivities are calculated using the following ODE, which can be derived from the original model equations (Eq. (3.1)) using the chain rule:

$$\frac{\mathrm{d}\phi_\theta}{\mathrm{d}t} = \frac{\partial f(C; \boldsymbol{\theta})}{\partial C}\phi_\theta + \frac{\partial f(C; \boldsymbol{\theta})}{\partial \theta}, \quad \phi_\theta(0) = 0.$$

We plot the sensitivities with respect to r, δ and K in Fig. 3.13. A larger value of $|\phi_\theta(t)|$ means the solution is more sensitive to a change in the parameter θ at time t, while holding the other

parameter values constant. We have $\phi_\theta(0) = 0$, since the initial condition is fixed. Define the sensitive interval for parameter θ as

$$\mathcal{T}_\theta = \left\{ t \in [0, T] : |\phi_\theta(t)| \geq \frac{1}{2} \max_{t' \in [0,T]} |\phi_\theta(t')| \right\}. \tag{3.18}$$

This interval represents the time span during which the solution is most sensitive to the parameter. For K, we have $\mathcal{T}_K \approx [13 \text{ h}, 25 \text{ h}]$. As a heuristic for optimal control, we should ensure that the control window $[\tau_0, \tau_0 + \tau]$ covers roughly half of $\mathcal{T}_K$, which gives us two sensible options: one with $\tau_0 < 13.5$ h and $\tau_0 + \tau \approx 19$ h, and another with $\tau_0 \approx 19$ and $\tau_0 + \tau = 25$. These two options each correspond to a local minimum of Δr with respect to τ_0 and τ as we found in Fig. 3.12.

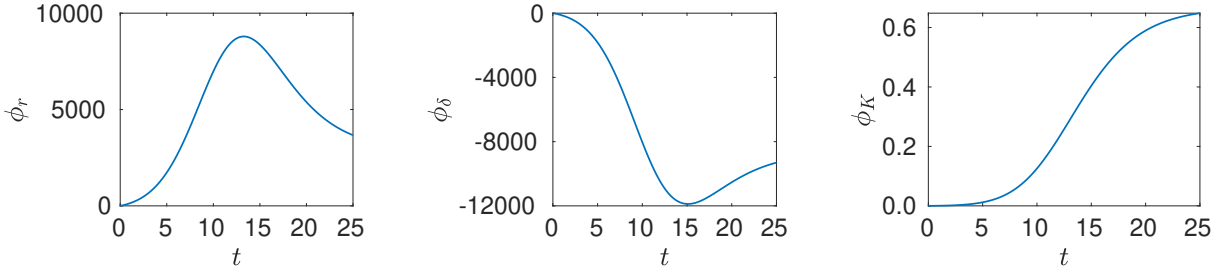


Figure 3.13: Local sensitivities of the model solution C_{model} (Eq. (3.5)) with parameter values from Eq. (3.14), with respect to the model parameters r, δ, K, as defined in Eq. (3.17). Units are $[t] = \text{h}$, $[\phi_r] = [\phi_\delta] = \text{cells} \cdot \text{h/mm}^2$, $[\phi_K] = 1$.

Since the local sensitivities ϕ_θ, and therefore $\mathcal{T}_\theta$, are much cheaper to compute compared to $\Delta\theta$, this heuristic is a cheaper alternative to the profile likelihood-based approach. A major disadvantage of the heuristic method is that it only uses local information in the parameter space at a single point, and does not make use of any global information, which is the same major weakness of using the FIM (Eq. (2.15)) to perform identifiability analysis, hence all criticisms of FIM also apply here. Furthermore, the heuristic method does not take into account the ways the parameters compensate for each other (which is encoded in the second mixed partial derivatives of the likelihood function with respect to the parameters, that is, the off-diagonal elements of the FIM). Therefore, we recommend using the heuristic method only as an intuitive guide to understand the output of the profile-likelihood based method for finding an optimal control, or when the model consists of a large number of equations, which may make the profile likelihoods too expensive to compute.

3.5.2 Using u_δ as the control variable

With u_δ being the control variable to be applied, the conditions on the parameter values to be able to perfectly reproduce the unperturbed data given in Eq. (3.15), that is,

$$g_a(C; r, \delta, 1, K; 0) = g_a(C; \hat{r}, \hat{\delta}, 1, \hat{K}; 0) \ \forall C \geq 0,$$
$$g_a(C; r, \delta, 1, K; u_{\max}) = g_a(C; \hat{r}, \hat{\delta}, 1, \hat{K}; u_{\max}) \ \forall C \geq 0,$$

amount to only two equations,

$$r_{\text{eff}} = r - \delta = \hat{r} - \hat{\delta},$$
$$K_{\text{eff}} = K(1 - \delta/r) = \hat{K}(1 - \hat{\delta}/\hat{r}).$$

Therefore, with three free parameters (r, δ, K), it is easy to see that there is an infinite family of parameter sets, all capable of exactly reproducing the unperturbed data. This means under the additive control model (Eq. (3.6)), controlling the death rate, δ, is completely useless for improving parameter identifiability, as the model remains structurally non-identifiable for any choice of u_δ.

3.5.3 Using u_r as the control variable

With u_r as the control variable, the conditions for a set of parameter values to be able to perfectly reproduce the unperturbed data (Eq. (3.15)) can be written as three equations, namely

$$r_{\text{eff}} = r - \delta = \hat{r} - \hat{\delta}, \tag{3.19a}$$
$$K_{\text{eff}} = K(1 - \delta/r) = \hat{K}(1 - \hat{\delta}/\hat{r}), \tag{3.19b}$$
$$K\left(1 - \frac{\delta}{r + u_{\max}}\right) = \hat{K}\left(1 - \frac{\hat{\delta}}{\hat{r} + u_{\max}}\right). \tag{3.19c}$$

The first two conditions are the same as in the u_K case (Eq. (3.16a), Eq. (3.16b)), but the third is different. It can be shown via a similar calculation to the case of u_K in Sec. 3.5.1 that $(r, \delta, K) = (\hat{r}, \hat{\delta}, \hat{K})$ is the unique solution to Eq. (3.19), so a non-zero u_r will ensure structural global identifiability.

As in Sec. 3.5.1, we explore how Δr changes as the control parameters change. We centre our explorations around $u_{\max} = 0.02$ h^{-1}, $\tau = 10$ h, $\tau_0 = 10$ h, which makes the model identifiable, and vary one or two of these parameters at a time. In Fig. 3.14, we plot Δr as one of the three control parameters vary, with the other two fixed, and in Fig. 3.15, we plot Δr as a function of τ and τ_0, with $u_{\max}$ fixed.

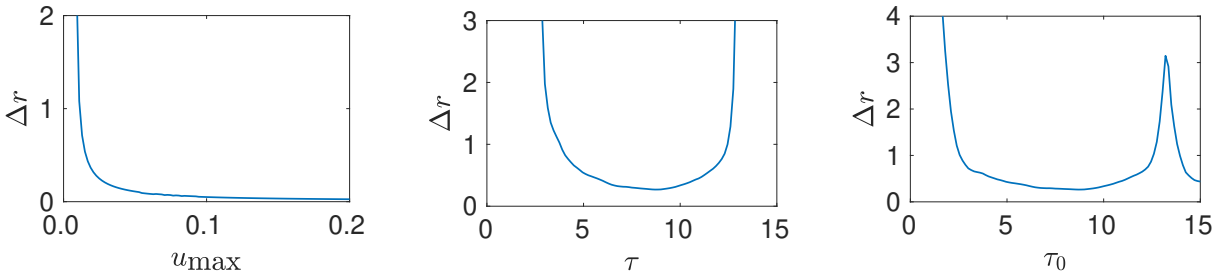


Figure 3.14: Similar to Fig. 3.11, except the applied control variable is u_r instead of u_K. The widths of the confidence intervals of r, Δr are shown as a function of one of the control parameters for u_r as defined in Eq. (3.13), with the other two held fixed at $u_{\max} = 0.02, \tau_0 = 10, \tau = 10$. The units are $[\tau] = [\tau_0] = \text{h}, [\Delta r] = \text{h}^{-1}$.

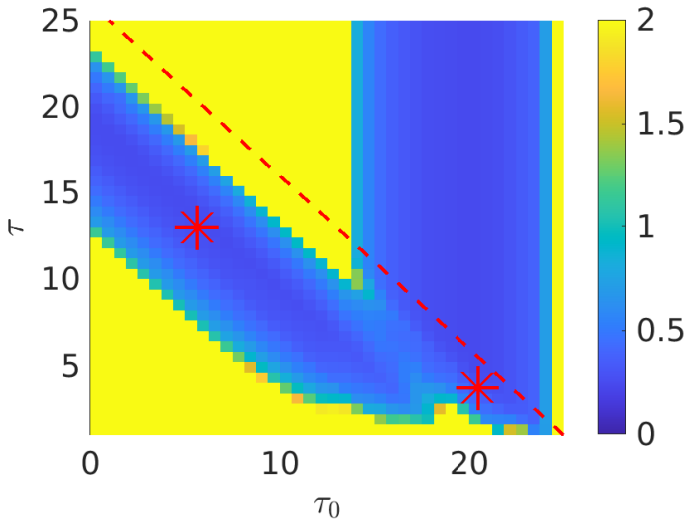


Figure 3.15: Same as Fig. 3.14, but with Δr shown as a function of τ_0 and τ, represented as a heat-map, with $u_{\max} = 0.02$ fixed. The red dashed line marks the line $\tau_0 + \tau = 25$. The two red stars denote the local minima of Δr at (τ_0, τ) =(5.6 h, 13.0 h), (20.5 h, 4.5 h). The units are $[\tau] = [\tau_0] = \text{h}, [\Delta r] = \text{h}^{-1}$.

Observe that the overall shape of Δr in the u_r case shown in Fig. 3.14 and 3.15 is similar to that in the u_K case shown in Fig. 3.11 and 3.12. A higher $u_{\max}$, the magnitude of the control, is always helpful for improving identifiability. The combinations of (τ_0, τ) that give the highest degree of identifiability satisfy either $\tau_0 < 16$ and $\tau_0 + \tau \approx 18$, or $\tau_0 > 16$ and $\tau_0 + \tau \approx 25$. The sensitive interval (Eq. (3.18)) for r is $\mathcal{T}_r \approx [8, 22]$, and the above combinations represent windows that roughly overlap with half of this interval, so our sensitivity-based heuristic is still valid in this case.

The results in this section show that an intelligent choice of the control variable, along with a strategic choice of the control, can be effective at removing structural non-identifiability and improving practical identifiability. This choice can be informed by using profile likelihoods to

inspect the dependence of uncertainty in parameter estimates on the control. We have also shown a heuristic way to select the control, which involves computing local sensitivities of the parameters and some simple calculations. This heuristic is intuitive to understand, and is much less computationally expensive.

3.6 Optimal control for model discrimination

In this section, we seek to optimally distinguish two models as in Sec. 3.4, but now by modulating the control variable instead of choosing an initial condition. We will mostly focus on the case where only one control is allowed at a time, since it may be impractical to assume we can apply multiple controls simultaneously in an experimental setting. Let $\boldsymbol{u}(t) = (u_r(t), u_\delta(t), u_K(t))$ denote the vector of control variables, where two of the controls are set to zero for any particular experiment. Let $\alpha_r, \alpha_\delta, \alpha_K \geq 0$ represent the cost of applying the associated control. It is reasonable to introduce a cost for turning the control on, since in general, depending on the context of the experiment, applying the control can be costly in terms of materials or effort. Furthermore, let $C_{\text{model},i}(t; u),\ i = 1, 2$ be the solution to the model

$$\frac{\mathrm{d}C(t)}{\mathrm{d}t} = g(C; \boldsymbol{\theta}_i; \boldsymbol{u}(t)), \tag{3.20}$$

where the g can be either g_a in Eq. (3.6), or g_b in Eq. (3.7). The problem we aim to solve can be formulated as

$$\max_{\boldsymbol{u}\in\mathcal{U}} J[\boldsymbol{u}] = \max_{\boldsymbol{u}\in\mathcal{U}} \int_0^T \left[C_{\text{model},1}(t; \boldsymbol{u}(t)) - C_{\text{model},2}(t; \boldsymbol{u}(t))\right]^2 - \boldsymbol{\alpha} \cdot \boldsymbol{u}(t)\mathrm{d}t, \tag{3.21}$$

where $\mathcal{U}$ denotes the set of values the control variable is allowed to take. We consider both the case of continuous control, and bang-bang control. In continuous control, each of the control variables may take any value between zero and an imposed upper bound, that is, $\mathcal{U} = [0, u_{r,\max}] \times [0, u_{\delta,\max}] \times [0, u_{K,\max}]$ if all three controls are applied. The upper bounds represent what is realistically achievable in an experimental setting, or a "safe" level beyond which we might alter the system dynamics too much for the model to be valid. In bang-bang control, the control variable is a piecewise constant function that may only take the value of zero or the upper bound, that is, $\mathcal{U} = \{0, u_{r,\max}\} \times \{0, u_{\delta,\max}\} \times \{0, u_{K,\max}\}$ if all three controls are applied. While continuous control allows more flexibility and greater room for optimisation, in an experiment it is often difficult to fine-tune the precise value of the control, so bang-bang control may be more applicable.

3.6.1 Applying PMP

We will be applying Pontryagin's Maximum Principle (PMP), introduced in general terms in Sec. C.1 (Theorem 1), to the control problem formulated in Eq. (3.20) and Eq. (3.21). PMP is an optimal control method suitable for this problem, since the control problem is posed as an open-loop problem with a somewhat unusual objective function, which nonetheless can be naturally handled by PMP. We also consider using the direct optimisation approach for solving the same problem in Appendix G. Direct optimisation, including reinforcement learning as a special case, is more flexible than PMP, but ignores the mathematical structure of the control problem. They are introduced in more detail in Ch. 5. Recall that in the statement of PMP, $\boldsymbol{x}$ is the vector of state variables, which is $\boldsymbol{x} = (C_1, C_2)$ for the current problem, and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ are the adjoint variables. The initial conditions of the state variables are fixed, $\boldsymbol{x}(0) = (C_0, C_0)$, so S_0, the set of allowed initial conditions, consists of a single point. Consequently, the transversality condition on the adjoint variables is simply $\boldsymbol{\lambda}(0) \in \mathbb{R}^2$, i.e., it is free. There are no conditions prescribed for the state variables at the final time, hence $\boldsymbol{x}(T) \in \mathbb{R}^2$ and $\boldsymbol{\lambda}(T) = (0, 0)$. The dynamics of the adjoint variables are given in terms of H, the Hamiltonian. The optimal control $u^*(t)$ is given by maximising H as a function of u at every point in time. The abnormal multiplier λ_0 can either be negative in the generic case, in which case it can be set to -1 without loss of generality, or zero in the degenerate case.

For the control problem at hand, it can be shown that $\lambda_0 = 0$, together with the transversality condition, implies $\boldsymbol{\lambda}(t) \equiv \mathbf{0}$ for all t, which means $H = 0$ is independent of all inputs, giving us no information on the control u, and rendering the conclusions of the theorem trivial. Hence we will not need to consider the degenerate case from now on, and focus on the generic case by setting $\lambda_0 = 1$. With these considerations, we can simplify the five PMP conditions in Theorem 1 to the following three conditions:

1. (Canonical equations and the Hamiltonian)

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{u}) = -(C_1 - C_2)^2 + \boldsymbol{\alpha} \cdot \boldsymbol{u}, \tag{3.22a}$$

$$H(C_1, C_2, \lambda_1, \lambda_2; \boldsymbol{u}) = (C_1 - C_2)^2 - \boldsymbol{\alpha} \cdot \boldsymbol{u} + \lambda_1 g(C_1; \boldsymbol{\theta}_1; \boldsymbol{u}) + \lambda_2 g(C_2; \boldsymbol{\theta}_2; \boldsymbol{u}), \tag{3.22b}$$

$$\frac{\mathrm{d}C_i}{\mathrm{d}t} = g(C_i; \boldsymbol{\theta}_i; \boldsymbol{u}(t)), \quad i = 1, 2, \tag{3.22c}$$

$$\frac{\mathrm{d}\lambda_i}{\mathrm{d}t} = \Lambda_i(\boldsymbol{x}; \boldsymbol{\theta}_i; \boldsymbol{u}(t)) = (-1)^i 2(C_1 - C_2) - \lambda_i \frac{\partial g(C; \boldsymbol{\theta}_i; \boldsymbol{u})}{\partial C}\bigg|_{C = C_i(t)}, \quad i = 1, 2, \tag{3.22d}$$

where g is either the model with additive control in Eq. (3.6) or the model with multiplicative control in Eq. (3.7).

2. (Point-wise maximum condition)

$$\boldsymbol{u}^*(t) = \underset{\boldsymbol{u}\in\mathcal{U}}{\operatorname{argmax}}\, H(\boldsymbol{x}^*(t), \boldsymbol{\lambda}^*(t); \boldsymbol{u}),\ \forall t \in [0, T]. \tag{3.23}$$

3. (Transversality conditions)

$$C_i(0) = C_0, \quad C_i(T) \text{ free}, \quad \lambda_i(0) \text{ free}, \quad \lambda_i(T) = 0. \tag{3.24}$$

The complexity of these conditions means little analytical progress can be made, but there is at least one simple fact we can prove regarding the optimal control, which is that it must vanish at $t = T$. This is because the boundary condition for $\boldsymbol{\lambda}(T) = (0, 0)$ in Eq. (3.24) means at $t = T$, the last two terms in the Hamiltonian H (Eq. (3.22b)) disappear, and H depends on $\boldsymbol{u}$ only via the $-\boldsymbol{\alpha} \cdot \boldsymbol{u}$ term, which implies that the maximum of H occurs at $\boldsymbol{u}^*(T) = \mathbf{0}$. As turning on the control near the end of the simulation would do little to increase the difference between the two models, this is expected.

3.6.2 Numerical methods

It will be impossible, or at least difficult, to analytically find a solution that satisfies the PMP conditions in Eqs. (3.22–3.24), so we will focus on using numerical methods to solve the control problem. The numerical methods are introduced in general terms in Sec. C.2. We will implement the FBS algorithm for the problem of continuous control, and direct optimisation for bang-bang control. We additionally consider the direct optimisation and hybrid methods for continuous control in Appendix G.

The FBS algorithm is outlined in Algorithm 2. We use an initial update rate of $\omega^0 = 0.1$, and $\epsilon = 1$ as the termination criterion, which is very small compared to the typical magnitude of $J[\boldsymbol{u}^*]$ in our case, which is around 10^6–10^7 $(\text{cells/mm}^2)^2$. The ansatz for the control used to initialise the FBS algorithm is chosen to be $\boldsymbol{u}^1(t) = \boldsymbol{u}_{\max}/2$. The initial value problems in step (III.1) and (III.2) of Algorithm 2 are solved with MATLAB's *ode45*. We impose a maximum step size of $T/100$, then store the solutions at $n_t = 100$ evenly spaced points in $[0, T]$ by linearly interpolating the output of *ode45* at these points. The objective function $J[u]$ is calculated with the trapezium rule using the same points in step (III.3), and the control iterate $\boldsymbol{u}^{\text{new}}(t)$

is again calculated at the same points in step (III.5), using MATLAB's *globalsearch* to perform the point-wise optimisation.

In the bang-bang control scenario, we restrict the control to be a window function as in Sec. 3.5 in order to simplify the problem, and parameterise it with two control variables: τ_0, the onset of control, and τ, the length of the window during which the control is turned on. The height of the window function is set to $u_{\max}$. This restriction is reasonable because in most settings, we expect the number of jumps to be small, and the results of the continuous control problem suggest that the optimal bang-bang control will take the form of such a window function.

3.6.3 Logistic model with additive controls

We seek to optimally discriminate between two logistic models represented by the following parameter values:

$$(r, \delta, \gamma, K)_1 = (0.45, 0.15, 1, 3900), \quad (r, \delta, \gamma, K)_2 = (0.3, 0, 1, 2600). \tag{3.25}$$

The values of the first parameter set coincide with Eq. (3.14), and the values of the second parameter set are the same as in Eq. (3.9). These two models have the same r_{eff} and K_{eff} as in Eq. (3.4), which means they give rise to identical solutions without control. We might be interested in discriminating these two models from a biological perspective to determine whether the mechanism represented by the linear death term genuinely exists within the system.

We first consider the model with additive control (Eq. (3.6)). As discussed in Sec. 3.5, in this case, the addition of u_δ does not remove structural identifiability, as both models yield the exact same solution for any u_δ. Therefore, we consider only applying u_K or u_r.

In the case of applying u_K, we take $u_{K,\max} = 1200$ cells/mm^2 and $\alpha_K = 0.03$ cells/mm^2. A much larger α_K (e.g., $\alpha_K = 0.1$ cells/mm^2) results in $u_K^*(t) \equiv 0$, as the cost of the control is too great to be compensated by the reward of being able to discriminate the models. On the other hand, a much smaller α_K (e.g., $\alpha_K = 0.005$ cells/mm^2) results in $u_K^*(t) \equiv u_{K,\max}$, as a larger u_K is always helpful in discriminating between the two models in this case, and the cost of turning u_K on is negligible. Therefore, there is a range for α_K to make the problem non-trivial.

In Fig. 3.16, we show the result of applying the FBS algorithm to the optimal control problem, with u_K being the applied control variable. The strategy implied by the resulting optimal control can be summarised as allowing both solutions to reach roughly $K_{\text{eff}}/2$ without much intervention, then abruptly turn on the control. In the early stages of the experiment (for

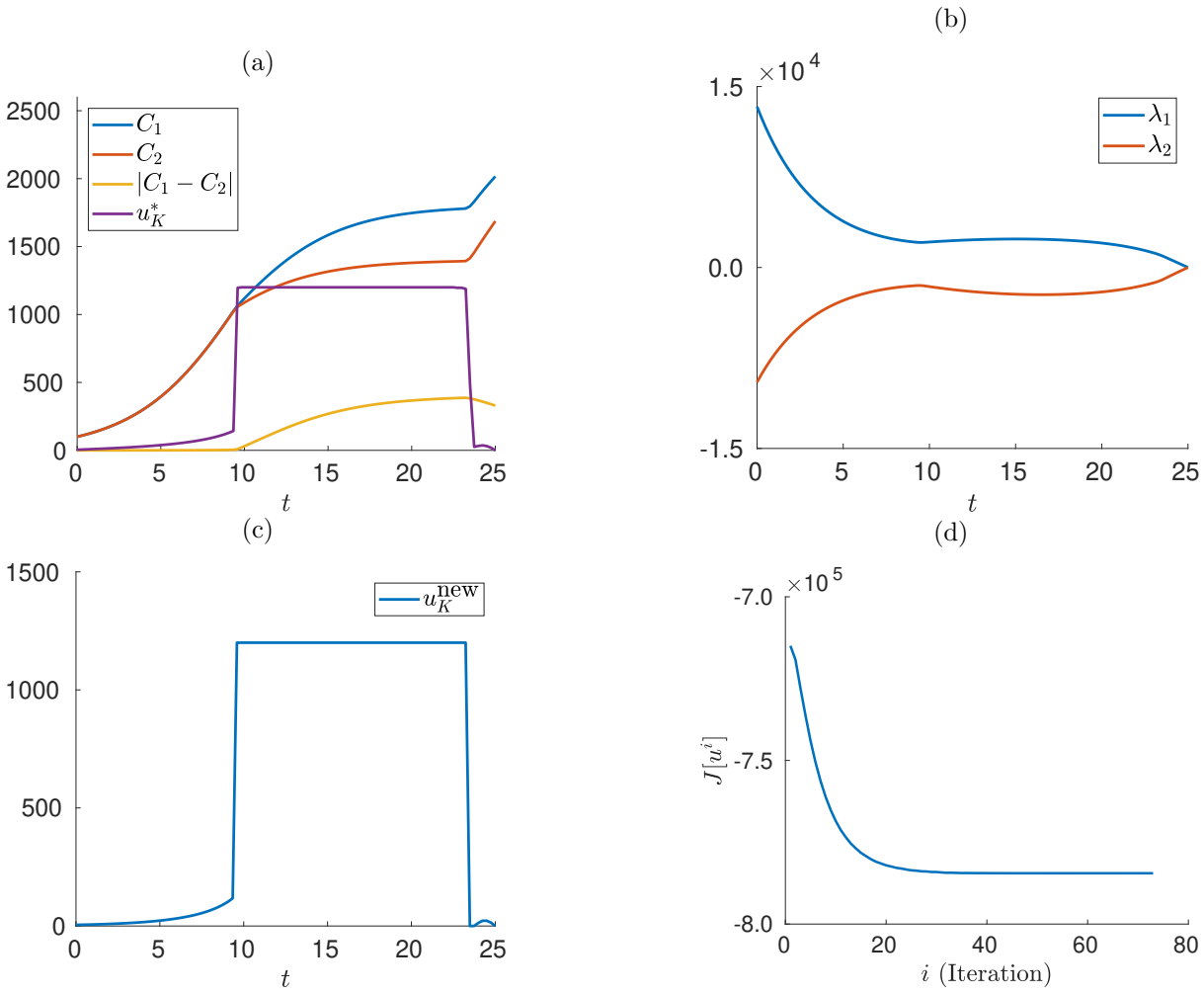


Figure 3.16: Results of the FBS algorithm (Algorithm 2) applied to the optimal control problem (Eqs. (3.22–3.24)) with the additive control model (Eq. (3.6)), and $\boldsymbol{u} = u_K$, $u_{K,\max} = 1200$ cells/mm^2, and $\alpha_K = 0.03$ cells/mm^2. (a) The optimal control $u_K^*(t)$, as found by the algorithm, and the corresponding trajectories of the state variables C_1, C_2, as well as their difference over time. (b) The trajectories of the adjoint variables λ_1, λ_2 corresponding to the optimal control. (c) u_K^{new}, the update to the control, computed at the final iteration before termination of the algorithm. It is very close to u_K^*, suggesting that further iterations will not lead to significant changes in the control. (d) The cost functional $J[u]$ over the iterations, showing its convergence near the final iterations. The units are given in Eq. (3.2), and $[u_K]$ =cells/mm^2, $[J] = (\text{cells/mm}^2)^2$.

small t), the carrying capacity K (and therefore the control variable u_K) has relatively little impact on the solutions of both models, so the control remains low. From $t \approx 9.5$ h, the impact of the control on discriminating the two model solutions is great enough to overcome the cost of applying the control, and u_K is turned on to the maximum, where it remains until $t \approx 23.3$ h. After this point, the control is almost turned off completely until $t = T$, when it is exactly zero, as we know it must be. We can relate this control strategy to local sensitivity, ϕ_K. In Fig. 3.13(c), we can observe that ϕ_K is monotonically increasing with time. The period of time where $u_K^* = u_{K,\max}$ roughly coincides with the time window where ϕ_K is large. This is expected,

since the control is likely to be most effective in distinguishing between the model solutions at points where the models are sensitive to the parameter associated with the control.

Interestingly, the optimal control has a small bump near $t \approx 24$ h, the reason for its presence is not immediately apparent. Manually removing the bump results in a higher cost functional, suggesting that it is valid and not simply a numerical artifact. We have also applied direct optimisation and the hybrid method, introduced in Sec. C.2, to this control problem, to compare the different numerical methods. The resulting controls have nearly identical shape to the control found using the FBS algorithm. We present the results for the alternative methods in Appendix G.

Next, we consider a few variations of the optimal control problem, to determine how various factors impact the optimal control. Changing the value of α_K does not result in a qualitative change in the optimal control. Fig. 3.17 shows that with a higher value of $\alpha_k = 0.05$ cells/mm^2, the control is turned on over a shorter window between $t \approx 12$ h and $t \approx 22.3$ h, but retains the same shape as in Fig. 3.16. The small bump near $t = T$ is still present. The optimal bang-bang control for the same α, as found by direct optimisation, is turned on at $t \approx 9.4$ h, almost the same time as when the optimal continuous control is ramped up sharply to its maximum, and turned off at $t \approx 23.4$ h, very close to the time when the optimal continuous control decreases sharply to near zero. The bang-bang optimal control achieves an objective value of $J = -779401$ (cells/mm^2)^2, only slightly worse than the continuous optimal control, which is $J = -784470$ (cells/mm^2)^2. These results show that in this case, we do not gain much from allowing the control to take continuous values, and a bang-bang optimal control is both easier to compute, and experimentally easier to implement.

The optimal control for model discrimination is quite different from the optimal control for improving parameter identifiability, discussed in Sec. 3.5. Once more, this illustrates that model discrimination and improving identifiability are distinct problems. As argued in Sec. 3.3, in the model discrimination problem, we are essentially proposing a new experiment in addition to the experiment with which we calibrated our two models. For the new experiment to be the most useful, we want to observe the system behaviour in an environment as different to the first experiment as possible, i.e., with the control turned to a maximum, whereas in the problem of identifiability improvement, we are essentially conducting two experiments at once, so the data should be split between when the control is on or off.

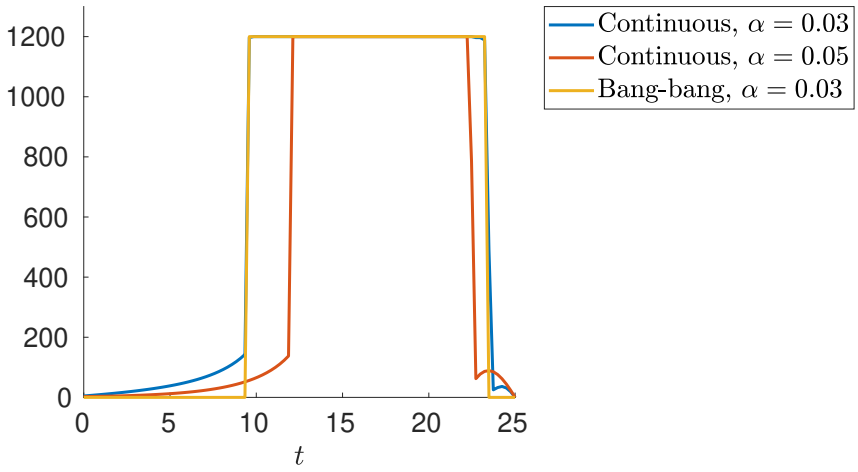


Figure 3.17: Comparison of the solutions of variations of the optimal control problem. The blue curve is the same control as in Fig. 3.16. The orange curve is the optimal continuous control with a different value for the weight of control cost, α, and the yellow curve is the optimal bang-bang control using the same value of α as the blue curve. The units are $[t] = \text{h}, [u_K] = \text{cells/mm}^2$.

Next, we consider the case of applying u_r as the control variable. The optimal control found by the FBS algorithm is shown in Fig. 3.18. The optimal control is roughly constant, at $u_r \approx 0.7$ h^{-1} over most of the domain, with a noticeable bump near the inflection point of the solutions, and ramps down smoothly to zero as $t \to T$. The strategy seems to be to turn on the control early so that both solutions reach their carrying capacity quickly, then maintain the control in order to observe the difference in K_{eff} for the two models. The reason for a local maximum in u_r^* near the inflection point of the two model solutions is not immediately clear, but a possible explanation can be given using local sensitivity. Examine the plot of local sensitivity ϕ_r (Fig. 3.13(a)) and the associated model solution (yellow curve in Fig. 3.7), we can observe that the inflection point of the model solution is near the global maximum of ϕ_r. Therefore, the local maximum of the control, u_r^*, is close to where both model solutions are most sensitive to the parameter r, which is sensible for the objective of distinguishing the two model solutions. The window in time where the optimal bang-bang control is turned on covers most of the time span of the simulation, but it is turned off a short time before $t = T$, around the same time as the optimal continuous control is ramped down. The bang-bang control achieved an objective value of $J = -4618503$ (cells/mm^2)^2, compared to $J = -6388826$ (cells/mm^2)^2 for the optimal continuous control, which is a noticeably worse performance.

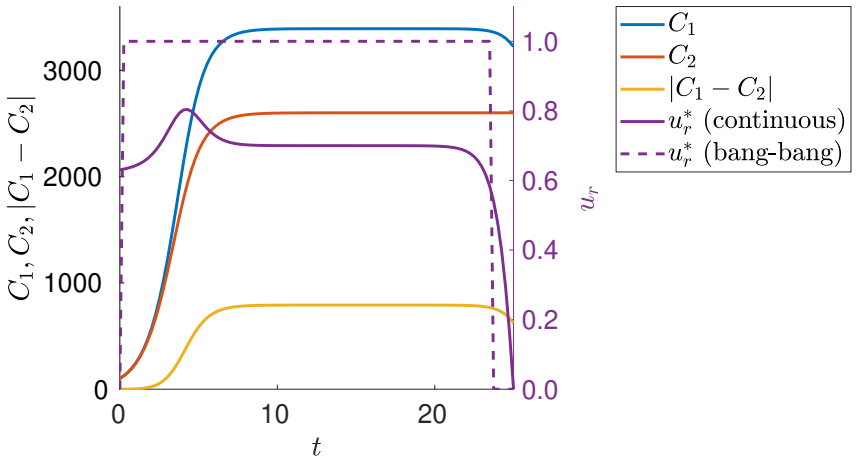


Figure 3.18: Results of FBS (Algorithm 2) applied to the optimal control problem (Eqs. (3.22–3.24)) with the additive control model (Eq. (3.6)), and $\boldsymbol{u} = u_r$, $u_{r,\max} = 1\ \mathrm{h}^{-1}$, and $\alpha_r = 500000$ h(cells/mm^2)^2, showing the optimal control $u_K^*(t)$ found by the algorithm, the corresponding trajectories of the state variables C_1, C_2, and their difference over time. The optimal bang-bang control is also shown for comparison. The units are $[C] = \text{cells/mm}^2$, $[t] = \text{h}$, $[u_r] = \text{h}^{-1}$.

3.6.4 Logistic model with multiplicative controls

Next, we look at the case of multiplicative controls for the logistic model, where the model equation is given by Eq. (3.7), and the two model-parameter combinations to be discriminated remain the same as the previous section, given in Eq. (3.25). In contrast to the case of additive controls, where u_δ is ineffective in removing structural non-identifiability and u_r, u_K are effective, as shown in Sec. 3.5, in the multiplicative case u_K is ineffective in removing structural non-identifiability, whereas u_r, u_δ are effective. We present the results obtained by the FBS algorithm in Fig. 3.19. For u_r, the strategy implied by the optimal control seems to be to turn the control on during most of the time span of the simulation, then ramp it down smoothly to zero as $t \to T$. This is because u_r has an impact on the difference between the two solutions throughout the time span of the simulation, but towards the end of the simulation, the cost of keeping it on outweighs the gain in the difference between model solutions. Comparing the shape of u_r^* in Fig. 3.19(a) to that in the additive control case shown in Fig. 3.18, we can observe that in both cases, u_r^* has a local maximum near the inflection points of the two model solutions but, in the multiplicative case, u_r^* decreases more rapidly after that point, whereas it stays near a constant value for a period of time in the additive case. This is because u_r has a larger effect on the steady state of the models in the additive case compared to the multiplicative case. Therefore, as the model solutions approach the steady state K_{eff}, the effect of u_r on model difference diminishes in the multiplicative case, while the effect remain significant in the additive case after the model

solution approaches the steady state.

In the case of u_δ, the optimal control strategy involves setting its value to the maximum for most of the simulation's time span, followed by a sharp decrease in the control as t approaches T. This can be explained by the fact that the model solution C_2 is entirely unaffected by u_δ, due to its parameter δ_2 being equal to zero, whereas C_1 is influenced by u_δ at all time. Maintaining a high value of u_δ depresses C_1 relative to C_2, which is helpful for distinguishing the two model solutions in all stages of the experiment.

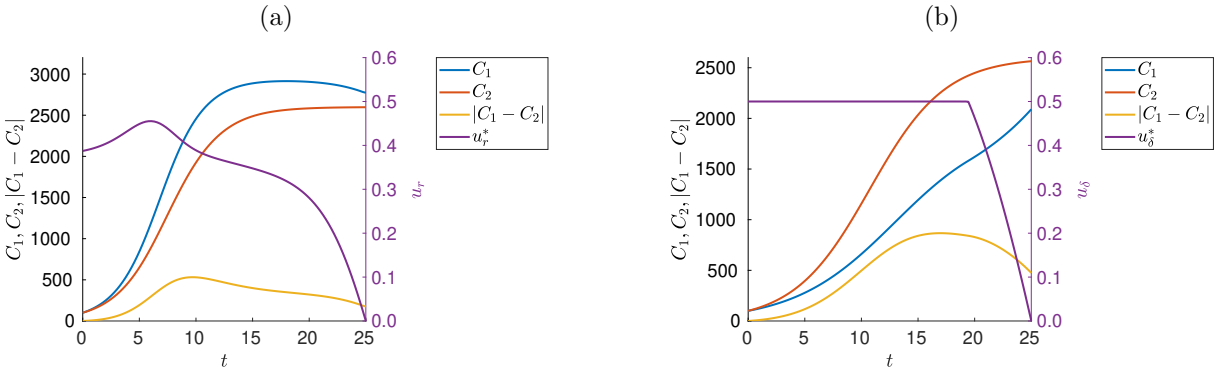


Figure 3.19: Results of the FBS (Algorithm 2) applied to the optimal control problem (Eqs. (3.22–3.24)) with the multiplicative control model (Eq. (3.7)). (a) The control variable is u_r, with $u_r^{\max} = 0.5$, $\alpha_r = 700000$ (cells/mm^2)^2. (b) The control variable is u_δ, with $u_\delta^{\max} = 0.5$, $\alpha_\delta = 1300000$ (cells/mm^2)^2. The units are $[C] =$ cells/mm^2, $[t] =$ h, and u_r and u_δ are dimensionless.

Summarising the results in Sec. 3.6.3 and Sec. 3.6.4, we find that it is possible to design an optimal control to enable us to differentiate between two models that were previously indistinguishable without such intervention. We can understand the optimal control strategies intuitively by looking at how the control variable affects the difference between the models at various points in time, using local sensitivities of the parameters associated with the control variable as a rough guide. Whether the control variables act on model parameters additively or multiplicatively can make a big difference, as an efficient control strategy in one scenario may prove entirely ineffective in the other.

3.6.5 Richards model

We now consider the problem of discriminating between two Richards models with the following parameter sets:

$$\begin{aligned}(r, \delta, \gamma, K)_1 &= (0.225\ \text{h}^{-1}, 0\ \text{h}^{-1}, 8, 2381\ \text{cells/mm}^2),\\ (r, \delta, \gamma, K)_2 &= (0.235\ \text{h}^{-1}, 0\ \text{h}^{-1}, 3, 2433\ \text{cells/mm}^2).\end{aligned} \tag{3.26}$$

These parameters are chosen such that their solutions in the absence of controls are very similar, but not identical. The two solutions only differ noticeably as they approach the carrying capacity: the model with larger γ is sharper between the inflection point and K (in Fig. 3.7, the yellow curve corresponds to the first parameter set in Eq. (3.26), and the solution of the second parameter set is very similar to the blue curve in the same figure). Both models have the coefficients of the death term, δ, set to zero. This is so we can illustrate a scenario where the model without control is structurally identifiable, but we might not have sufficient data for it to be practically identifiable. We will only present results for the additive control case, which is sufficient to illustrate the method.

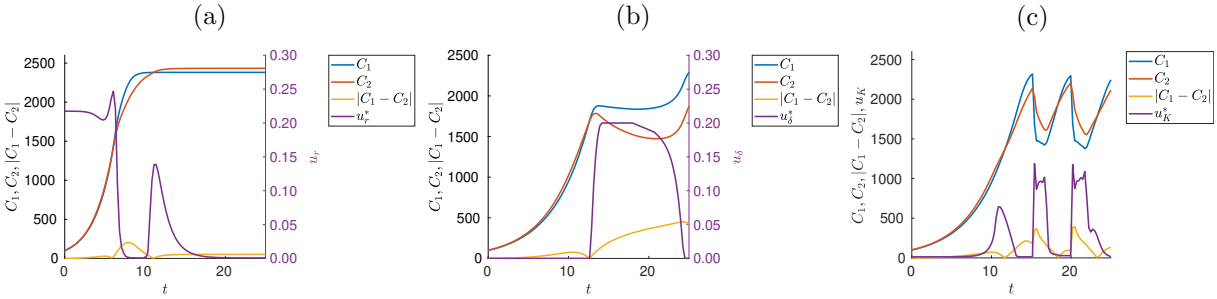


Figure 3.20: Results of the FBS algorithm (Algorithm 2) applied to the optimal control problem (Eqs. (3.22–3.24)) with the model parameters in Eq. (3.26), using the additive control model (Eq. (3.6)). (a) The control variable is u_r, with $u_r^{\max} = 0.4$ h^{-1}, $\alpha_r = 30000$ h(cells/mm^2)^2. (b) The control variable is u_δ, with $u_\delta^{\max} = 0.2$ h^{-1}, $\alpha_\delta = 500000$ h(cells/mm^2)^2. (c) The control variable is u_K, with $u_K^{\max} = 1200$ cells/mm^2, $\alpha_K = 0.03$ cells/mm^2. The units are $[C] =$ cells/mm^2, $[t] =$ h, and u_r, u_δ and u_K are dimensionless.

In Fig. 3.20, we present the optimal control obtained with the FBS algorithm in the case where one of the three control variables, u_r, u_δ, and u_K, is applied individually. The weight of the cost of control, α_θ, and the imposed upper bound on the controls, $u_\theta^{\max}$ are given in the caption. Out of the three, u_r helps the least to distinguish between the two models. The optimal control u_r^* never hits the imposed upper bound $u_r^{\max}$, and it is turned on relatively briefly. The difference between the models in the presence of the control is negligibly greater than that in the absence of the control.

Both u_δ and u_K seem to be much more effective at inducing noticeable differences in the two model solutions. The control strategy implied by u_δ^* allows both models to approach their respective carrying capacity, then turns u_δ on abruptly to maximum. This is a sensible strategy, the effect of u_δ is similar for both models in the earlier phase, where both model solutions, C_1 and C_2, behaves approximately as exponential growth, and u_δ lowers the effective growth rate r_{eff} by the same amount for both models, so it is useless to turn on the control in that phase.

The control is turned on during the later saturation phase, where both model solutions approach their respective steady states. The steady state, with u_δ treated as a constant, can be written as

$$K\left(1-\frac{\delta+u_\delta}{r}\right)^{1/\gamma}.$$

A greater value of γ diminishes the effect of u_δ on the steady state, so the steady state of model 2, which has a lower γ, is decreased much more by a positive u_δ compared to that of model 1. The difference in steady states means turning the control on in the saturation phase allows the two models to be easily distinguished.

In contrast, u_K modifies the K_{eff} of both models by the same amount, which means the case of using u_K as the control variable requires a different strategy from the case of using u_δ. The strategy in the case of u_K is to first allow both models to get close, but not too close, to their respective carrying capacities, then cyclically and abruptly turn u_K on and off so that both model solutions stay near, but not too close, to their respective carrying capacities. Recall that in the absence of the control, the two model solutions are best distinguished by their behaviour just before they approach K. This control strategy maximises the amount of time spent within the regime where the behaviours of the two models are most different, and thereby maximises the difference between the two model solutions.

3.7 Discussion

This chapter examined two objectives: maximising the practical parameter identifiability of a model, and discriminating between two models to the greatest extent, either by choosing the most appropriate initial conditions, or by optimally designing a control. This work improved upon existing research by considering a more general class of continuous control, which necessitates using a more sophisticated numerical algorithm to solve the optimal control problem. We also investigated whether a continuous control can be more effective at achieving the objectives compared to simpler controls, such as bang-bang controls. This work also clarifies the similarities and distinctions between improving identifiability and model discrimination.

In Sec. 3.3, we demonstrated that the parameters in a model can have their own, different optimal initial conditions that maximise their identifiability. For example, r in the logistic model is more identifiable if the initial cell density is low, whereas the optimal initial density for maximising the identifiability of K is higher. Hence, there might not be a single optimal initial condition that maximises the identifiability of all parameters in the model. Nonetheless,

by plotting the relationship between the initial conditions and the widths of the confidence intervals for parameter values, we can try to find an initial condition that compromises between the different parameters, so that all parameters are at least practically identifiable, which can serve as the first experiment in an iterative experiment—parameter inference—experimental design cycle. Alternatively, we may propose a set of experiments, equal in number to the parameters, with each experiment focusing on optimising the identifiability of one parameter at a time.

We have established that the problems of model discrimination and improving identifiability, although related, represent distinct objectives. The optimal initial conditions and control for achieving those objectives are different. This difference arises from the fact that the two objectives rely on different amounts of prior knowledge concerning the system. In tackling the problem of improving identifiability, we assumed little prior knowledge about parameter distributions, other than the bounds of a very wide range of plausible values for the parameters. In contrast, for model discrimination, we used at least one set of data from prior experiments to fix the model-parameter combinations.

Structural non-identifiability can potentially be addressed by adding a control variable to the model, and incorporating the corresponding external stimulus into the experiment, as we have seen in Sec. 3.5. The logistic model with a death term is structurally non-identifiable, as a change in δ, the death rate, can be perfectly compensated by a corresponding change in r and K. Adding the appropriate control variable allows the system to be observed in a state that otherwise would not be accessible, providing additional information on the underlying mechanisms. For example, using u_r as a control variable, with a window function for control, allows us to observe the system in two states, a state where the control is off, and another where the control is on. In this scenario, a change in δ can no longer be compensated by changing r and K to obtain identical behaviour in both states. However, this approach can only succeed if knowledge of how the stimulus impacts the mechanisms (e.g., whether it induces an additive or multiplicative change in a parameter, and to what extent) is available. Without this knowledge, introducing the control variable introduces additional unknowns, which prevents the derivation of additional information on the parameters. For example, suppose we wish to introduce an external stimulus that modulates the growth rate, represented by the control variable u_r, but do not know how much this stimulus changes r in advance. It would be necessary to introduce another unknown parameter, say ρ, as coefficient of u_r in the model. This ρ must be estimated

in the same way as the original model parameters r, δ and K. As a result, the inference problem becomes more complicated and requires more data, preventing any gains from introducing the control variable. One possibility of circumventing this issue is to test the control mechanism (say a growth signal, or a toxin) on a system with known dynamics (e.g., a cell type whose growth is fully understood). The mechanism might not have the same effect on the system of interest (e.g., the toxin might not affect the cell type of interest in the same way), but this approach provides a starting point to iteratively improve our understanding of the system with more experiments.

Section 3.6 demonstrated that the optimal control for model discrimination depends on the control variable. The optimal strategy for control variables acting on different model parameters can be completely different. For example, in Fig. 3.20, the optimal strategies for the three control variables are qualitatively and quantitatively distinct. In some cases, the shape of the optimal continuous control is close to a window function, such as the case of an additive control u_K for distinguishing two logistic models with death terms in Fig. 3.16, therefore its performance is only a minor improvement upon the optimal bang-bang control. In this case, we might prefer to implement the bang-bang control in practice, since it requires relatively simple experimental techniques, whereas continuous control requires the ability to modulate the control mechanism more precisely. In other cases, the optimal continuous control significantly out-performs the optimal bang-bang control, due to the greater flexibility we have in its design. Such is the case for an additive u_r in Fig. 3.18, where the optimal continuous control has a significantly lower cost compared to the optimal bang-bang control. In these cases, the implementation of the continuous control is preferred, if it is feasible to carry out in the experiment. Note that bang-bang controls, or more generally piecewise constant controls, are more commonly used in control applications, due to the fact that they are both easier to compute and to implement compared to continuous controls. The analysis in this chapter provides a way to determine how much can be gained by considering continuous controls over bang-bang controls.

The FBS algorithm with adaptive update rate, based on PMP, is adequate for solving continuous optimal control problems, at least for the examples considered in Sec. 3.6. In Appendix G, we additionally implemented direct optimisation for the same control problems, and found that the quality (as measured by the cost function) of the control found by direct optimisation is comparable to those found by the FBS algorithm. However, direct optimisation is more susceptible to introducing numerical artifacts, often manifesting as high-frequency oscillations in the

control it produces. A significant drawback of direct optimisation lies in its sensitivity to the initial ansatz for the control. A poorly chosen initialisation can result in sub-optimal controls. This suggests that a hybrid method, which uses the output of the FBS algorithm to initialise direct optimisation, may overcome this weakness. We demonstrated this hybrid method in Appendix G, and found that it can provide a control with a lower cost compared to the output of both the FBS algorithm and a naively initialised direct optimisation procedure, without requiring excessive computational time. We have also considered the case of simultaneously applying multiple control variables, which led to significant computational challenges for all three methods considered. The partial results for this case are given in Appendix G.

In the future, the scope of experimental design can be extended to PDE models. For example, for models of cell invasion assays in Chapter 2, we might ask what is the optimal shape of the initial cell population, which is inherently a spatial problem that requires a PDE model. While this extension is conceptually straight-forward, it poses significant computational challenges, especially in finding the global optimum of likelihood functions during the computations for profile likelihoods, a challenge we have previously encountered.

The problem of finding optimal initial conditions for model discrimination can also be formulated in the PMP framework, despite the fact that no control variable is present. Compared to the conditions we obtain from applying PMP to the optimal control problem (Eqs. (3.22)–(3.24)), the Lagrangian $\mathcal{L}$ and Hamiltonian H no longer depend on $\boldsymbol{u}$, the point-wise maximum condition does not apply, and the transversality conditions become

$$C_1(0) = C_2(0), \quad \lambda_1(T) = -\lambda_2(t).$$

While for ODE models, it is much easier to optimise directly rather than using the PMP conditions, the PMP conditions might prove useful in the PDE case.

Having investigated experimental design and its role in improving parameter identifiability, the next chapter shifts focus towards exploring pattern formation in developmental biology. Given what we have learned about parameter identifiability, we consider a scenario where an appropriate model for pattern formation has been selected, its parameters are identifiable, and accurate estimates of the parameter values have been obtained through a series of carefully designed experiments. By exploring the interactions between pattern formation and a phenomenon known as a wave of competency, we pave the way towards using control theory to engineer desired patterns in biological systems.

