



THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e.g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Phenomenological Modelling

Statistical abstraction methods for Markov chains

Michalis Michaelides



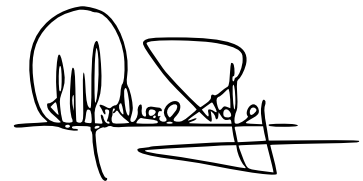
Doctor of Philosophy
Institute of Adaptive and Neural Computation
School of Informatics
University of Edinburgh
2019

*To my family
for their resolute faith
and unwavering support.*

Declaration

I declare that this thesis has been composed by myself and that this work has not been submitted for any other degree or professional qualification. I confirm that the work submitted is my own, except where work which has formed part of jointly-authored publications has been included. My contribution and those of the other authors to this work have been explicitly indicated at the beginning of each chapter. I confirm that appropriate credit has been given within this thesis where reference has been made to the work of others.

Edinburgh, 2019

A handwritten signature in black ink, appearing to read 'Michalis Michaelides', written over a horizontal line.

Michalis Michaelides
2019

Acknowledgements

It is not an easy task to appropriately credit everyone who in any way brought about this work; I must nonetheless try. Firstly, I would like to express my immense gratitude to my supervisor Guido Sanguinetti, for providing the freedom to explore and the necessary resources to survive the many rigours and challenges of the academic landscape. Any elegance in this work stems from him, any clumsiness is my own. On the same note, Jane Hillston has been a pillar of support and a beacon of guidance. Her experience has refined this work and shaped it to a comprehensive whole. On both a personal and academic level, they proved to be the best of mentors.

There is a vibrant group to be found at all times under Guido's aegis. To everyone who populated this group in the past four years with stimulating conversation, ranging from philosophy to technical matters, I would like to extend my thanks for a productive and positive atmosphere. The morning coffee sessions sparked creativity, and our social events are now cherished memories. In particular, Andreas Kapourani and Dimitrios Milios have been excellent companions, in research and otherwise; I owe them much. Some people in the forum who, although not part of Guido's group, improved my work and well-being are Rafael, as well as Ludovica and Paul from Jane's group. People in my office contributed a pleasant environment in which to work and create.

Finally, this thesis would be impossible without the emotional support and faith that I received from all my friends and family. A special thanks goes to Demetris Hadjimichael, who ensured that the philosophical positions of this thesis are viable and defensible. My sister, my parents, and my godfather have kept me sane and afforded me every comfort and understanding.

This work was supported by the Institute for Adaptive and Neural Computation (ANC) of the Informatics School of The University of Edinburgh, funded by the UK Engineering and Physical Sciences Research Council (EPSRC) and the European Research Council (ERC).

Abstract

Continuous-time Markov chains have long served as exemplary low-level models for an array of systems, be they natural processes like chemical reactions and population fluctuations in ecosystems, or artificial processes like server queuing systems or communication networks. Our interest in such systems is often an emergent macro-scale behaviour, or phenomenon, which can be well characterised by the satisfaction of a set of properties. Although theoretically elegant, the fundamental low-level nature of Markov chain models makes macro-scale analysis of the phenomenon of interest difficult. Particularly, it is not easy to determine the driving mechanisms for the emergent phenomenon, or to predict how changes at the Markov chain level will influence the macro-scale behaviour.

The difficulties arise primarily from two aspects of such models. Firstly, as the number of components in the modelled system grows, so does the state-space of the Markov chain, often making behaviour characterisation untenable under both simulation-based and analytical methods. Secondly, the behaviour of interest in such systems is usually dependent on the inherent stochasticity of the model, and may not be aligned to the underlying state interpretation. In a model where states represent a low-level, primitive aspect of system components, the phenomenon of interest often varies significantly with respect to this low-level aspect that states represent.

This work focuses on providing methodological frameworks that circumvent these issues by developing abstraction strategies, which preserve the phenomena of interest. In the first part of this thesis, we express behavioural characteristics of the system in terms of a temporal logic with Markov chain trajectories as semantic objects. This allows us to group regions of the state-space by how well they satisfy the logical properties that characterise macro-scale behaviour, in order to produce an abstracted Markov chain. States of the abstracted chain correspond to certain satisfaction probabilities of the logical properties, and inferred dynamics match the behaviour of the original chain in terms of the properties. The resulting model has a smaller state-space which is interpretable in terms of an emergent behaviour of the original system, and is therefore valuable to a researcher despite the accuracy sacrifices.

Coarsening based on logical properties is particularly useful in multi-scale modelling, where a layer of the model is a (continuous-time) Markov chain. In such models, the layer is relevant to other layers only in terms of its output: some logical property evaluated on the trajectory drawn from the Markov chain. We develop here a framework for constructing a surrogate (discrete-time) Markov chain, with states corresponding to layer output. The expensive simulation of a large Markov chain is therefore replaced by an interpretable abstracted model. We can further use this framework to test whether a posited mechanism could be the driver for a specific macro-scale behaviour exhibited by the model.

We use a powerful Bayesian non-parametric regression technique based on Gaussian process theory to produce the necessary elements of the abstractions above. In particular, we observe trajectories of the original system from which we infer the satisfaction of logical properties for varying model parametrisation, and the dynamics for the abstracted system that match the original in behaviour.

The final part of the thesis presents a novel continuous-state process approximation to the macro-scale behaviour of discrete-state Markov chains with large state-spaces. The method is based on spectral analysis of the transition matrix of the chain, where we use the popular manifold learning method of diffusion maps to analyse the transition matrix as the operator of a hidden continuous process. An embedding of states in a continuous space is recovered, and the space is endowed with a drift vector field inferred via Gaussian process regression. In this manner, we form an ODE whose solution approximates the evolution of the CTMC mean, mapped onto the continuous space (known as the fluid limit). Our method is general and differs significantly from other continuous approximation methods; the latter rely on the Markov chain having a particular population structure, suggestive of a natural continuous state-space and associated dynamics.

Overall, this thesis contributes novel methodologies that emphasize the importance of macro-scale behaviour in modelling complex systems. Part of the work focuses on abstracting large systems into more concise systems that retain behavioural characteristics and are interpretable to the modeller. The final part examines the relationship between continuous and discrete state-spaces and seeks for a transition path between the two which does not rely on exogenous semantics of the system states. Further than the computational and theoretical benefits of these methodologies, they push at the boundaries of various prevalent approaches to stochastic modelling.

Lay Summary

Processes from chemical reactions and population fluctuations in ecosystems to server queuing systems, are well described by non-deterministic (stochastic) formal models known as Markov chains. Our interest in such systems is often an emergent macro-scale behaviour, or phenomenon, which can be characterised by the satisfaction of a set of properties. Although theoretically elegant, the fundamental low-level nature of Markov chains makes the analysis of such phenomena of interest difficult. Particularly, it is not easy to determine the driving mechanisms for the emergent phenomenon, or to predict how changes at the Markov chain level will influence the macro-scale behaviour.