Chapter 4

Control of diffusion-driven pattern formation behind a wave of competency

Having explored parameter identifiability in the preceding chapters, our attention now turns toward exploring the behaviour of a PDE model of pattern formation, to focus on the alignment of stripe patterns in biology, and its relation to the "wave of competency". A wave of competency, in the context of developmental biology, refers to a phenomena where tissues mature behind a moving front, and becomes capable of responding to certain signals. Pattern formation takes place behind a wave of competency in several biological contexts, such as the plumage patterns of birds [11] and stripes on certain species of fish [204]. Currently, the effects of a wave of competency on the patterning outcome is not well-understood. In this study, we use Turing's diffusion-driven instability (DDI) model to study pattern formation behind a wave of competency, under a range of wave speeds. Numerical simulations show that in one spatial dimension a slower wave speed drives a sequence of peak splittings in the pattern, whereas a higher wave speed leads to peak insertions. In two spatial dimensions, we observe stripes that are either perpendicular or parallel to the moving boundary under slow or fast wave speeds, respectively. We argue that there is a correspondence between the one- and two-dimensional phenomena, and hypothesise that (as others have) pattern formation behind a wave of competency can account for the pattern organisation observed in many biological systems. The work in this chapter is published as Y. Liu, P. K. Maini, and R. E. Baker, *Control of diffusion-driven pattern formation behind a wave of competency* in Physica D: Nonlinear Phenomena [117]. The author of this thesis conducted all of the analysis in the paper.

4.1 Biological motivations of diffusion-driven pattern formation

The DDI mechanism proposed by Turing [186] has been used as a canonical model for pattern formation in mathematical biology [135]. Originally, the model was proposed as a mechanism for cell fate specification, but it has since been adopted to describe a wide range of biological phenomena [39, 90, 204]. In this framework, cell fate is hypothesised to be determined by underlying spatially pre-patterned chemical cues. Turing termed these chemicals morphogens, and proposed that different rates of diffusion of these morphogens in the system destabilise a spatially homogeneous steady state, leading to spatially inhomogeneous chemical profiles (patterns). The basic applications of DDI models are on fixed domains, with kinetic parameters constant throughout the entire domain (see e.g., [135, Ch. 2.1]).

There have been many extensions to the basic DDI model proposed in order to obtain a richer, and more robust range of patterning behaviours, reflecting those observed in biology [103]. One such extension is domain growth. The most basic form of growth, where the underlying tissue grows uniformly, results in dilution effects on the morphogens, which can lead to interesting behaviours not found on a fixed domain, as reported by [15, 49, 50, 124, 189], among others.

In contrast, it has been hypothesised that, in certain cases, a related mechanism called a "wave of competency" may serve as a more biologically relevant way to generate interesting patterning behaviours. In this setting, the overall domain size is fixed, but it is divided into two subdomains by a propagating front representing the so-called wave of competency. Patterning can take place only in the subdomain that lies behind the propagating wave of competency, and this subdomain expands as the wave advances. In the context of patterning via a DDI, we assume that the model parameters are such that the part of the domain behind the wave of competency is capable of displaying a DDI, while the part ahead of the wave is not. The goal of this chapter is to understand the patterning behaviour under different speeds of propagation of the wave of competency.

One possible biological interpretation for the wave of competency is as follows. The overall fixed domain represents the entirety of the underlying tissue. The subdomain in front of the wave represents immature tissue that cannot support patterning until it matures further, i.e., becomes "competent", and the subdomain behind the wave represents mature tissue, forming the effective domain of patterning. Experimental support for the existence of a wave of competency comes from the plumage patterning of birds. For example, Bailleul et al. [11] argued that early

mesodermal development leads to the formation of feather fields (the patterning subdomains), which expand on a similar time scale to that of pattern formation. In an earlier study, Jung et al. [87] identified possible molecular candidates encoding the wave of competency for feather formation, and Jiang et al. [81] provided further experimental evidence. Mou et al. [133] examined a similar mechanism that restricts feather formation to one side of a front.

The study by Bailleul et al. [11] found that while a DDI model on a fixed domain could reproduce the final pattern of follicles in birds, it was not able to reproduce the correct sequence of emergence of the feather follicles, nor the consistent orientation of the stripe-shaped transient structures known as feather tracts, which exist prior to the formation of follicles. This is because with biologically realistic initial conditions, which are usually taken to be the homogeneous steady state plus a noisy perturbation, patterns emerge simultaneously throughout the domain. Hence, they proposed a model which combined chemotaxis and DDI mechanisms by coupling the dynamics of two morphogens with the evolution of cell density. In their model, the cell population undergoes logistic growth and diffusion, in addition to responding chemotactically to one of the morphogens. The morphogen concentrations are governed by a DDI model, which is coupled to the cell density model via the reaction terms. If the cell density is assumed to be a constant parameter, then the DDI system possesses a stable spatially uniform steady state, which becomes unstable when the cell density becomes sufficiently high. This means that the region of high cell density forms the patterning subdomain, which is initially small, but expands outwards under the effects of chemotaxis and diffusion. This model exhibits sequential pattern formation behaviour, in agreement with observations of feather formation.

Here, we seek to better understand this type of patterning scenario by isolating the effect of increases in the size of the patterning subdomain on the dynamics of a model capable of exhibiting a DDI. The resulting model can be viewed as a classical DDI model with a minimal modification that represents a wave of competency, which is analogous to cell density in the feather model developed by [11]. In essence, the wave of competency takes the form of a moving boundary that divides the domain into two subdomains. Within each subdomain, the system behaves as a classical reaction-diffusion system, but the dynamics differ between the subdomains, with a DDI only possible for the subdomain that lies behind the moving boundary. We are interested in understanding whether such a modification could enable the model to reproduce sequential pattern formation.



Zebrafish (*Danio rerio*) | Golden Trevally (*Gnathanodon speciosus*) | Catfish (*Pseudoplatystoma tigrinum*)

Figure 4.1: Example of fishes with different pigmentation stripe orientations. The zebrafish has horizontal stripes [9], whereas the Golden Trevally has vertical stripes ([213], image under CC-BY license), and a certain species of catfish has a more complex pattern where the stripes are not strictly aligned (figure reprinted with permission from [17]).

In the classical fixed domain setting, typical patterns arising from DDI in two spatial dimensions take the form of spots, stripes, or labyrinths. We will focus on understanding the effect of a wave of competency on stripe or labyrinthine patterns, since stripe patterns can exhibit greater variation through their directionality and alignment than typically observed in spot patterns. Furthermore, in the natural world, some stripe patterns exhibit certain preferential directions, and stripes may or may not be aligned throughout the domain. To illustrate stripe directionality and alignment, we use pigmentation patterns on fishes as an example. As shown in Fig. 4.1, the stripes on a zebrafish are always aligned head-to-tail, along the antero-posterior axis, whereas some other fish species, such as the banded trevally and certain types of angelfish, have their stripes aligned vertically instead, along the dorso-ventral axis. However, certain catfish species can develop stripe patterns that are not strictly aligned, but nonetheless display a preference for directionality. It is still an open question as to the precise mechanisms driving the alignment of these patterns. For zebrafish, Kondo and Asai [98] suggest that the stripes are the result of a Turing mechanism, however [62, 140, 197] suggest an alternative mechanism based upon cell-cell interactions. Here, we will use Turing's DDI model to explore stripe directionality and alignment in general.

In contrast to the aligned stripes and consistent directionality seen in nature, the typical patterns produced by a DDI model under spatial perturbations to a homogeneous steady state, however, tend to be disordered labyrinthine patterns without clear alignment or directionality (examples can be seen in Fig. 4.7, last column). This raises the question of how, and whether, aligned stripes and consistent directionality might be achieved. In this chapter, we suggest that expansion of the patterning subdomain behind a wave of competency, where different wave speeds select different preferred directions, can effectively control pattern alignment and

directionality.

There have been several other proposed mechanisms for pattern alignment. Shoji et al. [173] showed that anisotropic diffusion can account for directionality, while Hiscock and Megason [74] showed that a gradient in the reaction parameters, or anisotropic growth, can do the same, and Nakamasu et al. [138] were able to produce aligned stripes with a specific initial condition. Page et al. [141] considered parameters that vary across space in more complicated ways, and showed that the DDI model, in this case, is able to produce a larger variety of patterns. Finally, Frohnhöfer et al. [62] suggested prepatterning as another mechanism. We will compare the patterns arising from implementations of these different mechanisms with the one we are proposing in the discussion.

Although the main motivation behind the study of DDI models comes from biology, it is very challenging to experiment directly with proposed DDI systems in this context due to the complexity of living tissues. Therefore, chemical systems have been used to experimentally validate DDI models, since their dynamics are better understood, and more amenable to manipulation. In Konow et al. [100], the authors experimentally observed and simulated the CDIMA (chlorine dioxide-iodine-malonic acid) reaction, which is capable of displaying DDI, on a patterning subdomain bounded by a circular boundary. This boundary is then made to expand radially outwards, serving as an analogue of a wave of competency. In both the chemical experiment and numerical simulations of the corresponding mathematical model, Konow et al. found that the stripes produced behind such a wave of competency tend to be concentric circles aligned parallel to the moving boundary if the expansion speed is fast, and radial stripes aligned perpendicular to the boundary if the expansion speed is slow.

In this chapter, we seek to determine whether this effect is specific to the setting in [100], or if it is generalisable to a wider range of scenarios. We will mainly consider Schnakenberg kinetics [167], a simple model that allows insight into the observed dynamics, as well as CDIMA kinetics. We will show that, for the CDIMA model, we can replicate the behaviours observed in [100]. Considering both models allows us to draw conclusions on whether the observed phenomena are particular to the CDIMA model, or may be observed more generally in models that can exhibit a DDI. By comparing the behaviours of these models in one dimension versus two dimensions, we are also able to provide insight into why variations in the wave speed affect stripe directionality and alignment. We also define a speed for the natural propagation of

patterns, and illustrate its significance in determining patterning outcome. This has, to our knowledge, not been considered in previous studies.

In Section 4.2, we describe the models to be used, and provide details of how the wave of competency and the expanding subdomain are implemented. In Section 4.3, we use a series of numerical simulations to explore the behaviour of the model, and to identify and distinguish the cases where different wave speeds lead to qualitatively distinct patterns. Finally, we summarise our findings in Section 4.4.

4.2 Models and methods

The basic two-morphogen DDI model in two spatial dimensions, with an imposed wave of competency, can be written as the following system of PDEs,

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u + f(u, v, W), \quad (x, y) \in \Omega, t > 0, \tag{4.1a}$$

$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + g(u, v, W), \quad (x, y) \in \Omega, t > 0, \tag{4.1b}$$

$$W = W(x, y, t), \tag{4.1c}$$

$$\frac{\partial u}{\partial n} = \frac{\partial v}{\partial n} = 0, \quad (x, y) \in \partial\Omega,$$

$$u(x, y, 0) = u_0(x, y), \quad v(x, y, 0) = v_0(x, y),$$

where $u(x, y, t)$ and $v(x, y, t)$ are the concentrations of the morphogens U and V, respectively, D_u and D_v are their respective diffusion coefficients, and f and g the reaction functions, which differ for the two models we consider. We use the function $W(x, y, t)$ to encode the effects of a wave of competency. We impose zero-flux boundary conditions, since in most applications the morphogen cannot cross the boundary of the domain. The model is said to display a DDI (also known as a Turing instability) if it possesses a spatially homogeneous steady state that is stable in the absence of diffusion, but can be destabilised in the presence of diffusion by a spatially non-homogeneous perturbation [186]. This condition is checked in Appendix H. A necessary condition for a DDI is that the diffusion coefficients must be different, and, without loss of generality, we take $D_u \ll D_v$.

We model the wave of competency as a piecewise constant function W, such that $W = 0$ on the patterning-competent subdomain $\Omega'(t)$, and $W = W_{\max}$ on the patterning-incompetent subdomain $\Omega \setminus \Omega'(t)$. The wavefront represents the boundary separating the two subdomains. The values for $W_{\max}$ and the other model parameters are chosen to ensure that the system is

capable of undergoing a DDI in $\Omega'(t)$, but not in $\Omega \setminus \Omega'(t)$. The domain and subdomains are illustrated in Fig. 4.2.

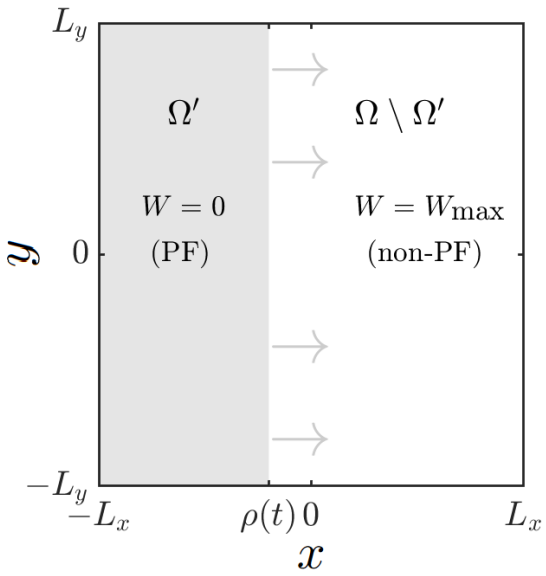


Figure 4.2: Illustration of the effective patterning domain, Ω', which is bounded by the wave of competency at $x = \rho(t)$ that propagates to the right with constant speed α. (PF – pattern forming)

In [100], Ω' was chosen to be a circular region with the radius increasing linearly with time, at constant speed α. Naturally, we expect the system to be attracted to its homogeneous steady state in $\Omega \setminus \Omega'$ and Turing patterns to form inside Ω', with the patterns expanding outwards in synchrony with the expansion of the region Ω'. Both chemical experiments and numerical simulations carried out by [100] observed this expected behaviour. The interesting observation is that the type of pattern that forms depends on the wave speed, α, which determines the rate of expansion of Ω'. For the CDIMA chemical reaction system (see later in this section) on a circular expanding subdomain, the authors mainly observed the formation of concentric ring patterns at higher wave speeds, and stripes perpendicular to the circular front at slower wave speeds, with a mixture of the two behaviours for intermediate values of α. The same behaviours are observed in numerical simulations of a DDI model for this reaction. A reproduction of the numerical simulation results can be found in Fig. 4.12.

The findings of [100] raise several outstanding questions. While the authors mapped out parameter regimes where distinct modes of pattern formation occur, they left open the question of why the wave speed impacts pattern formation in the manners observed, which we aim to answer in this chapter. Furthermore, the authors also did not address whether the observed

phenomenon is particular to the CDIMA model, or extends more broadly to stripe patterns formed under a DDI. We will address this by comparing the patterning outcome under the Schnakenberg model with that of the CDIMA model. We further expand upon existing research by investigating the robustness of patterning behaviour with respect to the shape of the patterning subdomain, and the presence of structural noise.

It is known that a curved domain boundary can impact pattern formation [16, 31, 191]. Intuitively, zero-flux boundary conditions force any stripes near the boundary to be either perpendicular or parallel to it, therefore a curved boundary is likely to result in curved stripes, which will impact pattern formation in the interior. Thus, in order to remove any potential boundary curvature effects, in this work we will simulate the DDI model described in Eq. (4.1) on a rectangular domain. We take

$$\Omega = [-L_x, L_x] \times [-L_y, L_y], \quad \Omega'(t) = [-L_x, \rho(t)] \times [-L_y, L_y], \quad \rho(t) = \min(-L_x + \alpha t, L_x), \tag{4.2}$$

and recall that

$$W(x, y, t) = \begin{cases} 0 & \text{if } (x, y) \in \Omega'(t), \\ W_{\max} & \text{otherwise.} \end{cases}$$

This choice of domain is illustrated in Fig. 4.2. Mathematically, the function W, together with Eq. (4.1), forms a non-autonomous reaction-diffusion system on a fixed domain.

In this chapter, we consider two specific sets of reaction kinetics: those for the Schnakenberg model, and those that describe the CDIMA reaction, in order to explore whether the effect of a wave of competency depends on the form of the reaction kinetics, or the phase of the patterns. For the Schnakenberg model, the profiles for u and v are out-of-phase, whereas for the CDIMA model they are in-phase.

4.2.1 Schnakenberg model

The model we will primarily focus on is the Schnakenberg model, which was proposed by [167] for a hypothetical reaction system, as a special case of the Gierer–Meinhardt model [67]. It consists of very simple kinetics, and is one of the most well-studied DDI models [78]. The non-dimensional reaction terms can be written as

$$f(u, v) = a - u + u^2 v + W, \tag{4.3a}$$

$$g(u, v) = b - u^2 v. \tag{4.3b}$$

Using linear stability analysis (see Appendix H) and simulations on two-dimensional fixed domains as a guide, we take the following as default parameters:

$$D_u = 1, \quad D_v = 20, \quad a = 0.05, \quad b \in \{1.4, 1.6\}, \quad L = 100, \quad W_{\max} = 1. \tag{4.4}$$

In this model, u and v represent the concentrations of some abstract morphogens, and we take W to represent an increase in the production rate of u. The system possesses the following unique homogeneous steady state when W is treated as a constant,

$$u_* = a + b + W, \quad v_* = b/(a + b + W)^2, \tag{4.5}$$

which cannot exhibit a DDI when $W = W_{\max} = 1$, but can display a DDI when $W = 0$.

We consider two values for b, while all other parameters are kept fixed. On a fixed domain without a wave of competency, with all other parameters taking the values given in Eq. (4.4), selecting $b = 1.4$ tends to give rise to patterns that consist of a mixture of spots and stripes on a fixed, square, two-dimensional domain, while selecting $b = 1.6$ mostly gives rise to patterns that consist of stripes (see Fig. 4.7, right-most column). The conditions which favour spots over stripes or *vice versa* have been investigated on a fixed two-dimensional domain in previous studies [53, 137]. Here we are interested to see if imposing a wave of competency can change this behaviour, and whether the wave of competency interacts differently with spots and stripes. The values for these parameters are chosen so that the model exhibits the desired patterning behaviours, which serves to illustrate the effects of a wave of competency. In this chapter, we do not use the Schnakenberg model to represent any concrete biological system, and the parameter values are not inferred from any experiments.

4.2.2 CDIMA model

We also consider the kinetics derived for a model of the CDIMA chemical reaction taken from Konow et al. [100], which is a modified version of the model by Lengyel and Epstein [111]. The non-dimensional form of the model is given by

$$f(u, v) = a - u - \frac{4uv}{1 + u^2} - W, \tag{4.6a}$$

$$g(u, v) = \sigma b \left(u - \frac{uv}{1 + u^2} + W \right), \tag{4.6b}$$

with default parameter values taken from [100]:

$$D_u = 1, \quad D_v = \sigma, \quad a = 12, \quad b = 0.31, \quad d = 1, \quad \sigma = 50, \quad L_x = L_y = 100, \quad W_{\max} = 1.5. \tag{4.7}$$

Physically, u and v represent the concentrations of iodine and chlorine dioxide, two of the chemical species in the CDIMA reaction, respectively. W represents the amount of illumination applied to the reactor. This model assumes that iodine promotes the formation of chlorine dioxide at a constant rate, and both are consumed together in a reaction that is inhibited by an abundance of iodine. Other chemical species present in the CDIMA reaction are not explicitly tracked in the model. When $W < a/5$ is treated as a constant, this system possesses a unique positive homogeneous steady state at

$$u_* = \frac{a - 5W}{5}, \quad v_* = \frac{(u_* + W)(1 + u_*^2)}{u_*}, \tag{4.8}$$

which cannot exhibit a DDI when $W = 0$, but can exhibit DDI when $W = W_{\max} = 1.5$.

This model is able to capture the patterning behaviours of the CDIMA chemical system, qualitatively matching the orientation and alignment of the stripes [100]. Since the illumination can be readily controlled, as opposed to the physical size of the reactor, manipulating the illumination provides an easy way of observing a DDI behind a wave of competency.

4.2.3 Artificial noise

Throughout this chapter, we will add noise to the models throughout their simulation, at each time step. The reasons for this are as follows. Firstly, intrinsic noise is present in almost all biological systems. Therefore, adding noise enables us to better reflect reality. Secondly, there have long been criticisms that in many situations, the DDI model is very sensitive to noise and changes in the initial conditions, and so the patterns it produces are unreliable [14]. By adding noise to model simulations, we can ensure that our observations and conclusions are robust to noise, and applicable to biologically realistic situations. Finally, it is known that for the type of models we are analysing, the numerical scheme chosen for their simulation can have an impact on both the transient behaviour and the final pattern [124], with lower-order methods prone to producing aliasing artifacts [164]. In fact, as we will see in Section 4.3.2, the choice of mesh can also have an impact on the qualitative behaviour. The addition of artificial noise allows us to verify that the observed behaviours are not merely numerical artifacts, since the numerical noise will be subsumed as a part of the overall noise.

In the next section, we will present and compare results first from simulations without, then with, on-going artificial noise. For the simulations with artificial noise (refereed to as "noisy simulations" henceforth), we add noise to u at each grid point and each time step, independently and identically distributed as $N(0, \mu^2)$, with the standard deviation of the noise $\mu = 0.01$. This

magnitude of noise is sufficiently strong to break symmetries and lead to different behaviours compared to the noiseless case, and sufficiently weak so that it does not overwhelm the system, i.e., solutions known to be stable on a fixed domain persist in the presence of this noise. See Sec. 4.3.5 for discussion on the impact of noise on patterning behaviours. The random numbers are generated with Matlab's *normrnd* function. All solutions presented are produced with the Mersenne Twister algorithm [128] with seed set to 0, which is the default setting.

4.3 Numerical simulations

In this section, we carry out numerical simulations of the models outlined in Section 4.2. We use a standard finite difference scheme for the spatial discretisation, placing equally spaced mesh points at $x_n = -L_x + n\Delta x$, with $\Delta x = 0.5$, and similarly for the y direction in two dimensions. For the time stepping, we use an implicit-explicit scheme (IMEX) [7, 38, 164], where we implement the diffusion term using the implicit Crank–Nicolson method, and use the explicit Euler method for the nonlinear reaction term, with $\Delta t = 0.01$. This scheme is the same as the one used for PDE simulations in Chapter 2, and details are given in Appendix A. The Crank–Nicolson scheme has been shown to be stable even in the presence of noise [88, 214]. We have verified that all described behaviour still occurs for a much finer time-stepping with $\Delta t = 0.00125$ or a finer grid size with $\Delta x = 0.25$, and that the numerical scheme indeed converges as Δt is decreased, as shown in Appendix A.1.1. Model simulations are implemented in Matlab, with the code, as well as the animations corresponding to the solutions, provided in the supplementary materials and on Github at `github.com/liuyue002/turing_expanding_domain`.

In Section 4.3.1, we begin by simulating the models on a one-dimensional domain to identify and understand the basic behaviours of the models and guide our choice for the range of wave speeds. Then in Section 4.3.2, we move to two-dimensional domains in order to explore the full range of possibilities for pattern behaviours. Finally, in Section 4.3.3, we use a very narrow two-dimensional domain to closely examine the structure of the stripes in the patterns observed in the earlier simulations. The dynamics on this narrow two-dimensional domain are effectively one-dimensional, which allows us to explain the differences between the distinct patterns observed under varying speeds of propagation of the wave of competency in two dimensions, by relating them to behaviours in one dimension. Besides the simulations shown in this chapter, we have also attempted to analyse the models using bifurcation analysis and several other techniques.

These analyses ultimately did not bring significant additional insight into the behaviour of the models, hence their presentation is omitted.

4.3.1 Simulations in one spatial dimension

Although the focus of this study is on two-dimensional pattern structures, we begin in one dimension, where the pattern and the transient behaviours are simpler to visualise and understand. We simulate the PDE system (4.1) on the one-dimensional domain $\Omega = [-L, L]$. Analogous to the two-dimensional case, the patterning subdomain is defined as $\Omega'(t) = [-L, \rho(t)]$, where $\rho(t)$ is given in Eq. (4.2). For comparison, we also run simulations in the classical (fixed domain) setup without an expanding patterning subdomain, so that $\Omega' = \Omega$ and $W \equiv 0$.

For the noiseless case, we initialise the system at its homogeneous steady state $u = u_*$, $v = v_*$ (Eq. (4.5)), with a perturbation to u at the left-hand boundary, created by setting $u(-L, 0) = 2u_*$. For simulations with an expanding subdomain, we found that for the range of the wave speed α we tested, this initial condition produces the same behaviour as for the simulations when a noisy perturbation is added to the initial condition across the whole domain Ω. This is because, at the beginning of the simulation, most of the domain is not capable of supporting a pattern, so the noise far from the left-hand boundary simply decays and has no further impact. The simulation is run up to 100 time units after the patterning subdomain, Ω', has expanded to cover the entire domain, Ω, which we found, by visual inspection, is sufficient time for the pattern to evolve to its steady state in both models.

In one spatial dimension, in the regime where $D_u/D_v \ll 1$, the only possible patterns in both the Schnakenberg and CDIMA models are multi-spike solutions, which consist of a series of N equally-spaced, tall, localised peaks, known as spikes (Fig. 4.3). The stability properties of multi-spike solutions for the Schnakenberg model have been well-studied with asymptotic analysis [78]. Notably, for a given domain length, there is a range of integers N for which the N–spike solutions are stable.

This means that the final pattern, regardless of the wave speed, α, will be a multi-spike solution. Indeed, we found that in one dimension, with expanding patterning subdomains, the final patterns for different wave speeds, α, are all multi-spike patterns, which are qualitatively similar to the pattern shown in Fig. 4.3, except with possibly different numbers of spikes. Despite the fact that the form of pattern does not depend on α, we will show that the transient behaviour changes as α varies. This transient behaviour will be relevant when we relate the one-dimensional

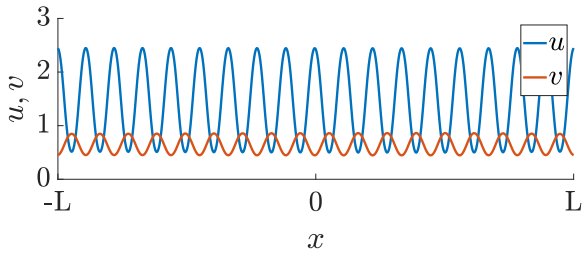


Figure 4.3: A typical multi-spike steady state solution of the Schnakenberg model (Eq. (4.3)) in one dimension. Notice that u and v are exactly out-of-phase, so we will only consider the profile of u henceforth. This simulation is produced with parameters from Eq. (4.4) with $b = 1.4$ on a fixed domain (i.e., $\Omega' = \Omega$, $W \equiv 0$) without on-going noise. The development of the system over time leading up to this pattern is illustrated in Fig. 4.4(d).

behaviours to two-dimensional behaviours in Section 4.3.3, and it is also interesting from a biological perspective since the transient distribution of the morphogens may have consequences on developmental processes. We will first discuss the transient behaviours for the Schnakenberg model, shown in Fig. 4.4, and then note the differences between these and those exhibited by the CDIMA model. In one dimension, the behaviour of the Schnakenberg model is qualitatively very similar when $b = 1.4$ and $b = 1.6$, therefore for brevity, we only present the results for $b = 1.4$.

First, let us consider the behaviour of the Schnakenberg model on a fixed domain without a wave of competency, that is, with $\Omega' = \Omega$ (Fig. 4.4(d)), to provide a comparison to later simulations. This is simply a standard, classical reaction-diffusion model exhibiting a DDI. Here the spikes naturally form in sequence, one after another, at a steady rate, hence the pattern propagates away from the initial perturbation at a finite speed. We thus refer to this speed, ν, as the speed of natural pattern propagation as it measures the speed at which an initial perturbation to a homogeneous steady state spreads. This is an intrinsic property of the PDE system and its parameter values, independent of the wave of competency which will be imposed later. We estimate ν from the fixed domain simulation as follows. We use τ, the first time at which $|u(L,\tau) - u_*| > 0.5$, as a measure of the amount of time it takes for the pattern to cover the entire domain. For the Schnakenberg model, we find $\tau = 160.93$, and with a domain length of $2L = 200$, we thus estimate $\nu = 2L/\tau \approx 1.2428$. This steady speed of spike propagation is visualised by the green dashed line in Fig. 4.4(d).

Next, we implement a wave of competency advancing at a constant speed α, located at $\rho(t) = -L + \alpha t$. We will consider a range of values of α relative to the natural pattern propagation speed, ν. First we consider the case where the wave of competency moves more quickly than the

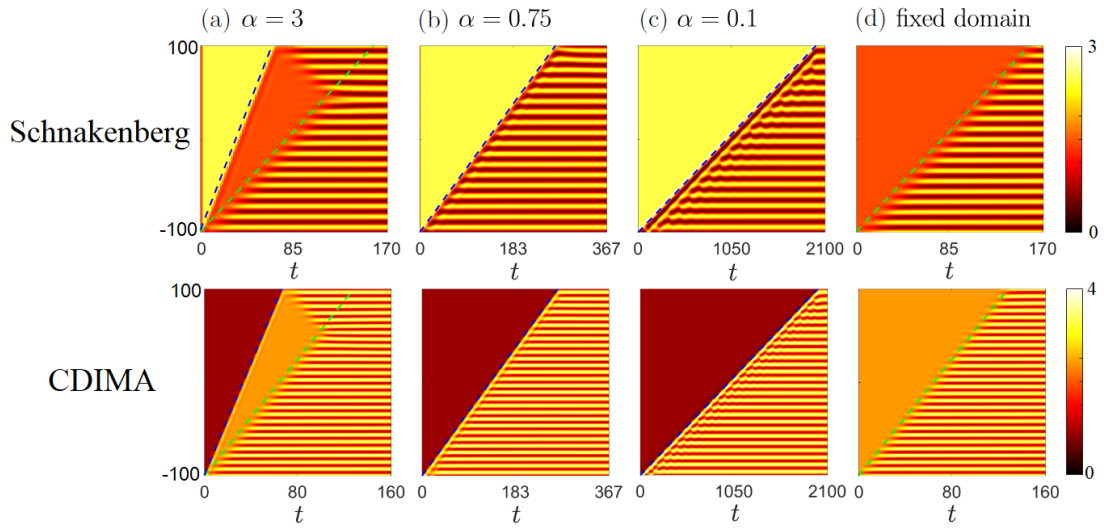


Figure 4.4: Kymograph representation of u in the solution of the Schnakenberg model (Eq. (4.3)) and the CDIMA model (Eq. (4.6)) in one spatial dimension, with parameters from Eq. (4.4) and $b = 1.4$, in the absence of on-going noise. Each yellow horizontal stripe corresponds to a spike, the red stripes correspond to the valleys between them, and regions of solid colours represent places where the system is locally spatially homogeneous. The system is initialised at the homogeneous steady state with a fixed perturbation at the left-hand boundary, as detailed in the text. Notice that there is a clear divide between the patterning subdomain $\Omega'(t)$ on the lower right (orange with yellow/red stripes), and the non-patterning part of the domain $\Omega \setminus \Omega'(t)$ on the upper left (solid yellow for Schnakenberg, dark red for CDIMA), with the dark blue dashed line representing the wave of competency at $x = \rho(t) = -L + \alpha t$ separating them. The green dashed lines in (a,d) are $x = -L + \nu t$, where the natural speed of pattern propagation is $\nu \approx 1.2428$ for the Schnakenberg model, and $\nu \approx 1.5564$ for the CDIMA model, see text for details. In all four cases, the final pattern is a multi-spike pattern similar to that in Fig. 4.3.

natural speed of propagation, $\alpha > \nu$, as in Fig. 4.4(a). In this case, the subdomain Ω' expands so rapidly that spike formation, which propagates at speed ν, lags behind the wave of competency, leaving a part of Ω' (which has width $(\alpha - \nu)t$) just behind the moving boundary momentarily devoid of pattern. This region corresponds to the small region of solid orange in the plot that sits in between the green and blue dashed lines. In this case, we see that the moving boundary has no meaningful impact upon the formation of the pattern. Consequently, we observe that the rate and manner of spike formation, represented by the time of emergence and the shape of the stripes in Fig. 4.4(a), are nearly identical to what we observe in Fig. 4.4(a), except near the right-hand boundary ($x = 100$). Here we see that when the front reaches the boundary at $t = 100$, the higher level of u in the vanishing region $\Omega \setminus \Omega'$ acts as a perturbation to the homogeneous steady state, triggering the formation of a sequence of spikes from right to left.

Next, we consider a moderately high wave speed, $0 < \alpha \lesssim \nu$ (denoting a value of α that is below, but close to, ν), as in Fig. 4.4(b). In this case, the pattern propagates at the same speed as the wave speed α, since pattern formation is restricted to behind the moving front. Here new

spikes form via insertion between the moving front and the previous leading spike, and remain mostly static after their formation. In contrast, with an even slower wave speed, $\alpha \ll \nu$, as in Fig. 4.4(c), there is a leading spike that travels at the same speed, α, as the moving front, and new spikes form behind it by splitting off the leading spike. After the split, the leading spike continues to move in tandem with the moving boundary. For an intermediate range of wave speeds, $0.2 < \alpha < 0.3$ for both models, we observe a mixture of behaviours in the Schnakenberg model, where both peak insertion and splitting occurs. We therefore define $\alpha_c \approx 0.25$ to be a threshold that approximately separates the two cases for the Schnakenberg model.

We denote these observed behaviour cases by A, B, and C, and summarise them in Table 4.1, including the fixed domain case (case (D)) for comparison. In all of these cases, the final pattern is the same, and we observe sequential formation of spikes from left to right (except near the right-hand boundary, where potential boundary effects may disturb the pattern), either at speed ν or α, whichever is slower. However, there are important differences in the transient behaviours. In Fig. 4.4(a), we have a part of the domain that momentarily remains at the homogeneous steady state after it has become capable of supporting a pattern, whereas in Fig. 4.4(b,c) the pattern always fills the entirety of Ω'.

Cases	Range of α	Behaviour in 1D simulations	Behaviour in noisy 2D simulations
A	$\alpha > \nu$	Peaks propagate at speed ν (noiseless, Fig. 4.4(a)) Disordered peak formation (noisy, Fig. 4.5(a))	Labyrinthine patterns (Fig. 4.7, 4.8, first column)
B	$0 < \alpha_c \ll \alpha < \nu$	Peak insertion (Fig. 4.4(b), 4.5(b))	Stripes parallel to moving front (Fig. 4.7, 4.8, second column)
C	$0 < \alpha \ll \alpha_c$	Peak splitting (Fig. 4.4(c), 4.5(c))	Stripes transverse to moving front (Fig. 4.7, 4.8, fourth column)
D	$\alpha = 0$ (fixed domain)	Peaks propagate at speed ν (noiseless, Fig. 4.4(d)) Disordered peak formation (noisy, Fig. 4.5(d))	Labyrinthine patterns (Fig. 4.7, 4.8, fifth column)

Table 4.1: Summary of the four cases identified for the Schnakenberg model in Section 4.3.1, and their behaviours observed in simulations on one-dimensional domains (without on-going noise (Fig. 4.4), and with on-going noise (Fig. 4.5)), and two-dimensional domains, with on-going noise (Fig. 4.7, 4.8). Two critical thresholds separate the cases: ν, the natural speed of pattern propagation defined in Section 4.3.1, and α_c, which was estimated using a parameter sweep.

The results for the CDIMA model are largely similar to the observations described above for the Schnakenberg model. We were able to estimate $\nu \approx 1.5564$ for the CDIMA model for the parameter values in Eq. (4.7), which separates case A from cases B and C in a similar way. The

fact that the value of ν for the CDIMA model is close to that of the Schnakenberg model is a consequence of the choice for the model parameters, which was made so that patterns form and propagate at similar rates under both models. One key difference is that in both cases B and C, the CDIMA model exhibits peak splitting behaviour similar to case C of the Schnakenberg model. It does not appear to exhibit peak insertion behaviours for any of the tested values of α within the relevant range. This means that we can only observe one, instead of two, distinct behaviours for the range $0 < \alpha < \nu$ in the CDIMA model.

Next, we consider the impact of on-going noise, as discussed in Section 4.2. In case A, as well as the fixed domain case D, we would expect this noise to trigger the formation of new spikes in the region of Ω' where the system would remain close to the homogeneous steady state in the absence of noise, disrupting the orderly sequential formation of spikes. On the other hand, in cases B and C, we do not expect this additional noise to have a major effect. This is because the pattern always fills the entirety of Ω', so the noise decays in Ω' because the multi-spike solution is stable, and it also decays in $\Omega \setminus \Omega'$ due to the spatially uniform steady state being stable in that subdomain.

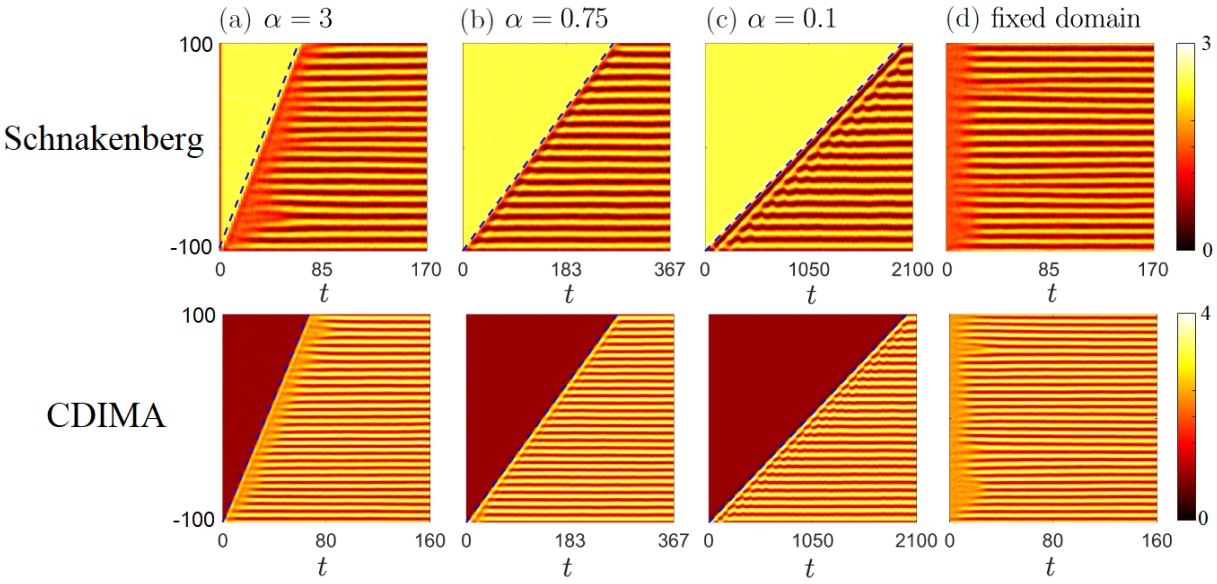


Figure 4.5: Solution of the Schnakenberg and CDIMA models, with the same parameters, wave speeds and initial conditions as in Fig. 4.4, except with on-going noise. The blue dashed lines indicate the boundary of $\Omega'(t)$. Observe that in (a) and (d), the spikes no longer form in an orderly sequence, whereas (b,c) are virtually identical to the corresponding cases in Fig. 4.4.

The numerical results from our simulations confirm our expectations. In Fig. 4.5, we observe that, indeed, the sequence of spike formations is disrupted in case A (Fig. 4.5(a)). Once again, the behaviour in this case closely resembles the behaviour on a fixed domain (case D, Fig. 4.5(d)),

where the pattern fills the patterning subdomain Ω', which coincides with entire domain Ω in the latter case. In contrast, the behaviours in cases B and C (Fig. 4.5(b,c)) remain virtually unchanged compared to the noiseless case in Fig. 4.4, as predicted. In the presence of on-going noise, spike formation is initiated everywhere in the patterning-competent subdomain Ω' simultaneously, instead of spreading outwards from a single point at speed ν. Nonetheless, ν remains the separator of the distinct behaviours in cases A and B, since the size of α relative to ν determines whether the pattern always fills Ω'.

The same behaviours are observed for the CDIMA model in the noisy case, except for the difference noted before, where peak splitting is observed instead of peak insertion in case B.

From the numerical simulations in one dimension, we are able to draw the following conclusions for the Schnakenberg model. First, if the wave speed, α, which determines the rate of expansion of the patterning subdomain, is faster than the natural speed of pattern propagation, ν (case A), then there will be little interaction between the moving front and the pattern forming mechanism, in which case we expect similar behaviour to the case where the whole domain can undergo a DDI. However, for $\alpha < \nu$ we can have very different behaviours. A high wave speed, α, leads to the formation of new spikes via the insertion of peaks behind the moving front (case B), whereas for a sufficiently low wave speed, α, the spikes form via the splitting of a leading, moving spike (case C). Moving forward into two dimensions, these observations suggest that in the presence of on-going noise, case A will lead to disorganised patterns, matching the behaviour on a fixed domain, whereas cases B and C may lead to the formation of more organised, but possibly distinct, patterns, following the wave of competency. These predictions apply to the CDIMA model as well, except we might expect fewer differences in behaviour between cases B and C.

4.3.2 Simulations in two spatial dimensions

With the knowledge gained from the four cases observed in one dimension, we now investigate whether we observe the same distinct behaviours in two dimensions, and if the cases in one and two dimensions have any correspondence between them. We run all simulations long enough for Ω' to expand to cover the entire domain Ω, plus a further 200 time units. By numerical experimentation, we found that this is sufficient time for the system to reach a steady state. For comparison, we also run simulations where we set $\Omega' = \Omega$ (that is, the whole domain is capable of undergoing DDI immediately), in which case we run the simulations until $t = 200$. We present

only the final profile for u in the two-dimensional simulations, since the profile for v is either in- or out-of-phase with respect to u, depending on the kinetic terms, so the alignment and orientation of the stripes will be the same for both morphogens. We will first briefly mention the observations in the noiseless case, then investigate how the results change with the addition of on-going noise.

For both the Schnakenberg and CDIMA models, we use the domain described in Section 4.2, with the parameters from Eqs. (4.4) and (4.7), and we initialise the system by perturbing u on the left-hand boundary by setting $u(x = -L, y) = 2u_*$. This choice is consistent with the initial condition used in the one-dimensional simulations, which can be recovered by taking any horizontal cross-section with y constant. By observing the simulation on a fixed domain (Fig. 4.6, see supplementary materials for the corresponding animation), we found the same value of ν as we did in one dimension. Without on-going noise, we found that for the range of wave speeds $0.1 \leq \alpha \leq 2$, the system always displays vertical stripes parallel to the moving boundary for both models (Fig. 4.6). Even if we consider a larger perturbation to the initial condition, e.g., adding uniformly distributed noise in $(-0.3, 0.3)$ to each point in the domain, the behaviour remains the same. The reason for this result is that since initially Ω', the subdomain capable of supporting Turing patterns, is very small, the noise in the initial condition decays, similar to what we have found in Section 4.3.1, leaving the system essentially at its homogeneous steady state in $\Omega \setminus \Omega'$. This, combined with the regular square mesh employed in the numerical simulations, means that there is no symmetry breaking mechanism in the $y-$direction. Consequently, the patterns we obtain are simply the multi-spike pattern in the $x-$direction, but constant in the $y-$direction. This reduces the system back to the one-dimensional case. While we do observe differences in the transient behaviour as α changes, analogous to the observations in Section 4.3.1, the final pattern is always the same.

We also attempted to recover the results of [100] for the CDIMA model, which uses a circular domain that grows radially. The simulations without on-going noise were only able to produce circular rings parallel to the growing boundary, and not the other patterns observed in [100]. Once again, this is an artifact of the regularity of the mesh used; [100] used a finite element method with an irregular mesh, which naturally acts as a symmetry-breaking mechanism. However, simulating the model with on-going noise, we can reproduce all patterns from [100] (see Sec. 4.3.4).

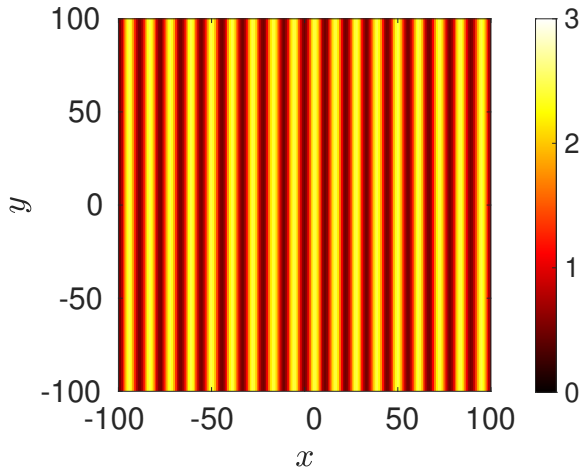


Figure 4.6: The final pattern produced with the Schnakenberg model on a two-dimensional domain with parameters from Eq. (4.4) and $b = 1.4$, $\alpha = 0.75$, initialised by perturbing the homogeneous steady state at each mesh point, without any on-going noise. The resulting pattern is a series of perfectly vertical stripes. Similar results were obtained for other values of α in the range $[0.1, 2]$, as well as on a fixed domain, and for the CDIMA model.

For the simulations with on-going noise, we first simulate the system of PDEs (4.1) with both sets of kinetics, initialised at the homogeneous steady state. We use the same initial condition as before for consistency, where we initialise at the homogeneous steady state with a perturbation to u at the left-hand boundary. However, we note that the specific form of perturbation makes little difference in this case, and we recover the same final patterns if a different perturbation to the homogeneous steady state is used in the initial condition, e.g., by adding a uniformly distributed noise at each mesh point in the domain. As explained before, we expect the initial condition to have little influence on the final pattern since it almost entirely decays away before the onset of pattern formation. The results are shown in Fig. 4.7, where the rows correspond to the different kinetics, and the columns are arranged in the order of decreasing wave speed, α. In the first column, which is the case when $\alpha > \nu$, corresponding to case A in Table 4.1, we can observe that the pattern initially takes the form of vertical stripes, but quickly transitions to become disordered and labyrinthine, matching the behaviour on a fixed domain (case D), as shown in the fifth column. This is reminiscent of the one-dimensional behaviour for the same case in Fig. 4.5(A), where the sequential formation of spikes is disrupted and the pattern becomes disordered, matching the behaviour in Fig. 4.5(D).

In the second column of Fig. 4.7, we have $0 < \alpha \lesssim \nu$, corresponding to case B in Table 4.1. With a moderately high α, for both the CDIMA and Schnakenberg models, multiple repeated simulations with different random seeds produced only stripes aligned vertically, parallel to

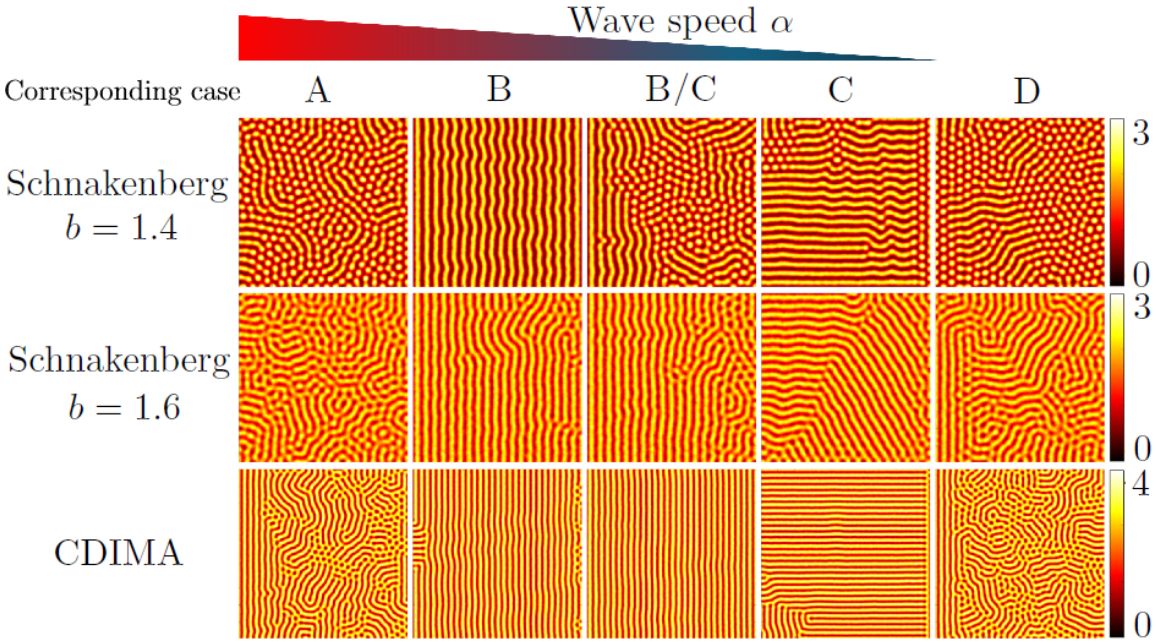


Figure 4.7: Examples of the steady state patterns produced by the Schnakenberg model (4.3), and the CDIMA model (4.6), with the choice of domain as given in Eq. (4.2), and the wave speed decreasing from left to right. The last column shows the fixed domain case. For both models, $\alpha = 3, 0.75, 0.5, 0.2$. Notice that in the first column, corresponding to case A in Table 4.1, the pattern begins as vertically aligned stripes on the left side, but quickly transitions to a labyrinthine pattern as pattern formation lags behind the wave of competency. The second column corresponds to case B, and the patterns mostly take the form of vertical stripes. The third column, corresponding to an intermediate regime between cases B and C, exhibits a mixture of the behaviours from the second and fourth columns, while the fourth column (corresponding to case C) contains regions where the stripes are slanted or horizontal. Finally, the last column shows the labyrinthine patterns produced for the case where $\Omega' = \Omega$.

the moving boundary of the patterning subdomain representing the wave of competency. This strongly suggests that vertical stripes are the only pattern possible in this case.

For sufficiently small values of α, as in the fourth column, corresponding to case C in Table 4.1, the stripes may be aligned horizontally, or at a slanted angle (most clearly visible in the Schnakenberg $b = 1.4$ pattern). The CDIMA model tends to form a few, large regions where the stripes are in alignment, with two main regions, one with vertical stripes, and the other with horizontal stripes. This can be seen in Fig. 4.7, fourth column, third row, where a small region in the bottom-left corner exhibits vertical stripes, and horizontal stripes cover the rest of the domain. In comparison, the Schnakenberg model tends to exhibit a greater number of such regions. For both models, the final pattern is highly dependent on the noise. The number and location of these regions containing aligned stripes, as well as the direction of stripes within them, can be completely different when a different random seed is used. The third column appears to exhibit a mixture of its two neighbouring columns. The Schnakenberg model produced

some slanted stripes, but not as many as in the fourth column.

We also observe in Fig. 4.7 that the Schnakenberg model with $b = 1.4$ is much more prone to producing spots interspersed between stripes than the same model with $b = 1.6$, or the CDIMA model. However, with an expanding patterning subdomain, the model tends to produce more aligned stripes compared to a fixed domain.

These results suggest that the vertical stripe pattern is the only stable pattern in case B (moderately high α), while in case C (low α), we observe multi-stability where stripes aligned at a variety of angles are possible, including horizontal stripes. This leads us to ask whether the horizontal stripes are stable at low α, and if so, how high does α need to be for the horizontal stripes to become unstable?

To answer this question, we simulate the same models, but using initial conditions chosen as horizontal stripes in the form of cosines, with the amplitude and period set to the estimated values from the final steady state of the previous simulations. Specifically, we set

$$u(x, y, 0) = m_u + A_u \cos(qy), \quad v(x, y, 0) = m_v + A_v \cos(qy), \tag{4.9}$$

where m_u, A_u, m_v, A_v and q are estimated to match the patterns in the second column of Fig. 4.7 by measuring the average wavelength and amplitude of the final patterns. For the CDIMA model, we used

$$m_u = 2.41, \quad A_u = 1.16, \quad m_v = 6.67, \quad A_v = 0.71, \quad q = 0.91.$$

For the Schnakenberg model, we used

$$m_u = 1.45, \quad A_u = 0.98, \quad m_v = 0.65, \quad A_v = -0.20, \quad q = 0.57.$$

This set of initial conditions, unsurprisingly, leads to the same pattern as before, for the aforementioned reason that, with our choice of $\rho(t)$, the initial condition decays to the homogeneous steady state before pattern formation can take place. Therefore, instead of the form of $\rho(t)$ as defined in Eq. (4.2), we take

$$\rho(t) = \min(-L_x + L_0 + \alpha \max(t - t_0, 0), L_x), \quad L_0 = 0.2L_x, \quad t_0 = 50,$$

meaning that the wave of competency remains at L_0 up to time t_0, allowing a pattern to develop inside Ω' before the wave begins to advance at constant speed α.

The results are shown in Fig. 4.8. For $\alpha > \nu$ (first column, corresponding to case A), we observe the same disordered labyrinthine patterns as we did in Fig. 4.7, since the initial stripe

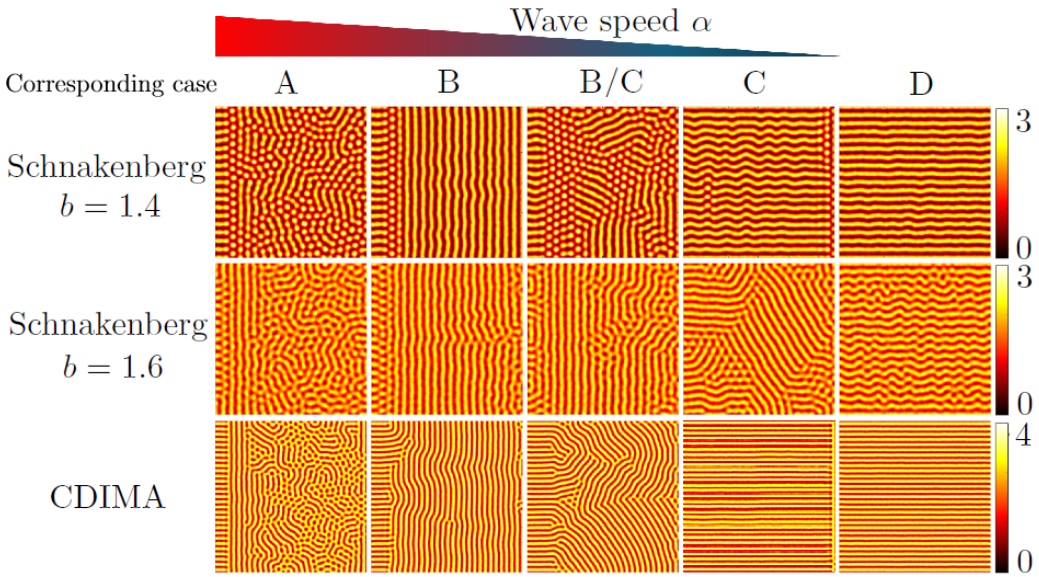


Figure 4.8: Examples of the steady state patterns produced by the Schnakenberg model (4.3), and the CDIMA model (4.6), where the wave speed decreases from left to right. The parameter values and wave speeds are the same as in Fig. 4.7, except the initial conditions are set to horizontal stripes, as described in the text. In the first column, for high $\alpha > \nu$, we observe disordered labyrinthine patterns, as before. In the second column, at moderately high α, the pattern abruptly switches to vertical stripes for both models, once again suggesting that this is the only stable pattern at this wave speed. In the third column, a medium value of α leads to a mixture of behaviours. In the CDIMA model, the stripes curve smoothly to become more vertical, while the Schnakenberg model with $b = 1.4$ forms an abundance of spots, which separates regions of aligned stripes. However, for $b = 1.6$, a similar pattern without spots is produced by the Schnakenberg model. In the fourth column, where α is low, we see that horizontal stripes persist for both models, although in the Schnakenberg model, the stripes may bend away at a slanted angle. Finally, in the fifth column, unsurprisingly, we observe that the horizontal stripes persist in the fixed domain case.

pattern in $\Omega \setminus \Omega'$ quickly decays to the homogeneous steady state, essentially recreating the same initial conditions as in the earlier simulations. For $0 < \alpha \lesssim \nu$ (second column, case B), both models produce vertical stripes, and the transition from horizontal stripes to vertical stripes is abrupt. Some spots appear near the transition, but in both the CDIMA model and the Schnakenberg model with $b = 1.6$ the spots are eventually absorbed into new stripes.

For $\alpha \ll \nu$ (fourth column, case C), the horizontal pattern persists for the CDIMA model and for the Schnakenberg model with $b = 1.4$. For the Schnakenberg model with $b = 1.6$, the stripes bend away in some parts of the domain and align at an angle, but nonetheless, horizontal stripes dominate. Whether or not this bending occurs is noise-dependent and differs between realisations. For medium α (third column), in both the CDIMA model and the Schnakenberg model with $b = 1.6$, the stripes curve away smoothly from horizontal to become more vertical. For the Schnakenberg model with $b = 1.4$, the curved stripes are replaced with spots.

For the fixed domain case (fifth column, case D), the on-going noise (recall that the noise has standard deviation $\mu = 0.01$) was unable to destabilise the initial stripe pattern. This is the only scenario where we obtain different outcomes in cases A and D. However, for the fixed domain case, the pattern can be destroyed if the noise strength is increased sufficiently, see Sec. 4.3.5 for details. Note that in both Fig. 4.7 and Fig. 4.8, some of the stripes are not perfectly straight, but rather are a little curved. This is a consequence of the noise added at each time step.

We also note that with an extremely low wave speed α below the value used in case C, the timescale for pattern formation is much shorter than the timescale for the expansion of the competent subdomain Ω', so the dynamics inside Ω' can be seen as quasi-static. Therefore, the effect of the wave of competency decouples from the pattern formation process. Over the long time it takes for the wave of competency to traverse the full domain, the pattern will eventually be destabilised by the on-going noise and become disordered. This effect is more easily seen for the Schnakenberg model with $b = 1.4$ since, compared to the CDIMA model or the Schnakenberg model with $b = 1.6$, it has a greater tendency to exhibit stripes that break up into spots, and therefore the stripe pattern is more easily destabilised by the on-going noise. An example is given in Fig. 4.9 where, for a very low value of α, the initial horizontal stripes do not persist very far beyond the left-hand boundary, and the final pattern consists of a mixture of disordered spots and stripes, similar to the fixed domain case with uniform initial conditions (Fig. 4.7, final column). Since we are mostly interested in cases where the two timescales are comparable, we will not consider this case further.

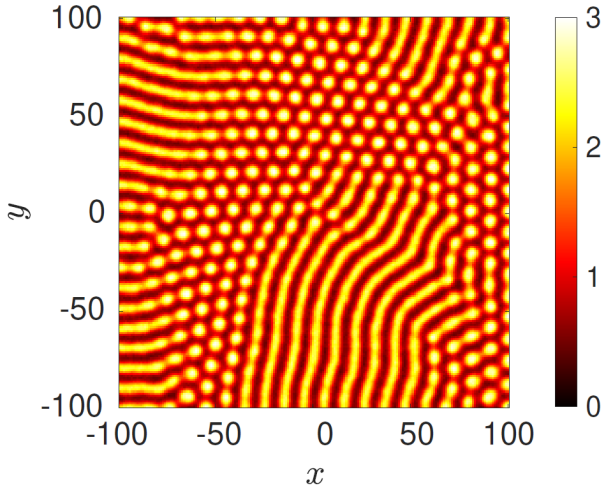


Figure 4.9: Final pattern for the Schnakenberg model with $b = 1.4$ and $\alpha = 0.05$, with initial conditions set to horizontal stripes, as in Fig. 4.8. Note that the pattern is mostly disordered.

These simulations show a clear correspondence with the cases in one dimension. First, for case A ($\alpha < \nu$; first column in Figs. 4.7 and 4.8), for both sets of initial conditions, we obtain labyrinthine patterns similar to case D with homogeneous initial conditions (Fig. 4.7, final column). This is analogous to the results observed in one dimension, where similar transient behaviours were observed in cases A and D. For case B ($0 < \alpha \lesssim \nu$), our simulations suggest that the only stable pattern is vertical stripes parallel to the moving front, whereas for case C ($\alpha \ll \nu$), patterns that are transverse to the moving front, including horizontal and slanted stripes, appear to be stable.

Moreover, for the Schnakenberg model, we found that the switch between cases B and C happens at approximately the same critical value of α in both one and two dimensions, suggesting that the distinction between peak insertion and peak splitting in one dimension may correspond to the distinction between vertical stripes and transversal stripes in two dimensions. For the CDIMA model, despite our observations that it exhibits the same qualitative behaviour in cases B and C in one dimension, we find it exhibits different outcomes in two dimensions that largely agree with the behaviour of the Schnakenberg model, where the preferred directionality of the stripes changes. While this is a surprising contrast to the one-dimensional behaviours, it is consistent with the results in [100], where distinct patterns analogous to those we have observed at different values of α were also observed.

While the critical threshold between cases A and B appears to be at $\alpha = \nu$, the threshold between cases B and C is much less clear. We will explore how these two cases can be distinguished, at least for the Schnakenberg model, in the next section.

4.3.3 Single stripes on a narrow two-dimensional domain

The simulations so far on one- and two-dimensional domains have allowed us to identify three distinct patterning regimes under the effect of a wave of competency, denoted by cases A, B, and C. While a promising candidate for the critical value of α separating cases A and B has been found, we still need to delineate cases B and C.

By examining animations of the simulations in two dimensions, we observe that during the formation of stripes transversal to the moving front (including horizontal and slanted stripes), there is usually a spot-like structure trailing right behind the front, with the rest of the stripe extending behind it. This structure can be seen in the snapshot presented in Fig. 4.10. Since this behaviour is observed for horizontal stripes and slanted stripes, we will classify both of these

orientations together as the same type of pattern henceforth. In contrast, a vertical stripe forms spontaneously when the space between existing stripes and the front becomes sufficiently large. This behaviour is common to both models. It turns out that, for the Schnakenberg model, the key to distinguishing between cases B and C is to look closely at the structure of the stripe as it lengthens.

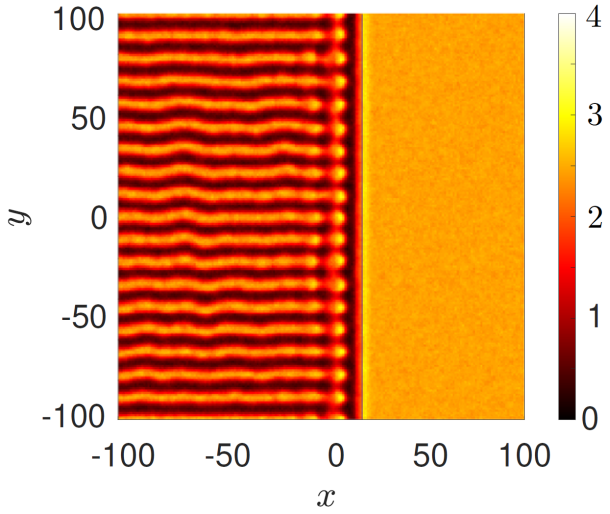


Figure 4.10: Snapshot of the simulation corresponding to the pattern presented in Fig. 4.8, for the Schnakenberg model with $b = 1.4, \alpha = 0.2$, taken at $t = 550$. Here the spot structure at the head of each stripe is clearly visible.

To do so, we modify the subdomain capable of undergoing a DDI to be a narrow rectangle that elongates over time. That is, we now consider

$$\Omega' = [-L_x, \rho(t)] \times [-0.1L_y, 0.1L_y], \tag{4.10}$$

where $\rho(t)$ is as before. We will focus on the Schnakenberg model with $b = 1.4$, then describe how the behaviour of the CDIMA model differs. The narrow width of the patterning subdomain means that no more than one horizontal stripe can be supported. We also found that the stripe does not break up into a row of spots for either model, and the final pattern is a single horizontal stripe for all α in the range $[0.1, 0.75]$. Typical snapshots of the pattern can be found in Fig. 4.11. We will focus on the transient behaviour during the process of stripe formation, just as we have done in one dimension.

In this case, on-going noise is not necessary to capture the interesting aspects of the behaviours. In fact, we obtain the same qualitative results regardless of the presence of the on-going noise. This is not surprising since, as discussed for the one-dimensional case (Section 4.3.1), with $\alpha < \nu$ the pattern always fills Ω' in the noiseless simulations. Without any freedom to extend

the pattern in the y–direction, the noise has little effect on the pattern. Therefore, we will only present the results from simulations without noise.

We illustrate the transient behaviour with a kymograph of u along the horizontal cross-section at $y = 0$, along the ridge of the single stripe. The results are presented in Fig. 4.11. We have also included a snapshot of the stripe during its formation for visualisation.

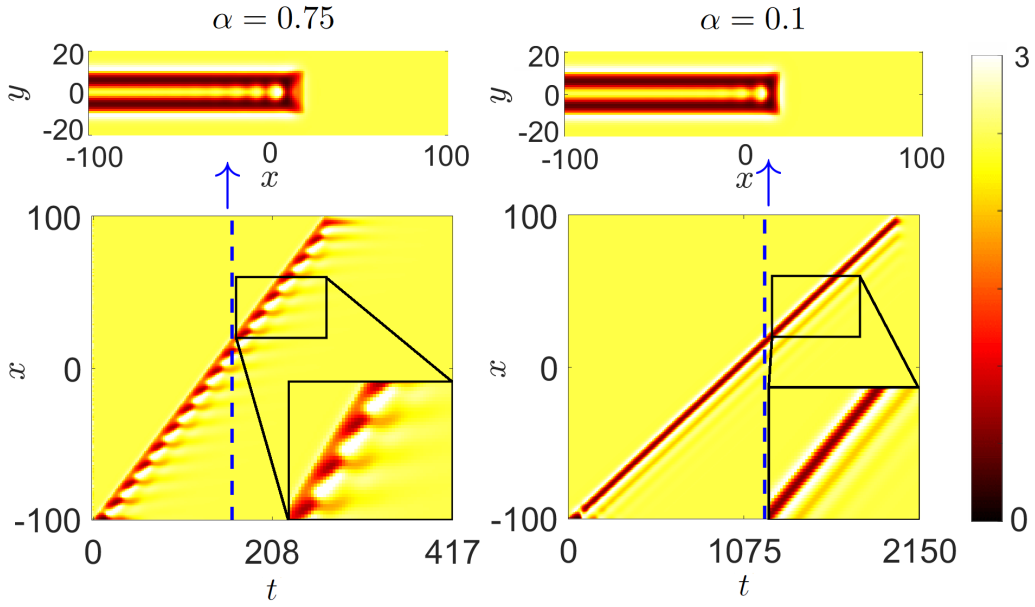


Figure 4.11: Simulations of the Schnakenberg model (Eq. (4.3)) with $b = 1.4$ on a narrow patterning subdomain as in Eq. (4.10), represented with kymographs of u along the cross-section $y = 0$, and a snapshot of the two-dimensional stripe during its formation. The vertical blue line indicates where the snapshot was taken. For $\alpha = 0.75$, which corresponds to case B, we can observe the formation of a series of spots via insertion between the moving boundary and the trailing stripe. They persist for only a short time before being absorbed into the stripe. In contrast, for $\alpha = 0.1$, which corresponds to case C, we observe a moving spot travelling at the same speed as the moving boundary, forming a travelling wave solution along the cross-section $y = 0$.

For sufficiently slow wave speeds, $\alpha \ll \alpha_c$, where α_c represents a critical threshold, there is a single leading spot right next to the moving boundary, as observed in the earlier simulations in two dimensions. This spot moves in tandem with the front, with the rest of the stripes trailing behind it, remaining at a constant distance away. This can be seen in Fig. 4.11 (right), where the two white diagonal lines, representing the location of the front and the leading spot, are parallel to each other. The solution along the horizontal cross-section at the middle of the domain, $y = 0$, is effectively a travelling wave, moving at the same speed, α, as the moving boundary.

In contrast, for faster wave speeds, $\alpha \gg \alpha_c$, the leading spot remains stationary, and the trailing stripe expands by engulfing the spot. New leading spots are inserted periodically be-

tween the moving boundary and the previous leading spot, as the space between them becomes sufficiently large. This behaviour is evident in Fig. 4.11 (left), where the locations of all local maxima remain roughly fixed. We performed a parameter sweep, simulating the system for a range of values of α, and found that travelling-wave-like behaviour is observed for $\alpha < 0.31$, while spot-insertion behaviour occurs for $\alpha > 0.45$. For the intermediate values of α lying between these values, we observe a mixture of behaviours. Our analysis allows us to make a crude estimate of the threshold $\alpha_c \approx 0.37$ for the Schnakenberg model in this context. Notice that this value for α_c is higher than that found with simulations on one-dimensional domains. This difference is likely due to the diffusion in the y–direction having a non-trivial impact on the dynamics at the cross-section. This value of α_c serves as a rough guide to delineate cases B and C for the Schnakenberg model. In a sufficiently wide two-dimensional domain (as in Section 4.3.2), we expect stripes parallel to the moving front to form if $\alpha \gg \alpha_c$, and stripes transverse to the moving front to arise if $\alpha \ll \alpha_c$. Just as we have seen in Section 4.3.2, the boundary between the two cases is not clear-cut, as we observe mixed behaviours for α near α_c in the third column.

For the CDIMA model, in the narrow two-dimensional domain described in Eq. (4.10), we observe the same travelling wave-like behaviour in both cases B and C, which is consistent with the observations made in one dimension.

4.3.4 Simulations on two-dimensional circular patterning subdomain

In this section, we simulate the CDIMA (Eq. (4.6)) and Schnakenberg models (Eq. (4.3)) on a circular patterning subdomain that expands radially at constant speed, that is,

$$\Omega'(t) = \{(x, y) | x^2 + y^2 \leq \rho(t)^2\}, \quad \rho(t) = \min(\alpha t, L_x/2, L_y/2), \tag{4.11}$$

instead of the rectangular patterning subdomain used in earlier sections. This choice matches the experimental and simulation set up of [100], which enables us to compare our results.

The steady state patterns are shown in Fig. 4.12. For simulations without on-going noise, the resulting patterns are concentric rings regardless of the wave speed α. With on-going noise, for the CDIMA model, we observe mostly concentric rings when the wave speed α is sufficiently high, and mostly stripes transversal to the circular boundary when the wave speed is low. This is consistent with the observations from [100].

The Schnakenberg model with $b = 1.6$ exhibits mostly the same qualitative behaviour as the CDIMA model. An interesting difference is that, with a low wave speed α, despite producing a pattern similar to the CDIMA model at first, the stripes near the circular boundary eventually

evolve into rings once the wave of competency stops, while the stripes in the centre of the domain remain in a radial direction. With $b = 1.4$, the Schnakenberg model tends to produce more spots. These observations are consistent with our observations in Section 4.3 on a rectangular patterning subdomain, where the vertical stripes parallel to the linear wave front play the same role as the rings in the circular setting.

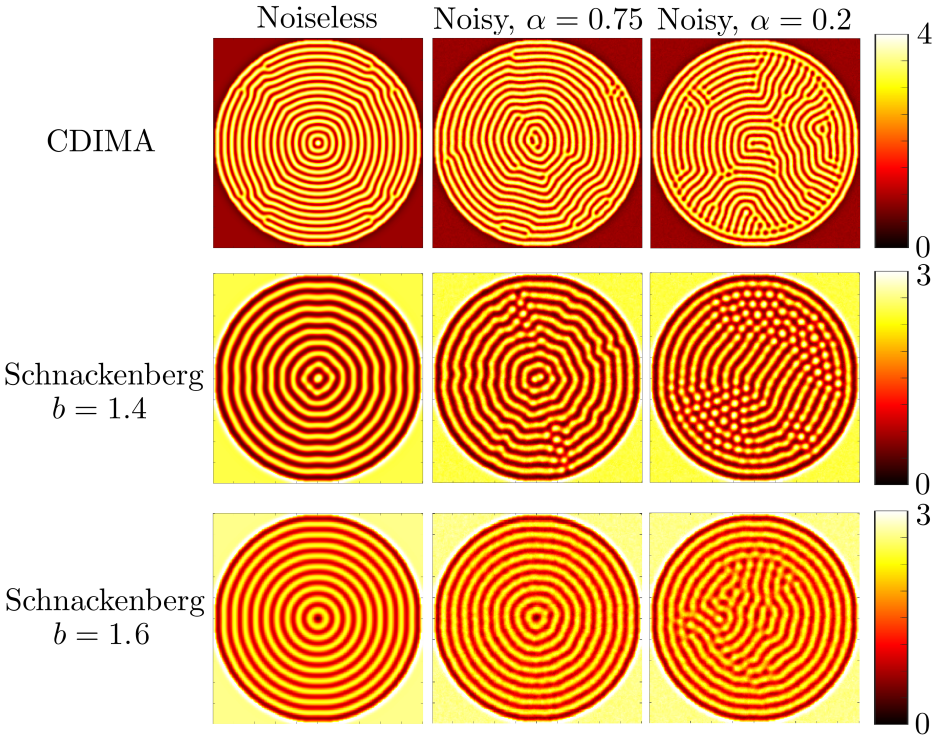


Figure 4.12: The steady state patterns produced with a circular patterning subdomain, as described in Eq. (4.11). The system is initialised by perturbing the homogeneous steady state with noise. In the first column, without on-going noise, the final pattern takes the form of concentric rings regardless of the value of α (the plot shown is produced with $\alpha = 0.75$). In the second and third columns, with the on-going noise, we observe different qualitative behaviours for different values of α.

4.3.5 Simulations with higher noise magnitude

In this section, we demonstrate the effect of higher levels of on-going noise on the pattern. We simulate the Schnakenberg model on a fixed domain (that is, $\Omega' = \Omega$), with $b = 1.4$, and other parameter values from Eq. (4.4). We initialise the system with horizontal stripes as described in Eq. (4.9).

When the noise is sufficiently strong, the system does not settle to a steady state, so in Fig. 4.13 we chose to present snapshots of the system at $t = 350$, which gives sufficient time for a steady state pattern to develop from perturbations to the homogeneous steady state in a fixed domain without on-going noise. Since on a two-dimensional domain the system is multi-stable,

over an infinitely long time scale even weak noise will eventually perturb the pattern into another steady state, disrupting the initial regular pattern. However, in biology, cells have a finite time window in which to respond to morphogen signals, and model behaviours over an infinite time span are therefore not biologically relevant. As such, we terminate the simulation on a time scale comparable to that of pattern formation from a perturbed uniform state to final steady state in a noiseless setting.

We observe that with the standard deviation of the noise $\mu = 0.01$, the initial horizontal pattern persists mostly undisturbed, only becoming slightly bent in certain places. With a stronger noise $\mu = 0.03$, the horizontal stripes are still recognisable but significantly disrupted, and with $\mu = 0.05$ the stripes are entirely unrecognisable, and the pattern evolves into a mess of spots and stripes that constantly change and shift as time continues to progress. These observations informed our choice of $\mu = 0.01$ for the simulations in the earlier sections, so that the noise does not overwhelm the system.

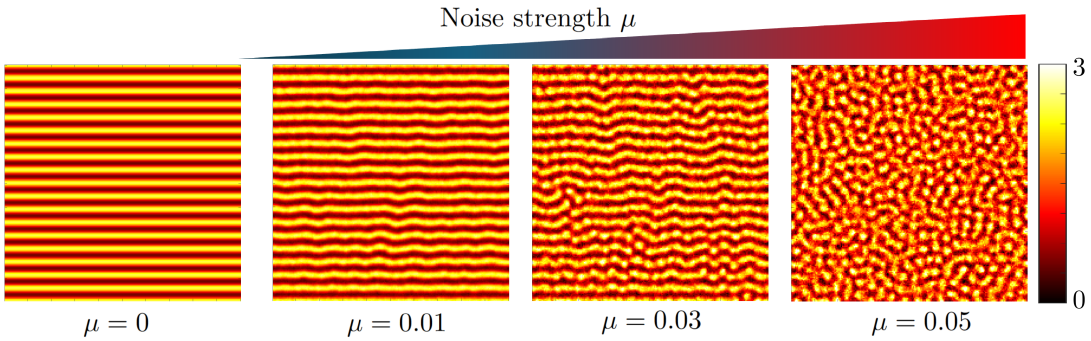


Figure 4.13: Snapshots of the simulations for the Schnakenberg model on a fixed domain, initialised at horizontal stripes, with $b = 1.4$ and varying noise strength μ, taken at $t = 350$.

4.4 Discussion

The aim of this chapter was to explore the impact of a wave of competency upon DDI models of pattern formation, using the Schnakenberg and CDIMA models as exemplar kinetics. Through extensive numerical exploration, we have shown that the speed of the wave of competency can determine the resulting pattern. Specifically, the alignment and directionality of striped patterns in two dimensions can be selected to be either parallel or perpendicular to the wave of competency, by an appropriate choice of the wave speed. Moreover, this phenomenon holds true for both the CDIMA and Schnakenberg kinetics, with minor differences in the model behaviours in one dimension. The agreement between the two models suggests that our results for two-dimensional behaviours may possibly be generalised to a larger class of models capable

of exhibiting a DDI, i.e., that the phenomenon does not depend on the specific form of the reaction terms, nor on whether the morphogens are in- or out-of-phase. Our observations are consistent in the presence of noise, giving us confidence that the results on stripe alignment and directionality are robust.

We have observed three main patterning regimes on expanding patterning subdomains with respect to the value of the wave speed α. These cases (A, B, C), along with the behaviour on the fixed domain (case D) for comparison, are summarised in Table 4.1. We have defined the natural speed of pattern propagation, ν, as the natural speed at which a pattern spreads from a spatially localised perturbation on a fixed domain. Our first main conclusion, which holds for both models, is that in order for stripe patterns in two dimensions to align in the presence of noise, the wave speed must be slower than the speed at which the pattern propagates naturally, that is $\alpha < \nu$. Otherwise, we obtain labyrinthine patterns that are essentially the same as those observed on fixed domains.

Our second main conclusion, again holding for both models, is that given $\alpha < \nu$, a higher α leads to stripes that are parallel to the moving front. These stripes arise as a two-dimensional analogue of peak insertion between the moving front and the existing pattern. In contrast, a lower α permits the formation of stripes that are transverse to the moving boundary, either perpendicular or at a slanted angle. Along the ridges of the stripes, the extension of the stripes as the front moves forward can be seen as a two-dimensional analogue of a travelling wave. For the Schnakenberg model, we were able to obtain an estimate for α_c, the threshold for α that separates these two cases, by observing the structure of the stripe that evolves when the region competent to form a pattern is restricted in the y–direction, so that it admits complicated patterns only in the x–direction. However, the separation between the two patterning regimes is not clear-cut, as we observe mixed behaviours for $\alpha \approx \alpha_c$. Moreover, despite its agreement in behaviour with the Schnakenberg model on broad two-dimensional domains, the CDIMA model does not display the peak splitting behaviour observed in the Schnakenberg model in one-dimensional and narrow two-dimensional domains.

Moreover, we found that in a model where both spot and stripe patterns are possible (e.g., Schnakenberg with $b = 1.4$), spots appear mainly when the stripes bend and break apart. Since in cases B and C the wave of competency favours the formation of straighter, more aligned stripes, spots appear less often with these wave speeds.

In the future, it will be interesting to understand why the two models behave differently on one-dimensional domains, but nonetheless exhibit similar behaviours on non-narrow two-dimensional domains. Another research direction, beyond the scope of this study, is to analytically determine the value of the critical threshold α_c. We speculate that this could be done by approximating the dynamics of the two-dimensional system along a stripe as a modified one-dimensional system, and using a wavelet analysis to determine the existence of a travelling wave solution.

We now compare the pattern selection mechanism investigated in this chapter with other mechanisms proposed in earlier works. In our model, we have chosen a wave of competency that advances in the x–direction only. However, the conclusions regarding alignment of the stripes with respect to the wave remain the same if the pattern subdomain expands isotropically (such as an expanding circle, see 4.3.4), meaning that this mechanism does not rely on any anisotropy. In contrast, for the anistropic diffusion mechanism considered by Shoji et al. [173], where the two morphogens have potentially differing preferred directions of diffusion, the anisotropy is essential for alignment. Hiscock and Megason [74] considered two mechanisms for pattern alignment, which were the existence of spatial gradients in the kinetic parameters, and anisotropies in diffusion or growth, which was also considered by Krause et al. [102], along with the geometric curvature of the underlying domain. These mechanisms were able to select between disordered labyrinthine patterns and one particular alignment for the stripes, but not between multiple possible directions, as our mechanism is capable of doing. Therefore, the mechanism we have proposed behaves differently compared to the previously proposed mechanisms for controlling the alignment and directionality of stripes.

We would also like to compare the wave of competency mechanism with DDI models posed on growing domains. While the expanding competent subdomain behind a wave of competency might be conceptually similar to growing domain models, they have different biological motivations, are not mathematically equivalent, and differ in behaviours. Models with isotropic growth, such as those studied in [35, 50, 105, 189], consider a scenario where the tissue underlying the patterns is itself expanding. This mechanism results in a dilution effect which acts on every point in the domain, causing patterns in the interior to continue to change as the domain grows. In contrast, a wave of competency only affects patterns near the boundary between the competent and incompetent subdomains, and interior patterns remain static once the wavefront has moved sufficiently far away. In one spatial dimension, Crampin et al. [50] showed that

the multi-spike pattern in the Schnakenberg model with exponential domain growth undergoes frequency-doubling, where every spike splits into two once the domain has grown sufficiently large. This is in contrast to the behaviour behind a wave of competency, where only the leading spike can split. In two spatial dimensions, Van Gorder et al. [189] showed that the Schnakenberg model with linear domain growth is able to produce disorganised, mixed spot-stripe patterns at low growth rate, and stripes parallel to the domain boundary at high growth rate. This behaviour is largely similar to our cases A and B. However, Van Gorder et al. did not observe stripes perpendicular to the domain boundary (our case C). Crampin et al. [49] considered apical growth, which is a form of non-uniform growth that only happens near the domain boundary. Although apical growth bears some resemblance to the wave of competency, and produces similar behaviours in one dimension, it is not mathematically equivalent to the wave of competency model, and it remains to be seen whether these models have similar behaviours in two dimensions.

Further generalisations of the wave of competency model are possible. In this chapter, the wave of competency has a simple shape, and its movement is independent of the morphogen concentrations. In a real biological system, the wave speed and the shape of the wave front may be coupled with the morphogen concentrations. For example, in [118], the patterning domain is interpreted as the interior of a cell, and the morphogens, through interactions with the actin cytoskeleton, deform the cell. It would be interesting to compare behaviours in this kind of dynamic domain under zero-flux boundary conditions, with behaviours in a domain bounded by a moving wave of competency. The findings in this chapter highlight the natural speed of pattern propagation as an important and intuitive quantity that plays a role in determining model behaviour. Therefore, it would be useful to investigate the way this speed depends on the model dynamics and parameter values. In [94], Klika et al. estimated the speed of pattern propagation using asymptotic analysis in the regime close to the onset of DDI, but it remains to generalise the results to broader regimes.

While the predictions from the analysis in this chapter concur with observations in chemical experiments conducted by Konow et al. [100], these predictions remain to be tested on real biological systems, with parameter values estimated from real data. There are several challenges that need to be resolved before experimental verification of the predictions is possible. First, while a few biological systems are postulated to be driven by a DDI mechanism behind a wave of competency, none are confirmed as such. This confirmation is required for finding a

suitable biological system as the target of an experiment. Second, inferring parameter values and performing identifiability analysis for DDI models is very difficult, since the pattern tends to be sensitive to slight perturbations to the system, and a small change in a parameter value can lead to a large quantitative change in the pattern, even if the pattern remain qualitatively similar. This means that the classical inference methods, such as those introduced in Ch. 2, are unsuitable in such cases. Some more sophisticated and specialised techniques for inferring parameter values for DDI models have been developed in [168]. While promising, in their current forms, these techniques are difficult to use, and cannot be used to perform identifiability analysis. Third, it might be difficult to alter the speed of the wave of competency to a sufficiently large degree in a biological system so that it crosses one of the critical wave speeds to produce a different qualitative behaviour as the models suggest. Therefore, overcoming these challenges, and experimentally verifying the predictions of this chapter with a biological system, remain future work.

In summary, we have demonstrated that modulation of the speed of propagation of a wave of competency can robustly select patterns in DDI models. These results have implications for patterning in a number of biological systems where a wave of competency has been either identified, or could exist. Our results also pave the way for future studies that aim to control the directionality and alignment of DDI patterns by modulating the wave speed. More complex patterns may be obtained by also varying the shape of the wave front.

This chapter provided an example of a mechanism, the wave of competency, that can act as a control to a complex system. The next chapter shifts the focus toward obtaining an optimal control into another kind of complex system — agent-based models.

Chapter 5

Controlling biological systems through machine learning

The previous chapters discussed parameter identifiability, investigated strategies for improving parameter identifiability through experimental design, and explored mechanisms that can control the behaviour of biological systems. After obtaining accurate estimates for parameter values and understanding the internal mechanisms driving system behaviour, and identifying an external mechanism that could be modulated experimentally to affect changes in system behaviour, we are now ready to begin to consider the optimal way to apply the control in order to produce a desired outcome.

Compared to ODE and PDE-based models, it is considerably more challenging to apply control to ABMs, and the techniques are relatively less developed, leaving ample room for exploration. This chapter proposes methods to overcome the challenges associated with controlling ABMs, offers preliminary results in the first steps towards developing this method, and speculates on potential future directions.

5.1 Agent-based modelling and control

Applying control to agent-based models (ABMs) is of particular interest, since the ABM framework is natural for describing a wide range of interesting phenomena in biology, including cell interaction and movement [30, 196], collective animal behaviour [6, 192], and fine-grained models of the spread of infectious disease in a population [207]. In many of these cases, a certain outcome is more desirable than others, which necessitates the need to develop methods for effectively controlling these models. For example, in a model of infectious disease, we might wish to optimally suppress the spread of the disease with certain interventions acting as control mechanisms. At the cellular level, the ability to control gene expression in single cells is useful for

personalised medicine and synthetic biology [188]. In another example, experimental apparatus that herd electrotactic cells by applying an electric field has been developed in [212]. Such devices open up ways to control cell migration in order to accelerate wound healing or inhibit tumour growth.

A major challenge faced when applying optimal control to ABMs is that the main techniques for optimal control, such as PMP and HJB introduced in Appendix C, have been formulated for differential equation-based models, and few techniques are directly applicable to ABMs. This chapter explores a potential direction towards solving this challenge, by obtaining differential equation-based models that approximate the behaviour of an ABM. Such models can serve as a surrogate for applying control strategies, bridging the gap between control theory and ABMs.

5.1.1 Surrogate model

In this chapter, we assume that the ABM of interest is a stochastic model, which is the case for most ABMs used in mathematical biology. There are a few existing methods for applying control to ABMs, but they each have major disadvantages. First, many ABMs can be viewed as a very large system of SDEs, with one or more equations describing the evolution of each agent, and it is, in theory, possible to apply PMP or other techniques to this system. However, the number of agents in a typical ABM can be very large, so this system of SDEs would have a very high dimension, making it unwieldy for the purpose of applying control theory. Direct optimisation is also possible, where we parameterise the control function in some way, view the cost as a function of the control parameters, then minimise the expected value of the cost using a general-purpose optimisation algorithm. For example, An et al. [3] demonstrated the use of genetic algorithms to find an optimal control for a simple ABM. A specific case of the direct optimisation approach is reinforcement learning. Reinforcement learning has been applied to a variety of control problems in engineering, such as controlling the movement of vehicles [149, 199] and airplanes [150]. Reinforcement learning applied in this context is also known as policy gradient [183]. In this approach, the control is typically represented as a neural network, which typically contains a large number of control parameters. This network is then trained using some variant of gradient descent. Training data are obtained by simulating the ABM with a certain control strategy, then evaluating the cost function. Reinforcement learning is capable of handling systems with stochastic dynamics [34], making it particularly suitable for applying control to ABMs, which tend to be stochastic. The problem with direct optimisation

in general, which includes reinforcement learning, is that many simulations of the ABM are required to accurately estimate the expected value of the cost function, and the cost function needs to be evaluated many times throughout the optimisation procedure, making this approach computationally expensive. Alternative approaches that exploit the mathematical structure of a control problem have the potential to be much more computationally efficient.

The approach considered in this chapter is to use a simpler, equation-based model to act as a surrogate for applying control to an ABM. This approach is illustrated in [3]. The main idea is to treat data generated from the ABM in the same way as we would treat data obtained from experiments, and derive an equation-based model to capture the most important characteristics of the ABM. While in certain cases we might be able to formulate an equation-based model for the biological system directly, having the ABM as an intermediary between the actual biological system and the equation-based model can be useful in cases where direct formulation of an equation-based model is impractical. In cases where the ABM arises from a biological system, it is generally much easier to generate additional data from the ABM than to obtain more experimental data from the underlying system. Once a surrogate equation-based model is obtained, the optimal control to the surrogate model can be derived using control theory techniques such as PMP or HJB, introduced in Appendix C. This optimal control then needs to be transferred to the ABM and tested to ensure that it performs well, which we expect it to do if the equation-based model is a good approximation to the ABM.

We discuss two approaches for obtaining such a surrogate equation-based model that are relevant to the ABM context, with a simple example of each given in [3]. First, an ODE or PDE model can be derived by formally coarse-graining the ABM. In many ABMs, the positions of the agents are the most important state variables. For some simpler ABMs, in the limit where the number of agents, N, approaches infinity, a PDE that approximates the density of agents can be formally derived. In many cases, such derivations use a mean-field approximation, hence the resulting PDE model is called the mean-field model. The canonical example for this is that for a large number of agents undergoing a random walk according to standard Brownian motion, the density evolves according to the diffusion equation [89]. For more complex ABMs, such as those including multiple types of agents [47], non-local interactions between agents [43, 142], or additional state variables for the agents, deriving a mean-field model can be much more difficult. Even if such a model can be derived, it will likely be very complicated, and applying PMP to the model can result in a numerically intractable problem. Furthermore, a mean-field model is

only guaranteed to approximate the ABM well in the regimes where the assumptions made in the derivation of the model (e.g., $N \to \infty$) are valid. In the regimes where these assumptions are not satisfied, the mean-field model may become ill-posed [139], and can no longer represent the behaviour of the ABM. Therefore, controlling an ABM by formally deriving a mean-field model is practical only in simpler cases, which calls for an alternative approach.

The second approach we discuss here is to derive an equation-based model empirically, by fitting an ODE or PDE system with simple forms and unknown coefficients to the output of the ABM. As an example, An et al. [3] fitted a linear PDE to a relatively simple ABM. This approach prioritises simplicity, aiming to fit observations of the underlying system (the ABM in this case), while deprioritises explaining the mechanisms responsible for the behaviour. As a result, this approach tends to produce a mostly phenomenological model. The hope is that for the purpose of applying control, the surrogate model does not need to be mechanistically accurate, nor retain all information in the ABM. Rather, it may suffice to approximate the behaviour of the variables relevant for the goal of control. We will investigate whether that is the case in this chapter.

The advantage of this second approach is that it requires much less manual calculation. The resulting surrogate model is also simple by design, which makes it easier to apply control. On the other hand, it might not be obvious what form of model should be chosen, and whether the resulting model approximates the ABM well. Indeed, it might very well be that a simple equation-based model that can approximate a particular ABM well does not exist.

This chapter focuses on the second approach, where we of empirically fit simple equation-based models to ABM data, rather than formally deriving mean-field models. To this end, we propose the development of a method based on machine learning that may be effective for finding an appropriate, simple, surrogate model.

5.1.2 Equation learning

In recent years, the utilisation of machine learning in mathematical modelling has experienced rapid growth. While machine learning-driven statistical modelling can be seen as a "competitor" to mechanistic modelling, there have been calls to unite the two approaches in mathematical biology to achieve better prediction accuracy, and to gain additional insight into the biological system beyond what is possible with either approach alone [2, 12]. Equation learning is one of the areas where machine learning and mechanistic modelling meet. Equation learning refers to

a class of machine learning algorithms that aim to find equations that best represent data, and is an area which has been flourishing in recent years [23, 29, 165, 166]. For this chapter, we specifically aim to discover governing equations that serve as the right-hand side of an ODE or PDE system that describes the dynamics of a set of data generated from an ABM. Equation learning algorithms typically parameterise the function space with a large number of parameters, then search through the function space to find a system of equations that minimise a certain loss function. The loss function typically involves terms that measure fitting error, regularisation terms, and other application-specific terms.

While superficially similar to traditional model fitting, equation learning is different in a few ways which, it can be argued, make it more attractive for the purpose of obtaining an approximate model for ABMs, but there are also drawbacks. First, in traditional model fitting, as was carried out in Chapter 2, a functional form of the governing equations is fixed *a priori*, with relatively few unknown parameter values to be determined through inference. To propose such a functional form requires knowledge or insight into the mechanistic processes driving the system behaviours. In contrast, equation learning assumes much less knowledge of the system, and searches through a much larger function space compared to typical model fitting, which makes it likely to be able to find governing equations that approximate the data well. This is helpful for complicated ABMs, where it can be difficult to propose a good functional form for the governing equations. The drawback is that the larger function space is parameterised by a much larger number of parameters, and the dimension of the search space is equal to the number of parameters. As we have seen in Chapter 2, even a small increase in search dimension can result in a large increase in computational cost, the well-known "curse of dimensionality" [21, Ch. 1.4]. This means that, in practice, we often have to be content with finding a local optimum, since finding the global optimum can be intractable. This also raises the issue of practical identifiability. As we found in Ch. 2, complicated models with a higher number of parameters tend to be less identifiable. Given the large number of parameters used in equation learning, we expect that the surrogate model being fitted to the ABM data will be practically non-identifiable. However, whereas parameter identifiability is critical for gaining an accurate understanding of the biological system, it is less crucial in the context of applying control. The surrogate model need not be a mechanistically correct description of the ABM dynamics nor the underlying biology, as it merely serves as a vehicle for obtaining an optimal control. It is possible that a surrogate model that can approximate the dynamics of the ABM well can lead to a good control

for the ABM, even if its terms are not mechanistically correct. We will demonstrate the extent to which this is true using a simple example in Sec. 5.3.

Second, model fitting methods typically compare the data to the solution of a proposed model to evaluate its goodness of fit. Doing so is problematic for stochastic models, which include most ABMs, since two realisations of a model with the same parameter values may be quantitatively very different, making direct comparison difficult. Such cases necessitate running a large number of simulations of the model to obtain a distribution of model solutions. In contrast, some equation learning methods (such as the ones considered in this chapter) directly compare the proposed governing equations to the time derivatives of the data. This makes it more straight-forward to evaluate the goodness of fit. The drawback is that the data coming from a stochastic ABM are noisy by definition. Therefore, computing the time derivative naively with finite differences can magnify the noise. This is recognised by Lagergren et al. [107], who evaluated several de-noising methods that can be helpful in this situation.

In this chapter, we will focus on adapting Sparse Identification of Nonlinear Dynamics (SINDy), an equation learning method developed in [29], for the task of obtaining approximate equation-based models for ABMs. Given time series data arising from a system governed by an unknown ODE, SINDy represents the right-hand side of the ODE as a linear combination of user-specified library functions with unknown coefficients, and aims to discover an ODE model that well approximates the dynamics of the unknown ODE while maintaining sparsity, i.e., with few non-zero coefficients for the library functions, since many biological and physical systems tend to have only a small number of important terms in their governing equations, and sparse regression makes the algorithm more robust against noise. The library functions, by default, consists of all polynomial functions of the state variables, up to a maximum degree specified by the user. Additional terms, such as those involving trigonometric functions, or rational functions, can be added to the library if appropriate for the application. A sparse set of coefficients of the library functions is obtained by training the model with data, using an appropriate sparse regression algorithm. It is not always clear which training algorithm is the most appropriate, as this depends on the problem.

Another difficulty associated with SINDy is that the library functions must be chosen appropriately to allow for the existence of a sparse representation of the true dynamics. This is mitigated by an update to the method [40], referred to as Autoencoder-SINDy. An autoencoder is a neural network architecture used in unsupervised learning, consisting of two individual

multi-layer neural networks, called the encoder and decoder. The encoder maps the input data to a lower-dimensional latent representation, and the decoder does the same in reverse by reconstructing the original data from the latent representation. The idea is that in many cases, complicated data can be represented by relatively few informative features, akin to applying a nonlinear coordinate transform to the data, hence the latent variables are also called coordinate representations. The Autoencoder-SINDy method simultaneously trains an autoencoder for the data to obtain a low-dimensional coordinate representation, together with SINDy, which learns the governing equations for the latent variables. By training both components simultaneously, the algorithm will try to favour a latent representation (or coordinate transformation) for the data, such that the governing equations can be expressed with a combination of a small number of library functions in the new coordinate, i.e., a sparse representation of the dynamics exists. In this way, the choice of library functions is less important (as long as they are sufficiently expressive, i.e., span a sufficiently large space of functions), as the learning process will try to pick an appropriate latent representation according to the chosen library functions. The variables in the latent representation can be thought of as summary statistics of the data. These summary statistics are selected automatically by the algorithm, relieving the user from a non-trivial task. The sparse representation of the dynamics makes interpreting the resulting model easier, since there are fewer, simpler terms (represented by library functions) on the right-hand-side of the equations, and makes it more tractable to apply control.

5.1.3 Overview

In this chapter, we will investigate the suitability of adapting SINDy to develop a method for learning a surrogate model for ABMs. The method consists of using finite differences to compute a set of user-specified summary statistics for the ABM, then using a modified version of SINDy to learn a set of ODEs that governs the dynamics of the summary statistics. We will introduce the method in general terms in Sec. 5.2, and demonstrate the method by implementing it to apply control to a simple ABM of motile cells in Sec. 5.3. We propose ways to further expand, generalise, and automate the method in Sec. 5.4. Finally, we summarise the findings, and alternatives to the proposed methods, in Sec. 5.5.

5.2 Learning surrogate model using specified summary statistics

Developing a systematic and automatic approach to obtain surrogate models for a broad category of ABMs poses significant challenges. Therefore, it is necessary to break down the development of such a method into more manageable stages. In this chapter, we demonstrate the first stage towards the development of a method for learning surrogate models, centred around a machine learning algorithm.

5.2.1 Notation

We will use the following notations. Let N be the number of agents, which may change during the course of a simulation (e.g., modelling cell division or death). Let n_t be the number of time points in the data. For simplicity, we assume that data are provided at equally spaced points in time, $t_i = i\Delta t,\ i = 1, \ldots, n_t$, with the terminal time $T = n_t \Delta t$, although it is straight-forward to generalise to the case where the time points are not evenly spaced. Suppose that each agent has m state variables (e.g., their position coordinates), so there are Nm state variables overall. Let $\boldsymbol{x}_i(t) \in \mathbb{R}^m$ be the state of agent i at time t, and $\boldsymbol{x}(t) = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N) \in \mathbb{R}^{Nm}$ the state of the whole system, and $X \in \mathbb{R}^{n_t \times Nm}$ the data matrix, such that the i^{th} row of X is $\boldsymbol{x}(t_i)$. Let $\boldsymbol{u}(t)$ be a vector of control variables, with their values specified by, and known to, the user. The eventual goal is to find an optimal control $\boldsymbol{u}^*(t)$ to minimise an objective functional of the form

$$J[\boldsymbol{u}] = \int_0^T \mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\mathrm{d}t, \tag{5.1}$$

where $\mathcal{L}$ is the Lagrangian, initially defined as a function of the state of the agents, $\boldsymbol{x}$.

We assume that there exists a set of $k \ll Nm$ summary statistics computed from the state of the agents, $\boldsymbol{z}(t) = \phi(\boldsymbol{x}(t)) \in \mathbb{R}^k$, that sufficiently summarises the behaviour of the ABM (the precise meaning of this is discussed later), and simultaneously evolves according to governing equations in the form of ODEs,

$$\frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{z}, \boldsymbol{u}, t; \boldsymbol{\xi}),$$

where $\boldsymbol{\xi}$ are the parameters whose values are to be learned through machine learning, and $\boldsymbol{f}$ is the right-hand side of the governing equations to be learned. For the rest of the chapter, the dot notation (e.g., $\dot{\boldsymbol{z}}$) denotes derivatives in time.

5.2.2 SINDy with control

The goal of the algorithm we implement in this chapter is to learn the governing equations for the dynamics of user-specified statistics. The number of summary statistics, k, is much smaller than the number of state variables in the ABM, Nm. Learning the dynamics of the summary statistics yields a low dimensional surrogate model, which is a feasible target for applying PMP, whereas applying PMP to the original Nm-dimensional model can be highly intractable. The machine learning algorithm receives the time series of the summary statistics, $\boldsymbol{z}(t)$, and the control, $\boldsymbol{u}(t)$, as input, and outputs the right-hand side of the governing equations $\boldsymbol{f}(\boldsymbol{z}, \boldsymbol{u}, t; \boldsymbol{\xi})$. We also assume that it is known how to mathematically state the control problem, therefore the Lagrangian, in terms of the chosen summary statistics, $\boldsymbol{z}$.