The difficulties arise primarily from two aspects of such models. Firstly, as the number of components in the modelled system grows, so does the state-space of the Markov chain, often making behaviour characterisation untenable under both simulation-based and analytical methods. Secondly, in a model where states represent primitive aspects of system components, the phenomenon of interest often varies significantly with respect to this low-level aspect that states represent.

This work focuses on providing methodological frameworks that circumvent these issues by developing abstraction strategies, which preserve the phenomena of interest. To begin with, we express behavioural characteristics of the system in terms of logical properties, verifiable on simulations of the Markov chain model. This allows us to group states according to their probability of satisfying the logical properties that characterise macro-scale behaviour, in order to produce a simplified (abstracted) Markov chain. We similarly abstract Markov chains which are parts of an overarching multi-scale system; our abstraction aims to preserve any output which is relevant to other parts of the system for the emergence of some macro-scale behaviour of interest. The final part of the thesis presents a novel continuous-state process approximation to the macro-scale behaviour of discrete-state Markov chains with large state-spaces.

Overall, this thesis contributes novel methodologies that emphasize the importance of macro-scale behaviour in modelling complex systems. Further than the computational and theoretical benefits of these methodologies, they push at the boundaries of various prevalent approaches to stochastic modelling.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Overview | 1 |
| 1.2 | Context | 1 |
| 1.3 | Contribution | 2 |
| 1.4 | Structure | 3 |
| 2 | Background | 5 |
| 2.1 | A sample of probability theory | 5 |
| 2.2 | Markovian systems | 6 |
| 2.2.1 | Memoryless-ness | 7 |
| 2.2.2 | Discrete-time Markov chains | 8 |
| 2.2.3 | Continuous-time Markov chains | 9 |
| 2.2.4 | Markov processes | 11 |
| 2.2.5 | The Chapman-Kolmogorov equation | 12 |
| 2.3 | Statistics | 19 |
| 2.3.1 | The problem of induction | 20 |
| 2.3.2 | Frequentism | 22 |
| 2.3.3 | Bayesianism | 23 |
| 2.3.4 | Gaussian processes | 25 |
| 2.4 | Logic and modelling | 27 |
| 2.4.1 | Formal modelling and verification for CTMCs | 28 |
| 3 | Coarsening discrete systems | 31 |
| 3.1 | Population models | 33 |
| 3.2 | Behaviour through logical properties | 34 |
| 3.3 | High level method description | 34 |
| 3.4 | Satisfaction probability as a function of initial conditions | 37 |
| 3.5 | Dimensionality reduction of behaviours | 38 |

| | | |
|----------|---|-----------|
| 3.6 | Clustering and structure discovery | 41 |
| 3.7 | Constructing coarse dynamics | 41 |
| 3.8 | Discussion | 47 |
| 4 | Statistical abstraction for multi-scale spatio-temporal systems | 49 |
| 4.1 | Background | 50 |
| 4.1.1 | Spatio-temporal agent models | 50 |
| 4.1.2 | Multi-scale models | 51 |
| 4.1.3 | Simulating multi-scale systems | 51 |
| 4.2 | Methodology for statistical abstraction | 52 |
| 4.2.1 | Statistical abstraction framework | 52 |
| 4.2.2 | Approximating the underlying probability function | 53 |
| 4.3 | Related work | 55 |
| 4.4 | The case of <i>E. coli</i> chemotaxis | 55 |
| 4.4.1 | Simulating chemotaxis in <i>E. coli</i> | 58 |
| 4.4.2 | Abstracting the <i>E. coli</i> chemotaxis pathway | 59 |
| 4.4.3 | Results | 63 |
| 4.5 | Closing the information loop | 66 |
| 4.5.1 | Original model | 69 |
| 4.5.2 | Abstracted model | 71 |
| 4.5.3 | Results | 74 |
| 4.6 | Discussion | 75 |
| 5 | Continuous approximations to discrete systems: the geometric fluid approximation | 81 |
| 5.1 | Background theory and related work | 82 |
| 5.1.1 | Continuous relaxation and the <i>fluid limit</i> | 82 |
| 5.2 | Methodology | 85 |
| 5.2.1 | Eigen-embeddings of CTMCs | 85 |
| 5.2.2 | Diffusion maps | 88 |
| 5.2.3 | Gaussian processes for inferring the drift vector field | 90 |
| 5.2.4 | Consistency result | 91 |
| 5.3 | Empirical observations | 92 |
| 5.3.1 | Models | 93 |
| 5.3.2 | Assessing fluid solution and mean trajectory in embedding space | 95 |
| 5.3.3 | Embedding a subset of the system | 100 |
| 5.3.4 | First passage times | 100 |

| | | |
|----------|--|------------|
| 5.4 | Conclusion | 103 |
| 6 | Conclusions and future directions | 105 |
| | Bibliography | 107 |
| | Appendix A Supplementary material for the statistical abstraction frame- work | 115 |
| A.1 | Gaussian process classification details | 115 |
| A.2 | Simulation schemes for <i>E. coli</i> model | 117 |
| A.2.1 | Simulation scheme for original <i>E. coli</i> model | 117 |
| A.2.2 | Simulation scheme for abstracted <i>E. coli</i> model | 117 |
| A.3 | Constants of <i>D. discoideum</i> model | 117 |
| | Appendix B Supplementary material for the Geometric Fluid Approxi- mation | 121 |
| B.1 | Diffusion maps for Markov chains | 121 |
| B.1.1 | Undirected graphs | 121 |
| B.1.2 | Embedding unweighted, undirected, grid graphs | 124 |
| B.1.3 | Directed graphs | 127 |

Chapter 1

Introduction

1.1 Overview

Stochastic dynamical systems are ubiquitous in nature. It is therefore desirable to model such systems mathematically to facilitate understanding, prediction, and control. Markov chains are often a natural choice for modelling these systems — the properties of memoryless-ness and non-teleology, where system evolution is not driven by any past experience or future goal but only the present state, are particularly attractive for physical models. There exist many variations of Markov chains, differing in either the interpretation of time or the nature of possible configurations (states) available to the system. Most of this thesis builds upon the continuous-time Markov chain (CTMC) model, with a finite discrete set as the state space and exponentially distributed transition times for each possible transition from state to state.

Description of a physical system in this formalism often produces very large state spaces if the system has a large number of possible configurations, making subsequent analysis of the system particularly onerous. These large models are usually a result of a theoretical, reductionist-driven understanding of the system, translated into the Markov chain formalism. In reality, many a time the interest is for a particular behaviour emerging from this low-level model, which may be described by a simpler, phenomenological model of the original system.

1.2 Context

This work follows in the spirit of many others to establish connecting links between system descriptions at different levels. This has been achieved in numerous fields, the most

notable example of which is the use of statistical ensembles to relate microscopic particle states and dynamics to macroscopic thermodynamic states and dynamics. However, such connexions are notoriously difficult to achieve in complex systems with non-linearly interacting components.