Under the SINDy approach, let $\dot{\boldsymbol{z}}(t)$ be the empirical time derivative of $\boldsymbol{z}(t)$, which can be computed by applying an appropriate finite difference scheme to $\boldsymbol{z}(t)$ directly. After specifying a set of s library functions $g_i(\boldsymbol{z}, \boldsymbol{u}, t), i = 1, \ldots, s$, we aim to represent the dynamics of each summary statistic as a linear combination of this same set of library functions, with coefficients $\xi_{i,j}$ to be found by minimising a certain loss function during the training process:

$$\frac{\mathrm{d}z_j}{\mathrm{d}t} = f_j(t) \approx \sum_{j=1}^{s} \xi_{i,j} g_i(\boldsymbol{z}(t), \boldsymbol{u}(t), t), \tag{5.2}$$

where z_j is the j^{th} summary statistic. Note that in the original SINDy algorithm from [29], the library functions depend on $\boldsymbol{z}$ only.

Let the data derivative matrix be $\dot{Z}_{i,j} = \dot{z}_j(t_i)$, $\dot{Z} \in \mathbb{R}^{n_t \times k}$, the library matrix be $\Theta_{i,j} = g_j(\boldsymbol{z}(t_i), \boldsymbol{u}(t_i), t_i)$, $\Theta \in \mathbb{R}^{n_t \times s}$, and the coefficient matrix be $\Xi_{i,j} = \xi_{i,j}, \Xi \in \mathbb{R}^{s \times k}$. Then Eq. (5.2) can be rewritten as

$$\dot{Z} \approx \Theta \Xi. \tag{5.3}$$

The training for the machine learning algorithm will attempt to minimise a loss function with respect to Ξ, so that Eq. (5.3) will approximately hold. The simplest choice for the loss function is

$$L_1(\Xi) = \frac{1}{n_t k} ||\dot{Z} - \Theta \Xi||_F^2, \tag{5.4}$$

where $|| \cdot ||_F$ denotes the Frobenius norm. The $1/(n_t k)$ normalisation factor arises from the dimension of the matrices inside the norm. This loss function arises from the assumption that the discrepancy between the two sides of Eq. (5.3) arises from normally distributed process noise present in the dynamics. The solution for the coefficients Ξ can be found by solving a

linear least squares problem, for which efficient and well-known algorithms exist. However, this solution does not guarantee that the coefficients are sparse, i.e., most entries of Ξ are likely to be non-zero. This results in a non-sparse representation of the dynamics, which in turn constitutes a complicated model that is likely to be unwieldy to apply control. Without any regularisation, the resulting model is also prone to overfitting to the noise in the data. There are several different training methods that promote sparsity in the solution (sparse regression), and we now describe three of them.

The first training method, recommended in [29], is to apply sequential thresholding to the coefficients $\xi_{i,j}$. This training method requires the user to specify a minimum threshold $\epsilon > 0$ for the coefficients. In each iteration, the linear least squares problem is solved, and any coefficient such that $|\xi_{i,j}| < \epsilon$ is eliminated from the dynamics equations. The least squares problem with fewer variables is then solved again in the next iteration. The training continues until either none of the remaining coefficients have magnitude below the threshold, or for a specified maximum number of iterations.

The first training method relies on an appropriately chosen value of ϵ. If ϵ is too small, then the sequential thresholding procedure might fail to achieve sparsity by retaining too many non-zero coefficients. If ϵ is too large, then terms important to the dynamics might be inappropriately eliminated. To avoid requiring the specification of any threshold, we propose a second training method, which we call Greedy Elimination (GE). This method requires partitioning the data into a training set and a validation set, which is common practice in machine learning. In each iteration, we solve the linear least squares problem. Then, for each j, instead of eliminating any coefficients falling below a threshold, we eliminate the coefficient with the smallest magnitude, $\min_i |\xi_{i,j}|$. This choice is greedy, as it looks at the current iteration only, and takes the locally optimal choice. The training method continues for s iterations, so that in the final iteration, only a single coefficient remains in the governing equation for each summary statistic. We compute the training and validation losses, that is L_1 in Eq. (5.4), for the training and validation datasets associated with each summary statistic at each iteration. The training loss is non-decreasing from one iteration to the next, since we are essentially fitting a model which is a strict subset of the model from the previous iteration to the same data. However, eliminating unnecessary coefficients reduces overfitting, so the validation loss might decrease. For each j, we choose the governing equation that gives the smallest validation loss across all iterations to be included in the final model. This training method is summarised as Algorithm 1. We partition $\{1, \ldots, n_t\}$

into two subsets, $I_T,\ I_V$, as the indices of the data points in the training and validation datasets. We use the notation X_{S_1,S_2} to denote the submatrix of a matrix X with rows in the set of indices S_1, and columns in S_2. We use $\Xi^{(l)}$ to denote the coefficient matrix at iteration l, and $S_j^{(l)} \subseteq \{1,\dots,s\}$ to denote the set of remaining coefficients for the dynamics of z_j in iteration l.

Algorithm 1 GE algorithm for training SINDy

(1) For $j = 1,\dots,k$, initialise $S_j^{(1)} = \{1,\dots,s\}$

(2) For $l = 1,\dots,s$:

(2.1) For $j = 1,\dots,k$:

(2.1.1) Solve linear least square problem,

$$\Xi^{(l)}_{S_j^{(l)},\ j} = \min_{\xi_{S_j^{(l)}}} ||\dot{Z}_{I_T,\ j} - \Theta_{I_T,\ S_j^{(l)}}\Xi_{S_j^{(l)},\ j}||_F^2, \quad \Xi^{(l)}_{\{1,\dots,s\}\setminus S_j^{(l)},\ j} = 0.$$

(2.1.2) Compute validation loss

$$\text{ValLoss}_j^{(l)} = ||\dot{Z}_{I_V,\ j} - \Theta_{I_V,\ S_j^{(l)}}\Xi_{S_j^{(l)},\ j}||_F^2.$$

(2.1.3) Eliminate smallest remaining coefficient,

$$S_j^{(l+1)} = S_j^{(l)} \setminus \{q\}, \text{ where } q = \underset{i\in S_j^{(l)}}{\text{argmin}} \left\{|\xi_{i,j}^{(l)}|\right\}.$$

(3) For $j = 1,\dots,k$:

(3.1) Return the coefficients at the iteration with lowest validation loss,

$$\xi_{i,j} = \xi_{i,j}^{(l_{min})}, \text{ where } l_{min} = \underset{l\in\{1,\dots,s\}}{\text{argmin}} \text{ValLoss}_j^{(l)}.$$

The first two methods rely on the fact that the loss function can be minimised by solving a linear least squares problem. In the later stages, we will augment SINDy with an additional neural network architecture in order to accomplish additional tasks, making the loss function more complicated, and methods based on linear least squares would not be sufficient. While the first two methods are not generalisable, the third method, which is based on least absolute selection and shrinkage (LASSO) and illustrated in [40], is generalisable. Here, instead of Eq. (5.4), the loss function is modified to include an l^1 regularisation term,

$$L_1(\Xi) = \frac{\lambda_0}{n_t k}||\dot{Z} - \Theta\Xi||_F^2 + \frac{\lambda_1}{ks}||\Xi||_1, \tag{5.5}$$

where $||\Xi||_1$ denotes the sum of the absolute values of all entries, and λ_0, λ_1 are user-specified weights for the two terms. The weights are normalised by the number of entries in the matrices

or vectors inside the norms. Including the additional l^1 term is a common tactic to induce sparsity. Then, the loss function is minimised using Adam (Adaptive moment estimation [93]), which is a method derived from gradient descent, and has seen wide-spread usage in machine learning [162]. Sequential thresholding is applied at regular intervals during the training process, such that any sufficiently small $|\xi_{i,j}|$'s are fixed to zero in all subsequent iterations, to further induce sparsity. This method is slower than the first two methods, but its generalisability means it will be the most useful method as we develop the method further to incorporate automatic selection of summary statistics in the future. We will implement these training methods in Sec. 5.3 to illustrate their performance.

5.3 Demonstrating the method on a simple ABM

Having formulated a preliminary method in Sec. 5.2, in this section, we demonstrate it on a simple ABM for the behaviour of motile cells. All computational codes for this section are written in Python, and provided on Github at `https://github.com/liuyue002/abmEncoder`. Parts of the codes are modified from Champion et al. [40]. The ABM is a very simple model of non-interacting motile cells, which undergo a random walk where the mean velocity of the cells is influenced by a control mechanism. This control mechanism could represent, e.g., an applied electric field. There has been recent interest in controlling cell movement using electricity. An experimental apparatus that allows programmable application of an electric field to cells has been developed in [212], and Martina-Perez et al. [126] have modelled the movement of MDCK cells in response to the applied electric field.

The ABM considered in this section consists of N agents, representing cells, undergoing a biased random walk in one spatial dimension. As this is an abstract and idealised model, all quantities in the model are dimensionless. The only control variable $u(t)$ propels the agents to the left when positive, and to the right when negative. More specifically, the ABM is simulated in discrete time, with a time step $\Delta t = 0.01$, from $t = 0$ to $t = T = 100$, so there are $n_t = 10000$ time points. The position of agent j, x_j, evolves as

$$x_j(t_{i+1}) = x_j(t_i) - u(t)\Delta t + \sqrt{2D_0\Delta t}\,\eta_{i,j}, \quad x_j(0) \sim \text{Unif}(0,1), \tag{5.6}$$

where $\eta_{i,j} \sim \mathcal{N}(0,1)$ are i.i.d standard normal random variables, $\text{Unif}(0,1)$ denotes the uniform distribution on $(0,1)$, and $D_0 = 5$ is a model parameter that scales the degree of randomness in cell movement, which act as a "diffusion coefficient", see Eq. (5.7) below. The ABM stated in

Eq. (5.6) is a discrete time model, with no interactions between the agents. In the limit $\Delta t \to 0$, the trajectories of the agents follow the SDE

$$\mathrm{d}x_{j,t} = -u(t)\mathrm{d}t + \sqrt{2D_0}\mathrm{d}W, \quad j = 1, \ldots, N,$$

where $\mathrm{d}W$ denotes the standard Brownian motion. This SDE has the corresponding Fokker-Planck equation for the distribution of the position of agents, $p(x,t)$, in the form of an advection-diffusion PDE:

$$\frac{\partial p}{\partial t} = D_0 \frac{\partial^2 p}{\partial x^2} + u(t)\frac{\partial p}{\partial x}. \tag{5.7}$$

The state variables of the ABM consist of the position of the agents, $\boldsymbol{x}(t) = \{x_j(t)\}_{j=1}^N$. We consider two summary statistics, $M(t)$, the mean of agent positions, and $V(t)$, the variance of agent positions. The function ϕ that maps the ABM state variables to the summary statistics can be written as

$$\boldsymbol{z}_{\mathrm{ABM}}(t) = \begin{bmatrix} M(t) \\ V(t) \end{bmatrix} = \phi(\boldsymbol{x}(t)) = \begin{bmatrix} \frac{1}{N}\sum_{j=1}^N x_j(t) \\ \sum_{j=1}^N (x_j(t) - M(t))^2 \end{bmatrix}. \tag{5.8}$$

Through elementary calculations, it can be shown that the summary statistics obey the following mean-field ODEs in the limit of $N \to \infty$:

$$\dot{\boldsymbol{z}}_{\mathrm{MF}}(t) = \begin{bmatrix} \dot{M}_{\mathrm{MF}}(t) \\ \dot{V}_{\mathrm{MF}}(t) \end{bmatrix} = \begin{bmatrix} -u(t) \\ 2D_0 \end{bmatrix}. \tag{5.9}$$

The mean-field ODE has the explicit solution

$$\boldsymbol{z}_{\mathrm{MF}}(t) = \begin{bmatrix} M_{\mathrm{MF}}(t) \\ V_{\mathrm{MF}}(t) \end{bmatrix} = \begin{bmatrix} M(0) - \int_0^T u(t)\mathrm{d}t \\ V(0) + 2D_0 t \end{bmatrix}. \tag{5.10}$$

We use finite differences to calculate $\dot{\boldsymbol{z}}_{\mathrm{ABM}}(t)$ from $\boldsymbol{z}_{\mathrm{ABM}}(t)$. More specifically, we chose to use the upwind first-order finite difference scheme for its simplicity. Let $\dot{\boldsymbol{z}}_{\mathrm{FD}}$ denote $\dot{\boldsymbol{z}}$ as calculated using the one-sided first-order finite difference scheme, that is,

$$\dot{\boldsymbol{z}}_{\mathrm{FD}}(t_i) = \frac{\boldsymbol{z}(t_{i+1}) - \boldsymbol{z}(t_i)}{\Delta t},$$

where $\Delta t = 0.01$ is the same as the time stepping used in the ABM simulations. This scheme is simple and easy to implement, however, a finite difference scheme will magnify any noise in $\boldsymbol{z}$, which may make equation learning more difficult. The main source of noise in $\boldsymbol{z}$ will be the randomness in cell movement. To quantify how much the noise will be amplified, we can use

Eqs. (5.6) and (5.8) to show that

$$\mathbb{E}[\dot{M}_{\mathrm{FD}}] = -u(t), \tag{5.11}$$

$$\mathbf{Var}[\dot{M}_{\mathrm{FD}}] = \frac{2D_0}{\Delta t N}, \tag{5.12}$$

$$\mathbb{E}[\dot{V}_{\mathrm{FD}}] = 2D_0, \tag{5.13}$$

$$\mathbf{Var}[\dot{V}_{\mathrm{FD}}] = \frac{4D_0^2}{N} + \frac{4D_0}{\Delta t N}(V + M^2) + O(N^{-2}) + O(N^{-2}\Delta t^{-1}), \tag{5.14}$$

where $O(\cdot)$ denotes the big-O notation with respect to N and Δt. Eq. (5.11) and Eq. (5.13) show that the mean of the finite difference approximation matches that of the mean-field ODE (Eq. (5.9)), so $\dot{\boldsymbol{z}}_{\mathrm{FD}}$ is an unbiased approximation to $\dot{\boldsymbol{z}}$. Eq. (5.12) and Eq. (5.14) show that the variance of $\dot{M}_{\mathrm{FD}}$, as well as the dominant term in the variance of $\dot{V}_{\mathrm{FD}}$, are inversely proportional to Δt, so computing the finite difference using a finer time step will amplify the noise, as expected. On the other hand, increasing the number of agents N will decrease the variance of both $\dot{M}_{\mathrm{FD}}$ and $\dot{V}_{\mathrm{FD}}$. In many cases considered, the $V + M^2$ factor in Eq. (5.14) is large, so we expect the data for $\dot{V}$ to be much noisier than for $\dot{M}$. We will show how the noise affects equation learning in the next section.

5.3.1 Learning the dynamics

We now implement a modified version of SINDy that incorporates the control variable, and investigate whether it can recover the ground-truth ODE model for the summary statistics, Eq. (5.9). The library functions consist of all polynomial combinations of the summary statistics and control up to order 2, that is,

$$\{g_i(\boldsymbol{z}, \boldsymbol{u}, t)\}_{i=1}^{s} = \{1, M, V, u, M^2, MV, Mu, V^2, Vu, u^2\},$$

for a total of $s = 10$ library functions. We implement two training algorithms, the greedy elimination (GE) method, and the LASSO method, as described in Sec. 5.2. For the LASSO method, we minimise the loss function L_1 defined in Eq. (5.5), with the weights of the two loss terms set to $\lambda_0 = 100$, and $\lambda_1 = 1$. The weight of the regularisation term, λ_1, is small enough compared to the weight of the model fit term, λ_0, so it will not significantly impact the quality of the fit, but is large enough to promote sparsity. The optimisation is done with Adam (adaptive moment estimation) [93], a widely-used method for optimisation in machine learning based on gradient descent. We train for 30000 epochs (in each epoch, we use every data point in the training dataset exactly once), and verify that this is sufficient to ensure convergence of

the training algorithm (in all experiments, both training and validation loss have plateaued by the last epoch, that is, the value of the loss function changed by less than 0.01% compared to the previous epoch). Sequential thresholding is applied every 500 epochs, with a threshold of $\epsilon = 0.01$. After the training epochs, 1000 epochs of coefficient refinement are applied, where we minimise the loss function L_1 with λ_1 set to zero, in order to improve the fit of the ODE model to the ABM data.

The datasets are generated by simulating the ABM according to Eq. (5.6). There are many choices to be made when generating the data. For example, the choice of the control function $u(t)$, the number of agents N, the number of simulations to be included in the dataset, and whether smoothing is applied to the data. We will generate a number of datasets with different choices for these factors, and explore the effects of these factors on the success of the equation learning algorithm. For the purpose of this section, we consider the equation learning algorithm successful if the terms retained in the dynamics equation are the same two terms as in the mean-field model Eq. (5.9).

In order to have any chance of learning the correct dynamics, the training data must include data points involving a range of different values of $u(t)$. If $u(t)$ is approximately constant in the data, then any library function involving u would not be distinguishable from those without u. For example, the effects of $g_1(\boldsymbol{z}, \boldsymbol{u}, t) = 1$ and $g_4(\boldsymbol{z}, \boldsymbol{u}, t) = u$ would not be distinguishable, and $\xi_{1,j}$ and $\xi_{4,j}$ will compensate each other as we fit the ODE model to the ABM data, causing practical non-identifiability. We consider two control functions for data generation, namely,

$$u_{\sin}(t) = \sin\left(\frac{2\pi}{T/2}t\right), \quad u_{\text{quad}}(t) = \frac{t^2}{T}. \tag{5.15}$$

These two control functions are chosen because their values vary significantly within the simulation time frame $t \in [0, T]$, with the value of $u_{\text{quad}}(t)$ varying over a larger range compared to $u_{\sin}(t)$. This variation provides more information on the effect of the control on the dynamics of the summary statistics, and we expect more success when using datasets generated with $u_{\text{quad}}(t)$ than with $u_{\sin}(t)$. These two control functions are not meant to be optimal at achieving any control objective.

Having more data is beneficial in any machine learning problem, so we consider a few larger datasets formed by concatenating the time series from multiple ABM simulations together. Noise in the data can mislead the training procedure, causing spurious terms to appear in the learned ODE model. A larger N, or appropriate data smoothing, can reduce noise, so we expect such

measures to improve the rate of success. For simplicity, we will consider smoothing by applying a running average operation to $\boldsymbol{z}_{\mathrm{ABM}}$ with a window width of 100 points, before applying the finite difference scheme to obtain $\dot{\boldsymbol{z}}_{\mathrm{FD}}$.

With these considerations, we generated eight datasets, each partitioned into a training dataset and a validation dataset, summarised in Table 5.1, along with the results of the training. All datasets consist of the time series $\boldsymbol{u}(t)$, $\boldsymbol{z}_{\mathrm{ABM}}(t)$, and $\dot{\boldsymbol{z}}_{\mathrm{FD}}(t)$ at $t = t_i, i = 0, \ldots, n_t$, where $\boldsymbol{z}_{\mathrm{ABM}}(t)$ is computed from individual agent positions using Eq. (5.8), and $\dot{\boldsymbol{z}}_{\mathrm{FD}}(t)$ is computed by applying the up-wind first-order finite difference scheme to $\boldsymbol{z}_{\mathrm{ABM}}(t)$. We train SINDy using the LASSO method five times for each dataset, randomly initialising the coefficients $\xi_{i,j}$ using Xavier initialisation [69], also called Glorot initialisation, which is a commonly used initialisation strategy that picks initial values for the coefficients independently from uniform distributions, with the bounds of the distribution chosen according to certain heuristics [180]. Since GE does not depend on any random factors, unlike LASSO, we train SINDy with GE only once for each dataset.

Dataset	$u(t)$	N	Smoothing	#Train	#Val	LASSO M	LASSO V	GE M	GE V
1	$u_{\sin}$	100	No	1	1	0.4	0	Yes	No
2	$u_{\sin}$	100	No	10	1	0.4	0	Yes	Yes
3	$u_{\sin}$	10000	No	10	1	0.4	0	Yes	Yes
4	$u_{\sin}$	100	Yes	10	1	0	0	Yes	No
5	u_{quad}	100	No	1	1	0.6	0.2	Yes	Yes
6	u_{quad}	100	No	10	1	1	0.8	Yes	Yes
7	u_{quad}	10000	No	10	1	1	1	Yes	Yes
8	u_{quad}	100	Yes	10	1	0.4	0.2	Yes	No

Table 5.1: Summary of the datasets generated by simulating the ABM, and the results of training SINDy with the corresponding datasets. #Train denotes the number of simulations in the training dataset, and #Val denotes the number of simulations in the validation dataset. Each simulation contains $n_t = 10000$ data points. The LASSO M and V columns indicate the success rate (number of successful experiments, divided by five) for SINDy to recover the correct dynamics (i.e., having the same terms as the mean-field model) for $M(t)$ and $V(t)$, respectively, computed from five experiments with random initialisations. The GE columns indicate whether the Greedy Elimination method was able to recover the correct dynamics for M and V.

Table 5.1 shows that the success of the equation learning algorithm depends crucially on the dataset. We observed that when LASSO or GE fails to recover the correct terms in the dynamics, the terms retained varied between trials and datasets. There is little consistency between the dynamics learned by LASSO even in different trials for the same dataset. However, if the dynamics contain the correct terms, their coefficients (an example is given in Eq. (5.16)) are very close to those in the mean-field model (Eq. (5.9)). Comparing the results for Datasets

1-4 with those of Datasets 5-8, we can see that the datasets generated with u_{quad} indeed result in greater success compared to equivalent datasets generated with u_{sin}, as predicted. Comparing the results for Datasets 3 and 7 with those of Datasets 2 and 6 shows that increasing N, the number of agents, makes SINDy more likely to successfully learn the correct governing equations. Comparison between Datasets 5 and 6 suggests that having more data increases the success rate when u_{quad} is used for control, which is the case we expect in theory, but comparing Datasets 1 and 2 show that it had little effect when u_{sin} is used for control. This could be because data generated using u_{sin} are not helpful when distinguishing the effects of the library functions involving u, resulting in practical non-identifiability, and simply having more uninformative data provide little benefit.

Comparing Datasets 4 and 8 with Datasets 2 and 6, we see that smoothing seems to, in fact, harm the performance, which is surprising. A closer examination reveals that applying a running average to $\dot{\boldsymbol{z}}_{\text{ABM}}$ tends to introduce spurious oscillations, as illustrated in Fig. 5.1(b,d). While smoothing the data using a running average achieved almost a 10-fold decrease in the magnitude of the noise, the operation also introduced temporal correlation in the noise, which was previously not present, leading to the spurious oscillations. The oscillations, in turn, may have caused the algorithm to incorrectly retain terms in order to overfit to the noise-induced spurious oscillations. This observation demonstrates that applying naive denoising methods to the data, such as smoothing with a running average, can do more harm than good. We have chosen this de-noising method for its simplicity. Identifying an appropriate de-noising method is a topic of active research, but it is outside the scope of this chapter. Some promising de-noising methods are discussed in [107], such as using cubic splines, or ones based on using neural networks to represent the data.

Another observation is that it is much harder for SINDy to learn the correct dynamics for V compared to M. This is mostly due to the fact that $\dot{V}_{\text{FD}}$ has a much higher noise level compared to $\dot{M}_{\text{FD}}$, which causes the algorithm to overfit to the noise. This agrees with the predictions in Eqs. (5.12) and (5.14), and can be clearly seen by comparing the vertical spread of the data points in Fig. 5.2(a,b).

GE could recover the correct dynamics for M using any dataset, but failed for $V(t)$ using Datasets 1, 4, and 8. The success of the two training methods is correlated, as Datasets 1, 4, and 8 also lead to the lowest success rates for LASSO. GE not only seems to succeed more often than LASSO, it is also much less computationally expensive. GE requires no more than 10

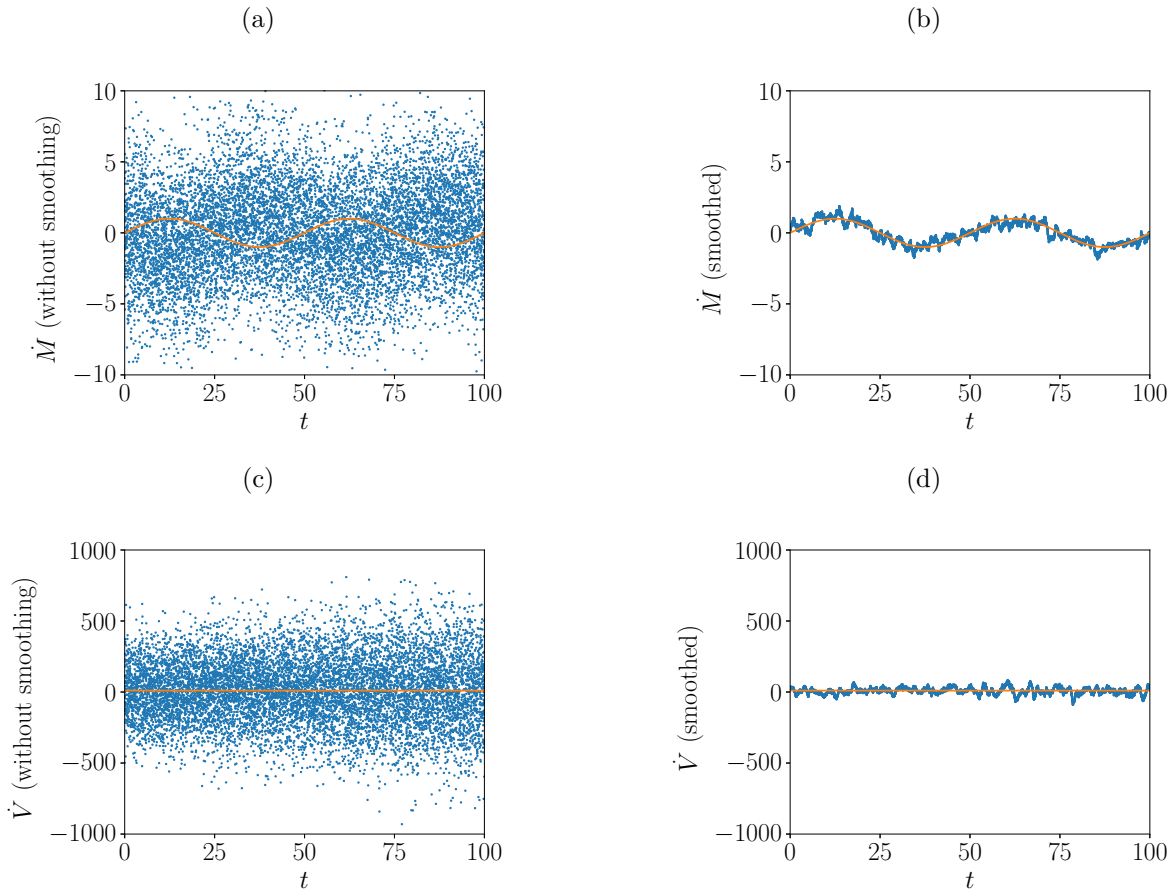


Figure 5.1: The effect of applying smoothing to data. The data presented in these plots come from the first ABM simulation in Dataset 5, which uses a sinusoidal $u_{\sin}(t)$ as the control. The blue dots in each figure represent: (a) $\dot{M}_{\mathrm{FD}}(t)$ calculated from the ABM simulation using finite differences without applying any smoothing; (b) Same as (a), but $M_{\mathrm{ABM}}(t)$ was smoothed first by applying a running average operation with a window width of 100 points (equivalent to a time interval of $100\Delta t = 1$), before applying the finite difference scheme to calculate $\dot{M}_{\mathrm{FD}}$; (c) $\dot{V}_{\mathrm{FD}}(t)$ computed without any smoothing; (d) $\dot{V}_{\mathrm{FD}}(t)$ computed with smoothing. In all figures, the orange curves represent $\dot{M}_{\mathrm{MF}}$ or $\dot{V}_{\mathrm{MF}}$ given by the mean-field ODE, Eq. (5.9). Notice the difference in the scales of the vertical axes, indicating a much greater spread from the theoretical mean (orange curves) in the non-smoothed data. However, there is a clearly visible high-frequency oscillation present in the smoothed data (the blue dots in (b) and (d)), which is not visible in the non-smoothed data.

seconds [1] for any dataset, compared to LASSO, which requires around 2 hours for Datasets 1, 5, and around 20-25 hours for Datasets 2–4 and 6–8 (which are larger than Datasets 1 and 5, with 10 times more data points). The computational complexity of both methods is asymptotically linear in the number of data points, so we expect the relation between their computational cost to hold even for much larger datasets. Despite the clear advantage of GE, it is not generalisable as it depends on the fact that the loss function can be minimised using linear least squares,

[1] All computations were carried out on a computer with a 12th Gen Intel Core i5-12500T CPU, with 12 cores and a clock speed of 4.4 GHz. The codes are written in Python.

which will no longer be the case as we extend the method further in Sec. 5.4.

To examine the outcome of equation learning more closely, we consider the following two ODE models learned from Dataset 2,

$$\begin{cases} \dot{M}^{(1)}_{\text{Learn}}(t) = -1.0042u, \\ \dot{V}^{(1)}_{\text{Learn}}(t) = 9.2541. \end{cases} \tag{5.16}$$

$$\begin{cases} \dot{M}^{(2)}_{\text{Learn}}(t) = -1.0046u - 0.0120u^2, \\ \dot{V}^{(2)}_{\text{Learn}}(t) = 8.0215 + 0.0172M + 2.5648u^2. \end{cases} \tag{5.17}$$

The first, in Eq. (5.16), is learned using GE, and has the correct terms, with the coefficients close to the mean-field model, Eq. (5.9), repeated here:

$$\begin{cases} \dot{M}_{\text{MF}}(t) = -u, \\ \dot{V}_{\text{MF}}(t) = 10. \end{cases} \tag{5.18}$$

The small discrepancy between the coefficients is due to the fact that the ABM simulation is random, and the process noise means the dynamics of the summary statistics might not exactly reflect the theoretical mean-field model. The validation loss for this model, that is, the function L_1 as defined in Eq. (5.5) evaluated using the validation dataset and the learned parameters in this model, is $L_1 = 26479$ (the value of the loss is not interesting in itself, its main purpose is to allow comparison of goodness-of-fit between models). The second model in Eq. (5.17) is learned using LASSO, and does not have the correct terms. Nonetheless, its validation loss is $L_1 = 27351$, only marginally worse than the first model. The two learned models are the result of fitting a larger, parent model with the full set of library functions to the data, and can be seen as two sets of parameter values for the parent model, with the parameters being the coefficients $\xi_{i,j}$. The fact that the parent model can fit the data similarly well with two distinct parameter sets implies that practical non-identifiability is present, which is to be expected since the parent model contains a large number of parameters that can compensate for each other.

In Fig. 5.2(a,b), we plot $\dot{\boldsymbol{z}}(t)$ given by both learned models, as well as the ABM data $\dot{\boldsymbol{z}}_{\text{FD}}(t)$. Notice that the dynamics given by both models appears almost indistinguishable, as their corresponding curves lie on top of each other. In Fig. 5.2(c,d), we plot $\boldsymbol{z}_{\text{Learn}}(t)$, as computed with the dynamics given by both models using a standard numerical ODE integrator, along with the ABM data. For M, despite the addition of the incorrect u^2 term, the trajectory $M^{(2)}_{\text{Learn}}(t)$ predicted by Model 2 is very close to that of Model 1. The predictions from both models deviate increasingly from the data for later times, which is to be expected as errors accumulate during the integration. Nonetheless, both are good approximations of the data. For V, while Model 1's prediction approximates the data closely, Model 2's prediction is wildly off. This observation

shows that even if a model can apparently fit the dynamics (i.e., the first derivative) well, it might still give very poor predictions for the trajectories, which is a critical drawback to SINDy, and other equation learning methods that fit a model to derivatives of the data rather than the trajectories. Due to this vulnerability, we must remember to verify the predictions of both the dynamics and the trajectories of the learned model against the data, and if a significant discrepancy is found, we should generate a larger dataset and re-train the model.

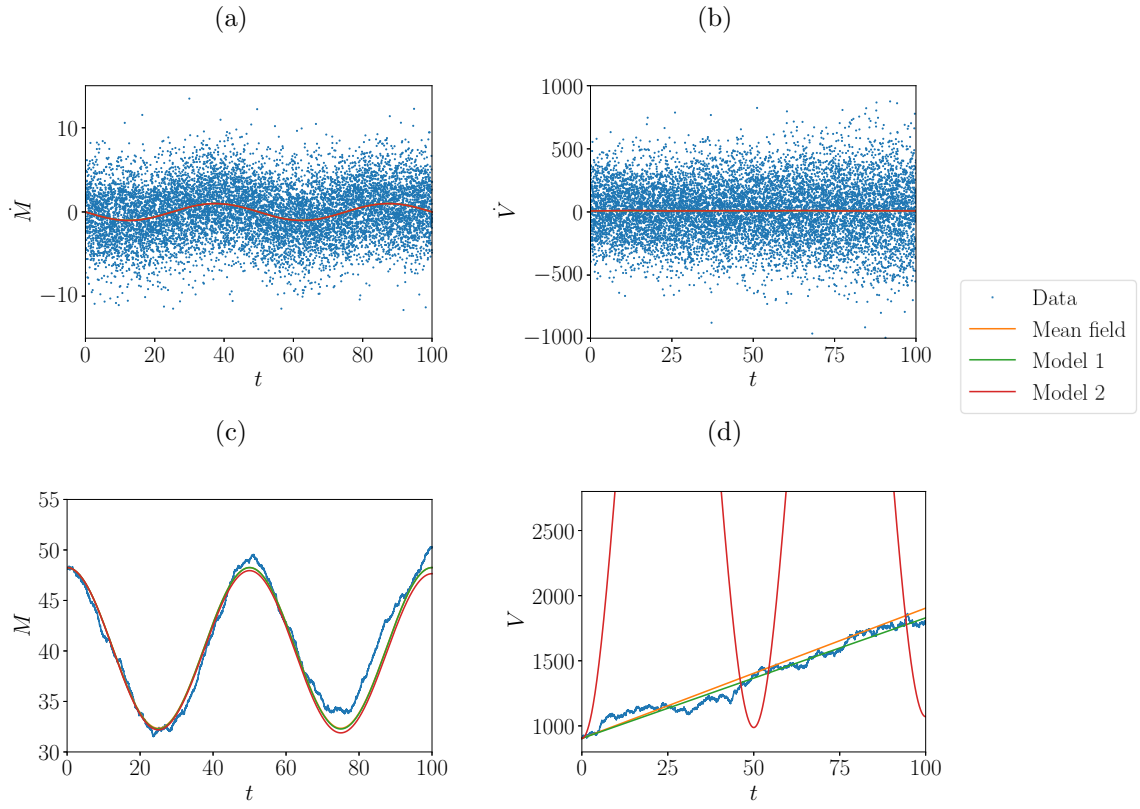


Figure 5.2: Comparison of the dynamics and trajectories of the summary statistics predicted by the mean-field model, $\boldsymbol{z}_{\text{MF}}$ (orange curve), given in Eqs. (5.9) and (5.10), and the two learned models, $\boldsymbol{z}^{(1)}_{\text{Learn}}$ (green curve) given in Eq. (5.16), and $\boldsymbol{z}^{(2)}_{\text{Learn}}$ (red curve) given in Eq. (5.17), with the data $\boldsymbol{z}_{\text{ABM}}$ (blue dots) from the first ABM simulation in Dataset 2. In (a) and (b), the dynamics $\dot{\boldsymbol{z}}$ given by the three models are indistinguishable as their corresponding curves lie on top of each other. In (c), the trajectories of M predicted by the three models are very close to each other, and close to the data. The discrepancies between the model predictions and the data are initially small in the simulation, but grow as t increases. In (d), the trajectories of V predicted by the mean-field model and Model 1 are close to the data and each other, while those of Model 2 diverge significantly from the rest.

5.3.2 Applying control

Having obtained surrogate ODE models for our ABM, we are finally ready to apply control. For simplicity, suppose that our goal is to push all agents towards the left as much as possible, while expending minimum effort on the control. In the optimal control framework (see Appendix C),

this goal can be captured by the following Lagrangian,

$$\mathcal{L}(\boldsymbol{x}(t), u(t), t) = \frac{1}{N}\sum_{i=1}^{N} x_i(t) + \alpha u(t)^2, \tag{5.19}$$

where $\alpha > 0$ is the weight of the cost of applying the control. The first term encodes the objective of pushing the agents toward the left, and the second term is a quadratic term that encodes the cost of applying the control. A quadratic cost term is commonly used in optimal control studies, since it makes later computations easier. This Lagrangian can be expressed in terms of the summary statistics as

$$\mathcal{L}(\boldsymbol{x}(t), u(t), t) = \tilde{\mathcal{L}}(\boldsymbol{z}(t), u(t), t) = M(t) + \alpha u(t)^2. \tag{5.20}$$

We will consider three surrogate models for applying control: the mean-field model (Eq. (5.18)), and the two learned models (Eqs. (5.16) and (5.17)). In this case, the control objective does not involve V, and the dynamics of M are not coupled to V in either Model 1 or Model 2, so we will ignore the V components of all models for the rest of this section. Model 2's prediction for M is reasonable, so it is hopeful that it can provide a reasonably good control in this case, despite the fact that it does not contain the expected terms.

Intuitively, turning on the control early in the simulation should be more effective at reducing the cost function than turning it on later, since the cost is calculated by integrating the Lagrangian over the entire time frame, so reducing M early on should have a greater impact on the cost. Hence, we expect the optimal control to be high near $t = 0$, and slowly ramp down to zero as $t \to T$. This means we expect both $u_{\text{sin}}(t)$ and $u_{\text{quad}}(t)$ to perform poorly. The optimal control problems for both the mean-field model and the two learned models are simple enough to be calculated analytically by hand. Applying PMP (introduced in Sec. C.1) to the mean-field model yields the following optimal control

$$u_{\text{MF}}(t) = \frac{T - t}{2\alpha}. \tag{5.21}$$

Alternatively, under the mean-field model, if we restrict $u(t)$ to be a constant, then we can compute the cost function explicitly, so the optimal constant control, u_{cons}, can be found without applying PMP, to be

$$u_{\text{cons}}(t) \equiv \frac{T}{4\alpha}, \tag{5.22}$$

which can serve as an easy-to-obtain benchmark for comparing the other candidates for the optimal control. Applying PMP to the learned Model 1 gives

$$u^{(1)}_{\text{Learn}}(t) = \frac{1.0042(T - t)}{2\alpha}, \tag{5.23}$$

which is very similar to u_{MF}, as the dynamics of the mean-field model and Model 1 are very close, so we expect their performance to be similar. PMP applied to Model 2 leads to a degenerate case where a singularity is present in the control at $t^* = T - \alpha/0.012$. For $t < t^*$, the optimal control $u(t)$ is infinite. This is because the u^2 term in Model 2's M equation means that the reduction in M by using a larger control u always out-weighs the increase in control cost (the u^2 term in $\mathcal{L}$). Such a situation is made possible since both cost terms are of the same order in u.

Having the control approaching infinity is unrealistic in any biological application. To circumvent this issue, we impose an upper bound on the control, chosen arbitrarily to be T/α, which is larger than, but around the same order of magnitude as u_{cons}. The resulting control function is

$$u^{(2)}_{\text{Learn}}(t) = \begin{cases} \dfrac{T}{\alpha} & 0 \le t \le T - \dfrac{\alpha}{0.012}, \\ \min\left(\dfrac{T}{\alpha}, \dfrac{1.0046(T-t)}{2(\alpha - 0.0120(T-t))}\right) & t > T - \dfrac{\alpha}{0.012}. \end{cases} \tag{5.24}$$

Although Eq. (5.24) appears very different from Eq. (5.21), it is easy to see that the behaviour of Eq. (5.24) becomes closer to that of Eq. (5.21) as α increases.

While $u_{\text{MF}}(t)$ and $u^{(1)}_{\text{Learn}}(t)$ are, respectively, optimal controls for two surrogate ODE models (and $u^{(2)}_{\text{Learn}}(t)$ is modified from the optimal control to another surrogate model), they are not necessarily optimal for the original ABM. The final task is therefore to transfer these candidate controls back to the original ABM and evaluate them. For each control function we consider, we simulate the ABM 1000 times with $N = 100$, and report the mean and standard deviation of the cost in Table 5.2. We do this for both $\alpha = 1$ and $\alpha = 10$, mostly to see if $u^{(2)}_{\text{Learn}}(t)$ performs better for larger α.

Control	$\alpha = 1$	$\alpha = 10$
$u \equiv 0$	4724 ± 184	4724 ± 184
u_{sin} (Eq. (5.15))	3983 ± 182	4422 ± 193
u_{quad} (Eq. (5.15))	121408 ± 184	1921867 ± 176
u_{cons} (Eq. (5.22))	-57794 ± 185	-1531 ± 184
u_{MF} (Eq. (5.21))	-78609 ± 185	-3603 ± 186
$u^{(1)}_{\text{Learn}}$ (Eq. (5.23))	-78612 ± 180	-3600 ± 188
$u^{(2)}_{\text{Learn}}$ (Eq. (5.24))	107412 ± 185	-3523 ± 181

Table 5.2: The mean and standard deviation of the cost of the ABM control problem with various control functions, for the cases $\alpha = 1$ and $\alpha = 10$. A lower (more negative) cost is more desirable.

Notice that the null control ($u \equiv 0$), $u_{\sin}$ and u_{quad} did not perform well, as we expect. The constant control u_{cons} performed much better than the aforementioned controls, but it is out-performed by u_{MF} and $u^{(1)}_{\text{Learn}}$, which achieved the lowest cost overall, in both cases, and are almost indistinguishable from each other. Finally, $u^{(2)}_{\text{Learn}}$ performed very badly in the $\alpha = 1$ case, but it performed well in the $\alpha = 10$ case, and only marginally behind u_{MF}. This observation shows that the performance of a surrogate model can depend on factors other than its ability to fit the data (it depends on α in this case), and a minor difference in model equations (the additional u^2 term in $u^{(2)}_{\text{Learn}}$, which has a relatively small coefficient) can result in a control that performs much worse after transferring to the original ABM, even if it does not have a large effect on the model fit.

We also note that the singularity in $u^{(2)}_{\text{Learn}}$ can be avoided by choosing a different cost function $\mathcal{L}$, for example replacing the u^2 term with u^4 would ensure that u tends to infinity is not an optimal control. The highest order of u in the learned model (Eq. (5.17)) is quadratic, so a u^4 term will dominate $\mathcal{L}$ in that limit and cause the cost to approach positive infinity. Non-sensible controls can be produced from a flawed surrogate model in many ways, and avoiding them by selecting the cost function would need to be done on an ad-hoc basis.

It is important to note that these results are not proof that u_{MF} or $u^{(1)}_{\text{Learn}}$ are the optimal controls for the ABM. Nonetheless, our study shows that a properly learned surrogate model can provide a control that performs reasonably well against controls chosen more naively, such as u_{cons}, and therefore is a promising way to find reasonably good control for more complicated ABMs, even if there is no guarantee that the resulting control is the optimal.

5.4 Further automating the equation learning method

We have seen that the equation learning method described in Sec. 5.2 and demonstrated in Sec. 5.3 can provide a surrogate model that leads to reasonably well-performing controls for the ABM, provided sufficient data are available. However, in the future, we envision that the method can be further developed into an automated, generalisable method that can be used to apply control to a broad range of ABMs, while requiring minimal user input. In this section, we outline a road-map to achieve this goal, by dividing the further developments of the method into three tasks. The first task is to learn to compute a set of specified summary statistics, while the second is to learn to express the Lagrangian in terms of these summary statistics. These two tasks lay the foundation for the final task, which involves extending the machine learning

algorithm so that it can automatically choose a set of summary statistics, without requiring the user to specify them.

5.4.1 Learning to compute summary statistics

The method we have discussed so far in Sec. 5.2 requires the user to specify a set of summary statistics, $\boldsymbol{z}$, and supply its time series to the machine learning algorithm. It would be advantageous if we could train a machine learning algorithm to compute the summary statistics automatically, thereby reducing the amount of effort required from the user. While in many cases computing the summary statistics is not difficult, the ability to do so automatically will be necessary for the final step of extending the method, as will be explained in Sec. 5.4.3, which is to allow the machine learning algorithm to automatically pick the set of summary statistics without requiring the user to specify them. The task of computing the summary statistics is independent from the task of learning their dynamics, so the machine learning algorithm discussed in this subsection is separate from the SINDy component in Sec. 5.2. However, these two components will be coupled together in Sec. 5.4.3.

The task of this subsection can be seen as finding a function $\phi(\boldsymbol{x}) = \boldsymbol{z}$ that receives the ABM data $\boldsymbol{x}$ as inputs, and outputs the summary statistics, $\boldsymbol{z}$. In the example in Sec. 5.3, the function to be found is ϕ given in Eq. (5.8). For this discussion, we will assume that the ABM contains only a single type of agent. If multiple types of agents are present in the model, then the summary statistics can be duplicated for each type of agent. Any summary statistic for an ABM should satisfy the following two properties. First, the summary statistics must be able to be computed regardless of the number of agents. Since the number of agents might increase or decrease during a simulation, ϕ must accept input of variable dimensions. This precludes representing ϕ as a simple multi-layer perceptron, which has an input layer of fixed width, or a combination of simple library functions that depends on a fixed number of inputs. Second, the summary statistics should be invariant with respect to re-labelling the agents, that is,

$$\phi(\boldsymbol{x}) = \phi((\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)) = \phi(\sigma(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)),$$

for any permutation σ. This is because all agents represent the same type of object (e.g., cells or organisms), so their ordering should not matter. The invariance property also implies that each agent should contribute to the summary statistics in the same way.

We propose the following representation for ϕ, which is mostly inspired by CNN. It shares similarities with algorithms proposed in [101, 211], as both algorithms aim to achieve permutation invariance in different ways. The representation has three parts. The first part is called the convolution layers, based on its similarity to convolutional layers in CNNs. These layers apply the transformation ϕ_c, represented as a multi-layer neural network, that maps $\boldsymbol{x}_i$, the state of each agent, to $\tilde{\boldsymbol{x}}_i$, which contains m_c features describing the state of a single agent. The data for each agent are transformed using the same function ϕ_c independently from the data for other agents. The features will be trained so that they will be the most helpful for computing the summary statistics. The second part is a single linear aggregation layer ϕ_a, that takes the $\tilde{\boldsymbol{x}}_i$'s as input and combines them to form the input to the next layer, denoted $\boldsymbol{y}$. The weights W_a for the inputs to ϕ_a are shared in a way such that the contribution from each agent is the same. The third part, ϕ_f, is comprised of multiple fully connected layers that take $\boldsymbol{y}$ as input, and output the summary statistics. These three parts can be written as follows:

$$\tilde{\boldsymbol{x}}_i = \phi_c(\boldsymbol{x}_i) = W_c^{(l_c)}\varphi(\cdots\varphi(W_c^{(2)}\varphi(W_c^{(1)}\boldsymbol{x}_i + \boldsymbol{b}_c^{(1)}) + \boldsymbol{b}_c^{(2)}) + \cdots) + \boldsymbol{b}_c^{(l_c)}, \quad \text{(convolutional layers)} \tag{5.25}$$

$$\boldsymbol{y} = \sum_{i=1}^{N} \varphi(W_a\tilde{\boldsymbol{x}}_i) + \boldsymbol{b}_a, \quad \text{(aggregation layer)} \tag{5.26}$$

$$\boldsymbol{z} = \phi_f(\boldsymbol{y}) = W_f^{(l_f)}\varphi(\cdots\varphi(W_f^{(2)}\varphi(W_f^{(1)}\boldsymbol{y} + \boldsymbol{b}_f^{(1)}) + \boldsymbol{b}_f^{(2)}) + \cdots) + \boldsymbol{b}_f^{(l_c)}, \quad \text{(fully connected layers)}, \tag{5.27}$$

where φ is a nonlinear activation function (popular choices include sigmoid, tanh or ReLU (rectified linear unit)), l_c and l_f are the number of convolutional and fully connected layers, respectively, W and $\boldsymbol{b}$ are the weights and biases, respectively, with the subscripts indicating from which of the three parts of the overall network these parameters come. It is easy to see that ϕ, defined in this way, can accept inputs from an arbitrary number of agents, and is invariant with respect to permuting the order of agents.

The neural network for ϕ can be trained simultaneously with SINDy, which gives us the following loss function to be minimised during training:

$$L_2(W, \boldsymbol{b}) = \lambda_2||\phi(\boldsymbol{x}) - \boldsymbol{z}||_2^2 + \lambda_3(||W||_F^2 + ||\boldsymbol{b}||_2^2), \tag{5.28}$$

where λ_2 and λ_3 are scalar weights for the loss for learning the summary statistics and regularisation for the parameters in the neural network for ϕ, respectively. W and $\boldsymbol{b}$ are short-hands, respectively, for all weights and biases across all layers, and the sum is over all layers in the three parts of the neural network.

5.4.2 Learning to express the Lagrangian in terms of the summary statistics

The surrogate model learned by the machine learning algorithm so far is in terms of the summary statistics, $\boldsymbol{z}$. However, in a control problem for ABMs, the Lagrangian, $\mathcal{L}$, which is the integrand of the objective functional, J (Eq. (5.1)), is expressed in terms of the state variables of the agents, $\boldsymbol{x}$. In order to apply control to the surrogate model, we need to express the Lagrangian in terms of $\boldsymbol{z}$, i.e., we need to find an expression $\tilde{\mathcal{L}}$ such that

$$\tilde{\mathcal{L}}(\boldsymbol{z}(\boldsymbol{x}(t)), \boldsymbol{u}(t), t) = \mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t).$$

In the example in Sec. 5.3, we manually reformulated the Lagrangian in terms of $\boldsymbol{x}$ (Eq. (5.19)) to be in terms of $\boldsymbol{z}$ (Eq. (5.20)). While this was easy to do manually for a simple ABM, it would be burdensome for more complicated ABMs. Furthermore, in the final stage of extending the method, as will be explained in Sec. 5.4.3, the machine learning algorithm will choose the summary statistics automatically, and the user will not know the form of the chosen $\boldsymbol{z}$ in advance. In this case, it is necessary for the machine learning algorithm to be able to express $\mathcal{L}$ in terms of $\boldsymbol{z}$ without user input.