Difficulties aside, there is significant literature on *coarsening* CTMCs. These are attempts to reduce the state-space of a CTMC by partitioning the original set of states into macro-states. For exact coarsening, the Markov chain must be *lumpable*: a property which imposes strict conditions on the partitioning with respect to the transition rates between aggregated states (Kemeny and Snell, 1960). It is in general not trivial to determine whether a chain is lumpable, or given that it is, to find appropriate partitions. To address this, there have been efforts to develop approximate methods (Abate et al., 2015; Buchholz and Kriege, 2014; Dayar and Stewart, 1997; Franceschinis and Muntz, 1994) with a focus on analytic bounds for the coarsening accuracy. Such methods are completely agnostic to downstream use of the chain, so that the coarsening is solely driven by the transition structure of the chain, not the utility of the coarsening. This imposes a heavy burden on the methods since they lack any sort of extrinsic guidance to construct the partitioning and coarse dynamics.

On the other side of the spectrum, we have approximations of a different nature: *continuous state approximations*. Instead of looking at reducing the size of the state-space, such approximations allow for a continuous relaxation of the discrete states (or some dimensions of the states). This allows one to construct continuous distributions over the state-space, parametrised by a small set of variables evolving over time (Kurtz, 1971; Schnoerr et al., 2017b). It is often computationally efficient to analyse the behaviour of a system in this fashion, and one can derive a shrinking approximation error to the true distribution under scaling. The latter advantages are the prime reasons for the wide application of such methods in chemical reaction networks, where particle numbers tend to be high enough to warrant a continuous state-space model. However, chemical reaction networks are highly structured systems, with obvious choices for the continuous state-space and dynamics. As a result, methods for the evolution of the approximating continuous distributions have a constrained application domain on CTMCs that can be expressed in a chemical reaction network formalism.

1.3 Contribution

In this thesis we circumvent many of the analytical issues that plague such systems by adopting a *phenomenological stance*. We formalise macro-scale behaviour of interest that

the system exhibits using logical properties interpreted on the low-level description. This allows us to examine the mechanisms responsible for that behaviour in the system, and so to construct surrogate abstracted models with a statistically consistent behaviour. The abstracted models have the benefit of a concise state-space and dynamics, interpretable in terms of the logical properties used to capture the behaviour of interest. In the same vein, we regard continuous approximations to discrete systems as abstractions, where the driving mechanism is a continuous relaxation of the state-space and dynamics of the discrete system. We are then able to develop a method for constructing continuous approximations for arbitrary discrete-state CTMCs, by utilising statistical tools from machine learning to infer the underlying continuous relaxation to the state-space and dynamics.

1.4 Structure

Some structure is often useful. We begin by laying out the fundamental definitions and theoretical tools in Chapter 2, necessary for the work that follows. In Chapter 3 we present an approach for coarsening a discrete state CTMC based on satisfaction of logical properties defined on CTMC trajectories. In Chapter 4 we utilise a similar coarsening in the context of a multi-scale system; we propose a framework to develop interpretable statistical surrogates for CTMCs which are layers in an overarching multi-scale model. The statistical surrogates are simpler chains which reflect a mechanistic link for observed behaviour at different scales in the model. The focus in Chapter 5 is the generalisation of continuous-state approximations to discrete-state CTMCs. We present a novel method for recovering a continuous relaxation of the state-space and endowing it with deterministic dynamics, which relies on statistical methods for inference. We highlight the connexion it bears to the classical *fluid limit* approximation and demonstrate its applicability on systems not amenable to the classical approximation. Finally, we end with some concluding remarks and future directions for this work in Chapter 6.

Chapter 2

Background

2.1 A sample of probability theory

One cannot talk about contemporary probability theory and stochastic processes without talking about Kolmogorov. Andrei Nikolaevich Kolmogorov (1903–1987) made contributions across the entire realm of mathematics, but was particularly instrumental in laying the foundations of probability theory as it is understood today. In 1933 he published *Grundbegriffe der Wahrscheinlichkeitsrechnung (The Foundations of Probability Theory)*, his axiomatic construction of probability theory building on Borel’s earlier work with measure theory, which was to be revered by the mathematical community for bringing probability theory under the fold of pure mathematics. Many useful results stemmed from this to form the rich canopy of probability theory we stand under today. Some of these were rigorous proofs or necessary conditions for empirically derived equations used by physicists for years (Fokker-Planck, Chapman, Smoluchowski, etc.). Kolmogorov’s contribution to the study of non-deterministic systems cannot be overstated.

We present here the three axioms. Consider a set of *outcomes* Ω , containing all possible outcomes of an experiment¹. Take F to be the *event space*² containing every possible *set of outcomes* E , including Ω . We say that the function $P(E) \in \mathbb{R}$ assigns *probability* to each event E if the following are true:

(i) $P(E) \geq 0, \forall E \in F$;

(ii) $P(\Omega) = 1$;

¹The notion of an *experiment*, or *trial*, is that used in probability theory here; it implies a procedure that can be infinitely repeated and each time results in a random *outcome* from a well-defined set.

²The *event space* F contains sets that form a closed system under the union and intersection set operations — i.e. it is a σ -algebra.

- (iii) any countable sequence $(E_i)_{i \in \mathbb{N}}$, of *non-overlapping* sets ($E_i \cap E_j = \emptyset \forall i \neq j$), satisfies $P(\cup_i E_i) = \sum_i P(E_i)$.

These three are all the axioms necessary³ to define a tuple (Ω, F, P) that satisfies them as a *probability space*, with Ω the *sample space* (the set of all possible outcomes), F the set of *events* (sets of outcomes), and $P : F \rightarrow \mathbb{R}_{\geq 0}$ the probability function mapping events to probabilities.

Since Kolmogorov there have been others to formalise probability in different ways, often motivated by a Bayesian approach⁴ (see Cox's theorem and de Finetti's theorem). Some of these alter the axioms above slightly, but for most cases the axiomatic bases remain consistent in terms of their implications regarding probability calculus. Hence, beyond these historical and philosophical aspects presented in the early sections of this work we shall follow the edict 'Let us calculate' as issued by Leibniz, and be content. There is a discussion to be had about the connexion of data to probability theory that is touched upon in Section 2.3. Markov systems as presented here are a purely theoretical construction abiding by the laws of probability calculus, and therefore we can suspend our worries until we are confronted with the task of fitting such models to *real data* (observations of Nature).

2.2 Markovian systems

Much of this work is concerned with constructing approximations to a wide class of systems characterised by Markovian dynamics. We therefore find it useful to revise here some of the fundamental theory for Markov processes in general, and in particular continuous-time Markov chains (CTMCs).

The theory of Markov chains was largely established by Andrei Andreevich Markov, and hence bears his name today (Basharin et al., 2004). The endeavour started as an attempt to correct a claim by Pavel Alekseevich Nekrasov, that independence is necessary for a collection of random variables to obey the weak law of large numbers. Fuelled by this, and in close conversation with Alexander Alexandrovich Chuprov, Markov first formalised the concept of a simple (first-order) binary state chain in a seminal paper

³Trivial but useful consequences to note are: (i) for $\bar{E} = \Omega \setminus E$, $P(\bar{E}) = 1 - P(E)$; and (ii) $P(\emptyset) = 0$.