To accomplish this task, we can use another SINDy module that operates independently from the components of the machine learning algorithm described so far. The inputs to this module are the time series of the summary statistics $\boldsymbol{z}(t)$, the control $\boldsymbol{u}(t)$ used to generate the data, and a time series of the Lagrangian $\mathcal{L}(t)$, calculated manually from ABM data. The output is an expression for the Lagrangian in terms of $\boldsymbol{z}$, in the form of a linear combination of library functions, g_i,

$$\mathcal{L}(t) \approx \tilde{\mathcal{L}}(\boldsymbol{z}(t), \boldsymbol{u}(t), t) = \sum_{i=1}^{s} \zeta_i g_i(\boldsymbol{z}(t), \boldsymbol{u}(t), t),$$

where ζ_i is the coefficient of the i^{th} library function to be learned by the algorithm. This module can be trained using the LASSO method discussed in Sec. 5.2, with the following loss function:

$$L_3(\boldsymbol{\zeta}) = \lambda_4 \sum_{i=1}^{n_t} |\mathcal{L}(\boldsymbol{x}(t_i), \boldsymbol{u}(t_i), t_i) - \tilde{\mathcal{L}}(\boldsymbol{z}(t_i), \boldsymbol{u}(t_i), t_i)|^2 + \lambda_5 ||\boldsymbol{\zeta}||_1, \tag{5.29}$$

where λ_4 and λ_5 are scalar weights for the fit loss term, and the l^1 regularisation term, respectively.

5.4.3 Automatic feature discovery for selecting summary statistics

The final piece of the puzzle towards developing an automated method to control an ABM is to have the machine learning algorithm choose a set of summary statistics, instead of requiring the user to specify them. In the example in Sec. 5.3, we chose the mean position of the agents as a summary statistic, whereas now, we would like the machine learning algorithm to make that choice by itself.

The necessary ingredients for such an algorithm are already discussed in previous subsections. Once a set of summary statistics, $\boldsymbol{z}$, is chosen, we have a CNN-like neural network for ϕ that computes $\boldsymbol{z}$ in terms of $\boldsymbol{x}$, a SINDy component that learns the dynamics of $\boldsymbol{z}$, and another SINDy component that learns an expression of the Lagrangian in terms of $\boldsymbol{z}$, thereby producing a surrogate model and a control problem in terms of $\boldsymbol{z}$, ready to be solved with PMP. However, the final remaining task of automatically choosing the summary statistics is difficult, since there are few clues for what these summary statistics can be. A useful idea is to take inspiration from the Autoencoder-SINDy algorithm in [40], couple the three previously independent components of the machine learning algorithm together, and train them simultaneously, which may provide sufficient clues for choosing the summary statistics.

In the Autoencoder-SINDy architecture, the choice of summary statistics is constrained by two things. The first is that there must exist an approximate inverse function ψ to ϕ, such that the original input data can be approximately reconstructed from $\boldsymbol{z}$, that is, $\psi(\boldsymbol{z}) \approx \boldsymbol{x}$. This ensures that the summary statistics contain enough information from the original data. The second is that the dynamics, i.e., the governing equations for $\boldsymbol{z}$ learned by the SINDy part of the algorithm, must have sparse representation in $\boldsymbol{z}$. This selects for coordinate representations that would actually be useful. Without sparsity, the problem for finding ϕ is very under-constrained, as the identity function would work for both ϕ and ψ.

The autoencoder idea is not directly applicable to data generated from ABMs. In most problems where we might want to apply control, we are interested in controlling some macro-scale behaviour of the system. The raw data from the ABM contain mostly micro-scale information about the individual agents, which would be irrelevant for applying control. If the micro-scale information is retained in the summary statistics $\boldsymbol{z}$, then the dynamics of $\boldsymbol{z}$ would likely be too complicated to have a sparse representation. However, if the micro-scale information is discarded, then we cannot expect to be able to reconstruct $\boldsymbol{x}$ from $\boldsymbol{z}$. We have implicitly chosen to discard the micro-scale information in stage 2, when we required the function ϕ to be invariant under

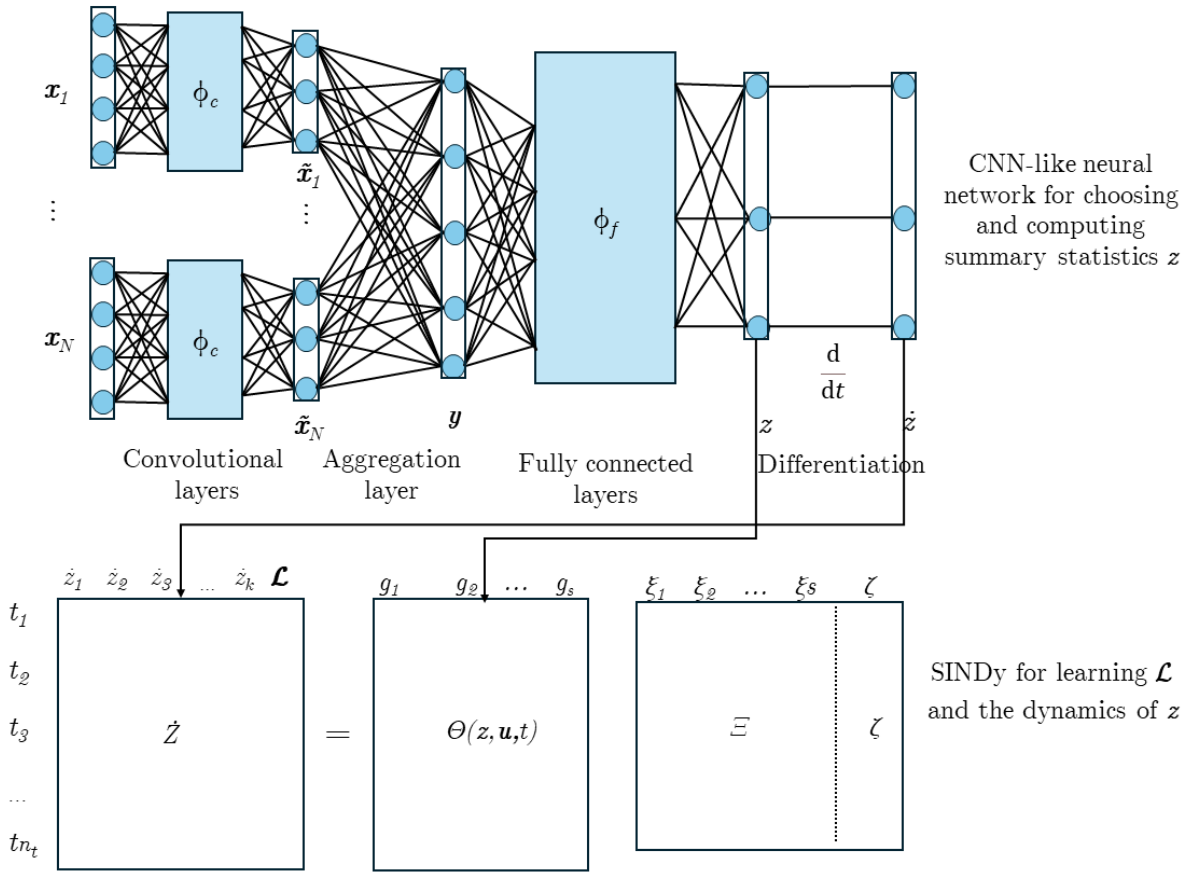


Figure 5.3: Schematic of the proposed machine learning architecture, consisting of a neural network for calculating the latent variables, as well as two extended SINDy-like structure for learning the dynamics of the latent variables and the Lagrangian in terms of the latent variables. The neural network itself is composed of multiple convolution-like layers that transform the state variables for an individual agent, a combination layer that combines the inputs from all agents, and fully connected layers to output the latent coordinate representation.

permutation of the agents. This is because most control objectives are concerned with macro-scale behaviours, so retaining the micro-scale information would only needlessly distract from, and over-complicate, the problem. This means we must abandon the hope of reconstructing $\boldsymbol{x}$ from $\boldsymbol{z}$, and lose one important source of constraints (and therefore, hints) for what $\boldsymbol{z}$ could be. However, compared to the problem in [40], we are additionally requiring the Lagrangian $\mathcal{L}$ to have a sparse representation in terms of $\boldsymbol{z}$, which can act as a constraint that replaces the reconstruction criterion we lost.

The final machine learning algorithm consists of the three components described in Secs. 5.2, 5.4.1, and 5.4.2, respectively, now coupled together through $\boldsymbol{z}$. The loss function to be minimised during training is formed by combining the loss functions for the three components (Eqs. (5.5),

(5.28)), and (5.29)),

$$L_4(\Xi, W, \boldsymbol{b}, \boldsymbol{\zeta}) = L_1(\Xi) + L_2(W, \boldsymbol{b}) + L_3(\boldsymbol{\zeta}).$$

The overall structure of the machine learning algorithm is summarised in Fig. 5.3. In [40], the authors found that training the autoencoder component and the SINDy component in an Autoencoder-SINDy simultaneously enabled the machine learning algorithm to choose a set of summary statistics that yield a sparse representation of the dynamics. The success of SINDy in earlier papers suggests that, for learning surrogate models to ABMs, it can be hoped that training the three components simultaneously will enable the algorithm to identify a set of summary statistics that not only have a sparse representation of their dynamics, but can also act as a vehicle for applying control to the ABM.

5.5 Discussion

In this chapter, we explored the approach of controlling an ABM by applying control to a surrogate model obtained via machine learning. We adapted SINDy to learn an ODE model for summary statistics of ABM simulations, and demonstrated its usage by applying it to a simple toy ABM for motile cells. In our demonstration, we showed that given sufficient data, the algorithm is capable of recovering the mean-field ODE describing the dynamics of the summary statistics. However, in most realistic cases, we expect that the model being fitted to the data by the equation learning algorithm, which contains a large number of free parameters, to be practically non-identifiable given a tractable amount of data. In such cases, the algorithm may produce a surrogate model with terms that are different from the mean-field model. Such a phenomenological surrogate model, despite being able to approximately fit the dynamics of the summary statistics, can nonetheless fail to approximate their trajectories, as seen in Fig. 5.2. Even when the surrogate model can approximate both the dynamics and trajectories well, applying PMP to it can still result in a control that is far from optimal for the original ABM. The potential for such outcomes underscores the importance of validating the control derived from the surrogate model by transferring it back to the ABM, which allows for an evaluation and comparison with simple control candidates to ensure its effectiveness. The topic of parameter identifiability in the context of equation learning should be explored in the future.

We found that simple denoising procedures, such as taking a running average of the time series of the summary statistics, can have significant drawbacks. Nonetheless, significant process noise is present in most ABMs, which means a more appropriate denoising method should be

able to improve the performance of the equation learning algorithms. Determining the extent to which denoising can impact equation learning in the context of ABMs, and finding suitable denoising methods, should be pursued in future research.

An issue with any machine learning-based method is that the outcome is dependent on hyper-parameters (such as the number and width of hidden layers, and the weights of the terms in the loss function), which often requires fine-tuning. The outcome is also dependent on the methodological choices made regarding the various aspects of the training algorithm. The impact of these choices in the context of equation learning should be investigated in more detail in the future.

We also proposed further extensions to this method, so that it can automatically choose a set of summary statistics without user intervention. By training a CNN-like neural network to compute the summary statistics, and SINDy modules that learn the dynamics and the control objective in the form of the Lagrangian at the same time, we hope the algorithm will be able to identify an appropriate set of summary statistics that will be useful for applying control to the ABM. We leave the implementation of these extensions to future work.

We used the SINDy algorithm for the equation learning components of the method developed in this chapter. SINDy has inspired a family of extensions and related algorithms, such as PDE-FIND [163], the analogous method for learning PDEs, and variants for learning the governing equations of stochastic dynamical systems [24, 80, 136]. In some cases it might be more natural to formulate a surrogate model for an ABM in terms of PDEs or SDEs rather than ODEs, which makes these SINDy variants suitable.

Biologically-Informed Neural Networks (BINNs) [108], an equation learning algorithm inspired by Physics-Informed Neural Networks (PINNs) [151], is another promising alternative to SINDy. The BINNs method aims to learn an interpretable PDE model from spatial-temporal data arising from a biological system. The right-hand sides of the PDEs are assumed to take a certain general form, e.g., reaction-diffusion equations, with the unknown parts of the model (e.g., the reaction function and diffusion coefficient) represented by multi-layer neural networks. The parameters in these neural networks are trained so that the dynamics described by the model approximates the data.

The BINNs method is suitable if the general form of the PDE is known. It has been demonstrated in [139] that, if a suitable general form of the PDE model is provided, then the PDE model learned by BINNs can do well in approximating the spatio-temporal dynamics of the

summary statistics of an ABM, such as the density of agents, and in making predictions. The learned model may even out-perform the formally derived mean-field model at predicting the behaviour of the ABM in certain cases, which makes it a good candidate for a surrogate model for applying control. By using neural networks to represent the model equations, BINNs can be more flexible than SINDy, which restricts the space of models fitted to ABM data to linear combinations of a finite number of library functions.

The original BINNs method learns a PDE model for agent density, and requires the user to specify a general form of the PDE model. In some ABMs, agents have state variables other than position. For example, in a model for the spread of infectious disease with agents representing individual people, each agent can have additional state variables describing their viral load or immune system status. Approximating the dynamics of the ABM in such cases requires a PDE surrogate model not only for agent density, but also for other quantities. However, determining which additional quantities to model with PDEs, and the form of those PDEs, may not be straight-forward. Extending the BINNs method to learn surrogate models for these types of ABMs could involve adding a machine learning component that automatically chooses the additional quantities to be modelled with PDEs, similar to the neural networks proposed to calculate the summary statistics.

We have shown that the surrogate model obtained through equation learning might not be a good representation of the dynamics of the original ABM over a long time horizon. This can be seen in Fig. 5.2, where the solutions of the surrogate model increasingly deviate from the summary statistics of the ABM as time progresses. This deviation casts doubt as to whether the optimal control for the surrogate model remains near-optimal for the ABM or the underlying biological system for later times, which is especially a concern if the time horizon of the control problem, T, is large. Model predictive control (MPC) is a possible method to address this concern. An overview of MPC is given in [75]. Under the MPC framework, we divide the time horizon $[0, T]$ into N sub-intervals of length τ. For each sub-interval $[n\tau, (n+1)\tau]$, where $n = 0, \ldots, N-1$, we take measurements of the biological system at the beginning of the sub-interval, $t = n\tau$, and use the measurements as initial conditions of the surrogate model to compute an optimal control on the time horizon $[n\tau, T]$. This control strategy is only implemented until $t = (n+1)\tau$, at which point we re-measure the biological system and compute a new control strategy to be used over the next sub-interval. Using the MPC approach, we periodically re-calibrate the surrogate model, with prevents the discrepancy between the surrogate model and

the real biological system from growing too large.

Aside from equation learning, model-free reinforcement learning is another possible approach to control ABMs through machine learning. Within the framework of model-free reinforcement learning methods, we parameterise the control strategy as a neural network, then optimise the objective function with respect to the weights and biases through trial-and-error [182, Ch. 8], where the objective function is evaluated for many trial control strategies by simulating the ABM. Such methods have no restriction on the form of the model, since the model is only used for evaluating the objective function. The likely disadvantage is that a large number of ABM simulations might required for reinforcement learning to yield a good control strategy, as measured by the objective function. It remains to be seen how the computational cost of using a reinforcement learning algorithm to control an ABM compares with the equation learning approach in conjunction with PMP, and how well the resulting control strategy performs as measured by the objective function.

Equation learning is a powerful tool that can contribute significantly toward the goal of applying control to ABMs, as demonstrated in this chapter. While this chapter provides a proof-of-concept for using equation-learning methods to control ABMs, the method as implemented so far in this chapter is still in its early stages of development. Further integration of the equation learning algorithms with control theory will make applying control to ABMs more practical in the future.

Chapter 6

Conclusions and future work

The inherent complexity of biological processes presents numerous mathematically interesting challenges at every stage of study, spanning from experimental design and data generation, to parameter inference and model selection, and further to exploration and analysis of model behaviours, as well as the application of control. Throughout this thesis, we have tackled a variety of challenges encountered in each of these steps. We have addressed the issue of assessing and improving the practical parameter identifiability of a model, from both the viewpoint of model selection and experimental design. We then explored a complex model that makes the application of many traditional methods of analysis difficult, and identified a mechanism suitable for applying control to this model. Finally, we worked toward developing a method for applying optimal control for ABMs, which have the potential to avoid the shortcomings of existing methods. These results contribute toward the aims of ensuring the accuracy of a model, improving our understanding of biological mechanisms, and enabling the application of control toward a wide range of biological systems.

6.1 Summary of contributions

In Chapter 2, we investigated factors impacting the practical parameter identifiability of a PDE model, and the connections of parameter identifiability to model selection, using PDE models of cell invasion as an example. We showed that certain models require more data to be practically identifiable. The minimum amount of data required for identifiability tends to be higher for more complicated models, but the threshold cannot be determined by looking at the number of parameters alone. Given a set of experimental data, we should perform identifiability analysis to determine which models are practically identifiable, and such models should be favoured over the non-identifiable models during model selection. Conversely, once

we have picked a model, identifiability analysis can tell us how much data we need to collect in order for the chosen model to be identifiable. We have also found that the profile likelihoods are a computationally cheaper method for identifiability analysis compared to Bayesian inference. Nonetheless, it can be computationally difficult to find the global maximum in certain cases during the computation of profile likelihoods. Another interesting finding is that the more complicated models tend to show greater variation in parameter estimates across experimental replicates, and their parameter estimates are also more sensitive to the subtle differences in the data. Biological knowledge can inform whether this variation is due to overfitting, or if it points to real variations in system behaviour, which might be appropriate to model using a mixed effects framework. The chapter achieved the aims of highlighting the importance of practical parameter identifiability at all stages of a modelling study, and demonstrated the use of profile likelihoods for investigating parameter identifiability.

In Chapter 3, we explored ways to optimally design an experiment to maximise the informativeness of the resulting data for parameter inference and model selection purposes, using ODE models of cell growth as an illustrative example, thereby contributing toward the aim of improving the efficiency of biological experiments to better study biological systems. We investigated two aspects of experimental design: the selection of initial conditions for the system and the implementation of external stimuli to control the system. The experiments were designed such that they would capture the system's behaviour across a wide range of conditions, providing more insight into the system. The optimally designed experiments can provide data that are significantly more useful for parameter inference or model discrimination compared to more naively designed experiments, demonstrating that the benefits of conducting experimental design prior to performing experiments can be significant, especially in cases where doing experiments is expensive or difficult. Furthermore, we discussed the conceptual similarities between maximising parameter identifiability and model discrimination. These two goals represent questions raised at different stages of a modelling study, and the optimal experimental design for one of these goals might not be good for the other. We have also compared the performance of the FBS algorithm for the numerical implementation of PMP and direct optimisation techniques for finding solutions to optimal control problems arising from an experimental design approach. We found that the controls obtained by both methods perform similarly well, but FBS tends to be faster. Combining both methods, where we initialise direct optimisation using the output of

the FBS algorithm, can yield controls that perform even better than those found by using either one of the methods alone.

In Chapter 4, we have shown that a wave of competency, a phenomenon found in developmental biology, can serve as a mechanism to control the directionality and alignment of stripe patterns in a Turing-type DDI model of pattern formation. For both the CDIMA and Schnakenberg models, we found that when the wave speed is very high, disorganised labyrinthine patterns are produced. At moderate wave speeds, stripes parallel to the wave front are produced. The patterns shift to stripes perpendicular to the wave front as the wave speed decreases further. The regime that produces labyrinthine patterns and the regime that produces stripes parallel to the wave front are separated by a critical threshold for the wave speed, which is related to the natural rate of pattern propagation, whereas the separation between the latter regime and the regime that produces stripes perpendicular to the wave front is related to the existence of travelling wave solutions to a reduced model, which is less well understood. Given that the same qualitative behaviours were observed for the Schnakenberg model with two different parameter sets, and also for the CDIMA model, which has a completely different functional form, these conclusions are likely broadly applicable to other DDI models of stripe pattern formation, and are not dependent on the value of model parameters. These findings pave the way for controlling the system to generate a specific desired pattern by modulating the wave of competency. The chapter demonstrated the usage of numerical simulations to explore, and provide intuitive insight into, the behaviour of a complex system.

In Chapter 5, we proposed a method for optimally controlling an ABM by applying PMP to a surrogate equation-based model, obtained by using SINDy to learn the governing equations for the dynamics of a set of summary statistics of the ABM data in the form of ODEs. This chapter made progress toward developing a general framework for applying optimal control to ABMs, thereby making it easier to apply control to a wide class of models. We illustrated the method by using it to apply control to a simple ABM of motile cells, where the summary statistics are manually specified and provided to the equation learning algorithm. We found that SINDy does not always learn the terms that correspond to the mean-field model, which we take as "ground-truth". Its success rate depends on the data generated from the ABM. The most important factor is that the control function used to generate the data needs to take values across a large range, so that the data can reveal the full range of effects of the control on the summary statistics, and allow the different potential terms on the right-hand side of the ODE

to be distinguished. Furthermore, the amount of data and the level of noise present in the data also play a role, but applying naive denoising methods can do more harm than good. We then sample two learned ODEs, apply PMP to them to obtain candidates for optimal control, and transfer those control candidates back to the original ABM to evaluate them. We found that the model with mechanistically inaccurate terms may nonetheless lead to a control that performs well (as measured by the objective function) in some cases, but poorly in other cases. This shows that controls obtained this way should not be trusted blindly, and it is necessary to evaluate the quality of the control using the original ABM. This is an area where further development and refinement of the machine learning approach might be able to make improvements.

6.2 Future work

There are a number of directions for investigation and unanswered questions related to each of the chapters in this thesis, and we briefly summarise these below.

Related to Chapter 2, for practical parameter identifiability in general, it would be interesting to investigate how model mis-specification impacts parameter identifiability and, conversely, whether it is possible to use identifiability analysis as a tool for detecting possible model mis-specification. We have discussed the possibility that, in the presence of model mis-specification, making a model more complicated may, in fact, improve parameter identifiability if it rectifies a flaw in the model. While it may be relatively simple to construct an artificial example demonstrating this, it remains to be seen if such situations are common in more practical contexts.

In some cases, it might be worthwhile to work with a non-identifiable model, and find ways to mitigate its identifiability issues. Finding identifiable parameter combinations would be useful for understanding the source of identifiability issues, and gaining insight into the dynamics of the biological system. Most methods for structural identifiability analysis naturally reveals structurally identifiable parameter combinations (see Appendix F and [200] for details), but finding practically identifiable parameter combinations is more difficult. In some cases, we might be able to find simpler identifiable combinations by examining the correlations between parameter estimates and inspecting the bivariate profile likelihoods. For example, the bivariate profile likelihood with a sharp, linear ridge shown in Fig. 2.7 implies that a linear combination of the two plotted parameters is more identifiable than either parameter individually. However, such inspections would not be adequate for finding more complicated identifiable combinations. In the

future, it would be useful to develop computationally efficient techniques for finding such parameter combinations. Once these combinations are found, we may consider a re-parameterisation of the model, where the model is reformulated such that redundant parameters are eliminated, and non-identifiable parameters are grouped together, so that the reformulated model is more identifiable. Such approaches have been considered in a structural identifiability context by [86, 131], and it would be useful to explore ways to do the same in a practical identifiability context.

For cell invasion in particular, it would be interesting to see if additional data, such as cell cycle information, can help improve parameter identifiability. Incorporating the cell cycle into the model will make it more complicated, which may harm parameter identifiability, but this effect could be compensated for by the increased amount of data.

We have highlighted that the error model used in Chapter 2, which assumes an i.i.d. Gaussian observation error, is unrealistic. The noise in the barrier assay data is likely auto-correlated. Not accounting for this auto-correlation properly can lead to underestimating the uncertainty in parameter estimates. There are a few potential ways to address this, such as using bootstrap resampling to calibrate the confidence intervals given by profile likelihoods (as illustrated in [84]), or allowing the noise to depend on cell density, time, and space to account for heteroscedasticity (as done in [68]), or even using a stochastic model to account for structural noise. It remains to be investigated which approach is the most appropriate for cell invasion models.

There are many aspects of data collection and experimental design that we did not consider in Ch. 2 and Ch. 3. For example, when and how frequently the data were collected, which observable quantities to measure, and how long to run the experiment for. It would be interesting to investigate the effects of these factors on practical parameter identifiability, and take them into consideration when designing an experiment. In relation to the finding in Ch. 2, where we found that the initial conditions of the experiments can impact parameter identifiability, it would be interesting to find the optimal initial condition for the PDE models of cell invasion, i.e., the shape of the barrier in the barrier assay experiments, for parameter inference.

The natural rate of pattern propagation was found to be a key quantity in determining the patterning outcome in an expanding domain scenario in Chapter 4. This quantity is currently not well understood. Klika et al. [94] derived asymptotic expressions for the rate of propagation in the Schnakenberg model in certain regimes. It would be interesting to investigate this quantity for a more general class of models. Furthermore, while the natural rate of pattern propagation determines the critical speed for the wave of competency that separates the regime

for labyrinthine patterns and stripes parallel to the wavefront, the critical wave speed separating the latter with the regime for stripes perpendicular to the wave front is not well understood. A rigorous travelling wave analysis along the ridge of a perpendicular stripe could provide further insight into this matter.

In Chapter 5, we proposed future directions to extend the method by adding machine learning components that automatically choose and compute a set of summary statistics useful for applying control, without requiring the user to specify them. This would reduce the amount of ad-hoc work needed to be done for each ABM, broaden the scope of applicability of the method, and make it accessible to a wider audience interested in controlling ABMs. Implementing these extensions, and verifying that the method can reliably produce effective controls for ABMs, require further investigation and development.

In summary, this thesis explored several aspects of modelling and controlling biological systems. The explorations contribute to improving modelling practices, particularly regarding model selection and parameter inference. The results also shed light on aspects of pattern formation relevant to developmental biology. Furthermore, progress has been made towards developing a general framework for applying control to ABMs. Each of these topics raises further questions, which warrant future research.

Appendix A

Numerical methods for PDE simulations

This appendix provides details on the numerical methods employed for simulating the PDE models in Chapters 2 and 4. All PDEs considered in this thesis fall into the category of reaction-diffusion equations. The method we use can be summarised as discretising both the temporal and spatial dimensions with finite differences, then applying an IMEX (implicit-explicit) time-stepping scheme by treating the diffusion term implicitly, and the reaction term explicitly.

The basic method can be summarised as follows. We state the scheme in two spatial dimensions, it is straight-forward to adapt to the case of one spatial dimension. Consider the reaction-diffusion PDE

$$\frac{\partial u(x,y,t)}{\partial t} = D_0 \nabla^2 u + f(u), \quad (x,y) \in \Omega, t \in [0,T], \tag{A.1}$$

where the spatial domain $\Omega = [0, L_x] \times [0, L_y] \subseteq \mathbb{R}^2$ is a rectangle. We discretise the domain into $n_x \times n_y$ rectangles, each of size $\Delta x \times \Delta y$, such that $L_x = n_x \Delta x$ and $L_y = n_y \Delta y$. We discretise the time interval $[0, T]$ into n_t sub-intervals of length $\Delta t = T/n_t$. We denote the points of discretisation as

$$x_i = i\Delta x, i = 0, \ldots, n_x, \quad y_j = j\Delta y, j = 0, \ldots, n_y, \quad t_k = k\Delta t, k = 0, \ldots, n_t,$$

and the solution of the PDE at these points by $u_{i,j,k} = u(x_i, y_j, t_k)$.

We use an implicit-explicit (IMEX) scheme [7, 164] for time-stepping, where the nonlinear reaction term $f(u)$ is treated with the explicit Euler method, and the diffusion term overall is treated with the implicit Crank-Nicolson method, which has second order convergence. The advantage of this scheme is that the explicit treatment of the nonlinear components of the equation allows us to avoid solving a nonlinear root-finding problem at every time step, which

would be necessitated by a fully implicit scheme. The implicit treatment of the diffusion term improves the stability of the scheme, and [7] showed that this class of schemes has reasonably low errors when the diffusion term is not vanishingly small, which is the case for us. The IMEX Crank-Nicolson time-stepping scheme works by approximating the diffusion term as an average of its value at the current and next time step,

$$\frac{\partial u}{\partial t}(\cdot, t_k) \approx \frac{u(\cdot, t_{k+1}) - u(\cdot, t_k)}{\Delta t} = \frac{1}{2} D_0 \left[\nabla^2 u(\cdot, t_{k+1}) + \nabla^2 u(\cdot, t_k)\right] + f(u(\cdot, t_k)). \tag{A.2}$$

Let the linear operator $\tilde{\Delta}$ denote the discrete Laplacian, obtained from discretising ∇^2 in the usual way. Then, after simplifications, the time-stepping scheme can be written as

$$\left(I - \frac{\Delta t D_0}{2}\tilde{\Delta}\right) u_{i,j,k+1} = \left[\left(I + \frac{\Delta t D_0}{2}\tilde{\Delta}\right) u_{i,j,k} + \Delta t f(u_{i,j,k})\right]. \tag{A.3}$$

This is a linear problem, which can be solved efficiently. The scheme is adapted slightly to solve the specific models in Ch. 2 and 4, which we describe in the following sections.

A.1 Numerical scheme for Chapter 4

The PDE models in Chapter 4 involve two coupled PDEs in the form of

$$\begin{aligned}\frac{\partial u(x,y,t)}{\partial t} &= D_u \nabla^2 u + f(u,v,t),\\ \frac{\partial v(x,y,t)}{\partial t} &= D_v \nabla^2 v + g(u,v,t).\end{aligned}$$

It is straight-forward to adapt the numerical scheme from Eq. (A.3) for this system:

$$\begin{aligned}\left(I - \frac{\Delta t D_u}{2}\tilde{\Delta}\right) u_{i,j,k+1} &= \left[\left(I + \frac{\Delta t D_u}{2}\tilde{\Delta}\right) u_{i,j,k} + \Delta t f(u_{i,j,k}, v_{i,j,k}, t_k)\right],\\ \left(I - \frac{\Delta t D_v}{2}\tilde{\Delta}\right) v_{i,j,k+1} &= \left[\left(I + \frac{\Delta t D_v}{2}\tilde{\Delta}\right) v_{i,j,k} + \Delta t g(u_{i,j,k}, v_{i,j,k}, t_k)\right],\end{aligned}$$

where $u_{i,j,k} = u(x_i, y_j, t_k)$, and similarly for v.

A.1.1 Verification of convergence

To verify the convergence of the chosen numerical scheme, we compute multiple simulations of the Schnakenberg model (Eqs. (4.1),(4.3)) without a wave of competency (that is, $W \equiv 0$) nor artificial noise, with the parameter values given in Eq. (4.4) and $b = 1.4$. We choose a domain with two spatial dimensions, $\Omega = [-L, L] \times [-L, L]$ where $L = 100$, and $t \in [0, T]$ with $T = 80$, which is sufficient time for the system to reach a steady state pattern. The system is initialised

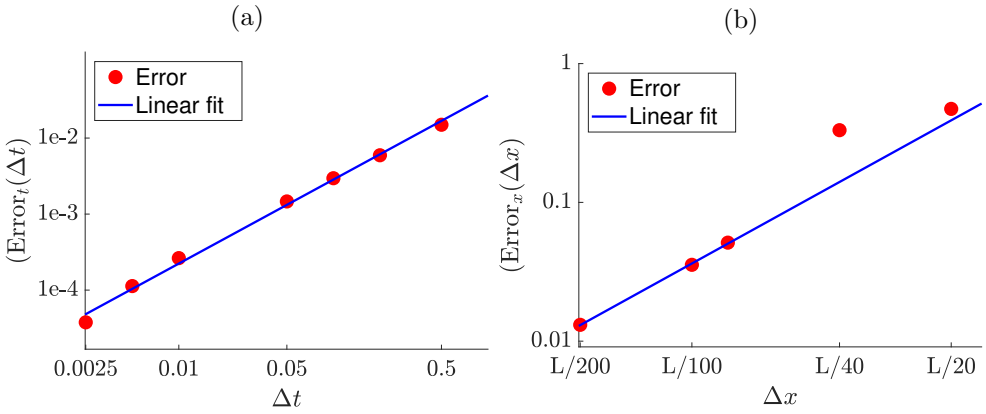


Figure A.1: Convergence verification for the finite difference scheme used in Chapter 4. Note the axes are logarithmic. (a) Plot of the error in the steady state pattern with respect to temporal discretisation, Δt, as defined in Eq. (A.4). (b) Plot of the error in the steady state pattern with respect to spatial discretisation, Δx, as defined in Eq. (A.5). The linear fit excludes the points at $\Delta x = L/20, L/40$, since with these values of Δx, the error is of the same order of magnitude as the solution, hence the spatial discretisation is too coarse.

at the HSS (Eq. (4.5)), and perturbed by Gaussian i.i.d. noise, which is generated with the same seed for each simulation.

Let $u(x, y, t; \Delta x, \Delta t)$ and $v(x, y, t; \Delta x, \Delta t)$ denote the model solution computed with the numerical scheme described in Sec. A.1, using the given Δx and Δt for spatial and temporal discretisation, respectively. Since we are primarily interested in the steady state pattern in Chapter 4, we define the error as the discrepancy between the final steady state of the model solutions at time T, computed with varying Δx and Δt, with that of a reference solution computed with smaller values of Δx and Δt, which we take to represent the "exact" solution. More specifically, we define

$$\begin{aligned}\mathrm{Error}_t(\Delta t) = \frac{1}{|\Omega|}\int_\Omega &|u(x, y, T; L/100, \Delta t) - u(x, y, T; L/100, 0.00125)| \\ &+ |v(x, y, T; L/100, \Delta t) - v(x, y, T; L/100, 0.00125)|\mathrm{d}x\mathrm{d}y, \end{aligned} \tag{A.4}$$

$$\begin{aligned}\mathrm{Error}_x(\Delta x) = \frac{1}{|\Omega|}\int_\Omega &|u(x, y, T; \Delta x, 0.01) - u(x, y, T; L/400, 0.01)| \\ &+ |v(x, y, T; \Delta x, 0.01) - v(x, y, T; L/400, 0.01)|\mathrm{d}x\mathrm{d}y. \end{aligned} \tag{A.5}$$

We plot Error_t and Error_x in Fig. A.1. The plots show a nearly linear relationship on a log-log plot with respect to Δt for $\Delta t \leq 0.5$, and Δx for $\Delta x \leq L/80$, indicating that the numerical scheme is convergent for these regimes. We have $\mathrm{Error}_t(0.01) = 2.63 \times 10^{-4}$ and $\mathrm{Error}_x(L/100) = 0.0354$, both of which are small compared to the typical magnitude of the solutions (which is of order 1), suggesting that the choices of $\Delta t = 0.01$ and $\Delta x = L/100$ used for the simulations in the main text are suitable. We have also verified that the steady

state pattern computed with $\Delta t = 0.01$ and $\Delta x = L/100$ has its number and location of stripes consistent with the more accurate simulations. Moreover, most simulations in Chapter 4 involve an on-going artificial noise of magnitude 0.01, which further renders the numerical errors insignificant.

A.2 Numerical scheme for Chapter 2

The scheme in Eq. (A.3) can be used directly for most models in Chapter 2, except the Porous Fisher model, which has a non-constant diffusion coefficient. The model can be written in the following form:

$$\frac{\partial u(x,y,t)}{\partial t} = \nabla \cdot (D(u)\nabla u) + f(u).$$

We follow the method in [202], where a central difference approximation with step size $\Delta x/2$ is used to approximate the second order spatial derivatives. This requires the evaluation of the non-linear diffusion coefficient $D(u)$ at half mesh points, which is then approximated as the average of D at the two adjacent mesh points, that is,

$$\begin{aligned} D(u_{i+1/2,j,k}) &= \frac{1}{2}(D(C_{i,j,k}) + D(C_{i+1,j,k})), \\ D(u_{i,j+1/2,k}) &= \frac{1}{2}(D(C_{i,j,k}) + D(C_{i,j+1,k})). \end{aligned}$$

We can then write the partial derivatives in the x–direction as

$$\begin{aligned} \frac{\partial}{\partial x}\left[D(u_{i,j,k})\frac{\partial u_{i,j,k}}{\partial x}\right] &\approx \frac{1}{\Delta x}\left[D(u_{i+1/2,j,k})\frac{u_{i+1,j,k}-u_{i,j,k}}{\Delta x} - D(u_{i-1/2,j,k})\frac{u_{i,j,k}-u_{i-1,j,k}}{\Delta x}\right], \\ &= \frac{1}{\Delta x^2}[D(u_{i+1/2,j,k})u_{i+1,j,k} - (D(u_{i+1/2,j,k}) + D(u_{i-1/2,j,k}))u_{i,j,k} \\ &\quad + D(u_{i-1/2,j,k})u_{i-1,j,k}], \end{aligned}$$

and similarly for the y–direction. This gives us the (u–dependent) discrete Laplace operator,

$$\begin{aligned} \tilde{\Delta}_{u_{i,j,k}} u_{i,j,k} &= \frac{\partial}{\partial x}\left[D(u_{i,j,k})\frac{\partial u_{i,j,k}}{\partial x}\right] \\ &= \frac{1}{\Delta x^2}[D(u_{i+1/2,j,k})u_{i+1,j,k} - (D(u_{i+1/2,j,k}) + D(u_{i-1/2,j,k}))u_{i,j,k} \\ &\quad + D(u_{i-1/2,j,k})u_{i-1,j,k}] \\ &\quad + \frac{1}{\Delta y^2}[D(u_{i,j+1/2,k})u_{i,j+1,k} - (D(u_{i,j+1/2,k}) + D(u_{i,j-1/2,k}))u_{i,j,k} \\ &\quad + D(u_{i,j-1/2,k})u_{i,j-1,k}]. \end{aligned}$$

At the points on the boundaries, to be consistent with the zero flux boundary conditions, we compute $D(u)$ using the value of u at that point. We adapt the Crank-Nicholson time-stepping

scheme to also treat the nonlinear $D(u)$ term explicitly, which gives us the following full numerical scheme for solving the Porous Fisher model:

$$\left(I - \frac{\Delta t}{2}\tilde{\Delta}_{u_{i,j,k}}\right) u_{i,j,k+1} = \left[\left(I + \frac{\Delta t}{2}\tilde{\Delta}_{u_{i,j,k}}\right) u_{i,j,k} + \Delta t f(u_{i,j,k})\right].$$

The problem of solving for $u_{\cdot,\cdot,k+1}$ in terms of $u_{\cdot,\cdot,k}$ remains linear in this scheme.

A.2.1 Verification of convergence

To verify the convergence of the numerical scheme, similar to Sec A.1.1, we compute multiple simulations of the Standard Fisher model with parameters from Eq. (2.13) using different values of Δx and Δt, and compare the discrepancies between the numerical solutions. The domain is $\Omega = [0, L] \times [0, L]$, where $L = 4380$ mm. The initial condition is $C = K$ within a circular region with a radius of 40 mm located at the centre of the domain, and $C = 0$ outside the circle. The choice of domain and initial conditions approximates the setup of the experiments with circular initial conditions.

Let $C(\cdot; \Delta x, \Delta t)$ denote the model solution computed with the numerical scheme described in Sec. A.2, using the given Δx and Δt for spatial and temporal discretisation, respectively, with $\Delta y = \Delta x$. In Chapter 2, we are interested in the model solution at all times, not just the final time, so we define the error to be the difference between the model solutions calculated with varying Δt or Δx with a more accurate, reference model solution, averaged over both time and space, as follows:

$$\text{Error}_t(\Delta t) = \frac{1}{T|\Omega|}\int_0^T \int_\Omega |C(x, y, t; L/150, \Delta t) - C(x, y, t; L/150, 1/240)| \, \mathrm{d}x\mathrm{d}y\mathrm{d}t, \tag{A.6}$$

$$\text{Error}_x(\Delta x) = \frac{1}{T|\Omega|}\int_0^T \int_\Omega |C(x, y, t; \Delta x, 1/30 \text{ h}) - C(x, y, t; L/600, 1/30 \text{ h})| \, \mathrm{d}x\mathrm{d}y\mathrm{d}t. \tag{A.7}$$

We plot $\text{Error}_t(\Delta t)$ in Fig. A.2(a). Observe that the errors for $\Delta t < 1/6$ h have a nearly linear dependence on Δt on a log-log plot, which implies that the numerical scheme is convergent in that regime. At the value of Δt used for the simulations in the main text, we have $\text{Error}_t(1/30 \text{ h}) = 1.604$ cells/mm^2, much lower than the carrying capacity, $K = 1300$ cells/mm^2, so we consider this error acceptable. The plot for $\text{Error}_x(\Delta x)$ is shown in Fig. A.2(b) which, once again, shows a nearly linear dependence on a log-log plot for $\Delta x \leq L/30$, implying that the scheme is convergent on this regime. We have $\text{Error}_x(L/150) = 3.1170$ cells/mm^2, which is within the range of acceptable accuracy. These results suggest that it is reasonable to use $\Delta t = 1/30$ h and $\Delta x = L/150$ for simulations in the main text.

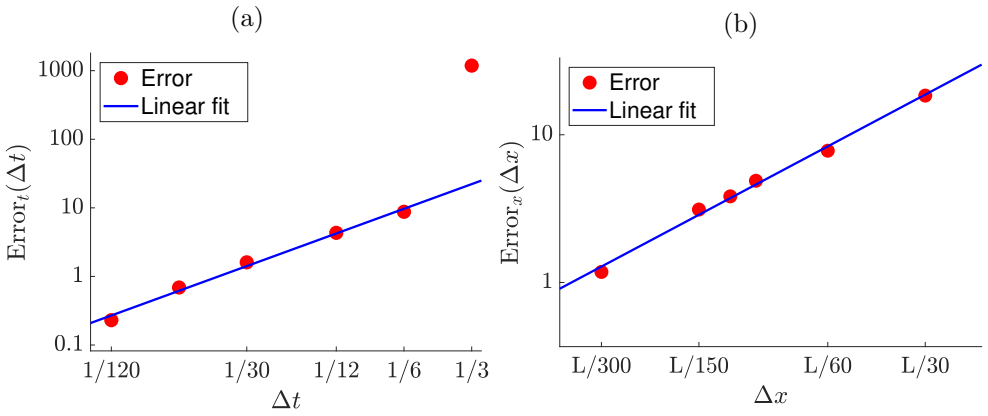


Figure A.2: Convergence verification for the finite difference scheme used in Chapter 2. Note that the axes are logarithmic. (a) Plot of the absolute error of the model solution of the Standard Fisher model as a function of temporal discretisation, Δt, as defined in Eq. (A.6). The linear fit ignores the point for $\Delta t = 1/3$ h, which is too large for the scheme to converge correctly. (b) Plot of the absolute error of the model solution of the Standard Fisher model as a function of spatial discretisation, Δx, as defined in Eq. (A.7).

Appendix B

Methods for parameter inference and identifiability analysis

This appendix introduces the three methods for parameter inference and identifiability analysis used in Chapter 2 and 3. These methods are described in general terms in this appendix, while the detail of implementing them for specific problems are given in the relevant chapters in the main text.

The purpose of these methods is to obtain estimates for the parameter values, as well as to quantify the uncertainty in the estimates by constructing a confidence interval. These methods are essentially different ways of exploring and approximating the likelihood function to varying degrees of efficiency or accuracy, and they vary in computational cost. Most of the computational cost is spent on evaluating the likelihood function, which has no closed form and is costly to evaluate. We introduce the first, maximum likelihood, in Sec. B.2. It is conceptually the simplest and computationally the cheapest, but the information it offers is limited, and insufficient for determining parameter identifiability. The second, profile likelihoods, is introduced in Sec. B.3. It is built upon maximum likelihoods, and is the method this study focuses on. We introduce the MH method, a widely known example of MCMC methods for Bayesian inference, in Sec. B.4. Finally, we offer a conceptual comparison between the three methods in Sec. B.5.

B.1 Notations

The notations used in this appendix are kept consistent with the notations in Sec. 2.3 whenever possible. We will state the methods in terms of a model consisting of a single PDE in a domain with two spatial dimensions,

$$\frac{\partial u}{\partial t} = f(u, \nabla u, \nabla^2 u, \ldots, x, y, t; \boldsymbol{\theta}), \quad (x, y) \in \Omega \subseteq \mathbb{R}^2, t \in [0, T], \tag{B.1}$$

where $u(x, y, t)$ is a state variable, and $\boldsymbol{\theta}$ is a vector containing all parameters in the model. It is straight-forward to generalise the method to cases where the spatial dimension of the domain differs from two, as well as to models consisting of multiple PDEs, or to ODE models. Let P be the number of parameters in the model. For example, for the Standard Fisher model given in Table 2.1, $P = 3$ and $\boldsymbol{\theta} = (D_0, r, K)$. Suppose the data for u, denoted $u_{\text{data}}(x_i, y_j, t_k)$, are available at discrete points $\{(x_i, y_j)\}_{i,j}$ in space, and $\{t_k\}_k$ in time, where $0 \leq i \leq n_x, 0 \leq j \leq n_y, 0 \leq k \leq n_t$. Let $u_{\text{model}}(x, y, t; \boldsymbol{\theta})$ be the solution to Eq. (B.1), solved with the initial condition

$$u_{\text{model}}(t = 0) = u_{\text{data}}(t = 0),$$

perhaps numerically. We assume that the data are generated by the model with a given parameter set $\hat{\boldsymbol{\theta}}$, and perturbed by an i.i.d. Gaussian observational noise at each data point that follows a Gaussian distribution with zero mean and unknown variance σ^2. That is,

$$u_{\text{data}}(x_i, y_j, t_k) = u_{\text{model}}(x_i, y_j, t_k; \hat{\boldsymbol{\theta}}) + \epsilon_{ijk}, \quad \epsilon_{ijk} \sim \mathcal{N}(0, \sigma^2). \tag{B.2}$$

This assumption was made in both Ch. 2 and 3. Let $\phi(\cdot, \mu, \sigma^2)$ denote the probability density function (pdf) of the Gaussian distribution with mean μ and variance σ^2, and let

$$\Delta u_{ijk}(\boldsymbol{\theta}) = |u_{\text{data}}(x_i, y_j, t_k) - u_{\text{model}}(x_i, y_j, t_k; \boldsymbol{\theta})|, \tag{B.3}$$

be the discrepancy between the data and model, and $R(\boldsymbol{\theta}) = \sum_{i,j,k} \Delta u_{ijk}^2(\boldsymbol{\theta})$ be the total squared residual. The likelihood function is defined as

$$\begin{aligned} L(u_{\text{data}}|\boldsymbol{\theta}, \sigma) &= \prod_{i,j,k} \phi(\Delta u_{ijk}(\boldsymbol{\theta}), 0, \sigma^2) \\ &= \frac{1}{\sigma^N (2\pi)^{N/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i,j,k} \Delta u_{ijk}^2\right) \\ &= \frac{1}{\sigma^N (2\pi)^{N/2}} \exp\left(-\frac{R(\boldsymbol{\theta})}{2\sigma^2}\right), \end{aligned} \tag{B.4}$$

where $N = n_x n_y n_t$ is the total number of data points in the dataset. It is usually much more convenient to work with the log-likelihood function instead, which is defined as

$$\hat{l}(\boldsymbol{\theta}, \sigma) = \log(L(u_{\text{data}}|\boldsymbol{\theta}, \sigma)) = -N \log(\sigma) - \frac{N}{2} \log(2\pi) - \frac{R(\boldsymbol{\theta})}{2\sigma^2}. \tag{B.5}$$

The likelihood function plays a central role in all three methods.

B.2 Maximum likelihood

We introduce the maximum likelihood method first, since it is the simplest and most widely known method, and it is the basis of the profile likelihood method. The maximum likelihood estimator (MLE) for the parameters, $\boldsymbol{\theta}^*$, as well as for the variance of the observational noise, σ^*, is defined as

$$(\boldsymbol{\theta}^*, \sigma^*) = \underset{\boldsymbol{\theta},\sigma}{\operatorname{argmax}}\, \hat{l}(\boldsymbol{\theta}, \sigma). \tag{B.6}$$

The optimal σ^* must satisfy $\mathrm{d}\hat{l}(\cdot, \sigma^*)/\mathrm{d}\sigma = 0$ which, due to the assumption of Gaussian i.i.d. observation error, can be simplified to

$$(\sigma^*)^2 = \frac{R(\boldsymbol{\theta}^*)}{N}, \tag{B.7}$$

which means the optimisation for σ can be done explicitly after the optimisation for $\boldsymbol{\theta}$. Substituting this σ^* into Eq. (B.5) motivates the definition

$$l(\boldsymbol{\theta}) = \hat{l}\left(\boldsymbol{\theta}, \frac{R(\boldsymbol{\theta})}{N}\right) = \frac{-N}{2}\left[\log\left(\frac{2\pi}{N} R(\boldsymbol{\theta})\right) + 1\right], \tag{B.8}$$

so that

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, l(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, R(\boldsymbol{\theta}).$$

This gives the well-known result that finding the MLE under the assumption of Gaussian i.i.d. error is equivalent to solving a least squares regression problem. Note that here we chose to use the biased estimator with $\sigma^2 = R/N$, instead of the bias-corrected estimator commonly used in statistics, $\sigma^2 = R/(N-1)$. In our case, N is large enough such that the difference between the two estimators is negligible. Using the bias-corrected estimator will complicate the later derivations for little practical gain.