⁴In his preface to a Russian translation of Bernoulli's *On the Law of Large Numbers* (*O Zakone Bolshikh Chisel* by Yu. V. Prokhorov in 1986) Kolmogorov writes: "The cognitive value of probability theory lies in the establishment of strict regularities resulting from the combined effects of mass random phenomena. The very notion of mathematical probability would have been fruitless if it were not realized as the frequency of a certain result under repeated experimentation. That is why the works by Pascal and Fermat can be viewed as only the prehistory of probability, while its true history begins with J. Bernoulli's law of large numbers."

(Markov, 1906), which was expanded by the time of publication (1907) to describe a general countable, finite, discrete state-space. Requiring ergodicity, Markov proved first that the Weak Law of Large Numbers (WLLN) and later that the Central Limit Theorem (CLT) hold for a sequence of chain-dependent random variables.

In 1913, Markov published the third edition of his textbook celebrating the 200th anniversary of the WLLN by Bernoulli (1713), where we find the first application of Markov chains: modelling a letter sequence as a binary state Markov chain (vowel / consonant). Markov constructed transition probabilities for vowel to vowel ($p_1 = 0.128$) and consonant to vowel ($p_2 = 0.663$) for A. S. Pushkin’s poem “Eugeny Onegin” (20,000 letters), and further calculated the stationary vowel distribution ($p = 0.432$). He did the same for S. T. Aksakov’s novel “The Childhood of Bagrov, the Grandson” of 100,000 letters. Today these are but trivial applications of Markov chains, as their application abounds in almost every quantitative field of study.

The definitions and derivations given below follow those found in (Gardiner, 2009; Norris, 1998), and we refer the reader to those texts for more detail where it might be lacking.

2.2.1 Memoryless-ness

The defining property of Markovian systems is that they are *memoryless*. This implies that given the current state of such a system, one has *maximal information* about its possible evolution. Any additional information about the history of the system, i.e. past events, has no bearing on predictions about the future. Mathematically, consider a system whose state is a time-dependent random variable $X(t)$, which we observe to take values $x_i \in \Xi$ at some times $t_i \in [0, \infty) = \mathbb{R}_{\geq 0}$, where Ξ is an arbitrary state-space, and $i \in \{0, 1, 2, \dots\} = \mathbb{N}_0$ is an index such that $t_{i+1} \geq t_i$. We assume that the system is fully described by the set of joint probability densities $p(x_0, t_0; x_1, t_1; x_2, t_2; \dots)$. Then the following statement

$$p(x_i, t_i; x_{i+1}, t_{i+1}; \dots | x_j, t_j; x_{j-1}, t_{j-1}; \dots) = p(x_i, t_i; x_{i+1}, t_{i+1}; \dots | x_j, t_j) \quad (2.1)$$

for $i \geq j$, expresses the fact that when considering the future, only the most recent condition is relevant and any further information about the past can be discarded without loss. This is the *Markov property* or *Markov assumption*, and is the basis of a *Markov process*. It follows from this that any joint conditional probability can be broken down

into a product:

$$p(x_0, t_0; x_1, t_1; \dots; x_n, t_n) = p(x_0, t_0) \prod_{i=1}^n p(x_i, t_i | x_{i-1}, t_{i-1}), \quad (2.2)$$

which is relevant in our investigation of how probability distributions over the state-space evolve over time in such a system.

2.2.2 Discrete-time Markov chains

We first look at discrete-time, where a random variable $X(n) = X_n$ evolves over fixed time steps $n \in \mathbb{N}_0$. Transitions in a discrete-time Markov chain (DTMC) are described by a *stochastic matrix* $P = (p_{ij} : i, j \in I)$, where each row is a discrete probability distribution over the countable state-space I . An initial probability distribution π over I is a probability distribution for the random variable X_0 , defining initial state probabilities for the DTMC.

Definition 2.2.1. A collection of random variables $\{X_n\}_{n \in \mathbb{N}_0}$ constitutes a *discrete-time Markov chain* with *initial distribution* π and *stochastic matrix* P (DTMC(π, P)), if and only if it satisfies

(i) $P(X_0 = i) = \pi_i$, and

(ii) $P(X_{n+1} = j | X_n = i, \dots, X_0 = k) = P(X_{n+1} = j | X_n = i) = p_{ij}$.

In the above: X_n takes values from a countable state-space $I = \{1, 2, 3, \dots\}$; π is a probability distribution over I , such that $\sum_{i \in I} \pi_i = 1$; and $P = (p_{ij} : i, j \in I)$ is a *stochastic matrix*, with $\sum_{j \in I} p_{ij} = 1$.

There are two discrete aspects in a DTMC, which we can relax to be continuous. A continuous-time setting gives rise to *continuous-time Markov chains* (CTMCs), where the time between transitions is also a random variable. A continuous state-space I can be trivially accommodated by replacing the transition matrix with a stochastic kernel; however, many of the results and properties we rely upon, such as that the chain converges to a stationary distribution if it is ergodic, are not trivially carried over⁵. Finally, a *Markov process* is a *stochastic process* which satisfies the *Markov assumption*. It is usually defined over a continuous-time domain and can in general take values from a continuous state-space — the other two cases (DTMCs and CTMCs) are encompassed in this general class.

⁵Theodore Harris defined recurrence conditions which are sufficient for the existence of stationary distributions in general state-space Markov processes (Baxendale, 2011). Such processes are termed *Harris recurrent*.

2.2.3 Continuous-time Markov chains

Here the discrete-time gives way to a continuous-time domain $t \in \mathbb{R}_{\geq 0}$, but we retain the discrete, countable state-space I . A system in some state $i \in I$ will jump to a different state $j \in I$ after some time S_i . Satisfaction of the Markov property demands that the time already spent in a state i does not affect the remaining time in that state. If the countable state-space is injected in \mathbb{R} , a time-dependent random variable $X(t) \in I$ which is a continuous-time Markov chain will be a càdlàg function of time. The stochastic process $X(t)$ is also known as a *jump process*. There are different equivalent definitions one can give for a continuous-time Markov chain; we give here the one in terms of its jump chain and holding times as given by Norris (1998).

Definition 2.2.2. Let $X(t)$, $t \in \mathbb{R}_{\geq 0}$, be a right-continuous process with values in a countable set I . Let Q be a Q -matrix on I with jump matrix Π , such that:

$$Q_{ij} \in \mathbb{R}_{\geq 0} \quad \forall i \neq j, \quad (2.3)$$

$$Q_{ii} = -\sum_{j \neq i} Q_{ij} \equiv -Q(i) \quad \forall i, \quad (2.4)$$

and

$$\Pi_{ij} = \begin{cases} Q_{ij}/Q(i) & \text{if } i \neq j \wedge Q_{ii} \neq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (2.5)$$

The process $X(t)$ is a *continuous-time Markov chain* with initial state distribution π and generator matrix Q (CTMC(π, Q)), if it satisfies:

- (i) $P(X(0) = i) = \pi_i$; and
- (ii) $X(t) = Y_n$ for $\sum_{k=0}^{n+1} S_k \geq t \geq \sum_{k=0}^n S_k$, such that its jump chain $\{Y_n\}_{n \in \mathbb{N}_0}$ is a DTMC(π, Π), $S_0 = 0$, and for each $n \geq 1$, $S_n \mid Y_{n-1}$ is an exponentially distributed variable with rate parameter $Q(Y_{n-1})$.