It is known that, given the assumption in Eq. (B.2) (which is that there exists a unique set of "true" parameters $\hat{\boldsymbol{\theta}}$, with which the models fitted to data can exactly replicate the model that generated the data, and that the observation noise is i.i.d. Gaussian), and under suitable regularity conditions, the MLE is asymptotically normal [99, Ch. 3.3]. As such, the distribution of the MLE converges in law (meaning that the cumulative distribution function converges pointwise) to

$$\boldsymbol{\theta}^* \sim \mathcal{N}\left(\hat{\boldsymbol{\theta}}, \frac{1}{N}\mathcal{I}(\boldsymbol{\theta}^*)^{-1}\right), \tag{B.9}$$

where $\mathcal{I}$ is the Fisher information matrix (FIM),

$$\mathcal{I}(\boldsymbol{\theta}^*) = -E\left[\left.\frac{\partial^2 l(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}^*}\right], \tag{B.10}$$

that is, the expected Hessian of the log-likelihood function. The FIM provides the local curvature of the log-likelihood at the MLE, which provides a sense of the accuracy of the estimator, where higher local curvature indicates that the log-likelihood decays faster away from the MLE, meaning the estimator is more accurate. The fact that the MLE converges to a Gaussian distribution motivates the idea to approximate the likelihood function, at least locally around the MLE, by the multivariate Gaussian distribution in Eq. (B.9), from which confidence regions for the parameter values can be derived. However, the confidence regions derived this way can be misleading, since the region in which the likelihood function is approximately Gaussian can be much smaller than the confidence region, which undermines the validity of the approximation. Such cases were encountered in Chapter 2, and also in [205]. In the next section, we will discuss how profile likelihoods can be used to construct a more accurate confidence region.

B.3 Profile likelihood

The profile likelihood method aims to produce a likelihood function for only a subset of parameters, usually a single one, instead of computing the joint likelihood for all parameters, which can be intractable. Essentially, for a parameter θ_i, we find the maximum likelihood of θ_i at a certain value θ_i' by performing the MLE calculation with θ_i fixed at θ_i'. We repeat the calculation over a range of θ_i' to produce a profile likelihood curve for θ_i.

In detail, let $\boldsymbol{\theta}_{-i}$ denote the parameter vector with θ_i, the i^{th} parameter, removed. We may also replace the subscript with the name of the replaced parameter for readability. For example, with the Standard Fisher model (see Table 2.1, where $\boldsymbol{\theta} = D_0, r, K$), we have $\boldsymbol{\theta}_{-r} = (D_0, K)$. The (non-normalised) profile log-likelihood function for a single parameter is then defined as

$$\tilde{l}_i(\theta_i') = \max_{\boldsymbol{\theta}_{-i}} \left[l(\boldsymbol{\theta})|_{\theta_i=\theta_i'} \right],$$

where l is the log-likelihood function defined in Eq. (B.8). It is common to normalise the profile log-likelihood function such that it has a maximum value of zero [175]. Clearly, the maximum of $\tilde{l}_i$ occurs at the MLE, θ_i^*, and $l_i(\boldsymbol{\theta}_i^*) = l(\boldsymbol{\theta}^*)$, so we define the normalised profile log-likelihood function as

$$l_i(\theta_i') = \tilde{l}_i(\theta_i') - l(\boldsymbol{\theta}^*), \tag{B.11}$$

which we refer to simply as the profile likelihood henceforth. A (normalised) profile (log-)likelihood function for m parameters can be defined similarly as

$$l_{i_1,\ldots,i_m}(\theta_{i_1}', \ldots, \theta_{i_m}') = \sup_{\boldsymbol{\theta}_{-i_1,\ldots,-i_m}} \left[l(\boldsymbol{\theta})|_{\theta_{i_1}=\theta_{i_1}',\ldots,\theta_{i_m}=\theta_{i_m}'} \right] - l(\boldsymbol{\theta}^*).$$

In most cases, we are interested in obtaining profile log-likelihood functions for each single parameter, as well as combinations of two parameters. Profile likelihoods with more than two parameters tend not to offer much additional insight, given the difficulty of visualising them. The rest of this section focuses on the $m = 1$ case for simplicity. The derivations for $m \geq 2$ are similar.

Next, we aim to use the profile likelihoods to construct confidence regions for the parameter values. Since we know that l_i has a maximum of zero at θ_i^*, a Taylor expansion gives

$$l_i(\theta_i) = \frac{(\theta_i - \theta_i^*)^2}{2} \left.\frac{\mathrm{d}^2 l_i}{\mathrm{d}\theta_i^2}\right|_{\theta_i^*} + O(|\theta_i - \theta_i^*|^3). \tag{B.12}$$

It is easy to see that the derivatives of $l_i(\theta_i)$ can be expressed in terms of the derivatives of $R(\theta_i)$. If the higher derivatives of $R(\theta_i)$ are small, which can happen if, for example, the model output is roughly linear with respect to parameter values, then it can be justified to ignore the higher order terms in Eq. (B.12), and approximate l_i by a quadratic. The likelihood function is then

$$L_i(\theta_i) = \exp(l_i(\theta_i)) \approx \exp\left[-\frac{(\theta_i - \theta_i^*)^2}{2\tilde{\sigma}^2}\right], \tag{B.13}$$

where $\tilde{\sigma}^2 = -1/l_i''(\theta_i^*)$ (this is the profile likelihood analogue of the FIM, Eq. (2.15)). This means that the posterior distribution for the parameter θ_i is approximately Gaussian of the form $\mathcal{N}(\theta_i^*, \tilde{\sigma}^2)$. Moreover, the variance of the MLE is equivalent to the curvature of the log-likelihood function. Intuitively, this means that a sharper log-likelihood curve implies a higher certainty in the parameter value. This result is closely related to the asymptotic normality of the MLE (Eq. (B.9)), and is discussed in more detail in [134].

Using Eq. (B.12), we can see that at the boundaries of the interval, $\theta_i^* \pm 1.96\tilde{\sigma}$, the log-likelihood function takes the value $-1.96^2/2 = -1.92$. Therefore, we take

$$\{\theta_i | l_i(\theta_i) > -1.92\}, \tag{B.14}$$

as an estimate for the 95% confidence region of the parameter θ_i, which would be an interval if the profile likelihood is uni-modal. We can similarly define an approximation to the 95% confidence region when there are m fixed parameters:

$$\{\theta_{i_1}, \ldots, \theta_{i_m} | l_{i_1,\ldots,i_m}(\theta_{i_1}, \ldots, \theta_{i_m}) > -\chi^2(0.95; m)/2\}, \tag{B.15}$$

where $\chi^2(\cdot; m)$ denotes the inverse of the cumulative distribution function (cdf) of a χ^2 distribution with m degrees of freedom, see [144, Ch. 9] and [134]. The threshold is approximately -1.92 for $m = 1$ and -3.00 for $m = 2$.

Note that in the $m = 1$ case, as an alternative to Eq. (B.14), the 95% confidence interval can be estimated as

$$[\theta_i^* - 1.96\tilde{\sigma}, \theta_i^* + 1.96\tilde{\sigma}].$$

This is equivalent to using only the MLE and FIM to construct a confidence interval, since this estimate does not use the profile likelihood at points other than the MLE. As discussed in Sec. B.2 and pointed out in [205], this estimate can be misleading if the parameter distribution is far from Gaussian, or equivalently, the profile likelihood is far from being a quadratic curve. As the results in Sec. 2.4 as well as in [176] show, this tends to be the case if the parameter is practically non-identifiable, hence the MLE and FIM do not provide reliable information on parameter identifiability by themselves.

B.4 Bayesian inference

In contrast to the maximum likelihood and profile likelihood approaches, which revolve around locating minima in the likelihood function, Bayesian inference aims to construct a posterior distribution for parameter values given data. The main idea of Bayesian inference in the context of parameter identification is to update our "degree of belief" in a certain parameter set based on observed data,

$$p(\boldsymbol{\theta}|C_{\text{data}}) \propto L(C_{\text{data}}|\boldsymbol{\theta})p(\boldsymbol{\theta}).$$

Here, $p(\boldsymbol{\theta}|C_{\text{data}})$ is the posterior distribution of parameters, $L(C_{\text{data}}|\boldsymbol{\theta})$ is the likelihood function defined in Eq. (B.4) (with σ taken as a constant), and $p(\boldsymbol{\theta})$ is the prior. Since it is usually very difficult to calculate the posterior directly, most Bayesian methods try to sample from the posterior distribution to approximate it. One of the simplest and most well-known methods is the MH algorithm (see [161, Ch. 4.5]), which belongs to the class of MCMC algorithms.

Briefly, the MH algorithm works as follows. We choose a certain initial point $\boldsymbol{\theta}_0$ in the parameter space. At each step, a candidate $\boldsymbol{\theta}'$ is proposed by sampling from a symmetric distribution $g(\boldsymbol{\theta}'|\boldsymbol{\theta}_s)$, known as the proposal density or the transition kernel, where $\boldsymbol{\theta}_s$ is the current step of the chain. Then, with probability $r = L(C_{\text{data}}|\boldsymbol{\theta}')/L(C_{\text{data}}|\boldsymbol{\theta}_s)$, the candidate is accepted as the next step of the chain by setting $\boldsymbol{\theta}_{s+1} = \boldsymbol{\theta}'$, and rejected otherwise, in which case the chain remains stationary at the current point, so $\boldsymbol{\theta}_{s+1} = \boldsymbol{\theta}_s$. The generated samples $\{\boldsymbol{\theta}_s\}_s$ form a Markov chain. The process terminates when the Markov chain satisfies a certain convergence criterion, discussed later, at which point the generated samples represent a

good approximation of the posterior, which can be used to construct confidence regions for the parameters.

There are several challenges in using sampling methods to construct posterior distributions in general, and MH in particular. Sampling methods, in general, require many samples to be collected before a good approximation to the true posterior can be constructed, especially if the dimension of the parameter space is high. This means high computational cost. For MH, it can be difficult to choose a good transition kernel g. The initial point might also be located in a region where the posterior likelihood is low. Therefore, the early samples within a "burn-in" period are discarded, since they are not representative of the true posterior. The length of the burn-in period is set by the user, which should be sufficiently large so that all non-representative samples are discarded. As a heuristic for setting the burn-in period, we may set a burn-in period sufficiently long so that the mean of the earliest retained samples is close to the mean of the later samples.

B.5 Conceptual comparison between the methods

The three methods we have introduced are related in many ways, but there are interesting conceptual differences. The inference outcome from the three methods for the parameter θ_i can be summarised in a way that best illustrates their similarities and differences using the following equations:

$$\begin{aligned}
\theta_i^* &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, L(C_{\text{data}}|\boldsymbol{\theta}) && \text{MLE for } \theta_i, \\
L_i(\theta_i') &= \underset{\boldsymbol{\theta}_{-i}}{\operatorname{argmax}}\, L(C_{\text{data}}|\boldsymbol{\theta})|_{\theta_i=\theta_i'} && \text{profile likelihood function for } \theta_i, \\
p(\theta_i|C_{\text{data}}) &= \int L(C_{\text{data}}|\boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}_{-i} && \text{Bayesian marginal posterior for } \theta_i.
\end{aligned}$$

Each method provides some information on the parameter set that can best represent the data, which can be used to construct a confidence interval for a given parameter, thereby quantifying uncertainty in estimates for parameter values. The methods do so by exploring the likelihood function $L(C_{\text{data}}|\boldsymbol{\theta})$ (Eq. (B.4), where the dependence on the nuisance parameter σ is ignored), but in different ways.

The maximum likelihood method stems from the frequentist philosophy of statistics, which posits that there is a definitive but unknown "true" value for the parameters, and $L(C_{\text{data}}|\boldsymbol{\theta})$ represents how likely it is that the true parameter value is located at $\boldsymbol{\theta}$. The method aims to find the global maximum of $L(C_{\text{data}}|\boldsymbol{\theta})$, while ignoring all other features of the likelihood landscape.

This means that the calculation of the MLE is a single optimisation problem in P–dimensional parameter space, and it is the computationally cheapest method out of the three. During the search for the MLE, the optimisation routine only evaluates points that are promising candidates for the global maximum. The MLE only provides information about the likelihood function at a single point. The Gaussian approximation to the likelihood function obtained using the FIM and MLE therefore paints an incomplete and potentially misleading picture, since for a nonlinear model like the ones considered here, there are no guarantees that the likelihood landscape is close to Gaussian. For this reason, the maximum likelihood method should not be considered a reliable way to quantify the uncertainty in parameter values, and so the information it provides is not sufficient to adequately determine parameter identifiability.

The profile likelihood method is also a frequentist method but, in contrast to maximum likelihood, it aims to provide global information on the likelihood landscape. The profile likelihood function can be thought of as an approximation to the marginal posterior distribution of that parameter in the Bayesian sense, the main discrepancy being that calculating the Bayesian marginal posterior entails integrating the likelihood function over the other parameters, instead of taking the maximum. In this project, we mostly focus on computing single-parameter (i.e., univariate) profile likelihood functions, which corresponds to finding the maxima of $L(\boldsymbol{\theta})$ on a series of $(P-1)$–dimensional slices of the parameter space. Since we compute n_p points on the profile likelihood curve of each parameter, the calculation involves $n_p P$ optimisation problems in $(P-1)$–dimensional space. Typically, we need at least $n_p \geq 10$ points for an accurate picture of the profile likelihood, which means that the computational cost of profile likelihoods is typically higher than that of the MLE, even for a modest number of parameters P. However, the information it provides gives a much more comprehensive picture of the likelihood landscape, so the confidence intervals constructed from profile likelihood functions are more trustworthy than one computed from the MLE and FIM for the purpose of determining parameter identifiability.

The Bayesian philosophy considers the parameters to be random variables with some prior distribution representing the initial degree of belief in the parameters taking on certain values, and this belief is updated upon observing the evidence (experimental data), resulting in a posterior distribution. Given the choice of a uniform distribution as the prior, the posterior distribution and the likelihood function are very similar, with the only difference being that the former averages over the other parameters, while the latter takes the maximum. Nonetheless, their interpretations are different. The computed posterior distribution provides global

information on the parameters, so it is suitable for determining parameter identifiability. The MH algorithm essentially explores the likelihood landscape with a random walk biased towards directions with higher likelihood. This is different from some gradient-based optimisation algorithms (such as the interior points method used by *fmincon*, which we use to implement the frequentist methods), which always move toward points with higher likelihood with no element of randomness. On the other hand, MH has a conceptual similarity to evolutionary algorithms such as CM-AES, which also explore the likelihood landscape with some degree of randomness. How MH compares to profile likelihoods in terms of computational costs is not obvious from theory. In the investigations in Sec. 2.4.6, we found that in practice, for the problems we have considered, computing profile likelihoods is cheaper than performing MH.

Appendix C

Theories of optimal control

We introduce the theories of optimal control in general in Sec. C.1, and the numerical algorithms for applying these theories in Sec. C.2. The theories and algorithms in this appendix are applied to specific problems in Chapter 3 and Chapter 5.

C.1 Optimal control theory for ODEs

The traditional optimal control problem for ODE models is formulated as follows. Suppose $\boldsymbol{x} \in \mathbb{R}^n$ is a set of n state variables whose dynamics, and boundary conditions at initial time ($t = 0$) and terminal time ($t = T$), are described by the system of ODEs

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, t), \quad \boldsymbol{x}(0) \in S_0, \quad \boldsymbol{x}(T) \in S_T,$$

where $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^n$ is continuous, and all of its partial derivatives are continuous, and S_0, S_T are prescribed subsets of $\mathbb{R}^n$ representing initial and terminal conditions. We are interested primarily in cases where the initial condition is specified while the terminal conditions are left free (e.g., $S_0 = \{\boldsymbol{x}_0\}, S_T = \mathbb{R}^2$), and the terminal time T is fixed, although the theory allows more general S_0 and S_T, and T can be left free for selection. Suppose $\boldsymbol{u} \in \mathbb{R}^m$ is a set of m control variables, whose trajectories are prescribed, which are allowed to take values in $\mathcal{U}$, and must be at least piecewise $\mathcal{C}^1$. We denote the class of piecewise $\mathcal{C}^1$ functions as $\mathcal{D}^1$. We aim to minimise the cost functional $J[\boldsymbol{u}]$, also known as the objective function,

$$\min_{\boldsymbol{u} \in \mathcal{U}} J[\boldsymbol{u}] = \min_{\boldsymbol{u} \in \mathcal{U}} \int_0^T \mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\mathrm{d}t + \Phi(\boldsymbol{x}(T)), \tag{C.1}$$

where $\mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ is known as the Lagrangian, which is assumed to be continuous in all its inputs, and all of its partial derivatives are continuous. The Lagrangian describes the cost incurred throughout the time span of the model, and usually involves one or more terms that are functions of $\boldsymbol{x}$, that describe our desired system behaviour, and terms that are functions of $\boldsymbol{u}$,

that describe the cost of applying the control. The function Φ describes the cost associated with the final state of the system. For example, in a model for leukaemia [171], the state variables represent the population of various healthy and cancerous cell types, while the control variable represents the chemotherapy dosage. The Lagrangian $\mathcal{L}$ indicates our desire to minimise both the count of cancer cells and the amount of chemotherapy administered over $[0, T]$, and the Φ term reflects the aim to achieve a low cancer cell count at the final time. Usually, the Φ term can be absorbed into the integral term by rewriting Eq. (C.1), resulting in a simpler system to analyse, so we will only consider the case $\Phi = 0$ henceforth.

There are two important theories for solving optimal control problems. The Hamilton-Jacobi-Bellman (HJB) equation (originally by Bellman [20], see [114, Ch. 5] for a modern introduction) applies the principle of dynamic programming, and provides a necessary and sufficient condition for a control to be optimal in the form of a system of PDEs, as well as a constructive method for finding the optimal control. However, the resulting PDE can be difficult to solve. The second theory is Pontryagin's Maximum Principle (PMP), (originally by Pontryagin [147], see [112, Ch. 1.3] and [114, Ch. 4.1] for a modern introduction) which has its origins in calculus of variations and Hamiltonian mechanics. It only provides a necessary condition for a control to be optimal, but it provides a computationally easier way to obtain optimal control than the HJB equation, since the condition is in the form of an ODE system. For this reason, PMP can be more practical for models of biological systems, which tend to be larger and more complicated (in the sense of having more equations and terms) than models found in mechanics, so we will focus on PMP in this thesis.

The PMP theory defines the Hamiltonian, H, as

$$H(\boldsymbol{x}, \boldsymbol{\lambda}; \lambda_0, \boldsymbol{u}) = \lambda_0 \mathcal{L}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda} \cdot \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}).$$

The variables $\boldsymbol{\lambda} \in \mathbb{R}^n$ are called the adjoint variables (also co-state variables), and the scalar λ_0 is called the abnormal multiplier. The dynamics of the state and adjoint variables can be described by the canonical equations

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \frac{\partial H}{\partial \boldsymbol{\lambda}}, \quad \frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}t} = -\frac{\partial H}{\partial \boldsymbol{x}}. \tag{C.2}$$

Then, under this framework, we have the following theorem, adapted from [114, Ch. 4.1.2].

Theorem 1 (Pontryagin's Maximum Principle (fixed terminal time)) *Let $u^*(t) \in \mathcal{D}^1$ be an optimal control to the problem described above, and let $\boldsymbol{x}^*(t)$ be the corresponding optimal*

trajectory for the state variables. Then, there exist a constant λ_0, and corresponding trajectories for the adjoint variables $\boldsymbol{\lambda}^*(t)$, such that

1. *(Canonical equations) $\boldsymbol{x}^*(t)$ and $\boldsymbol{\lambda}^*(t)$ satisfy the canonical equations, Eq. (C.2).*

2. *(Point-wise maximum condition) For every $t \in [0, T]$,*

$$\boldsymbol{u}^*(t) = \underset{\boldsymbol{u}' \in \mathcal{U}}{\operatorname{argmax}} H(\boldsymbol{x}^*(t), \boldsymbol{\lambda}^*(t); \lambda_0, \boldsymbol{u}'(t)). \tag{C.3}$$

   In other words, the control maximises the Hamiltonian at every point in time.

3. *(Transversality) The boundary conditions for the adjoint variables satisfy*

$$\boldsymbol{\lambda}(0) \perp \mathcal{T}_{\boldsymbol{x}^*(0)} S_0, \quad \boldsymbol{\lambda}(T) \perp \mathcal{T}_{\boldsymbol{x}^*(T)} S_T, \tag{C.4}$$

   where $\mathcal{T}_{\boldsymbol{x}} S$ denotes the tangent space of S at point $\boldsymbol{x}$. Recall that S_0 and S_T are admissible sets of initial and final conditions, respectively.

4. *(Conservation) $H(\boldsymbol{x}^*(t), \boldsymbol{\lambda}^*(t); \lambda_0, \boldsymbol{u}^*(t)) \geq 0$, and is constant in t.*

5. *(Abnormal multiplier) $\lambda_0 \leq 0$, and is constant in t.*

In practice, conditions (1–3) are the most helpful for finding the optimal control. Condition (4) tends to be satisfied automatically when the other conditions are satisfied, so it does not provide additional information for finding the optimal control. Condition (5) on the abnormal multiplier λ_0 can be simplified into two cases. The generic case has $\lambda_0 < 0$, in which case we can normalise H by dividing it by $|\lambda_0|$, which is equivalent to setting $\lambda_0 = -1$. The degenerate case has $\lambda_0 = 0$, which occurs only in rare situations.

Mathematically, PMP can be seen as the function-valued analogue of optimality conditions for (scalar or vector-valued) constrained optimisation problems. The canonical equations can be derived from the Euler-Lagrange equations, which are the equivalent of the first derivative test in a functional setting. The adjoint variables $\boldsymbol{\lambda}$ are analogous to Lagrange multipliers, with the purpose of enforcing the ODE constraint. Physically, the canonical equations arise from Hamiltonian mechanics, where the state variables represent the spatial coordinates of some mass, the adjoint variables represent the momentum, and the Hamiltonian represents the energy of the system, which is conserved in a closed system.

While PMP was originally formulated in terms of ODEs, it has been extended to more general settings, such as certain types of PDEs [113] and SDEs [122] (although there are no known

complete generalisations to all PDEs or SDEs), as well as certain instances of integro-differential equations [148] and nonlocal PDEs [22]. However, all of these extensions are formulated in terms of some variants of differential equation-based models, whereas ABMs represent an entirely different framework. There is no straight-forward way to generalise PMP (nor HJB) to make them applicable to ABMs. Addressing this challenge in applying PMP to ABMs will be the focus of Chapter 5.

C.2 Numerical optimal control

In most cases, it is impossible, or at least very difficult, to find solutions to the PMP conditions analytically, so numerical algorithms must be considered. It is not straight-forward to use these conditions to find a $\boldsymbol{u}^*$ satisfying the PMP conditions, since finding $\boldsymbol{x}^*$ and $\boldsymbol{\lambda}^*$ using Eq. (C.2) requires $\boldsymbol{u}^*$, and finding $\boldsymbol{u}^*$ using Eq. (C.3) in turn requires $\boldsymbol{x}^*$ and $\boldsymbol{\lambda}^*$, resulting in a "chicken-and-egg" problem. In many cases, including the problem considered later in Sec. 3.6, the boundary conditions for the state and adjoint variables are split between $t = 0$ and $t = T$, further complicating the problem. In this section, we review two numerical algorithms that utilise PMP to find the optimal control: shooting methods and forward-backward sweep (FBS), as well as direct optimisation, a numerical method for finding optimal control that does not rely on PMP.

Shooting methods and FBS are the two main classes of algorithms for finding numerical solutions to the PMP conditions (Eqs. (C.2)–(C.4)) [112]. Both are iterative methods that are initialised by replacing an unknown in the system with some reasonable value, enabling us to solve the system as an initial value problem (IVP). In each iteration, the solution to the IVP will violate one of the PMP conditions (unless the guess is exactly right). The guess is then iteratively updated based on the way the condition is currently violated. Ideally, the extent to which the condition is violated decreases after every iteration, eventually converging to zero as we approach the true solution to the optimal control problem.

Shooting methods [37, 91] work by guessing a value for the unknown initial value $\boldsymbol{\lambda}(0)$, then solving the initial value problem for $(\boldsymbol{x}, \boldsymbol{\lambda})$ forward in time from $t = 0$ to $t = T$, with $\boldsymbol{u}(t)$ at each time step calculated using the point-wise maximum condition, Eq. (C.3). The boundary condition for $\boldsymbol{\lambda}(T)$ will likely be violated, and the guess for $\boldsymbol{\lambda}(0)$ is iteratively updated. Shooting methods were found to be sensitive to the initial guess [58], and the update is non-trivial in $n > 1$ dimensions. The method also requires stringent conditions on the stability of the ODE system

to succeed [91], which are not satisfied for a broad range of boundary value problems arising from optimal control. For these reasons, we do not use shooting methods.

The FBS algorithm ([112, Ch. 4]) is initialised with an ansatz for the control, $\boldsymbol{u}^0(t)$, over the whole time interval. In each iteration, the current ansatz for the control, $\boldsymbol{u}^i(t)$, gives a closed system for $\boldsymbol{x}$, which is solved forward in time from $t = 0$ to $t = T$. This yields an $\boldsymbol{x}(T)$ which, along with the solution for $\boldsymbol{x}$, results in a closed system for $\boldsymbol{\lambda}$, which can be solved backward in time from $t = T$ to $t = 0$. We can then find a $\boldsymbol{u}^{\text{new}}(t)$ that satisfies the point-wise maximum condition, Eq. (C.3), given the solutions for $\boldsymbol{x}$ and $\boldsymbol{\lambda}$, and use an update rule to combine $\boldsymbol{u}^i(t)$ and $\boldsymbol{u}^{\text{new}}(t)$ to produce a control $\boldsymbol{u}^{i+1}(t)$ for the next iteration.

The simplest update rule for the control is to simply propose $\boldsymbol{u}^{\text{new}}(t)$ as the control in the next iteration. However, as found in [171], the method tends to not converge using this strategy. Instead, the authors recommend taking a weighted average between the previous $\boldsymbol{u}(t)$ and $\boldsymbol{u}^{\text{new}}(t)$ as the new control proposal, where ω, the weight of $\boldsymbol{u}^{\text{new}}(t)$, acts as an "update rate" parameter. They found that the method always converged for sufficiently small ω. This ω is directly analogous to the learning rate in machine learning, where a higher ω results in larger updates, so the control approaches the optimum quickly when the current ansatz for control is far from optimal, but can oscillate around the optimum once the control is near it. A smaller ω means the control approaches the optimum slower, but can get closer without oscillations. Therefore, we implement an adaptive update rate in Chapter 3, where the cost functional $J[u]$ is computed at each iteration, and we reduce ω by half whenever $J[u]$ increased compared to the previous iteration. We terminate the iterations when the difference in the cost functional between successive iterations is sufficiently small. We summarise the FBS algorithm, together with the adaptive update rule, in Algorithm 2.

There are possible ways to refine the FBS algorithm. For example, we can replace the adaptive update rate scheme in step (III.6) with a line search by setting

$$\omega = \underset{\omega}{\operatorname{argmin}}\, J[(1-\omega)\boldsymbol{u}^i + \omega \boldsymbol{u}^{\text{new}}].$$

However, this is much more difficult to implement since searching for the minimum of J can be computationally costly, so we choose to implement the aforementioned adaptive update instead. The forward-backward sweep method with $\omega = 1$ (instead of adaptive update) has been shown to be convergent under certain regularity conditions on $\boldsymbol{g}, \boldsymbol{\lambda}$ and H [129].

Algorithm 2 Forward-backward sweep with adaptive update rate

(I) Initialise $\boldsymbol{u}^1(t)$ to be some reasonable ansatz

(II) Set $\omega \leftarrow \omega^0, J[\boldsymbol{u}^0] \leftarrow \infty$

(III) For $i = 1, 2, \cdots$:

(1) Solve state equation, Eq. (3.22c), with $\boldsymbol{u} = \boldsymbol{u}^i$ and $\boldsymbol{x}(0)$ from Eq. (3.24), forward in time to obtain $\boldsymbol{x}^i(t)$

(2) Solve adjoint equation, Eq. (3.22d), with $\boldsymbol{u} = \boldsymbol{u}^i$, $\boldsymbol{\lambda}(T)$ from Eq. (3.24) and $\boldsymbol{x} = \boldsymbol{x}^i$, backward in time to obtain $\boldsymbol{\lambda}^i(t)$

(3) Compute cost functional, $J[\boldsymbol{u}^i]$, Eq. (3.21), using $\boldsymbol{u}^i, \boldsymbol{x}^i, \boldsymbol{\lambda}^i$

(4) Check termination criterion: if $|J[\boldsymbol{u}^i] - J[\boldsymbol{u}^{i-1}]| < \epsilon$: return $\boldsymbol{u}^i$ as the optimal control

(5) Apply point-wise maximum condition, Eq. (3.23), to compute

$$\boldsymbol{u}^{\text{new}}(t) \leftarrow \underset{\boldsymbol{u}' \in \mathcal{U}}{\operatorname{argmax}} H(\boldsymbol{x}^i(t), \boldsymbol{\lambda}^i(t); \boldsymbol{u}') \tag{C.5}$$

(6) Adaptive update rate: if $J[\boldsymbol{u}^i] > J[\boldsymbol{u}^{i-1}]$: set $\omega \leftarrow \omega/2$

(7) Set $\boldsymbol{u}^{i+1} \leftarrow (1-\omega)\boldsymbol{u}^i + \omega \boldsymbol{u}^{\text{new}}$

Direct optimal control is conceptually much simpler. This method dispenses with PMP (or any other optimal control theories) entirely. The idea is that we can reduce the original optimisation problem in an infinite-dimensional function space to an optimisation problem in finite-dimensional space by discretising the problem before optimisation, parameterising the control function with a finite number of control parameters, then minimising the control objective with respect to the control parameters using any general-purpose optimisation algorithm. The caveat of this approach is that a general-purpose algorithm cannot exploit the structure of the original control problem. We also need a sufficiently fine discretisation to be able to approximate the original continuous-time system well, which potentially requires a high number of control parameters. This leads to an optimisation problem in high dimensions, so we expect the optimisation algorithm to converge slowly, and to be sensitive to the choice of the initial ansatz as to where we start our search.

Both the FBS algorithm and direct optimisation have significant drawbacks, but combining them can result in an algorithm that has the potential to outperform both individually. Since the FBS algorithm can get close to the optimal control, but oscillates around it, and direct optimisation relies on a good initial ansatz, this motivates a hybrid method that involves taking the output of FBS and using it as the initial ansatz for direct optimisation.

In Sec. 3.6, we apply the FBS algorithm to solve an optimal control problem that arises from

the task of optimally designing an experiment. We apply direct optimisation and the hybrid method to the same problem in Appendix G, and compare their performances.

Appendix D

Technical description of the barrier assay experiments

This appendix provides a technical description of the experimental procedure for the barrier assays, which generated the data used in Chapter 2. This description complements the plain language version in Sec. 2.2.1. The experiments were conducted by Kevin Suh of Princeton University, who kindly provided the data as well as the technical description found in this appendix.

D.1 Cell culture

The first step of the experiment is the preparation of the cell culture. Madin-Darby Canine Kidney (MDCK-II) cells expressing the FUCCI cell cycle reporter were cultured in low-glucose Dulbecco's modified Eagle medium (DMEM) with 10% fetal bovine serum (FBS) and 1% Penicillin-Streptomycin. Cells were split every 1 to 2 days, depending on the conditions of the cells. MDCK cells were treated with TrypLE for 8 minutes to be detached from the tissue-culture treated (TC) plastic dish. Detached cells were 1:2 diluted with the culture media and centrifuged at 300 g at room temperature for 3 minutes. Cell pellets were then re-suspended with culture media and seeded in the new dish, which were then used for the barrier assay.

D.2 Barrier assay

The TC plastic dish was coated with 50 µg/mL Collagen 4 in PBS at 37°C for 30 min. To shape the monolayer tissue, a 250 µm thick polydimethylsiloxane (PDMS) membrane cut with a Silhouette Cameo vinyl cutter was deposited on the TC plastic dish. The membrane acts as the barrier, with the stencil cut out in the centre forming the initial confinement region.

A 2.25×10^6 cells/mL MDCK cell suspension was seeded in each stencil. The seeding volume was 0.44 µL per mm^2 of tissue area. The TC plastic dish was incubated at 37°C for 30 minutes to induce cell attachment to the collagen surface. Subsequently, the dish was flooded with the culture medium, and cells were incubated for 19 h to form a confluent monolayer. Stencils were removed from the dish with tweezers at the start of the barrier assay. The expansion of the tissue monolayer was imaged from this point onward every 20 minutes for 24 hours with 4× phase objective.

D.3 Image analysis

From the 4× phase images, nuclei images were reconstructed using a tool based on convolutional neural networks from [106]. The local cell density was calculated by counting the number of nuclei centroids in each density measuring box. The box size was 58.4 µm × 58.4 µm, with 50% overlap between neighboring boxes.

Appendix E

Detailed inference results for cell invasion models

In this appendix, we present the detailed inference results for the cell invasion models in Chapter 2. A selection of these results is presented in Sec. 2.4.

E.1 Inference results for experimental datasets

In this section, we present the MLE and 95% confidence intervals of the parameter values calculated for all experimental datasets in table format, and σ^*, the MLE for the noise parameter, as well as the AIC and BIC. Recall that we have cell density data from eight experiments, which we refer to as the full datasets. Experiments 1–4 have circular initial conditions, while Experiments 5–8 have triangular initial conditions. For Experiments 1–4 we also consider the radially-averaged datasets. All results are given to four significant figures. The units of the parameters are given in Eq. (2.4).

We also perform a χ^2-likelihood ratio test for nested models [169], and report the p-value. The Standard Fisher model is nested inside the Porous Fisher, Richards, and Generalised Fisher models, and this test provides a measure of whether the more complicated models have a significant improvement in maximum likelihood over the simpler model. We denote the MLE parameters of the Standard Fisher model by $\boldsymbol{\theta}_0^*$, and the MLE parameters of one of the more complicated models by $\boldsymbol{\theta}_1^*$. Let

$$\Lambda = -2\log\left[\frac{L(C_{\text{data}}|\boldsymbol{\theta}_0^*)}{L(C_{\text{data}}|\boldsymbol{\theta}_1^*)}\right] = 2\log L(C_{\text{data}}|\boldsymbol{\theta}_1^*) - 2\log L(C_{\text{data}}|\boldsymbol{\theta}_0^*),$$

be the test statistic based on the ratio of maximum likelihoods of the two nested models being compared. Then, under our assumption of normal i.i.d observation errors (Eq. (2.6)), by Wilks' theorem [206], $\Lambda \sim \chi^2_{(\text{df})}$, where the degrees of freedom, df, are equal to the number of additional

parameters in the more complicated model compared to the simpler model. This allows us to compute a p-value, $p = 1 - \Phi(\lambda)$, where Φ is the cdf of $\Lambda \sim \chi^2_{(\text{df})}$. The $p-$value represents how likely the observed improvement in likelihood will happen, if the simpler model were the true model underlying the data. According to this metric, the more complicated model should be accepted if the p-value is sufficiently small.

For all datasets and all three more complicated models, we have $p < 0.05$, suggesting that these models are a significant improvement upon the Standard Fisher model. However, this test is only accurate if the observation errors are indeed normal i.i.d. As discussed in the main text, the observation errors are likely correlated across space and time, hence the improvements in likelihoods are overestimated, and the p-values reported here are likely overestimates.

In all following tables, we present the MLE and 95% confidence interval computed from profile likelihoods using Eq. (B.15) in the first four columns. The MLE estimate for σ is given in the fifth column. The AIC and BIC scores are presented in the sixth and seventh columns. These information criteria select the model with the lowest score. The lowest scores for each datasets are highlighted in bold. Finally, the eighth column shows the $p-$value for the likelihood ratio test.

For the full dataset from Experiment 1, we also provide the FIM, $\mathcal{I}$, as defined in Eq. (2.15). Higher value of the diagonal entries of $\mathcal{I}$ means the profile likelihood curve is sharper near the MLE, hence the corresponding parameter is more identifiable. The off-diagonal entries are an indication of the correlation between parameter estimates.

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1287 [1267, 1307]	0.2707 [0.2683, 0.2731]	2620 [2614, 2625]		115.09	162289	162310	-
Porous Fisher	1361 [1306, 1419]	0.2686 [0.2658, 0.2714]	2622 [2616, 2628]	η : 0.0219[0.0069, 0.0372]	115.04	162283	162311	4.07×10^{-3}
Richards	1467 [1400, 1535]	0.2272 [0.2146, 0.2410]	2612 [2606, 2618]	γ : 1.3119[1.1950, 1.4416]	114.89	162258	162287	1.07×10^{-8}
Generalised Fisher	1391 [1321, 1465]	0.1429 [0.1130, 0.1758]	2664 [2652, 2678]	α : 1.1086[1.0779, 1.1434] β : 1.2034[1.1587, 1.2508]	114.36	**162175**	**162210**	$< 10^{-20}$

Table E.1: Experiment 1, radially-averaged dataset.

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1287 [1282, 1293]	0.2775 [0.2767, 0.2782]	2621 [2619, 2622]		459.97	26161393	26161430	-
Porous Fisher	1545 [1529, 1564]	0.2702 [0.2694, 0.2710]	2628 [2627, 2630]	η : 0.0719[0.0677, 0.0766]	459.88	26160486	26160536	$< 10^{-20}$
Richards	1402 [1385, 1418]	0.2462 [0.2425, 0.2502]	2616 [2615, 2618]	γ : 1.1972[1.1677, 1.2249]	459.97	26161172	26161221	$< 10^{-20}$
Generalised Fisher	1423 [1403, 1443]	0.1013 [0.0945, 0.1085]	2701 [2698, 2704]	α : 1.1733[1.1688, 1.1818] β : 1.3548[1.3437, 1.3663]	459.54	**26158023**	**26158085**	$< 10^{-20}$

Table E.2: Experiment 1, full dataset.

FIM for the full dataset from Experiment 1 are given in Eq. (E.1). The order of parameters is (D_0, r, K) for the Standard Fisher model, (D_0, r, η, K) for the Porous Fisher model, (D_0, r, γ, K)

for the Richards model, and $(D_0, r, \alpha, \beta, K)$ for the Generalised Fisher model. The subscripts "SF", "PF", "R", and "GF" denotes the four models.

$$\mathcal{I}_{SF} = \begin{bmatrix} 0.8362 & 6277 & 0.2855 \\ 6277 & 5.780 \times 10^7 & 5077 \\ 0.2855 & 5077 & 2.5294 \end{bmatrix}, \quad \mathcal{I}_{PF} = \begin{bmatrix} 0.5924 & 5164 & -1727 & 0.2247 \\ 5164 & 5.663 \times 10^7 & -1.425 \times 10^7 & 5081 \\ -1727 & -1.425 \times 10^7 & 5.302 \times 10^6 & -603.3 \\ 0.2247 & 5081 & -603.3 & 2.489 \end{bmatrix},$$

$$\mathcal{I}_R = \begin{bmatrix} 0.709 & 6404 & 466.7 & 0.256 \\ 6404 & 7.154 \times 10^7 & 6.095 \times 10^6 & 5595 \\ 466.7 & 6.095 \times 10^6 & 5.71 \times 10^5 & 656.3 \\ 0.256 & 5595 & 656.3 & 2.548 \end{bmatrix},$$

$$\mathcal{I}_{GF} = \begin{bmatrix} 0.7118 & 1.473 \times 10^4 & 9179 & -492.6 & 0.3304 \\ 1.473e+04 & 3.898 \times 10^8 & 2.515 \times 10^8 & -1.798 \times 10^7 & 1.639 \times 10^4 \\ 9179 & 2.515 \times 10^8 & 1.635 \times 10^8 & -1.209 \times 10^7 & 1.134 \times 10^4 \\ -492.6 & -1.798 \times 10^7 & -1.209 \times 10^7 & 1.183 \times 10^6 & -1467 \\ 0.3304 & 1.639 \times 10^4 & 1.134 \times 10^4 & -1467 & 2.466 \end{bmatrix}. \tag{E.1}$$

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1211 [1192, 1231]	0.2780 [0.2755, 0.2805]	2551 [2546, 2556]		114.64	162216	162238	-
Porous Fisher	1650 [1578, 1725]	0.2673 [0.2645, 0.2701]	2564 [2558, 2570]	η : 0.1270[0.1086, 0.1458]	113.35	162009	162038	$< 10^{-20}$
Richards	981 [938, 1026]	0.3675 [0.3474, 0.3891]	2560 [2554, 2565]	γ : 0.6808[0.6333, 0.7323]	114.00	162115	162143	$< 10^{-20}$
Generalised Fisher	806 [758, 857]	0.2787 [0.2374, 0.3219]	2873 [2838, 2913]	α : 1.0744[1.0532, 1.0984] β : 2.0151[1.9249, 2.1130]	106.94	**160935**	**160970**	$< 10^{-20}$

Table E.3: Experiment 2, radially-averaged dataset.

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1157 [1152, 1163]	0.2896 [0.2888, 0.2904]	2550 [2548, 2552]		451.90	26099801	26099838	-
Porous Fisher	1527 [1509, 1545]	0.2788 [0.2779, 0.2795]	2561 [2560, 2563]	η : 0.1099[0.1071, 0.1129]	451.56	26097458	26097507	$< 10^{-20}$
Richards	916 [907, 926]	0.4061 [0.4004, 0.4119]	2561 [2559, 2562]	γ : 0.6307[0.6196, 0.6419]	451.56	26097356	26097406	$< 10^{-20}$
Generalised Fisher	826 [816, 837]	0.2383 [0.2283, 0.2484]	2893 [2881, 2905]	α : 1.1091[1.1024, 1.1160] β : 2.1168[2.0860, 2.1485]	449.61	**26082314**	**26082376**	$< 10^{-20}$

Table E.4: Experiment 2, full dataset.

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1107 [1083, 1131]	0.3172 [0.3133, 0.3212]	2518 [2511, 2525]		167.47	169220	169242	-
Porous Fisher	1228 [1160, 1301]	0.3136 [0.3093, 0.3179]	2520 [2513, 2527]	η : 0.0394[0.0191, 0.0604]	167.34	**169207**	**169236**	1.11×10^{-4}
Richards	1200 [1117, 1288]	0.2860 [0.2622, 0.3122]	2514.7859 [2507, 2522]	γ : 1.1669[1.0249, 1.3367]	167.42	169217	169245	1.94×10^{-2}
Generalised Fisher	1146 [1064, 1237]	0.2391 [0.1941, 0.2866]	2534 [2516, 2553]	α : 1.0501[1.0231, 1.0811] β : 1.1005[1.0093, 1.1850]	167.33	169209	169245	4.77×10^{-4}

Table E.5: Experiment 3, radially-averaged dataset.

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1118 [1113,1122]	0.3198 [0.3191,0.3206]	2521 [2520,2523]		458.73	26152075	26152112	-
Porous Fisher	1300 [1286,1314]	0.3144 [0.3136,0.3153]	2524 [2523, 2526]	η : 0.0569[0.0529, 0.0609]	458.64	26151214	26151263	$< 10^{-20}$
Richards	1293 [1275,1307]	0.2633 [0.2597,0.2678]	2514 [2513,2516]	γ : 1.3493[1.3131, 1.3801]	458.68	26151457	26151506	$< 10^{-20}$
Generalised Fisher	1377 [1353,1398]	0.1999 [0.1919,0.2080]	2500 [2498,2504]	α : 1.0506[1.0439, 1.0576] β : 0.8671[0.8443, 0.8943]	458.64	**26151117**	**26151179**	$< 10^{-20}$

Table E.6: Experiment 3, full dataset.

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1239 [1221,1257]	0.2849 [0.2825,0.2873]	2784 [2779,2790]		113.29	161998	162019	-
Porous Fisher	1406 [1356,1458]	0.2800 [0.2773,0.2827]	2789 [2784,2795]	$\eta : 0.0499[0.0363, 0.0637]$	112.96	161945	161974	1.56×10^{-13}
Richards	1466 [1410,1523]	0.2273 [0.2168,0.2387]	2775 [2770,2781]	$\gamma : 1.4221[1.3136, 1.5407]$	112.82	161922	161951	1.14×10^{-18}
Generalised Fisher	1416 [1358,1476]	0.1464 [0.1216,0.1732]	2799 [2789,2810]	$\alpha : 1.1009[1.0761, 1.1284]$ $\beta : 1.0869[1.0457, 1.1293]$	112.75	**161913**	**161948**	4.32×10^{-20}

Table E.7: Experiment 4, radially-averaged dataset.

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1252 [1247,1258]	0.2865 [0.2858,0.2872]	2788 [2787,2790]		459.50	26157597	26157634	-
Porous Fisher	1440 [1425,1454]	0.2809 [0.2801,0.2817]	2794 [2792,2795]	$\eta : 0.0539[0.0500, 0.0575]$	459.37	26156816	26156866	$< 10^{-20}$
Richards	1565 [1548,1580]	0.2108 [0.2084,0.2137]	2775 [2773,2776]	$\gamma : 1.6398[1.6013, 1.6736]$	459.20	**26155625**	**26155675**	$< 10^{-20}$
Generalised Fisher	1552 [1533,1570]	0.1028 [0.0965,0.1093]	2797 [2794,2800]	$\alpha : 1.1489[1.1398, 1.1583]$ $\beta : 1.0701[1.0572, 1.0831]$	459.18	26155629	26155691	$< 10^{-20}$

Table E.8: Experiment 4, full dataset.

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1416 [1410,1422]	0.3085 [0.3077,0.3093]	2344 [2343,2346]		401.92	25693899	25693936	-
Porous Fisher	4102 [4059,4145]	0.2739 [0.2731,0.2747]	2377 [2375,2379]	$\eta : 0.5677[0.5609, 0.5746]$	396.80	25649571	25649620	$< 10^{-20}$
Richards	2697 [2688,2706]	0.1364 [0.1359,0.1367]	2336 [2334,2337]	$\gamma : 7.8055[7.7527, 7.9284]$	390.02	**25589713**	**25589762**	$< 10^{-20}$
Generalised Fisher	-	-	-	$\alpha : -$ $\beta : -$	-	-	-	-

Table E.9: Experiment 5, full dataset.

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1161 [1156,1166]	0.3244 [0.3235,0.3253]	2307 [2305,2308]		392.83	25614660	25614697	-
Porous Fisher	2713 [2685,2743]	0.2931 [0.2922,0.2940]	2333 [2331,2334]	$\eta : 0.4097[0.4041, 0.4157]$	389.78	25587515	25587565	$< 10^{-20}$
Richards	2290 [2282,2299]	0.1413 [0.1409,0.1416]	2288 [2286,2289]	$\gamma : 8.1042[8.0335, 8.1723]$	383.88	**25534767**	**25534816**	$< 10^{-20}$
Generalised Fisher	-	-	-	$\alpha : -$ $\beta : -$	-	-	-	-

Table E.10: Experiment 6, full dataset.

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1845 [1837,1853]	0.2260 [0.2254,0.2266]	2419 [2417,2421]		397.80	25658251	25658288	-
Porous Fisher	10061 [9894,10232]	0.1637 [0.1628,0.1646]	2627 [2622,2631]	$\eta : 1.0443[1.0313, 1.0573]$	390.06	25590211	25590260	$< 10^{-20}$
Richards	3180 [3170,3190]	0.1057 [0.1055,0.1059]	2353 [2352,2354]	$\gamma : \infty*$	382.43	**25521509**	**25521559**	$< 10^{-20}$
Generalised Fisher	-	-	-	$\alpha : -$ $\beta : -$	-	-	-	-

Table E.11: Experiment 7, full dataset. Note that for γ in the Richards model, the profile likelihood seems to be monotonically increasing up to the upper bound of $\gamma = 9$ which we have imposed for numerical stability, therefore the true MLE is likely to be very large or infinite.

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1448 [1442,1454]	0.2669 [0.2662,0.2676]	2294 [2292,2296]		384.08	25536527	25536564	-
Porous Fisher	3504 [3467,3543]	0.2374 [0.2367,0.2382]	2337 [2335,2339]	$\eta : 0.4475[0.4414, 0.4539]$	380.95	25508165	25508214	$< 10^{-20}$
Richards	2666 [2658,2675]	0.1199 [0.1196,0.1201]	2241 [2239,2242]	$\gamma : \infty*$	374.07	**25444985**	**25445034**	$< 10^{-20}$
Generalised Fisher	-	-	-	$\alpha : -$ $\beta : -$	-	-	-	-

Table E.12: Experiment 8, full dataset. Similar observations for γ as in Experiment 7.

E.2 Inference results for synthetic dataset

In this section, we present the MLEs and 95% confidence intervals of the parameter values calculated for the synthetic dataset with high noise. Recall that the synthetic dataset is generated with the following parameters (from Eq. (2.13)):

$$D_0 = 1300\ \mu\text{m}^2/\text{h}, \quad r = 0.3\ \text{h}^{-1}, \quad K = 2600\ \text{cell/mm}^2, \quad \alpha = \beta = \gamma = 1, \quad \eta = 0, \tag{E.2}$$

and perturbed with a Gaussian i.i.d. noise with standard deviation of $\sigma = 400$.

Model	D_0	r	K	Parameters unique to model	σ	AIC	BIC	$p-$value
Standard Fisher	1258 [1195,1323]	0.3068 [0.2968,0.3172]	2599 [2588,2611]		406.26	**169314**	**169336**	-
Porous Fisher	1260 [1197,1377]	0.3071 [0.2970,0.3175]	2599 [2588,2611]	$\eta : 0[0, 0.0290]$	406.22	169317	169346	0.4785
Richards	1333 [1173,1513]	0.2809 [0.2358,0.3370]	2599 [2587,2610]	$\gamma : 1.1340[0.8829, 1.4775]$	406.20	169315	169345	0.6679
Generalised Fisher	1304 [1157,1471]	0.3112 [0.2037, 0.4246]	2597 [2585,2609]	$\alpha : 0.9898[0.9430, 1.0529]$ $\beta : 0.9351[0.8286, 1.0446]$	406.19	169317	169354	0.5061

Table E.13: Inference results for the synthetic dataset. The units of the parameters are given in Eq. (2.4).

Note that both AIC and BIC selects the Standard Fisher model, and the χ^2-likelihood ratio test indicates that the three more complicated models do not have significantly greater likelihood compared to the Standard Fisher model. These observations are expected, since the Standard Fisher model is the true model used to generate the dataset.

E.3 Profile likelihoods computed from the experimental datasets

In this section, we present the profile likelihoods for Experiments 2–8 (those for Experiment 1 are presented in Fig. 2.3 and Fig. 2.4 of the main text). The units of the parameters are given in Eq. (2.4).

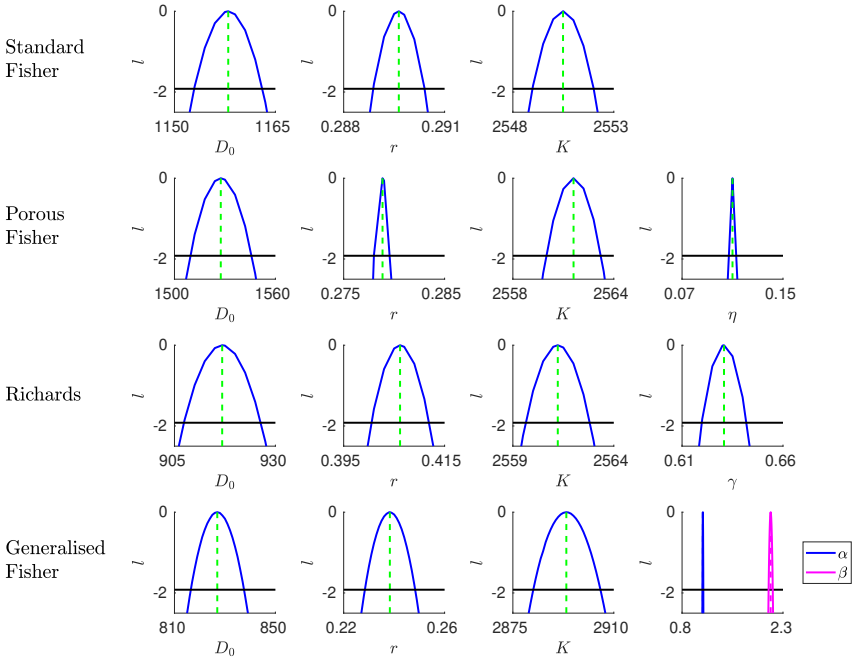


Figure E.1: Profile likelihoods for Experiment 2, full dataset.

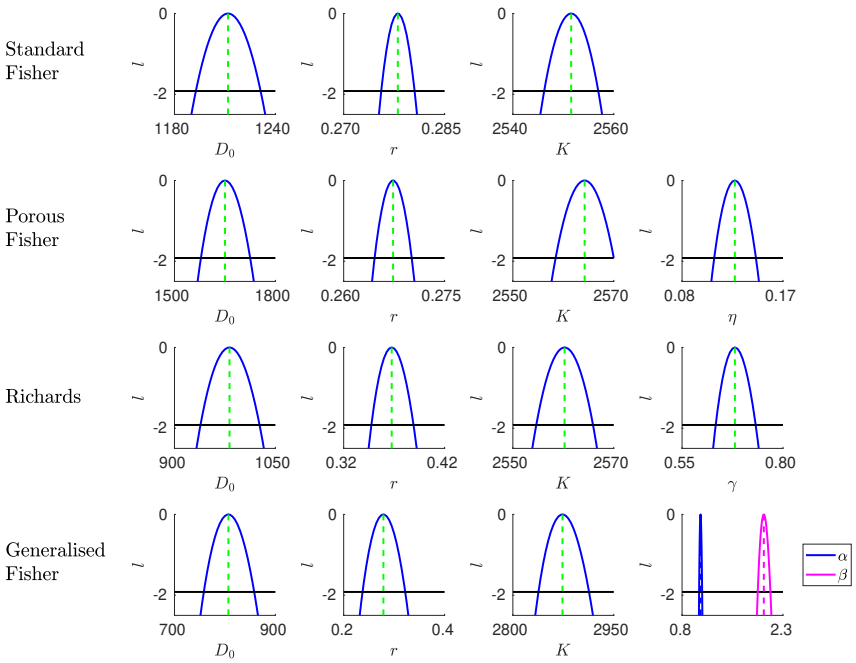


Figure E.2: Profile likelihoods for Experiment 2, radially averaged dataset.

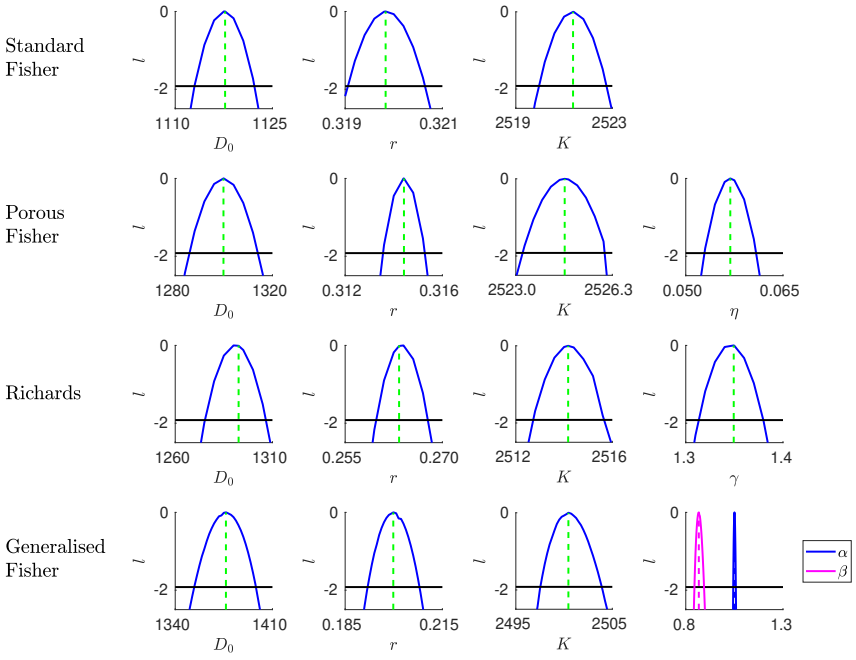


Figure E.3: Profile likelihoods for Experiment 3, full dataset.

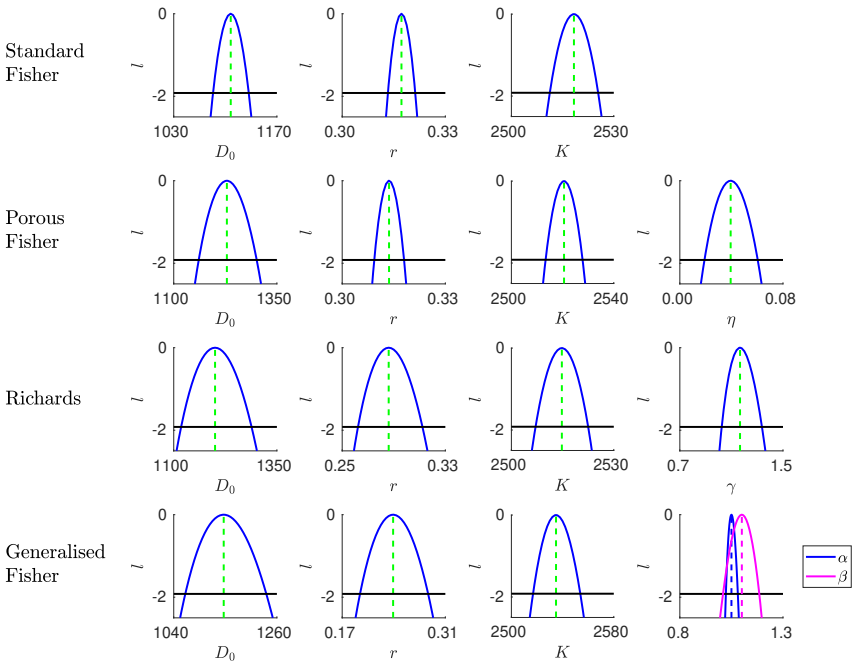


Figure E.4: Profile likelihoods for Experiment 3, radially averaged dataset.

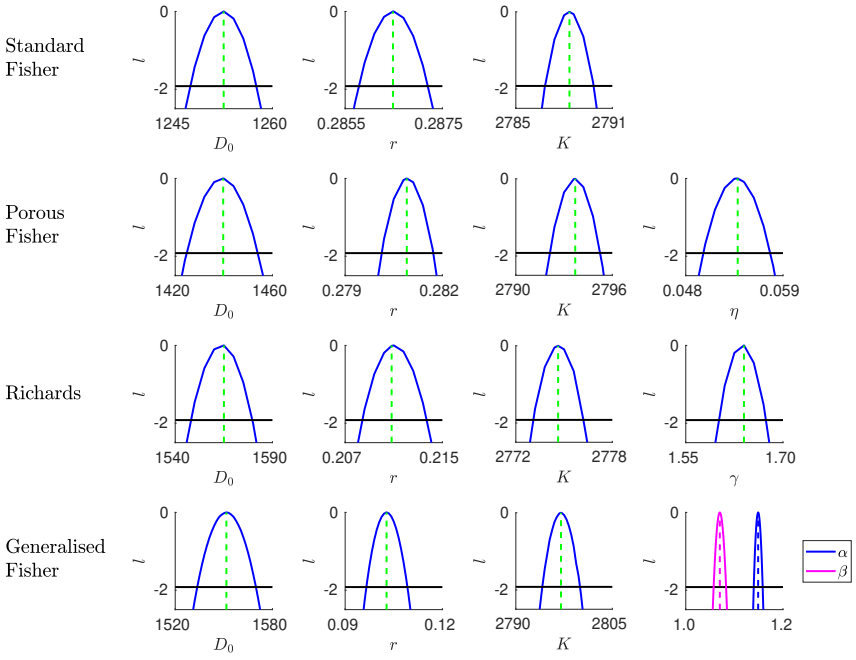


Figure E.5: Profile likelihoods for Experiment 4, full dataset.

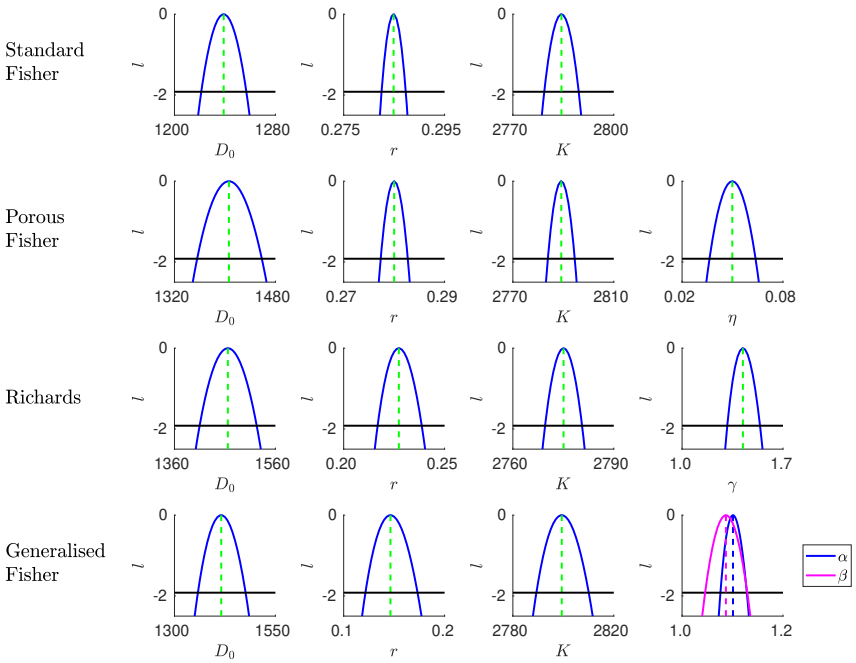


Figure E.6: Profile likelihoods for Experiment 4, radially averaged dataset.

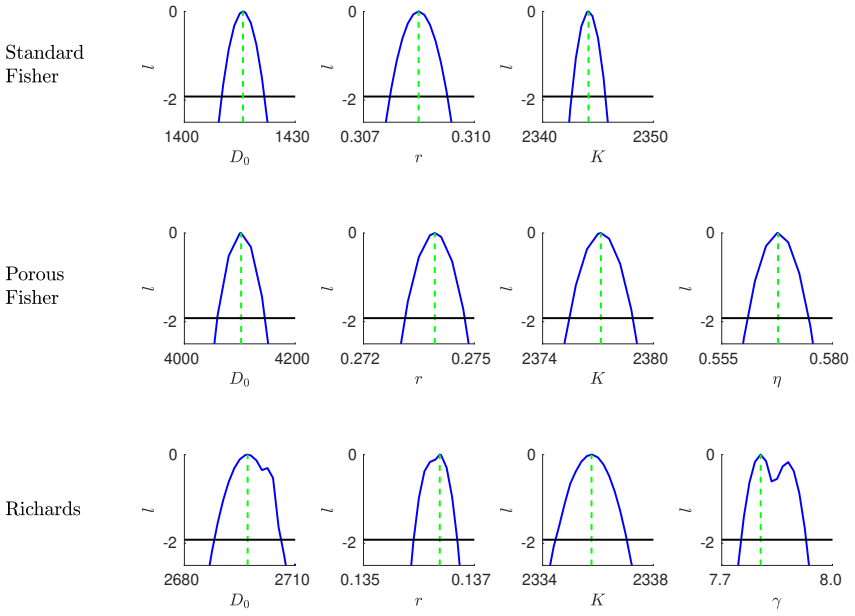


Figure E.7: Profile likelihoods for Experiment 5, full dataset.

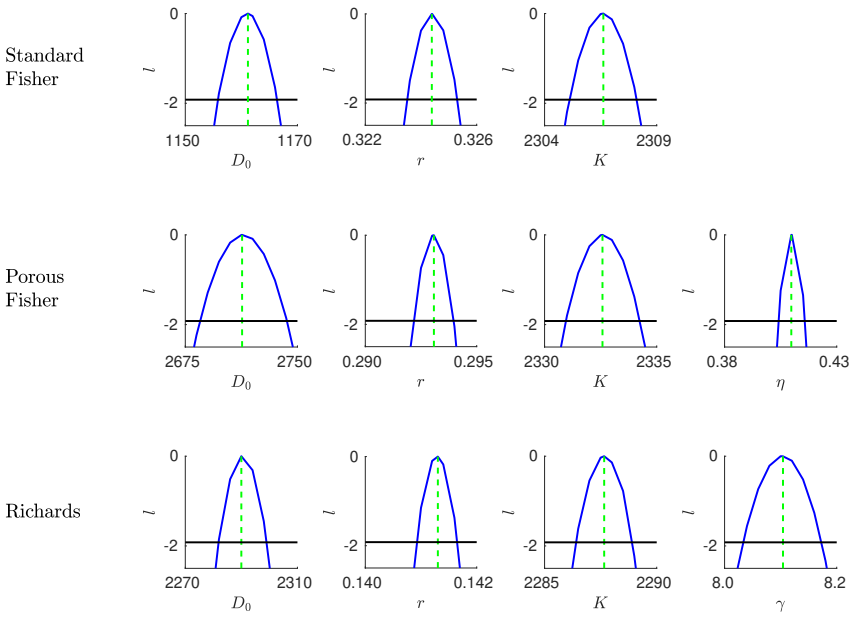


Figure E.8: Profile likelihoods for Experiment 6, full dataset.

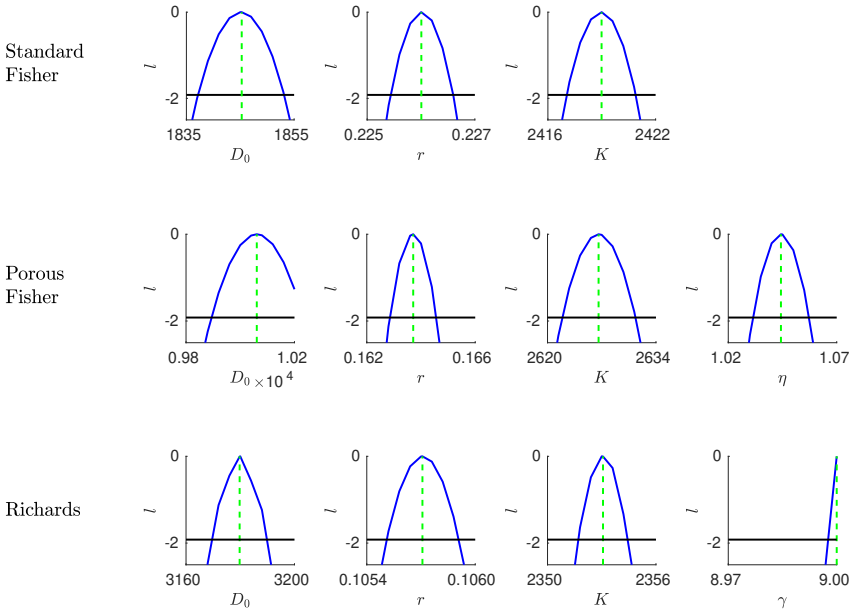


Figure E.9: Profile likelihoods for Experiment 7, full dataset.

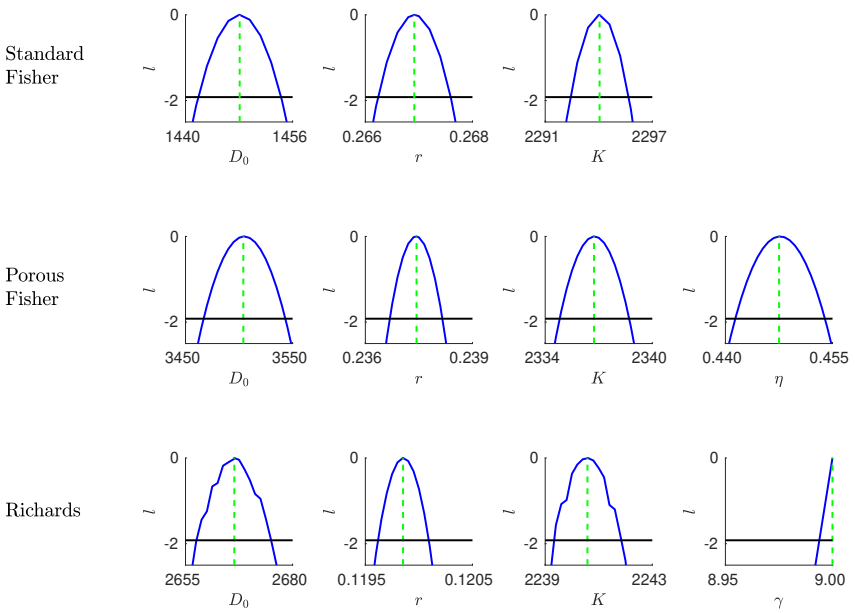


Figure E.10: Profile likelihoods for Experiment 8, full dataset.

E.4 Bivariate profile likelihoods

In this section, we present the bivariate likelihood plots computed from the radially-averaged dataset from Experiment 1. These plots provide intuition into the way the parameter estimates are correlated. The units of the parameters are given in Eq. (2.4). The bivariate profile likelihoods are represented as heat maps, where lighter colours (more yellow) indicates higher likelihood, and darker colours (more blue) indicates lower likelihood. The locations of the MLEs are indicated by the red stars. The black curve encircles the 95% confidence region, as defined in Eq. (2.19).

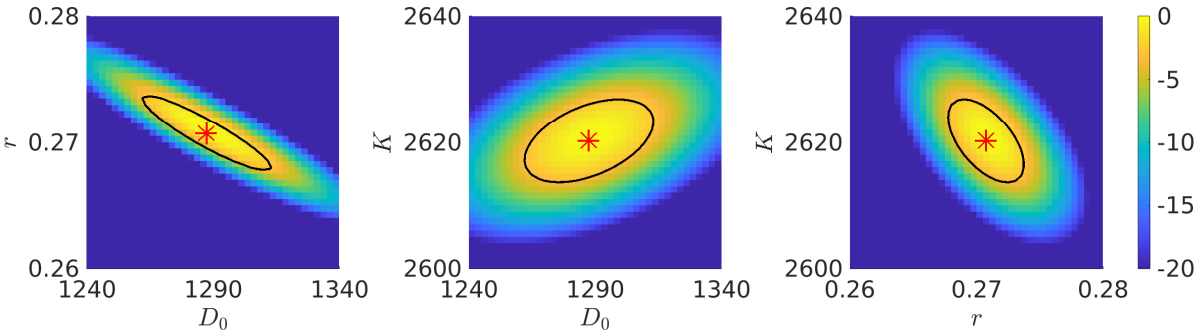


Figure E.11: Bivariate profile likelihoods of the parameters in the Standard Fisher model.

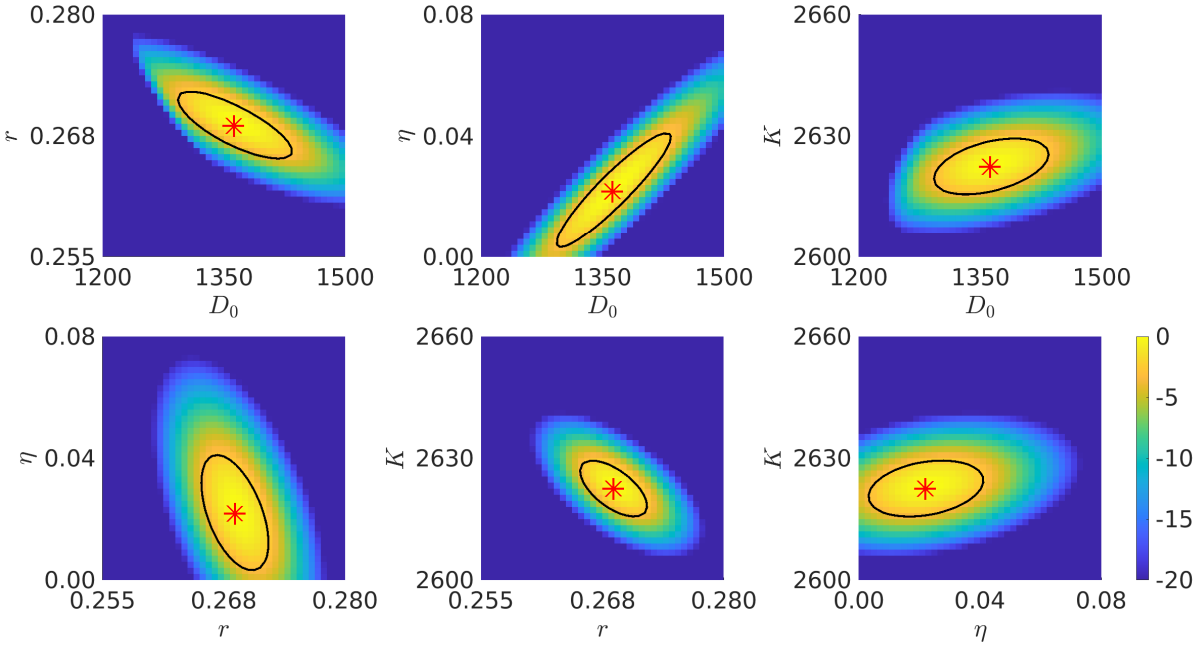


Figure E.12: Bivariate profile likelihoods of the parameters in the Porous Fisher model.

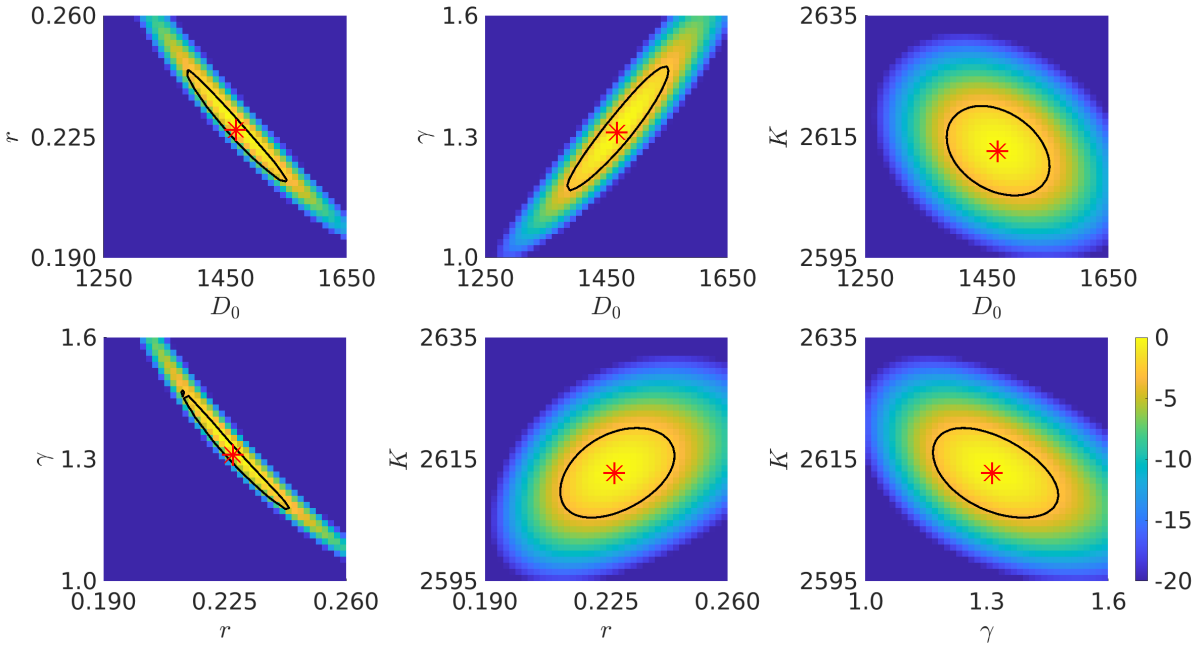


Figure E.13: Bivariate profile likelihoods of the parameters in the Richards model.

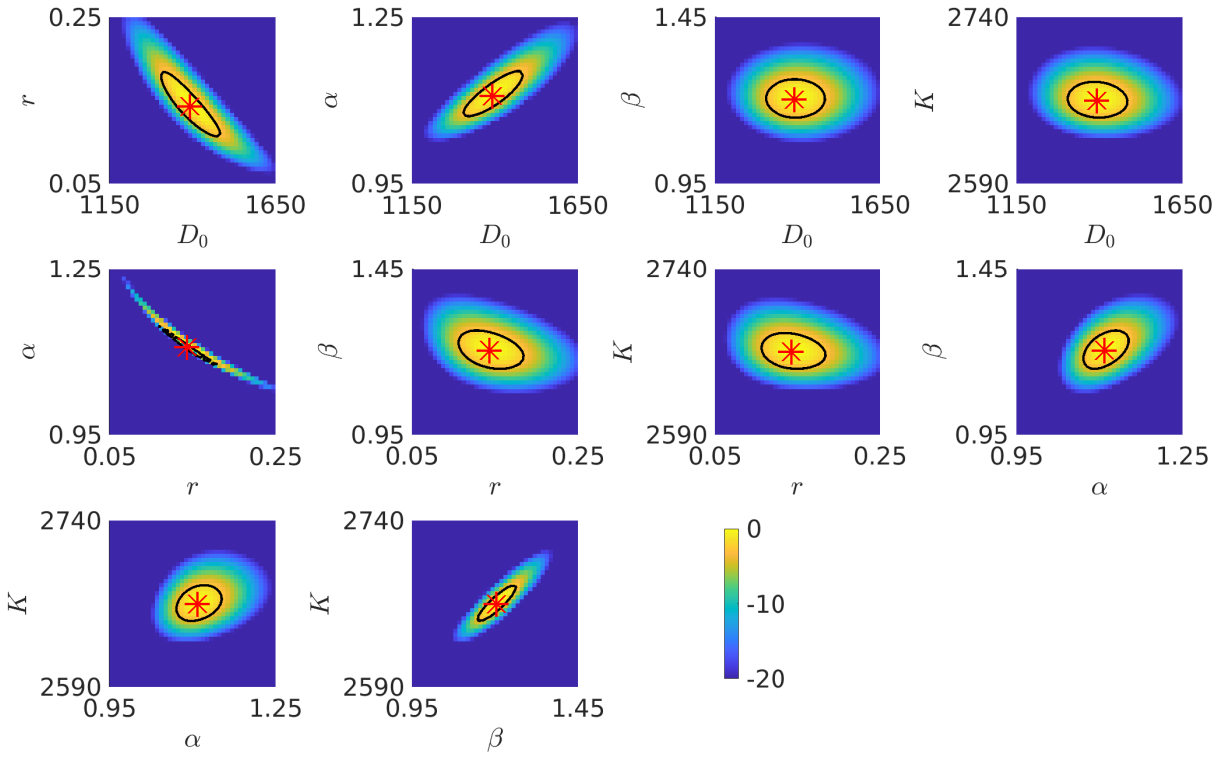


Figure E.14: Bivariate profile likelihoods of the parameters in the Generalised Fisher model.

E.5 Profile likelihoods for down-sampled data

In Fig. E.15, we present the profile likelihoods for the down-sampled datasets. A selection of these are presented in Fig. 2.8 in the main text. The units of the parameters are given in Eq. (2.4).

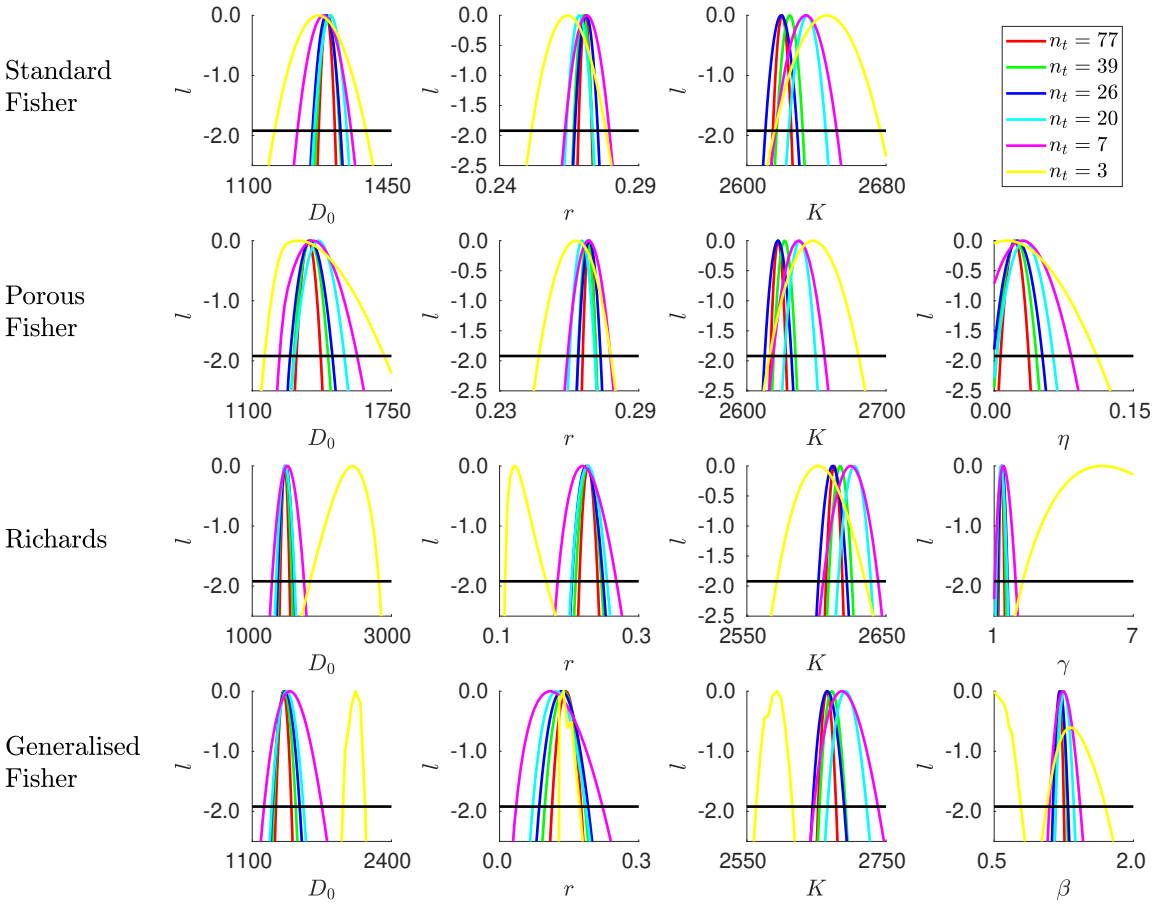


Figure E.15: Profile likelihoods for the down-sampled datasets.

Appendix F

Structural identifiability analysis for the logistic and Richard models

In this appendix, we discuss the notion of structural identifiability in more detail, and perform structural identifiability analysis for the logistic and Richards models used in Ch. 3.

F.1 The notion of structural identifiability

Structural identifiability refers to the ability to uniquely determine the values of the model parameters given a noise-free dataset of infinite precision. As such, it is an intrinsic property of the model itself, in contrast to practical identifiability, which only assumes the availability of a potentially noisy dataset of finite precision. In the context of differential equation models, structural identifiability can be restated as the question of whether there exist two distinct parameter sets that produce the exact same model solutions. The formal definition given below follows [41, 45]. We state the definition in terms of ODEs, but it is straight-forward to generalise to PDEs.

Formally, suppose

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = f(\boldsymbol{x};\boldsymbol{\theta}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0(\boldsymbol{\theta}), \tag{F.1a}$$

$$\boldsymbol{y} = h(\boldsymbol{x};\boldsymbol{\theta}), \tag{F.1b}$$

where $\boldsymbol{x}$ is a vector of state variables, $\boldsymbol{y}$ is a vector of observed quantities, $\boldsymbol{\theta}$ is a vector of P unknown parameters, f encodes the dynamical model, and h is the observation model. Let $\boldsymbol{x}(t;\boldsymbol{\theta})$ denote the model solution given a particular parameter set, and $\boldsymbol{y}(t;\theta) = h(\boldsymbol{x}(t;\boldsymbol{\theta});\boldsymbol{\theta})$ denote the observations corresponding to that parameter set. In Ch. 2, the dynamical model is the Generalised Fisher Model, Eq. (2.3), and in Ch. 3, the dynamical model is the Richards growth model, Eq. (3.1). In both cases, the state variables are assumed to be fully and directly

observable, so h is simply the identity function. The parameters $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^P$ are constrained to be within the set Θ, which encodes known restrictions on parameter values (e.g. certain parameters are known to be non-negative).

The model in Eq. (F.1) is called

- globally structurally identifiable, if for almost any $\boldsymbol{\theta} \in \Theta$,

$$\boldsymbol{y}(t;\boldsymbol{\theta}) = \boldsymbol{y}(t;\boldsymbol{\theta}') \ \forall t \Rightarrow \boldsymbol{\theta} = \boldsymbol{\theta}';$$

- locally structurally identifiable, if for almost any $\boldsymbol{\theta} \in \Theta$, there exist $\epsilon > 0$ such that

$$\boldsymbol{y}(t;\boldsymbol{\theta}) = \boldsymbol{y}(t;\boldsymbol{\theta}') \ \forall t, \text{ and } ||\boldsymbol{\theta} - \boldsymbol{\theta}'|| < \epsilon \Rightarrow \boldsymbol{\theta} = \boldsymbol{\theta}';$$

- structurally non-identifiable, if it is not locally structurally identifiable.

Note that globally structurally identifiable is a strictly stronger property than locally structurally identifiable. In the context of practical identifiability analysis with profile likelihoods, locally (but not globally) structurally identifiable models can be reflected by a multi-modal profile likelihood (as illustrated in Fig. 1.1(e)), and structurally non-identifiable models tend to have a profile likelihood (as in Fig. 1.1(c)). In this thesis, both of these cases are considered practically non-identifiable, as we do not make the same global-local distinction for practical identifiability.

There are a few methods for determining structural identifiability for ODEs, including the Taylor series approach [146], the Laplace transform approach [18] and the differential algebra approach [121]. Some of these are reviewed in [45] and demonstrated in [44].

F.2 Structural identifiability analysis with the Taylor series approach

In this section, we perform structural identifiability analysis for the logistic and Richards models used in Ch. 3, using the Taylor series approach developed in [146] and illustrated in [44]. The Taylor series approach is suitable for non-linear models considered here, and is relatively simple compared to the other methods for non-linear models. This analysis guides the experimental design in Ch. 3 by showing whether certain control inputs can, at least in theory, allow previously non-identifiable models to become identifiable.

Recall that the Richards model (Eq. (3.1)) can be written as

$$\frac{\mathrm{d}C}{\mathrm{d}t} = rC\left[1 - \left(\frac{C}{K}\right)^{\gamma}\right] - \delta C = f(C;(r,\delta,\gamma,K)), \quad C(0) = C_0, \tag{F.2}$$

where $\boldsymbol{\theta} = (r, \delta, \gamma, K)$ are the model parameters, and the logistic model is the special case with $\gamma = 1$ fixed.

The Taylor series approach for structural identifiability analysis assumes that the solution is smooth and can be expressed as a Taylor series in terms of time, and certain regularity conditions on the right-hand side of the ODE model holds. Under these assumptions, [146] showed that the model is structurally identifiable if the parameters can be uniquely determined if measurements of the system and all its derivatives in time at $t = 0$ are given, that is, $C(0)$ and $C^{(k)}$ for $k \geq 1$. Knowing the values of these derivatives is equivalent to knowing the coefficients of the Taylor series of the model solution at $t = 0$. The leading order derivatives of the model solution can be written as:

$$C(0) = C_0 \tag{F.3a}$$

$$C'(0) = rC_0(1 - (C_0/K)^\gamma) - \delta C_0 \tag{F.3b}$$

$$C''(0) = C'(0)[r - r(1+\gamma)(C_0/K)^\gamma - \delta] \tag{F.3c}$$

$$C'''(0) = C''(0)^2/C'(0) - C'(0)^2[r(\gamma+1)\gamma C_0^{\gamma-1}/K^\gamma] \tag{F.3d}$$

$$C''''(0) = \begin{cases} A - C'(0)^2[r(\gamma+1)\gamma(\gamma-1)C_0^{\gamma-2}/K^\gamma] & \text{if } \gamma \neq 1 \\ A & \text{if } \gamma = 1 \end{cases}, \tag{F.3e}$$

where $'$ denotes derivatives in time, and A denotes terms that can be written exclusively using known quantities. For both the Richards and logistic models, Eq. (F.3a) shows that the initial condition, C_0, can be determined from data, if it weren't already known.

For the Richards model, Eq. (F.3d) and Eq. (F.3e) together provides two polynomial constraints on γ and the parameter combination r/K^γ, which means that in general, there are at most a finite number of solutions for both, so γ and r/K^γ are at least locally identifiable. Given values for γ and r/K^γ, Eq. (F.3c) allows $r - \delta$ to be uniquely identified, so $r - \delta$ is at least locally identifiable. However, the parameters r, K, δ cannot be identified individually, since r always appears in the combinations of either $r - \delta$ or r/K^γ in Eq. (F.3b) or the higher derivatives, so the Richards model is overall non-identifiable. If the explicit death term is not present, or equivalently, $\delta = 0$ is fixed, then all parameters are at least locally identifiable.

For the logistic model, which has $\gamma = 1$ fixed, Eq. (F.3d) shows that the parameter combination r/K can be determined from data, which together with Eq. (F.3c), allows $r - \delta$ to be uniquely determined. None of the three parameters can be uniquely determined individually, for the same reason as for the Richards model. Therefore, the logistic models is structurally non-identifiable. However, knowing the value of one of r, δ, K will allow the other two parameters to

be uniquely identified. Therefore, the logistic models without the explicit death term (Eq. (F.2) with $\gamma = 1$ and $\delta = 0$ fixed) is globally identifiable.

Appendix G

Computational considerations for optimal control problems

In Sec. 3.6, we explored an optimal control problem, with the objective of maximising the difference between the solutions of two model-parameter combinations. There are three candidates for control variables in this problem. In the main text, we considered only the case of applying one control variable at a time, using the FBS algorithm to solve the resulting optimal control problem. In this appendix, we will implement the direct optimisation and the hybrid method, and compare their performance to the FBS algorithm. Furthermore, we will consider the case of applying multiple control variables at the same time.

We will focus on the logistic model with a death term and additive control in this appendix. Recall that the model can be written as

$$\begin{aligned}\frac{\mathrm{d}C(t)}{\mathrm{d}t} &= (r+u_r(t))C\left[1-\left(\frac{C}{K-u_K(t)}\right)^{\gamma}\right]-(\delta+u_\delta(t))C,\\ &= g_a(C;\boldsymbol{\theta},u_r(t),u_\delta(t),u_K(t)),\end{aligned} \tag{G.1}$$

and the objective of the control problem is

$$\max_{\boldsymbol{u}\in\mathcal{U}} J[\boldsymbol{u}] = \max_{\boldsymbol{u}\in\mathcal{U}} \int_0^T \left[C_{\mathrm{model},1}(t;\boldsymbol{u}(t))-C_{\mathrm{model},2}(t;\boldsymbol{u}(t))\right]^2-\boldsymbol{\alpha}\cdot\boldsymbol{u}(t)\mathrm{d}t. \tag{G.2}$$

G.1 Comparison of numerical algorithms

For the application of direct optimisation to the continuous control problem, we discretise the time interval $[0,T]$ into n_t evenly spaced points $t_i = i\Delta t$, where $\Delta t = T/n_t$. The control $u(t)$ is parameterised as its values at t_i, interpolated linearly between these values, where we choose $n_t = 100$. We use *fmincon* with the default interior point algorithm [36], with a termination criterion based on the Karush-Kuhn-Tucker (KKT) first-order optimality measure, which is akin to the first derivative condition in the presence of constraints. This is a reasonable choice, as

the interior-point algorithm was found to be suitable for directly optimising a variety of control problems in [110]. We initialise the optimisation at $u(t) \equiv 0$ when applying direct optimisation alone.

The caveat of direct optimisation is that a general-purpose algorithm cannot exploit the structure of the original control problem. We also need a sufficiently large n_t for the discrete-time system to be able to approximate the original continuous-time system well, which means that the discrete-time optimisation problem is very high-dimensional, and we expect the optimisation algorithm to converge slowly, and to be sensitive to the choice of initial point where we start our search. To partially mitigate these problems with direct optimisation, we also consider a hybrid method, where we initialise the aforementioned direct optimisation procedure at the output of the FBS algorithm.

In this section, we consider the case of using u_K as the control variable to control the model in Eq. (G.1), with the other two control variables, u_r and u_δ, fixed at zero. The outputs of direct optimisation and the hybrid methods are given in Fig. G.1, along with the output of the FBS algorithm, originally given in Fig. 3.16. Notice that the optimal controls given by all three methods are very close. There are some noticeable numerical artifacts in the form of oscillations in the output of the direct optimisation procedure. The FBS algorithm terminated after 73 iterations, and produced a control with the objective value of $J = -784470$ $(\text{cells/mm}^2)^2$. Direct optimisation terminated after 991 iterations of *fmincon*, although each of its iterations, which only require evaluating the objective function J and its partial derivatives with respect to the discretised control variables, is much faster than one iteration of FBS. Its output achieved an objective value of $J = -782476$ $(\text{cells/mm}^2)^2$, which is close to, but a little worse, than that of the FBS algorithm. Finally, the hybrid method required 67 iterations of *fmincon* (in addition to the computation of the FBS algorithm), and achieved an objective value of $J = -785081$ $(\text{cells/mm}^2)^2$, the best out of the three methods.

These results show that the FBS algorithm should be preferred over the direct optimisation algorithm with naive or uninformed initialisation. However, direct optimisation can be initialised at the output of the FBS algorithm to further refine the control if desired, forming what we termed the hybrid algorithm.

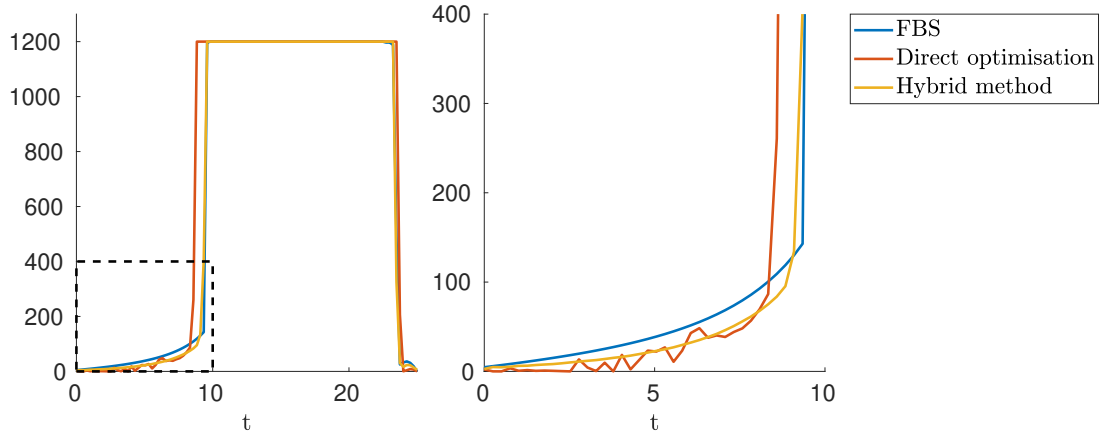


Figure G.1: Results from applying the FBS algorithm, direct optimisation, and the hybrid method to the optimal control problem in Eqs. (G.1)–(G.2). The plot on the right is a zoomed-in version of the area inside the dashed rectangle in the plot on the left, where the three solutions noticeably differ. The control found by the FBS algorithm is the same as the one presented in Fig. 3.16. The units are $[t] = \text{h}, [u_K] = \text{cells/mm}^2$.

G.2 Multiple control variables

In this section, we consider simultaneously using u_K and u_r as control variables in the optimal control problem in Eq. (G.1) and Eq. (G.2). We would like to see if applying both control variables can result in further gains in the control objective. The constraints we impose on the control variables are $u_{K,\max} = 1200$ cells/mm^2 and $u_{r,\max} = 1$ h^{-1}, and the cost of control is $\alpha_K = 0.1$ cells/mm^2, $\alpha_r = 700000$ h(cells/mm^2)^2.

We have encountered numerical challenges in solving the optimal control problem in this scenario. For the FBS algorithm, the $\boldsymbol{u}^{\text{new}}$ calculated in successive iterations jumps wildly between two values at a few points in time, and these points differ in each iteration. The final $\boldsymbol{u}$, which is a weighted average of these $\boldsymbol{u}^{\text{new}}$ over the iterations, appears jagged as a result. It achieved an objective value of $J = -6756694$ (cells/mm^2)^2. The reason for this is that the Hamiltonian H, viewed as a function of $\boldsymbol{u}$, tends to have two local maxima at these points in time. The value of H at these two local maxima is similar, but their location is far apart, so $\boldsymbol{u}^{\text{new}}$ in successive iterations can switch from one local maximum to another, causing the jumps. In this case, the conditions for convergence of the FBS algorithm given by [129] are not satisfied, namely, the continuity and differentiability of the function $h(\boldsymbol{x}, \boldsymbol{\lambda}) = \text{argmax}_{\boldsymbol{u}} H(\boldsymbol{x}, \boldsymbol{\lambda}; \boldsymbol{u})$ are violated. The discontinuity is caused by the existence of two local maxima of H at similar magnitude, and the global maximum shifts from one to another when $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ are perturbed slightly while the locations of the two local maxima remain far apart.

A similar issue is observed when applying direct optimisation, where the control in successive

iterations oscillates between the same two values as in the FBS algorithm. The objective value at the final iteration was $J = -7759570$ (cells/mm^2)^2, whereas the hybrid method achieved an objective value of $J = -8089009$ (cells/mm^2)^2.

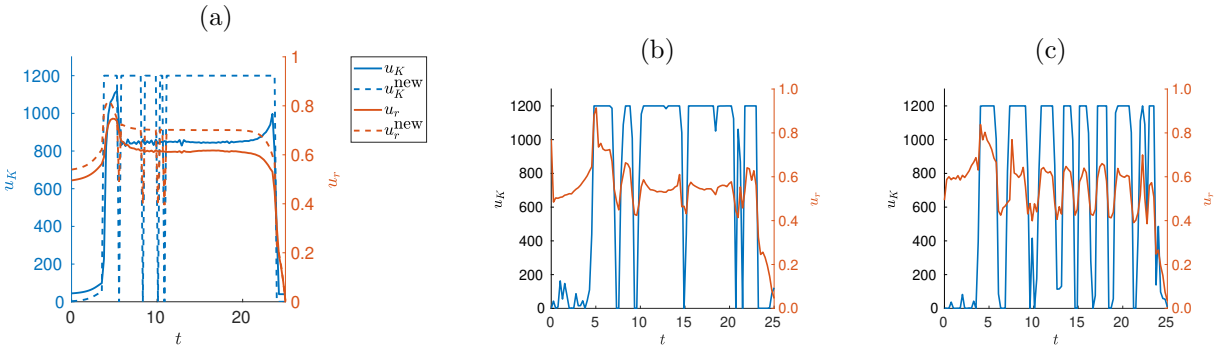


Figure G.2: The results from applying the three numerical optimisation methods to the optimal control problem in Eqs. (G.1) and (G.2). (a) The results of the FBS algorithm (Algorithm 2), showing $\boldsymbol{u}(t)$ (solid lines) and updates $\boldsymbol{u}^{\text{new}}(t)$ (dashed lines) at the last iteration. (b) The results of direct optimisation, initialised at $u_K = 500$ cells/mm^2, $u_r = 0.1$ h^{-1}, discretised with $n_t = 100$ points. (c) The results of the hybrid method, where direct optimisation was initialised with the output of FBS in (a). The units are $[t] = \text{h}$, $[u_K] = \text{cells/mm}^2$, $[u_r] = \text{h}^{-1}$.

With two control variables active at the same time, it is difficult to make sense of the control strategy intuitively. We postulate that the true optimal control for this problem might not be unique, and that the locations of the two local maxima of H represent two solutions an optimal control might take at those points in time.

Appendix H

Linear stability analysis for the Schnakenberg model of pattern formation

In this appendix, we perform a linear stability analysis of the Schnakenberg model. We used these results to guide parameter value selection in Ch. 4. The methods employed here follow [186], are entirely standard, and presented for completeness.