This definition emphasises the so-called *holding times* (also residence, or sojourn times), i.e. the random time that the system spends into any one state. The exponential distribution of the holding times is a simple consequence of the Markovian nature of the process; it also naturally suggests an exact algorithm to sample trajectories of CTMCs by drawing repeatedly exponential random numbers. This consideration forms the basis of the celebrated *Gillespie's algorithm*, widely used in the field of systems biology and known as the *stochastic simulation algorithm* (SSA) (Gillespie, 1977). Note that if there are no possible transitions out of a state k (i.e. $Q_{kk} = 0$) then k is an *absorbing state*,

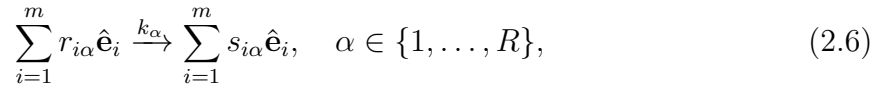
and the process will eventually end up in one such state if it exists to remain there for all future times; chains with absorbing states are not ergodic.

Population continuous-time Markov chains

A special case of such systems is *population* CTMCs, where the state-space is organised along populations. Population CTMCs (pCTMCs) are frequently used in many scientific and engineering domains; we will use here the notation of chemical reactions as it is widespread and intuitively appealing.

Definition 2.2.3. A *population CTMC* is a continuous-time Markov chain with a discrete state-space I , and an associated transition rate matrix Q . Each state in I counts the number of entities of each type or *species* in a population, $\xi \in \{\mathbb{N}_0\}^m$ for m species. Transitions in this space occur according to the rates given by Q .

The transitions can be regarded as occurrences of chemical reactions, written as



where the index α runs over the allowed reactions. In a reaction α , $r_{i\alpha}$ particles of species $\hat{\mathbf{e}}_i$ are consumed and $s_{i\alpha}$ particles of the species are created (species unit vectors are orthonormal: $\langle \hat{\mathbf{e}}_i | \hat{\mathbf{e}}_j \rangle = \delta_{i,j}$). The transition rate $q(\xi, \xi')$ between state $\xi = \sum_{i=1}^m x_{i\alpha} \hat{\mathbf{e}}_i$ and state ξ' , is given by

$$q(\xi, \xi') = \begin{cases} -\sum_{\alpha} \tau_{\alpha}(\xi) & \forall \xi = \xi', \\ \sum_{\alpha \in \mathcal{A}} \tau_{\alpha}(\xi), & \forall \xi \neq \xi' \wedge |\mathcal{A}| > 0, \\ 0 & \text{otherwise;} \end{cases} \quad (2.7)$$

where τ_{α} is the *propensity function* for reaction α , and \mathcal{A} is the set of all reactions where

$$\xi' = \sum_{i=1}^m (x_{i\alpha} - (s_{i\alpha} - r_{i\alpha})) \hat{\mathbf{e}}_i \quad \forall \alpha \in \mathcal{A}.$$

The propensity function $\tau_{\alpha}(\xi)$ of reaction α is derived from combinatorial considerations according to the kinetic laws governing the system. For the usual *mass action kinetics*, the propensity function of reaction α for state $\xi = \sum_{i=1}^m x_{i\alpha} \hat{\mathbf{e}}_i$ is given by

$$\tau_{\alpha}(\xi) = k_{\alpha} \Omega \prod_{i=1}^m \frac{x_{i\alpha}!}{\Omega^{r_{i\alpha}} (x_{i\alpha} - r_{i\alpha})!},$$

where Ω is the system size and k_α is the macroscopic reaction rate constant of reaction α . The transition rates $q(\xi, \xi')$ reconstruct the rate matrix Q of the CTMC.

Remark. In these definitions we have not specifically guarded against exploding CTMCs, non-ergodic chains, non-minimal chains and other such exotic creatures. We acknowledge their existence but let them lie while we explore the rest of the menagerie.

2.2.4 Markov processes

As a generalisation of continuous-time and continuous state-space systems we find *stochastic processes*; when these satisfy the Markov assumption, they are *Markov processes* and are amenable to analysis which often yields solutions or partial solutions to questions one can ask about the process. As we shall see, much of the analysis runs in the same line as that of Markov chains, and we can in fact formulate a CTMC as a Markov process where the measure integral is a step function with steps on the values of I .

We have already defined a general Markov process in Section 2.2.1. Here we impose an additional condition that the paths of the stochastic Markov process $X(t)$ be continuous. Continuous sample paths will allow us to derive differential equations to analyse the system and are therefore desirable. Even though the continuity condition may not be strictly true for the system in question, if the violations are small enough it can be a very good approximation.

Definition 2.2.4. Let $X(t)$ be a time-dependent random variable, which we observe to take values $x_i \in \Xi$ at some times $t_i \in \mathbb{R}_{\geq 0}$, where $\Xi \subseteq \mathbb{R}^n$ is a state-space, and $i \in \{0, 1, 2, \dots\} = \mathbb{N}_0$ is an index such that $t_{i+1} \geq t_i$. We assume that the system is fully described by the set of joint probability densities $p(x_0, t_0; x_1, t_1; x_2, t_2; \dots)$. Then $X(t)$ is a *continuous Markov process* if it satisfies:

(i) the Markov property:

$$p(x_i, t_i; x_{i+1}, t_{i+1}; \dots | x_j, t_j; x_{j-1}, t_{j-1}; \dots) = p(x_i, t_i; x_{i+1}, t_{i+1}; \dots | x_j, t_j)$$

for $j \geq i$, and

(ii) Lindeberg's condition:

$$\lim_{\Delta t \rightarrow 0} \left\{ \frac{1}{\Delta t} \int_{|x-z| > \epsilon} p(x, t + \Delta t | z, t) dx \right\} = 0 \quad (2.8)$$

for any $\epsilon > 0$ and uniformly in $z, t, \Delta t$.

Lindeberg's condition demands that the probability of finding $z = X(t + \Delta t)$ farther than a distance ϵ of $x = X(t)$, will vanish faster than Δt . Markov jump processes like pCTMCs violate the condition, since

$$P(X(t + \Delta t) | X(t) = z) = u(z) e^{Q\Delta t} = u(z) \sum_{k=0}^{\infty} \frac{Q^k (\Delta t)^k}{k!},$$

where $P(X(t))$ is now a row vector which represents the probability mass function over the countable state-space of the CTMC at time t , and $u(z) = \delta_{x,z}$ is a row vector which represents state z . Brownian motion, on the other hand, satisfies the continuity condition while being nowhere differentiable(!)

2.2.5 The Chapman-Kolmogorov equation

Chapman and Kolmogorov independently derived the equation which relates conditional probability distributions for different random variables of a Markov process to each other. Considering variables in a general stochastic process $\{x_i\}$ and as a consequence of the third axiom, we have that

$$p(x_2, t_2) = \int p(x_1, t_1; x_2, t_2) dx_1 = \int p(x_2, t_2 | x_1, t_1) p(x_1, t_1) dx_1, \text{ and} \quad (2.9)$$

$$p(x_2, t_2 | x_0, t_0) = \int p(x_2, t_2 | x_1, t_1; x_0, t_0) p(x_1, t_1 | x_0, t_0) dx_1. \quad (2.10)$$

The first equation is often referred to as *marginalisation*, where we reduce a probability distribution over a set of variables S to one over a subset of S . The second equation expresses that the same is possible for conditional probability distributions over a set S .

Introducing the Markov assumption to the stochastic process, where $t_{i+1} \geq t_i$ such that $p(x_2, t_2 | x_1, t_1; x_0, t_0) = p(x_2, t_2 | x_1, t_1)$, we now have that

$$p(x_2, t_2 | x_0, t_0) = \int p(x_2, t_2 | x_1, t_1) p(x_1, t_1 | x_0, t_0) dx_1, \quad (2.11)$$

which is the *Chapman-Kolmogorov equation* (CKE). For a discrete state-space the integral becomes a matrix multiplication:

$$P(x_2, t_2 | x_0, t_0) = \sum_{x_1} P(x_2, t_2 | x_1, t_1) P(x_1, t_1 | x_0, t_0). \quad (2.12)$$

Under assumptions related to the continuity of the process given below, one can derive the *differential Chapman-Kolmogorov equation* (dCKE)

$$\begin{aligned} \partial_t p(z, t | y, t') &= - \sum_i \partial_i [A_i(z, t) p(z, t | y, t')] \\ &\quad + \sum_{i,j} \frac{1}{2} \partial_i \partial_j [B_{ij}(z, t) p(z, t | y, t')] \\ &\quad + \int [W(z | x, t) p(x, t | y, t') - W(x | z, t) p(z, t | y, t')] dx, \end{aligned} \quad (2.13)$$

where the operators ∂_t , ∂_i are the differential operators $\frac{\partial}{\partial t}$, $\frac{\partial}{\partial z_i}$ respectively, and $t \geq t'$.

The quantities A_i , B_{ij} (generally space-time dependent) relate to continuous motion in dimensions i , j of the state-space of the process, whereas the function $W(x | z, t)$ relates to discontinuous motion (jumps), such that it must vanish $\forall x \neq z$ for a continuous process to emerge. These quantities are defined below in terms of the conditions the process must satisfy.

Conditions and implications

There are three conditions required to bring the original CKE to the differential form above, and additional *initial and boundary conditions* that must be satisfied for the existence of non-negative solutions to such a dCKE. The three conditions are that $\forall \epsilon > 0$:

(i)

$$\lim_{\Delta t \rightarrow 0} p(x, t + \Delta t | z, t) / \Delta t = W(x | z, t)$$

uniformly in x, z, t for $|x - z| \geq \epsilon$;

(ii)

$$\lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \int_{|x-z| < \epsilon} dx (x_i - z_i) p(x, t + \Delta t | z, t) = A_i(z, t) + O(\epsilon);$$

and

(iii)

$$\lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \int_{|x-z| < \epsilon} dx (x_i - z_i)(x_j - z_j) p(x, t + \Delta t | z, t) = B_{ij}(z, t) + O(\epsilon);$$

where the last two are uniform in z, ϵ, t . The *initial condition*

$$p(z, t | y, t) = \delta(y - z)$$

following from how probability density distributions are defined, and additional *boundary conditions* are necessary so that for a specific $A(x, t)$, p.s.d. $B(x, t)$, and non-negative $W(x | y, t)$, the solution to the dCKE exists, is non-negative, and satisfies the CKE. Gardiner points out that the *boundary conditions* are difficult to specify in the full equation, but for the *Fokker-Planck equation* (when $W(x | z, t) = 0$) it is possible to do so. Notice that higher-order quantities taken in the same manner as in the last two conditions necessarily vanish if those conditions are satisfied.

We now examine the three conditions above and link general Markov processes (described by the dCKE) to CTMCs (described by the *Master equation*), diffusion processes (the *Fokker-Planck equation*), and deterministic processes (*Liouville's equation*).

Jump processes: the Master equation

Assume that $A_i(z, t) = B_{ij}(z, t) = 0$ so that the dCKE reduces to:

$$\partial_t p(z, t | y, t') = \int [W(z | x, t) p(x, t | y, t') - W(x | z, t) p(z, t | y, t')] dx; \quad (2.14)$$

we term this the *Master equation* (ME). The first order in Δt approximation to the solution,

$$p(z, t + \Delta t | y, t) = \delta(y - z) \left[1 - \Delta t \int W(x | y, t) dx \right] + W(z | y, t) \Delta t,$$

suggests that sample paths of this process will consist entirely of discontinuous jumps from time to time with no continuous change, hence the term *pure jump process*.

When the function $W(x | z, t)$ is non-zero only at integer values, the ME becomes

$$\partial_t P(n, t | n', t') = \sum_m [W(n | m, t) P(m, t | n', t') - W(m | n, t) P(n, t | n', t')],$$

which is clearly a *pure jump process*, or a CTMC($\pi, Q(t)$) with the state-space indexed by integers. In the latter formalism, $W(m | n, t) = Q_{nm}(t)$ and $P(n, t | n', t') = P_{n'n}(t | t')$ such that the second term in the sum vanishes since $\sum_m Q_{nm}(t) = 0$ by definition, and we are left with

$$\partial_t P(t | t') = P(t | t') Q(t). \quad (2.15)$$

Diffusion processes: the Fokker-Planck equation

Assume that $W(z | x, t) = 0$ so that the dCKE reduces to:

$$\begin{aligned} \partial_t p(z, t | y, t') = & - \sum_i \partial_i [A_i(z, t) p(z, t | y, t')] \\ & + \sum_{i,j} \frac{1}{2} \partial_i \partial_j [B_{ij}(z, t) p(z, t | y, t')], \end{aligned} \quad (2.16)$$

which was already known as the *Fokker-Planck equation* (FPE). This partial differential equation (PDE) describes the forward evolution in time of a probability distribution over space for what is known mathematically as a *diffusion process*. NB: there is a distinction between the mathematical object termed *diffusion process* and the natural phenomenon (e.g. *Brownian motion*, the *Wiener process*, and *Brownian motion* of particles), although the former can often be used as satisfactory model of the latter. In the FPE, $A(z, t)$ is the *drift vector* and $B(z, t)$, which is necessarily p.s.d., is the *diffusion matrix*.