Firstly, we linearise the system of PDEs (Eq. (4.1)) around a spatially uniform steady state (u_*, v_*) by letting $(U, V) = (u, v) - (u_*, v_*)$. The linearized system can be written as

$$\frac{\partial}{\partial t}\begin{bmatrix} U \\ V \end{bmatrix} = D\frac{\partial^2}{\partial x^2}\begin{bmatrix} U \\ V \end{bmatrix} + J\begin{bmatrix} U \\ V \end{bmatrix}, \tag{H.1}$$

where $D = diag(D_u, D_v)$, and J is the Jacobian of the system at (u_*, v_*). For a DDI, we require that (u_*, v_*) is stable in the absence of diffusion, that is, the eigenvalues of J must have negative real parts, but is unstable in the presence of diffusion. To investigate linear stability, we impose the ansatz,

$$U = a\exp[\sigma t + i(k_x x + k_y y)],$$
$$V = b\exp[\sigma t + i(k_x x + k_y y)].$$

The zero-flux boundary conditions imply that only modes with

$$k_x = \frac{n^2\pi^2}{L_x^2}, \quad k_y = \frac{m^2\pi^2}{L_y^2},$$

for some integers m, n (where at least one of these is non-zero) are admissible. The real part of σ determines the linear stability of (u_*, v_*), that is, if a non-trivial solution with $Re(\sigma) > 0$ exists for any admissible k_x, k_y, then (u_*, v_*) is linearly unstable, and a DDI can occur.

If we let $k^2 = k_x^2 + k_y^2$, then it can be shown that $\sigma = \sigma(k^2)$ is the eigenvalue of $M = J - k^2 D$, and the condition for a DDI can be written as (for detailed derivation, see [52, Ch. 11.4])

$$J_{11} D_v + J_{22} D_u > 2\sqrt{D_u D_v \det(J)}, \tag{H.2}$$

where J_{ij} denote the $(i,j)^{th}$ entry of the Jacobian J.

For the Schnakenberg model (Eq. (4.3)), where the uniform steady state (u_*, v_*) is given in Eq. (4.5), we find that the inequality (H.2) simplifies to

$$\frac{2D_v b/u_* - D_u u_*^2 - D_v}{2\sqrt{D_v D_u}} - u_* > 0,$$

which holds for the parameter values given in Eq. (4.4) with $W = 0$, but does not hold when $W = W_{\max}$. This is the desired property for our choice of the parameters. Similar calculations can be done for the CDIMA model.

Moreover, in the case where a DDI is possible, by plotting the real parts of the eigenvalues of M as a function of k, we can determine the range of wave numbers k for which the corresponding mode is unstable. These are the only modes that will initially grow from the initial conditions. For the Schnakenberg model with $b = 1.4$, the range is $k \in (0.40, 0.82)$, and with $b = 1.6$, the range is $k \in (0.16, 0.24)$.

Bibliography

[1] K. Aho, D. Derryberry, and T. Peterson. Model selection for ecologists: the worldviews of AIC and BIC. *Ecology*, 95(3):631–636, 2014. arXiv:43495189.

[2] M. Alber, A. B. Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold, and E. Kuhl. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *npj Digital Medicine*, 2(1):1–11, 2019. doi:10.1038/s41746-019-0193-y.

[3] G. An, B. G. Fitzpatrick, S. Christley, P. Federico, A. Kanarek, R. M. Neilan, M. Oremland, R. Salinas, R. Laubenbacher, and S. Lenhart. Optimization and control of agent-based models in biology: A perspective. *Bulletin of Mathematical Biology*, 79(1):63–87, 2017. doi:10.1007/s11538-016-0225-6.

[4] C. Andrieu and J. Thoms. A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373, 2008. doi:10.1007/s11222-008-9110-y.

[5] M. Anguelova, J. Karlsson, and M. Jirstrand. Minimal output sets for identifiability. *Mathematical Biosciences*, 239(1):139–153, 2012. doi:10.1016/j.mbs.2012.04.005.

[6] G. Ariel and A. Ayali. Locust collective motion and its modeling. *PLOS Computational Biology*, 11(12):e1004522, 2015. doi:10.1371/journal.pcbi.1004522.

[7] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32:797–823, 1995. doi:10.1137/0732037.

[8] A. C. Atkinson. A comparison of two criteria for the design of experiments for discriminating between models. *Technometrics*, 23(3):301–305, 1981. arXiv:1267795, doi:10.2307/1267795.

[9] Azul. Zebrafish, 2005. From https://commons.wikimedia.org/wiki/File:Zebrafisch.jpg, image used under CC-0 license. URL: https://commons.wikimedia.org/wiki/File:Zebrafisch.jpg.

[10] F. Bachmann, G. Koch, M. Pfister, G. Szinnai, and J. Schropp. OptiDose: Computing the individualized optimal drug dosing regimen using optimal control. *Journal of Optimization Theory and Applications*, 189(1):46–65, 2021. doi:10.1007/s10957-021-01819-w.

[11] R. Bailleul, C. Curantz, C. D.-T. Dinh, M. Hidalgo, J. Touboul, and M. Manceau. Symmetry breaking in the embryonic skin triggers directional and sequential plumage patterning. *PLOS Biology*, 17:e3000448, 2019. doi:10.1371/journal.pbio.3000448.

[12] R. E. Baker, J.-M. Peña, J. Jayamohan, and A. Jérusalem. Mechanistic models versus machine learning, a fight worth fighting for the biological community? *Biology Letters*, 14(5):20170660, 2018. doi:10.1098/rsbl.2017.0660.

[13] S. Bandara, J. P. Schlöder, R. Eils, H. G. Bock, and T. Meyer. Optimal experimental design for parameter estimation of a cell signaling model. *PLOS Computational Biology*, 5(11):e1000558, 2009. doi:10.1371/journal.pcbi.1000558.

[14] J. Bard and I. Lauder. How well does Turing's theory of morphogenesis work? *Journal of Theoretical Biology*, 45(2):501–531, 1974. doi:10.1016/0022-5193(74)90128-3.

[15] I. Barrass, E. J. Crampin, and P. K. Maini. Mode transitions in a model reaction-diffusion system driven by domain growth and noise. *Bulletin of Mathematical Biology*, 68:981–995, 2006. doi:10.1007/s11538-006-9106-8.

[16] R. Barrio. A two-dimensional numerical study of spatial pattern formation in interacting turing systems. *Bulletin of Mathematical Biology*, 61:483–505, 1999. doi:10.1006/bulm.1998.0093.

[17] R. A. Barrio, R. E. Baker, B. Vaughan, K. Tribuzy, M. R. de Carvalho, R. Bassanezi, and P. K. Maini. Modeling the skin pattern of fishes. *Physical Review E*, 79:031908, 2009. doi:10.1103/PhysRevE.79.031908.

[18] R. Bellman and K. J. Åström. On structural identifiability. *Mathematical Biosciences*, 7(3):329–339, 1970. doi:10.1016/0025-5564(70)90132-X.

[19] R. Bellman, I. Glicksberg, and O. Gross. On the "bang-bang" control problem. *Quarterly of Applied Mathematics*, 14(1):11–18, 1956. doi:10.1090/qam/78516.

[20] R. Bellmann. *Dynamic Programming*. Princeton University Press, 1957.

[21] C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, 2006.

[22] T. Biswas, S. Dharmatti, and M. T. Mohan. Maximum principle for some optimal control problems governed by 2D nonlocal Cahn–Hillard–Navier–Stokes equations. *Journal of Mathematical Fluid Mechanics*, 22(3):34, 2020. doi:10.1007/s00021-020-00493-8.

[23] J. Bongard and H. Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007. doi:10.1073/pnas.0609476104.

[24] L. Boninsegna, F. Nüske, and C. Clementi. Sparse learning of stochastic dynamical equations. *The Journal of Chemical Physics*, 148(24):241723, 2018. doi:10.1063/1.5018409.

[25] D. Bortz and P. Nelson. Model selection and mixed-effects modeling of HIV infection dynamics. *Bulletin of Mathematical Biology*, 68(8):2005–2025, 2006. doi:10.1007/s11538-006-9084-x.

[26] A. P. Browning, T. D. Lewin, R. E. Baker, P. K. Maini, E. G. Moros, J. Caudell, H. M. Byrne, and H. Enderling. Predicting radiotherapy patient outcomes with real-time clinical data using mathematical modelling. *arXiv*, (arXiv:2201.02101), 2023. arXiv:2201.02101, doi:10.48550/arXiv.2201.02101.

[27] A. P. Browning and M. J. Simpson. Geometric analysis enables biological insight from complex non-identifiable models using simple surrogates. *arXiv*, (arXiv:2208.01868), 2022. arXiv:2208.01868, doi:10.48550/arXiv.2208.01868.

[28] A. P. Browning, M. Tască, C. Falcó, and R. E. Baker. Structural identifiability analysis of linear reaction-advection-diffusion processes in mathematical biology. *arXiv*, (arXiv:2309.15326), 2023. arXiv:2309.15326, doi:10.48550/arXiv.2309.15326.

[29] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi:10.1073/pnas.1517384113.

[30] J. A. Bull, F. Mech, T. Quaiser, S. L. Waters, and H. M. Byrne. Mathematical modelling reveals cellular dynamics within tumour spheroids. *PLOS Computational Biology*, 16(8):e1007961, 2020. doi:10.1371/journal.pcbi.1007961.

[31] B. Bunow, J.-P. Kernevez, G. Joly, and D. Thomas. Pattern formation by reaction-diffusion instabilities: application to morphogenesis in Drosophila. *Journal of Theoretical Biology*, 84:629–649, 1980. doi:10.1016/S0022-5193(80)80024-5.

[32] A. L. Burke, T. A. Duever, and A. Penlidis. Model discrimination via designed experiments: discriminating between the terminal and penultimate models on the basis of composition data. *Macromolecules*, 27(2):386–399, 1994. doi:10.1021/ma00080a011.

[33] K. P. Burnham and D. R. Anderson. Multimodel inference: understanding AIC and BIC in model selection. *Sociological Methods & Research*, 33(2):261–304, 2004. doi:10.1177/0049124104268644.

[34] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, and I. Palunko. Reinforcement learning for control: Performance, stability, and deep approximators. *Annual Reviews in Control*, 46:8–28, 2018. doi:10.1016/j.arcontrol.2018.09.005.

[35] A. Buttenschön, Y. Liu, and L. Edelstein-Keshet. Cell size, mechanical tension, and GTPase signaling in the single cell. *Bulletin of Mathematical Biology*, 82:28, 2020. doi:10.1007/s11538-020-00702-5.

[36] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999. doi:10.1137/S1052623497325107.

[37] J.-B. Caillau, R. Ferretti, E. Trélat, and H. Zidani. An algorithmic guide for finite-dimensional optimal control problems. In E. Trélat and E. Zuazua, editors, *Handbook of Numerical Analysis*, volume 24 of *Numerical Control: Part B*, pages 559–626. Elsevier, 2023. doi:10.1016/bs.hna.2022.11.006.

[38] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1988. doi:10.1007/978-3-642-84108-8.

[39] J. H. E. Cartwright. Labyrinthine Turing pattern formation in the cerebral cortex. *Journal of Theoretical Biology*, 217:97–103, 2002. doi:10.1006/jtbi.2002.3012.

[40] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, page 201906995, 2019. doi:10.1073/pnas.1906995116.

[41] M. J. Chapman, K. R. Godfrey, M. J. Chappell, and N. D. Evans. Structural identifiability of non-linear systems using linear/non-linear splitting. *International Journal of Control*, 76(3):209–216, 2003. doi:10.1080/0020717031000067420.

[42] B. H. Chen, S. P. Asprey, S. K. Bermingham, A. M. Neumann, and H. J. M. Kramer. On the design of optimally informative experiments for model discrimination among dynamic crystallization process models. In *4th Int. Conf. on Foundations of Computer-Aided Process Operations*, pages 455–458, 2003.

[43] L. Chen, K. Painter, C. Surulescu, and A. Zhigun. Mathematical models for cell migration: a non-local perspective. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375(1807):20190379, 2020. doi:10.1098/rstb.2019.0379.

[44] S. V. Chin and M. J. Chappell. Structural identifiability and indistinguishability analyses of the minimal model and a euglycemic hyperinsulinemic clamp model for glucose–insulin dynamics. *Computer Methods and Programs in Biomedicine*, 104(2):120–134, 2011. doi:10.1016/j.cmpb.2010.08.012.

[45] O.-T. Chis, J. R. Banga, and E. Balsa-Canto. Structural identifiability of systems biology models: A critical comparison of methods. *PLOS ONE*, 6(11):e27755, 2011. doi:10.1371/journal.pone.0027755.

[46] K.-H. Cho, S.-Y. Shin, W. Kolch, and O. Wolkenhauer. Experimental design in systems biology, based on parameter sensitivity analysis using a Monte Carlo method: A case study for the TNFα-mediated NF-κ B signal transduction pathway. *SIMULATION*, 79(12):726–739, 2003. doi:10.1177/0037549703040943.

[47] S. Christley, R. M. Neilan, M. Oremland, R. Salinas, and S. Lenhart. Optimal control of sugarscape agent-based model via a PDE approximation model. *Optimal Control Applications and Methods*, 38(4):473–497, 2017. doi:10.1002/oca.2265.

[48] C. Cobos, L. Muñoz, M. Mendoza, E. León, and E. Herrera-Viedma. Fitness function obtained from a genetic programming approach for web document clustering using evolutionary algorithms. In J. Pavón, N. D. Duque-Méndez, and R. Fuentes-Fernández, editors, *Advances in Artificial Intelligence – IBERAMIA 2012*, pages 179–188, Berlin, Heidelberg, 2012. Springer. doi:10.1007/978-3-642-34654-5_19.

[49] E. J. Crampin, W. W. Hackborn, and P. K. Maini. Pattern formation in reaction-diffusion models with nonuniform domain growth. *Bulletin of Mathematical Biology*, 64:747–769, 2002. doi:10.1006/bulm.2002.0295.

[50] E. J. Crampin, E. A. Gaffney, and P. K. Maini. Reaction and diffusion on growing domains: scenarios for robust pattern formation. *Bulletin of Mathematical Biology*, 61:1093–1120, 1999. doi:10.1006/bulm.1999.0131.

[51] A. de Pablo and A. Sánchez. Travelling wave behaviour for a Porous-Fisher equation. *European Journal of Applied Mathematics*, 9(3):285–304, 1998. doi:10.1017/S0956792598003465.

[52] L. Edelstein-Keshet. *Mathematical Models in Biology*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2005. doi:10.1137/1.9780898719147.

[53] B. Ermentrout and J. E. Enderby. Stripes or spots? Nonlinear effects in bifurcation of reaction-diffusion equations on the square. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 434:413–417, 1991. doi:10.1098/rspa.1991.0100.

[54] D. Espie and S. Macchietto. The optimal design of dynamic experiments. *AIChE Journal*, 35(2):223–229, 1989. doi:10.1002/aic.690350206.

[55] C. Falcó, D. J. Cohen, J. A. Carrillo, and R. E. Baker. Quantifying tissue growth, shape and collision via continuum models and Bayesian inference. *Journal of The Royal Society Interface*, 20(204):20230184, 2023. doi:10.1098/rsif.2023.0184.

[56] C. Falcó, D. J. Cohen, J. A. Carrillo, and R. E. Baker. Mechanical constraints and cell cycle regulation in models of collective cell migration. *arXiv*, (arXiv:2401.08805), 2024. arXiv:2401.08805, doi:10.48550/arXiv.2401.08805.

[57] D. Faller, U. Klingmüller, and J. Timmer. Simulation methods for optimal experimental design in systems biology. *Simulation*, 79(12):717–725, 2003. doi:10.1177/0037549703040937.

[58] P. J. Firnett and B. A. Troesch. Shooting-splitting method for sensitive two-point boundary value problems. In D. G. Bettis, editor, *Proceedings of the Conference on the Numerical Solution of Ordinary Differential Equations*, Lecture Notes in Mathematics, pages 408–433, Berlin, Heidelberg, 1974. Springer. doi:10.1007/BFb0066603.

[59] R. A. Fisher. The wave of advance of advantageous genes. *Annals of Eugenics*, 7(4):355–369, 1937. doi:10.1111/j.1469-1809.1937.tb02153.x.

[60] J. A. Flegg, S. N. Menon, H. M. Byrne, and D. L. S. McElwain. A current perspective on wound healing and tumour-induced angiogenesis. *Bulletin of Mathematical Biology*, 82(2):23, 2020. doi:10.1007/s11538-020-00696-0.

[61] F. Fröhlich, A. Reiser, L. Fink, D. Woschée, T. Ligon, F. J. Theis, J. O. Rädler, and J. Hasenauer. Multi-experiment nonlinear mixed effect modeling of single-cell translation kinetics after transfection. *npj Systems Biology and Applications*, 4(1):1–12, 2018. doi:10.1038/s41540-018-0079-7.

[62] H. G. Frohnhöfer, J. Krauss, H.-M. Maischein, and C. Nüsslein-Volhard. Iridophores and their interactions with other chromatophores are required for stripe formation in zebrafish. *Development*, 140:2997–3007, 2013. doi:10.1242/dev.096719.

[63] K. G. Gadkar, R. Gunawan, and F. J. Doyle. Iterative approach to model identification of biological networks. *BMC Bioinformatics*, 6(1):155, 2005. doi:10.1186/1471-2105-6-155.

[64] A. Gelman, W. R. Gilks, and G. O. Roberts. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997. doi:10.1214/aoap/1034625254.

[65] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, 1992.

[66] P. Gerlee. The model muddle: In search of tumor growth laws. *Cancer Research*, 73(8):2407–2411, 2013. doi:10.1158/0008-5472.CAN-12-4355.

[67] A. Gierer and H. Meinhardt. A theory of biological pattern formation. *Kybernetik*, 12:30–39, 1972. doi:10.1007/BF00289234.

[68] A. Gill, D. J. Warne, C. McGrory, J. M. McGree, and A. M. Overstall. Robust simulation design for generalized linear models in conditions of heteroscedasticity or correlation. In *2022 Winter Simulation Conference (WSC)*, pages 37–48, 2022. doi:10.1109/WSC57314.2022.10015326.

[69] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[70] R. N. Gutenkunst, J. J. Waterfall, F. P. Casey, K. S. Brown, C. R. Myers, and J. P. Sethna. Universally sloppy parameter sensitivities in systems biology models. *PLOS Computational Biology*, 3(10):e189, 2007. doi:10.1371/journal.pcbi.0030189.

[71] A. Habbal, H. Barelli, and G. Malandain. Assessing the ability of the 2D Fisher–KPP equation to model cell-sheet wound closure. *Mathematical Biosciences*, 252:45–59, 2014. doi:10.1016/j.mbs.2014.03.009.

[72] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 312–317, 1996. doi:10.1109/ICEC.1996.542381.

[73] N. Hansen. The CMA evolution strategy, 2014. https://cma-es.github.io/. URL: https://cma-es.github.io/.

[74] T. W. Hiscock and S. G. Megason. Orientation of Turing-like patterns by morphogen gradients and tissue anisotropies. *Cell Systems*, 1:408–416, 2015. doi:10.1016/j.cels.2015.12.001.

[75] K. S. Holkar and L. M. Waghmare. An overview of model predictive control. *International Journal of Control and Automation*, 3(4), 2010.

[76] W. G. Hunter and A. M. Reiner. Designs for discriminating between two rival models. *Technometrics*, 7(3):307–323, 1965. doi:10.1080/00401706.1965.10490265.

[77] B. Iooss and P. Lemaître. A review on global sensitivity analysis methods. In G. Dellino and C. Meloni, editors, *Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications*, pages 101–122. Springer US, Boston, MA, 2015. doi:10.1007/978-1-4899-7547-8_5.

[78] D. Iron, J. Wei, and M. Winter. Stability analysis of Turing patterns generated by the Schnakenberg model. *Journal of Mathematical Biology*, 49(4):358–390, 2004. doi:10.1007/s00285-003-0258-y.

[79] P. R. Jackson, J. Juliano, A. Hawkins-Daarud, R. C. Rockne, and K. R. Swanson. Patient-specific mathematical neuro-oncology: Using a simple proliferation and invasion tumor model to inform clinical practice. *Bulletin of Mathematical Biology*, 77(5):846–856, 2015. doi:10.1007/s11538-015-0067-7.

[80] M. Jacobs, B. W. Brunton, S. L. Brunton, J. N. Kutz, and R. V. Raut. Hyper-SINDy: Deep generative modeling of nonlinear stochastic governing equations. *arXiv*, (arXiv:2310.04832), 2023. arXiv:2310.04832, doi:10.48550/arXiv.2310.04832.

[81] T. Jiang, H. Jung, R. Widelitz, and C. Chuong. Self-organization of periodic patterns by dissociated feather mesenchymal cells and the regulation of size, number and spacing of primordia. *Development*, 126:4997–5009, 1999. doi:10.1242/dev.126.22.4997.

[82] W. Jin, K.-Y. Lo, S. Chou, S. W. McCue, and M. J. Simpson. The role of initial geometry in experimental models of wound closing. *Chemical Engineering Science*, 179:221–226, 2018. doi:10.1016/j.ces.2018.01.004.

[83] W. Jin, E. T. Shah, C. J. Penington, S. W. McCue, L. K. Chopin, and M. J. Simpson. Reproducibility of scratch assays is affected by the initial degree of confluence: Experiments, modelling and model selection. *Journal of Theoretical Biology*, 390:136–145, 2016. doi:10.1016/j.jtbi.2015.10.040.

[84] M. Joshi, A. Seidel-Morgenstern, and A. Kremling. Exploiting the bootstrap method for quantifying parameter confidence intervals in dynamical systems. *Metabolic Engineering*, 8(5):447–455, 2006. doi:10.1016/j.ymben.2006.04.003.

[85] D. Joubert, J. D. Stigter, and J. Molenaar. Determining minimal output sets that ensure structural identifiability. *PLOS ONE*, 13(11):e0207334, 2018. doi:10.1371/journal.pone.0207334.

[86] D. Joubert, J. D. Stigter, and J. J. Molenaar. An efficient procedure to assist in the re-parametrization of structurally unidentifiable models. *Mathematical Biosciences*, 323:108328, 2020. doi:10.1016/j.mbs.2020.108328.

[87] H.-S. Jung, P. H. Francis-West, R. B. Widelitz, T.-X. Jiang, S. Ting-Berreth, C. Tickle, L. Wolpert, and C.-M. Chuong. Local inhibitory action of BMPs and their relationships with activators in feather formation: implications for periodic patterning. *Developmental Biology*, 196:11–23, 1998. doi:10.1006/dbio.1998.8850.

[88] M. Kamrani and S. Mohammad Hosseini. The role of coefficients of a general SPDE on the stability and convergence of a finite difference method. *Journal of Computational and Applied Mathematics*, 234:1426–1434, 2010. doi:10.1016/j.cam.2010.02.018.

[89] I. Karatzas and S. E. Shreve. *Brownian Motion and Stochastic Calculus*, volume 113 of *Graduate Texts in Mathematics*. Springer, New York, NY, 1998. doi:10.1007/978-1-4612-0949-2.

[90] B. J. Kealy and D. J. Wollkind. A nonlinear stability analysis of vegetative Turing pattern formation for an interaction–diffusion plant-surface water model system in an arid flat environment. *Bulletin of Mathematical Biology*, 74:803–833, 2012. doi:10.1007/s11538-011-9688-7.

[91] H. B. Keller. *Numerical Solution of Two Point Boundary Value Problems*. Number 24 in CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, Pa, 4. print edition, 1976.

[92] D. L. Kern, S. Lenhart, R. Miller, and J. Yong. Optimal control applied to native–invasive population dynamics. *Journal of Biological Dynamics*, 1(4):413–426, 2007. doi:10.1080/17513750701605556.

[93] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, (arXiv:1412.6980), 2017. arXiv:1412.6980, doi:10.48550/arXiv.1412.6980.

[94] V. Klika, E. A. Gaffney, and P. K. Maini. On the speed of propagation in Turing patterns for reaction-diffusion systems. *arXiv*, (arXiv:2403.09247), 2024. arXiv:2403.09247, doi:10.48550/arXiv.2403.09247.

[95] A. N. Kolmogorov. On the empirical determination of a distribution function. In S. Kotz and N. L. Johnson, editors, *Breakthroughs in Statistics: Methodology and Distribution*, pages 106–113. Springer, New York, NY, 1992. doi:10.1007/978-1-4612-4380-9_10.

[96] A. N. Kolmogorov, I. G. Petrovsky, and N. S. Piskunov. A study of the diffusion equation with increase in the amount of substance, and its application to a biological problem. In V. M. Tikhomirov, editor, *Selected works of AN Kolmogorov*, pages 242–270. Springer, 1991.

[97] M. Komorowski, M. J. Costa, D. A. Rand, and M. P. H. Stumpf. Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. *Proceedings of the National Academy of Sciences*, 108(21):8645–8650, 2011. doi:10.1073/pnas.1015814108.

[98] S. Kondo and R. Asai. A reaction-diffusion wave on the skin of the marine angelfish Pomacanthus. *Nature*, 376:765–768, 1995. doi:10.1038/376765a0.

[99] S. Konishi and G. Kitagawa. *Information Criteria and Statistical Modeling*. Springer Series in Statistics. Springer, New York, 2008.

[100] C. Konow, N. H. Somberg, J. Chavez, I. R. Epstein, and M. Dolnik. Turing patterns on radially growing domains: Experiments and simulations. *Physical Chemistry Chemical Physics*, 21:6718–6724, 2019. doi:10.1039/C8CP07797E.

[101] I. Korshunova, J. Degrave, F. Huszar, Y. Gal, A. Gretton, and J. Dambre. BRUNO: A deep recurrent model for exchangeable data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7190–7198. Curran Associates, Inc., 2018.

[102] A. L. Krause, M. A. Ellis, and R. A. Van Gorder. Influence of curvature, growth, and anisotropy on the evolution of Turing patterns on growing manifolds. *Bulletin of Mathematical Biology*, 81:759–799, 2019. doi:10.1007/s11538-018-0535-y.

[103] A. L. Krause, E. A. Gaffney, P. K. Maini, and V. Klika. Introduction to 'Recent progress and open frontiers in Turing's theory of morphogenesis'. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379:20200280, 2021. doi:10.1098/rsta.2020.0280.

[104] C. Kreutz and J. Timmer. Systems biology: Experimental design. *The FEBS Journal*, 276(4):923–942, 2009. doi:10.1111/j.1742-4658.2008.06843.x.

[105] P. M. Kulesa, G. C. Cruywagen, S. R. Lubkin, P. K. Main, J. Sneyd, M. W. J. Ferguson, and J. D. Murray. On a model mechanism for the spatial patterning of teeth primordia in the Alligator. *Journal of Theoretical Biology*, 180:287–296, 1996. doi:10.1006/jtbi.1996.0103.

[106] J. LaChance and D. J. Cohen. Practical fluorescence reconstruction microscopy for large samples and low-magnification imaging. *PLOS Computational Biology*, 16(12):e1008443, 2020. doi:10.1371/journal.pcbi.1008443.

[107] J. Lagergren, J. T. Nardini, G. M. Lavigne, E. M. Rutter, and K. B. Flores. Learning partial differential equations for biological transport models from noisy spatiotemporal data. *arXiv:1902.04733 [math]*, 2019. arXiv:1902.04733.

[108] J. H. Lagergren, J. T. Nardini, R. E. Baker, M. J. Simpson, and K. B. Flores. Biologically-informed neural networks guide mechanistic modeling from sparse experimental data. *PLOS Computational Biology*, 16(12):e1008462, 2020. doi:10.1371/journal.pcbi.1008462.

[109] B. Lambert, C. L. Lei, M. Robinson, M. Clerx, R. Creswell, S. Ghosh, S. Tavener, and D. J. Gavaghan. Autocorrelated measurement processes and inference for ordinary differential equation models of biological systems. *Journal of The Royal Society Interface*, 20(199):20220725, 2023. doi:10.1098/rsif.2022.0725.

[110] J. Laurent-Varin, J. F. Bonnans, N. Bérend, M. Haddou, and C. Talbot. Interior-point approach to trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 30(5):1228–1238, 2007. doi:10.2514/1.18196.

[111] I. Lengyel and I. R. Epstein. Modeling of Turing structures in the chlorite–iodide–malonic acid–starch reaction system. *Science*, 251(4994):650–652, 1991. doi:10.1126/science.251.4994.650.

[112] S. Lenhart and J. T. Workman. *Optimal Control Applied to Biological Models*. Chapman and Hall/CRC, 2007. doi:10.1201/9781420011418.

[113] X. Li and J. Yong. *Optimal Control Theory for Infinite Dimensional Systems*. Springer Science & Business Media, 2012.

[114] D. Liberzon. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton university press, 2011.

[115] H. W. Lilliefors. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402, 1967. doi:10.1080/01621459.1967.10482916.

[116] T. Litwin, J. Timmer, and C. Kreutz. Optimal experimental design based on two-dimensional likelihood profiles. *Frontiers in Molecular Biosciences*, 9, 2022.

[117] Y. Liu, P. K. Maini, and R. E. Baker. Control of diffusion-driven pattern formation behind a wave of competency. *Physica D: Nonlinear Phenomena*, 438:133297, 2022. doi:10.1016/j.physd.2022.133297.

[118] Y. Liu, E. G. Rens, and L. Edelstein-Keshet. Spots, stripes, and spiral waves in models for static and motile cells. *Journal of Mathematical Biology*, 82:28, 2021. doi:10.1007/s00285-021-01550-0.

[119] Y. Liu, K. Suh, P. K. Maini, D. J. Cohen, and R. E. Baker. Dataset for: "parameter identifiability and model selection for partial differential equation models of cell invasion". *Zenodo*, 2024. doi:10.5281/zenodo.10594283.

[120] Y. Liu, K. Suh, P. K. Maini, D. J. Cohen, and R. E. Baker. Parameter identifiability and model selection for partial differential equation models of cell invasion. *Journal of The Royal Society Interface*, 21(212):20230607, 2024. doi:10.1098/rsif.2023.0607.

[121] L. Ljung and T. Glad. On global identifiability for arbitrary model parametrizations. *Automatica*, 30(2):265–276, 1994. doi:10.1016/0005-1098(94)90029-9.

[122] Q. Lü and X. Zhang. *General Pontryagin-Type Stochastic Maximum Principle and Backward Stochastic Evolution Equations in Infinite Dimensions*. SpringerBriefs in Mathematics. Springer International Publishing, Cham, 2014. doi:10.1007/978-3-319-06632-5.

[123] B. Macdonald and D. Husmeier. Gradient matching methods for computational inference in mechanistic models for systems biology: a review and comparative analysis. *Frontiers in Bioengineering and Biotechnology*, 3, 2015.

[124] A. Madzvamuse and P. K. Maini. Velocity-induced numerical solutions of reaction-diffusion systems on continuously growing domains. *Journal of Computational Physics*, 225:100–119, 2007. doi:10.1016/j.jcp.2006.11.022.

[125] P. K. Maini, D. L. S. McElwain, and D. Leavesley. Travelling waves in a wound healing assay. *Applied Mathematics Letters*, 17(5):575–580, 2004. doi:10.1016/S0893-9659(04)90128-0.

[126] S. F. Martina-Perez, I. B. Breinyn, D. J. Cohen, and R. E. Baker. Spatial heterogeneity in collective electrotaxis: continuum modelling and applications to optimal control. *bioRxiv*, 2024.02.28.580259, 2024. doi:10.1101/2024.02.28.580259.

[127] S. F. Martina-Perez, H. Sailem, and R. E. Baker. Efficient bayesian inference for mechanistic modelling with high-throughput data. *PLOS Computational Biology*, 18(6):e1010191, 2022. doi:10.1371/journal.pcbi.1010191.

[128] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, 1998. doi:10.1145/272991.272995.

[129] M. McAsey, L. Mou, and W. Han. Convergence of the forward-backward sweep method in optimal control. *Computational Optimization and Applications*, 53(1):207–226, 2012. doi:10.1007/s10589-011-9454-7.

[130] S. W. McCue, W. Jin, T. J. Moroney, K.-Y. Lo, S.-E. Chou, and M. J. Simpson. Hole-closing model reveals exponents for nonlinear degenerate diffusivity functions in cell biology. *Physica D: Nonlinear Phenomena*, 398:130–140, 2019. doi:10.1016/j.physd.2019.06.005.

[131] N. Meshkat, M. Eisenberg, and J. J. DiStefano. An algorithm for finding globally identifiable parameter combinations of nonlinear ODE models using Gröbner bases. *Mathematical Biosciences*, 222(2):61–72, 2009. doi:10.1016/j.mbs.2009.08.010.

[132] P. A. P. Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950. arXiv:2332142, doi:10.2307/2332142.

[133] C. Mou, F. Pitel, D. Gourichon, F. Vignoles, A. Tzika, P. Tato, L. Yu, D. W. Burt, B. Bed'hom, M. Tixier-Boichard, K. J. Painter, and D. J. Headon. Cryptic patterning of avian skin confers a developmental facility for loss of neck feathering. *PLOS Biology*, 9:e1001028, 2011. doi:10.1371/journal.pbio.1001028.

[134] S. A. Murphy and A. W. Van Der Vaart. On profile likelihood. *Journal of the American Statistical Association*, 95(450):449–465, 2000. doi:10.1080/01621459.2000.10474219.

[135] J. D. Murray. *Mathematical Biology II: Spatial Models and Biomedical Applications*, volume 3. Springer New York, 2001.

[136] A. Nabeel, A. Karichannavar, S. Palathingal, J. Jhawar, D. B. Brückner, D. R. M., and V. Guttal. Discovering stochastic dynamical equations from biological time series data. *arXiv*, (arXiv:2205.02645), 2024. arXiv:2205.02645.

[137] B. N. Nagorcka and J. R. Mooney. From stripes to spots: Prepatterns which can be produced in the skin by a reaction-diffusion system. *Mathematical Medicine and Biology: A Journal of the IMA*, 9:249–267, 1992. doi:10.1093/imammb/9.4.249.

[138] A. Nakamasu, G. Takahashi, A. Kanbe, and S. Kondo. Interactions between zebrafish pigment cells responsible for the generation of Turing patterns. *Proceedings of the National Academy of Sciences*, 106:8429–8434, 2009. doi:10.1073/pnas.0808622106.

[139] J. T. Nardini. Forecasting and predicting stochastic agent-based model data with biologically-informed neural networks. *arXiv*, (arXiv:2311.04709), 2023. arXiv:2311.04709, doi:10.48550/arXiv.2311.04709.

[140] J. P. Owen, R. N. Kelsh, and C. A. Yates. A quantitative modelling approach to zebrafish pigment pattern formation. *eLife*, 9:e52998, 2020. doi:10.7554/eLife.52998.

[141] K. M. Page, P. K. Maini, and N. A. M. Monk. Complex pattern formation in reaction-diffusion systems with spatially varying parameters. *Physica D: Nonlinear Phenomena*, 202:95–115, 2005. doi:10.1016/j.physd.2005.01.022.

[142] K. J. Painter, T. Hillen, and J. R. Potts. Biological modelling with nonlocal advection diffusion equations. *arXiv*, (arXiv:2307.14396), 2023. arXiv:2307.14396, doi:10.48550/arXiv.2307.14396.

[143] L. M. Păun, M. U. Qureshi, M. Colebank, N. A. Hill, M. S. Olufsen, M. A. Haider, and D. Husmeier. MCMC methods for inference in a mathematical model of pulmonary circulation. *Statistica Neerlandica*, 72(3):306–338, 2018. doi:10.1111/stan.12132.

[144] Pawitan, Yudi. *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford University Press, 2001.

[145] M. Peleg and M. G. Corradini. Microbial growth curves: what the models tell us and what they cannot. *Critical Reviews in Food Science and Nutrition*, 51(10):917–945, 2011. doi:10.1080/10408398.2011.570463.

[146] H. Pohjanpalo. System identifiability based on the power series expansion of the solution. *Mathematical Biosciences*, 41(1):21–33, 1978. doi:10.1016/0025-5564(78)90063-9.

[147] L. S. Pontrjagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The mathematical theory of optimal processes*. Interscience, 1962. English translation by Lucien W. Neustadt et al.

[148] C. Pouchol, J. Clairambault, A. Lorz, and E. Trélat. Asymptotic analysis and optimal control of an integro-differential system modelling healthy and cancer cells exposed to chemotherapy. *Journal de Mathématiques Pures et Appliquées*, 116:268–308, 2018. doi:10.1016/j.matpur.2017.10.007.

[149] X. Qi, Y. Luo, G. Wu, K. Boriboonsomsin, and M. Barth. Deep reinforcement learning enabled self-learning control for energy efficient driving. *Transportation Research Part C: Emerging Technologies*, 99:67–81, 2019. doi:10.1016/j.trc.2018.12.018.

[150] J. Rabault, M. Kuchta, A. Jensen, U. Réglade, and N. Cerardi. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of Fluid Mechanics*, 865:281–302, 2019. doi:10.1017/jfm.2019.62.

[151] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi:10.1016/j.jcp.2018.10.045.

[152] E. E. Ramírez-Torres, A. R. S. Castañeda, L. Rández, S. A. Sisson, L. E. B. Cabrales, and J. I. Montijano. Parameter estimation of Gompertz model for tumor growth: which likelihood to choose? *Research Square*, 2024. doi:10.21203/rs.3.rs-3999289/v1.

[153] A. Raue, V. Becker, U. Klingmüller, and J. Timmer. Identifiability and observability analysis for experimental design in nonlinear dynamical models. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(4):045105, 2010. doi:10.1063/1.3528102.

[154] A. Raue, C. Kreutz, T. Maiwald, J. Bachmann, M. Schilling, U. Klingmüller, and J. Timmer. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, 25(15):1923–1929, 2009. doi:10.1093/bioinformatics/btp358.

[155] A. Raue, J. Karlsson, M. P. Saccomani, M. Jirstrand, and J. Timmer. Comparison of approaches for parameter identifiability analysis of biological systems. *Bioinformatics*, 30(10):1440–1448, 2014. doi:10.1093/bioinformatics/btu006.

[156] A. Raue, C. Kreutz, F. J. Theis, and J. Timmer. Joining forces of Bayesian and frequentist methodology: A study for inference in the presence of non-identifiability. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110544, 2013. doi:10.1098/rsta.2011.0544.

[157] A. Ravasio, I. Cheddadi, T. Chen, T. Pereira, H. T. Ong, C. Bertocchi, A. Brugues, A. Jacinto, A. J. Kabla, Y. Toyama, X. Trepat, N. Gov, L. Neves de Almeida, and B. Ladoux. Gap geometry dictates epithelial closure efficiency. *Nature Communications*, 6(1):7683, 2015. doi:10.1038/ncomms8683.

[158] M. Renardy, D. Kirschner, and M. Eisenberg. Structural identifiability analysis of age-structured PDE epidemic models. *Journal of Mathematical Biology*, 84(1):9, 2022. doi:10.1007/s00285-021-01711-1.

[159] F. J. Richards. A flexible growth function for empirical use. *Journal of Experimental Botany*, 10(2):290–301, 1959. doi:10.1093/jxb/10.2.290.

[160] G. O. Roberts and J. S. Rosenthal. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351–367, 2001. doi:10.1214/ss/1015346320.

[161] S. Rogers and M. Girolami. *A First Course in Machine Learning*. Chapman and Hall/CRC, New York, 2011. doi:10.1201/9781466506299.

[162] S. Ruder. An overview of gradient descent optimization algorithms, 2017. arXiv:1609.04747.

[163] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017. doi:10.1126/sciadv.1602614.

[164] S. J. Ruuth. Implicit-explicit methods for reaction-diffusion problems in pattern formation. *Journal of Mathematical Biology*, 34:148–176, 1995. doi:10.1007/BF00178771.

[165] H. Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197):20160446, 2017. doi:10.1098/rspa.2016.0446.

[166] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009. doi:10.1126/science.1165893.

[167] J. Schnakenberg. Simple chemical reaction systems with limit cycle behaviour. *Journal of Theoretical Biology*, 81:389–400, 1979. doi:10.1016/0022-5193(79)90042-0.

[168] D. Schnörr and C. Schnörr. Learning system parameters from Turing patterns. *Machine Learning*, 112(9):3151–3190, 2023. doi:10.1007/s10994-023-06334-9.

[169] G. A. F. Seber and C. J. Wild. *Nonlinear Regression*. Wiley-Interscience, 2003.

[170] B. G. Sengers, C. P. Please, and R. O. Oreffo. Experimental characterization and computational modelling of two-dimensional cell spreading for skeletal regeneration. *Journal of The Royal Society Interface*, 4(17):1107–1117, 2007. doi:10.1098/rsif.2007.0233.

[171] J. A. Sharp, A. P. Browning, T. Mapder, K. Burrage, and M. J. Simpson. Optimal control of acute myeloid leukaemia. *Journal of Theoretical Biology*, 470:30–42, 2019. doi:10.1016/j.jtbi.2019.03.006.

[172] J. A. Sherratt, J. D. Murray, and B. C. Clarke. Models of epidermal wound healing. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 241(1300):29–36, 1990. doi:10.1098/rspb.1990.0061.

[173] H. Shoji, Y. Iwasa, A. Mochizuki, and S. Kondo. Directionality of stripes formed by anisotropic reaction-diffusion models. *Journal of Theoretical Biology*, 214:549–561, 2002. doi:10.1006/jtbi.2001.2480.

[174] A. Shyntar, A. Patel, M. Rhodes, H. Enderling, and T. Hillen. The tumor invasion paradox in cancer stem cell-driven solid tumors. *Bulletin of Mathematical Biology*, 84(12):139, 2022. doi:10.1007/s11538-022-01086-4.

[175] M. J. Simpson, R. E. Baker, S. T. Vittadello, and O. J. Maclaren. Practical parameter identifiability for spatio-temporal models of cell invasion. *Journal of The Royal Society Interface*, 17(164):20200055, 2020. doi:10.1098/rsif.2020.0055.

[176] M. J. Simpson, A. P. Browning, D. J. Warne, O. J. Maclaren, and R. E. Baker. Parameter identifiability and model selection for sigmoid population growth models. *Journal of theoretical biology*, 535:110998, 2022.

[177] M. J. Simpson, K. K. Treloar, B. J. Binder, P. Haridas, K. J. Manton, D. I. Leavesley, D. L. S. McElwain, and R. E. Baker. Quantifying the roles of cell motility and cell proliferation in a circular barrier assay. *Journal of The Royal Society Interface*, 10(82):20130007, 2013. doi:10.1098/rsif.2013.0007.

[178] B. Smucker, M. Krzywinski, and N. Altman. Optimal experimental design. *Nature Methods*, 15(8):559–560, 2018. doi:10.1038/s41592-018-0083-2.

[179] B. Steiert, A. Raue, J. Timmer, and C. Kreutz. Experimental design for parameter estimation of gene regulatory networks. *PLOS ONE*, 7(7):e40052, 2012. doi:10.1371/journal.pone.0040052.

[180] R. Sun. Optimization for deep learning: theory and algorithms, 2019. arXiv:1912.08957.

[181] M. Sunnåker, A. G. Busetto, E. Numminen, J. Corander, M. Foll, and C. Dessimoz. Approximate Bayesian Computation. *PLOS Computational Biology*, 9(1):e1002803, 2013. doi:10.1371/journal.pcbi.1002803.

[182] R. S. Sutton and A. G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachusetts, second edition edition, 2018.

[183] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

[184] K. K. Treloar and M. J. Simpson. Sensitivity of edge detection methods for quantifying cell migration assays. *PLoS ONE*, 8(6):e67389, 2013. doi:10.1371/journal.pone.0067389.

[185] A. Tsoularis and J. Wallace. Analysis of logistic growth models. *Mathematical Biosciences*, 179(1):21–55, 2002. doi:10.1016/S0025-5564(02)00096-2.

[186] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237:37–72, 1952. doi:10.1098/rstb.1952.0012.

[187] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí. Scatter search and local NLP solvers: A multistart framework for global optimization. *INFORMS Journal on Computing*, 19(3):328–340, 2007. doi:10.1287/ijoc.1060.0175.

[188] J. Uhlendorf, A. Miermont, T. Delaveau, G. Charvin, F. Fages, S. Bottani, G. Batt, and P. Hersen. Long-term model predictive control of gene expression at the population and single-cell levels. *Proceedings of the National Academy of Sciences*, 109(35):14271–14276, 2012. doi:10.1073/pnas.1206810109.

[189] R. A. Van Gorder, V. Klika, and A. L. Krause. Turing conditions for pattern forming systems on evolving manifolds. *Journal of Mathematical Biology*, 82:4, 2021. doi:10.1007/s00285-021-01552-y.

[190] J. Vanlier, C. A. Tiemann, P. A. J. Hilbers, and N. A. W. van Riel. A Bayesian approach to targeted experiment design. *Bioinformatics*, 28(8):1136–1142, 2012. doi:10.1093/bioinformatics/bts092.

[191] C. Varea, J. L. Aragón, and R. A. Barrio. Confined Turing patterns in growing systems. *Physical Review E*, 56:1250–1253, 1997. doi:10.1103/PhysRevE.56.1250.

[192] T. Vicsek and A. Zafeiris. Collective motion. *Physics Reports*, 517(3):71–140, 2012. doi:10.1016/j.physrep.2012.03.004.

[193] A. F. Villaverde, N. D. Evans, M. J. Chappell, and J. R. Banga. Input-dependent structural identifiability of nonlinear systems. *IEEE Control Systems Letters*, 3(2):272–277, 2019. doi:10.1109/LCSYS.2018.2868608.

[194] A. F. Villaverde, E. Raimúndez, J. Hasenauer, and J. R. Banga. Assessment of prediction uncertainty quantification methods in systems biology. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–12, 2022. doi:10.1109/TCBB.2022.3213914.

[195] S. T. Vittadello, S. W. McCue, G. Gunasingh, N. K. Haass, and M. J. Simpson. Mathematical models for cell migration with real-time cell cycle dynamics. *Biophysical Journal*, 114(5):1241–1253, 2018. doi:10.1016/j.bpj.2017.12.041.

[196] A. Volkening, M. R. Abbott, N. Chandra, B. Dubois, F. Lim, D. Sexton, and B. Sandstede. Modeling stripe formation on growing zebrafish tailfins. *Bulletin of Mathematical Biology*, 82(5):56, 2020. doi:10.1007/s11538-020-00731-0.

[197] A. Volkening and B. Sandstede. Modelling stripe formation in zebrafish: An agent-based approach. *Journal of The Royal Society Interface*, 12:20150812, 2015. doi:10.1098/rsif.2015.0812.

[198] U. von Toussaint. Bayesian inference in physics. *Reviews of Modern Physics*, 83(3):943–999, 2011. doi:10.1103/RevModPhys.83.943.

[199] N. Wang, Y. Gao, and X. Zhang. Data-driven performance-prescribed reinforcement learning control of an unmanned surface vehicle. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5456–5467, 2021. doi:10.1109/TNNLS.2021.3056444.

[200] L. Wanika, J. R. Egan, N. Swaminathan, C. A. Duran-Villalobos, J. Branke, S. Goldrick, and M. Chappell. Structural and practical identifiability analysis in bioengineering: a beginner's guide. *Journal of Biological Engineering*, 18(1):20, 2024. doi:10.1186/s13036-024-00410-x.

[201] D. J. Warne, R. E. Baker, and M. J. Simpson. Optimal quantification of contact inhibition in cell populations. *Biophysical Journal*, 113(9):1920–1924, 2017. doi:10.1016/j.bpj.2017.09.016.

[202] D. J. Warne, R. E. Baker, and M. J. Simpson. Using experimental data and information criteria to guide model selection for reaction–diffusion problems in mathematical biology. *Bulletin of Mathematical Biology*, 81(6):1760–1804, 2019. doi:10.1007/s11538-019-00589-x.

[203] D. J. Warne, O. J. Maclaren, E. J. Carr, M. J. Simpson, and C. Drovandi. Generalised likelihood profiles for models with intractable likelihoods. *arXiv*, (arXiv:2305.10710), 2023. arXiv:2305.10710, doi:10.48550/arXiv.2305.10710.

[204] M. Watanabe and S. Kondo. Is pigment patterning in fish skin determined by the Turing mechanism? *Trends in Genetics*, 31:88–96, 2015. doi:10.1016/j.tig.2014.11.005.

[205] F.-G. Wieland, A. L. Hauber, M. Rosenblatt, C. Tönsing, and J. Timmer. On structural and practical identifiability. *Current Opinion in Systems Biology*, 25:60–69, 2021. doi:10.1016/j.coisb.2021.03.005.

[206] S. S. Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, 1938. arXiv:2957648.

[207] L. Willem, F. Verelst, J. Bilcke, N. Hens, and P. Beutels. Lessons from a decade of individual-based models for infectious disease transmission: A systematic review (2006-2015). *BMC Infectious Diseases*, 17(1):612, 2017. doi:10.1186/s12879-017-2699-8.

[208] S. P. Williams, C. M. Gould, C. J. Nowell, T. Karnezis, M. G. Achen, K. J. Simpson, and S. A. Stacker. Systematic high-content genome-wide RNAi screens of endothelial cell migration and morphology. *Scientific Data*, 4(1):170009, 2017. doi:10.1038/sdata.2017.9.

[209] T. P. Witelski. Merging traveling waves for the Porous-Fisher's equation. *Applied Mathematics Letters*, 8(4):57–62, 1995. doi:10.1016/0893-9659(95)00047-T.

[210] Y. Xiao, L. Thomas, and M. A. J. Chaplain. Calibrating models of cancer invasion: parameter estimation using approximate Bayesian computation and gradient matching. *Royal Society Open Science*, 8(6):202237, 2021. doi:10.1098/rsos.202237.

[211] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3391–3401. Curran Associates, Inc., 2017.

[212] T. J. Zajdel, G. Shim, L. Wang, A. Rossello-Martinez, and D. J. Cohen. SCHEEPDOG: Programming electric cues to dynamically herd large-scale cell migration. *Cell Systems*, 10(6):506–514.e3, 2020. doi:10.1016/j.cels.2020.05.009.

[213] R. Zerpe. Golden trevally (Gnathanodon speciosus), 2019. From https://www.flickr.com/photos/krokodiver/47728328672, image used under CC-BY license. URL: https://www.flickr.com/photos/krokodiver/47728328672/.

[214] G. E. Zouraris. Crank–Nicolson finite element approximations for a linear stochastic fourth order equation with additive space-time white noise. *SIAM Journal on Numerical Analysis*, 56:838–858, 2018. doi:10.1137/15M1053098.