* * *

To relate the FPE to a physical process and its origins, we turn to a phenomenon that has long sought an explanation and accurate mathematical description: *pedesis*⁶. In the early 5th century BCE, the task of interpreting the world, and in essence all critical (or other) thinking, was subsumed under ‘philosophy’; under this umbrella, theories of matter, cosmological origins, politics, etc. abounded in the Hellenistic period. One of these was the *atomic hypothesis*, proposed by Leucippus and his student Democritus. The two philosophers suggested that all matter is composed of small, indivisible⁷, indestructible units which are eternally in motion, separated by void (empty space). Of course these ideas were far from the precise theories we have today. Since epistemology⁸ and the scientific method were at best fledgling notions, the atomic hypothesis was left untested, as many other theories of the time. The idea resurfaced a few times throughout history and found particular favour with John Dalton, who in 1808 formed a consistent theory of matter based on his and others’ experimental works; he hypothesised the existence of small particles constituting matter and combining to form more complex structures, giving rise to the established empirical laws for chemical reactions of the era. This formed

⁶An Ancient (and Modern) Greek word for ‘jumping’, $\pi\eta\delta\eta\sigma\iota\varsigma$ in this context refers to the jitter observed in the trajectories of diffusing particles.

⁷The greek word for indivisible, $\alpha\tau\omicron\mu\omicron\nu$, is where the theory gets its name.

⁸Epistemology had in fact occupied Democritus. He proposed that knowledge is not easily acquired since sensory information is biased, inaccurate, and cannot be trusted, however all evidence we have of the world comes from them. Reasoning has therefore the difficult task of figuring out the ‘truth’ from the corrupted account the senses provide.

the modern theory of atoms, although the community had not yet been convinced of its reality beyond a convenient framework.

In 1827, botanist Robert Brown sought to determine whether the erratic motion⁹ of particles released from pollen grains and suspended in water were indicative of ‘life’. Having observed the same motion in experiments with inorganic matter, he concluded otherwise. However, the problem of mathematically describing *Brownian motion*, as it was termed, and relating it to physical quantities (i.e. mean squared displacement of a particle after time t , and its relationship to a diffusion constant and Avogadro’s number) remained to garner the attention of Albert Einstein. In 1905, Einstein published his elegant treatment of the problem in a paper that essentially gave birth to stochastic modelling. His solution is so valuable because it rests upon the assumptions that: (i) the system is being driven by collective microscopic effects which (ii) are so complicated that their effect on the macroscopic scale (i.e. the pollen grain particle) can be modelled by independent probabilistic events (impacts). These assumptions allow for the analysis of such stochastic processes and are satisfied by many a physical system; we even pretend they are satisfied when we know otherwise¹⁰. Despite some simplifying assumptions (continuous time approximated by small discrete intervals), Einstein constructed special case examples for the FPE, the CKE, and the *Kramers-Moyal* approximation of a stochastic process containing jumps with a continuous one. Marian Smoluchowski derived similar results independently and published them the following year, taking up the mantle of developing the theory of *Brownian motion* and verifying experimentally many of the theoretical results put forth by Einstein and himself.

After reading Einstein’s results, Paul Langevin was convinced that he had an easier way to derive them. In 1908, Langevin published a paper starting from his eponymous equation, which is now interpreted as a stochastic differential equation (SDE) — the journey is examined in (Naqvi, 2005) and outlined here. The starting (Langevin) equation was the usual equation of motion for a particle moving in a viscous medium, perturbed by a random variable η :

$$m \frac{d^2x}{dt^2} = -6\pi\mu a \frac{dx}{dt} + \eta. \quad (2.17)$$

He proceeded to multiply by x and take ensemble averages, which gave rise to the quantity $\langle \eta x \rangle$. Perhaps because he wanted to re-derive the results of Einstein, he had no qualms

⁹The same motion had been gracefully described by the Roman poet and philosopher Lucretius in his well-known scientific poem ‘On the nature of things’ (c. 60 BCE), musing on the motion of dust particles in a sunbeam.

¹⁰The Black-Scholes pricing model essentially models the price of a financial instrument over time as a diffusion process. In reality, the diffusion assumptions are egregiously violated as it is unlikely for the fluxuations to be δ -correlated and the price to follow Markovian dynamics.

in making the following argument: “About the complementary force η , we know that it is indifferently positive and negative and that its magnitude is such that it maintains the agitation of the particle, which the viscous resistance would stop without it,” and that therefore “the average value of the term ηx is evidently null by reason of the irregularity of the complementary forces η ”. After this it becomes almost trivial to reach Einstein’s results by utilising the equipartition theorem. *However*, this reasoning collapses under appropriate scrutiny and in fact produces nonsensical results when carried over to the quantity $\langle v\eta \rangle$. It was Ornstein who produced the correct derivation in a series of papers¹¹, starting from the results of Mrs de Haas-Lorentz and formally integrating the velocity v in time to produce, not only the stationary limit of the rate of change of the mean squared distance $z_\infty = \lim_{t \rightarrow \infty} d \langle x^2 \rangle / dt$ associated to Einstein’s diffusion constant D , but also that it evolves in time according to

$$z(t) \propto 1 - e^{-\alpha t}.$$

These primitive versions of stochastic calculus and differential equations eventually crystallised into *Itô calculus*, which usually interprets the noise η in the stochastic differential equation as the derivative of the Wiener process, such that $\langle \eta(t)\eta(s) \rangle = \delta(t-s)$ as Ornstein earlier found necessary. An *Itô SDE* of the form

$$dx(t) = a(x(t), t) dt + b(x(t), t) dW(t) \quad (2.18)$$

is to be interpreted in terms of the stochastic integral

$$x(t) = x(t_0) + \int_{t_0}^t a(x(t'), t') dt' + \int_{t_0}^t b(x(t'), t') dW(t'), \quad (2.19)$$

where the last term is an integral of $b(x(t'), t')$ with respect to a sample path of the standard Wiener process. There are a couple of interesting things to note at this point. Firstly, the derivative of the Wiener process $dW(t)$ is associated with the term $\eta(t)dt$ of the Langevin equation 2.17; however, recall that the Wiener process, even though continuous, is nowhere differentiable, which leaves us with a paradox. We escape this by saying that the SDE is to be interpreted only in terms of its integral, adequately defined in terms of the mean square limit of a partial sum (with similarity to a Riemann sum). Secondly, the requirement that $\langle \eta(t)\eta(s) \rangle = \delta(t-s)$ and that it integrates to a

¹¹In his derivations, Ornstein appears to have also (inadvertently) developed delta calculus by constructing the function $\delta(x) = 0$ for $x \neq 0$, with the property $\int_{-\infty}^{\infty} \delta(x) dx = 1$. The function was later introduced with due attention by Dirac and hence now bears his name.

continuous process implies that the process must indeed be the Wiener process. This δ -autocorrelation is a theoretical construction which physically cannot occur since it implies infinite bandwidth for $\eta(t)$. Regardless, it is a useful idealisation which is termed *white noise* (since its frequency spectrum is flat, as for white light). The Gaussian nature of $\eta(t)$ is orthogonal to its spectrum but is often assumed.

Using Itô's formula for change of variables in an SDE, we can construct an associated FPE equation. Starting from the general multi-variable Itô SDE

$$dx(t) = \mu(x(t), t) dt + \sigma(x(t), t) dW(t), \quad (2.20)$$

we have that the conditional probability density $p := p(x, t | x_0, t_0)$ obeys the FPE

$$\partial_t p = - \sum_i \partial_i [\mu_i(x, t) p] + \frac{1}{2} \sum_{i,j} \partial_i \partial_j \left\{ [\sigma(x, t) \sigma^\top(x, t)]_{ij} p \right\}. \quad (2.21)$$

Note that the last term in the FPE implies that it is not unique to the SDE — we can construct the same FPE for an SDE with noise scaling $\tilde{\sigma} = \sigma U$, where $UU^\top = I$, so that $\tilde{\sigma} \tilde{\sigma}^\top = \sigma \sigma^\top$. The forward and backward FPEs can be considered to be a result of path integrals (intuitively a weighted sum of all possible realisations of the stochastic process described by an Itô SDE) expressed by the *Feynman-Kac formula*. The latter also allows calculation of solutions to some PDEs by stochastic simulation of the associated process.

Deterministic processes: Liouville's equation

Finally we take both $B(z, t), W(z | x, t) = 0$ in the dCKE 2.13 so that only the term $A(z, t)$ remains. In this case, all stochasticity is lost in the dynamics and we are left with deterministic trajectories for specified initial conditions. The evolution of a probability distribution $p(z, t | x, t')$ over initial conditions obeys the PDE

$$\frac{\partial p(z, t | x, t')}{\partial t} = - \sum_i \frac{\partial}{\partial z_i} [A_i(z, t) p(z, t | x, t')], \quad (2.22)$$

which is a special case of the *Liouville equation* in classical statistical mechanics. This tracks the trajectories of an ensemble of particles, each of which is the solution of the ODE

$$dx(t) = A(x(t), t) dt.$$

Starting from a single state, $p(z, 0 | y, 0) = \delta(z - y)$, the distribution at time t will be $p(z, t | y, 0) = \delta(z - x(t))$ (still a delta function). If the function $A(x, t)$ is highly

variable in x the process might be chaotic, such that uncertainty in initial conditions is amplified in time.

This concludes our examination of various processes as special cases of the dCKE. Note that in general, none of the components of the dCKE have to vanish, giving rise to a diffusion process with jumps.

2.3 Statistics

There seems to be an intimate link between probability theory and *non-deterministic* observations of Nature. *Non-deterministic* here means that we (experimenters or researchers) cannot exactly predict observations of a phenomenon, because of one or more of the following: (i) we lack sufficient information about the experimental set-up; (ii) we lack sufficient computational power to process all relevant information to produce exact predictions; (iii) parts of the data generation procedure (otherwise) introduce ambiguity.

To instantiate the above take the example of *Brownian motion*, as it pertains to the motion a massive particle (e.g. pollen grain) exhibits, amidst a sea of other lighter particles (e.g. water molecules). We generally cannot exactly predict the trajectory of the pollen grain because: (i) we do not have the position and momentum of every single molecule that will influence it; (ii) even if we did, the computational power required to apply the appropriate models (Newtonian mechanics) would be enormous, or non-existent for large enough systems; (iii) measuring particle positions is subject to ambiguity from the apparatus, Newtonian mechanics are only approximate models of reality (as is almost certainly every theoretical model of reality), and if we imagine our particles to become light enough so that quantum mechanical models become relevant, the ambiguity is inherent in the model.

How might one address these problems? The lack of information can be quantified as a source of ambiguity, and properly propagated in our predictive model to accompany our predictions. Depending on the level of accuracy required of the prediction, this might not even be necessary as the results will not substantially change to warrant it. The processing power problem can in cases be solved by enslaving the many aspects that need to be considered, to a summary description that they obey. For instance, statistical mechanics models of Brownian motion consider the effect of many light molecules to be reasonably close to random forces generated by a white Gaussian process, characterised by zero mean and a scaling co-efficient. This is not exactly right, in fact we know it is inaccurate, but it serves as an excellent approximation and makes our prediction task tractable. The last problem of other sources of ambiguity can be similarly addressed:

we can try to quantify the ambiguity source (e.g. measurement errors can reasonably be expected to be samples from a zero-mean Gaussian distribution) and take it into account when making predictions. In short, it might be fruitful to look towards a machinery that parsimoniously captures the ambiguity under the many forms it comes in, such that we can tractably make predictions for non-deterministic processes: that is *statistics*.

The natural question that arises then is how to correctly characterise the stochastic process that generates a set of observed data. This spawned a debate that is still ongoing, of what *probability* really is aside from the mathematical object. The prevailing schools of thought are: (i) Bayesianism, based on the view that probability corresponds to *degrees of belief* of a *rational agent*; and (ii) frequentism, which views probability as a *ratio of outcomes* in the limit of infinite repetitions of the experiment. Both views have merits and reasonable arguments, and both have a number of more nuanced doctrines embraced by various researchers (purposefully or inadvertently).

2.3.1 The problem of induction

Let us begin in 1739, with the great David Hume. In his *A Treatise of Human Nature* published that year, he raised what became known as *the problem of induction*¹² (PoI) which to this day remains unresolved. Hume was a Scottish philosopher when the country was going through its enlightenment period; a staunch Empiricist and Sceptic like others in his time, his problem was how humans come to form ‘ideas’ which generalise beyond the ‘experiences’ producing them¹³. This process is not *reasonably* (deductively) warranted and beyond the obvious fallacies it produces¹⁴, it lacks a sufficient driving principle other than induction itself. The issue strikes at the heart of *causal inference* since, as Hume notes, discovery of causal relations is based on observations of Nature and therefore on induction. In fact, a Uniformity Principle, which assumes that the same causal laws that governed past experiences will do so for future ones, underpins all causal inference. The Uniformity Principle relates to de Finetti’s exchangeability theorem, a cornerstone of predictive Bayesian inference.

The PoI is also deeply linked to philosophy of science because both examine how one should draw correct conclusions from observations. This becomes especially harder

¹²Hume was not the first to harbour such concerns. The Pyrrhonian Sceptic philosopher Sextus Empiricus posed the same (or a very much related) problem in the 2nd century. Hume’s name has however, become inextricably linked to the problem and his form of argument is more widely known so we follow him.

¹³After repeatedly observing an effect associated with the properties of an object, one assumes that all objects with similar properties will have the same effect.

¹⁴*Black swans* could and indeed did exist, even if they remained unobserved for a long time.

when the process is assumed to be non-deterministic (stochastic). As noted by Lawrence et al. (2010, chapter 5) in *A Brief Introduction to Bayesian Inference*, this problem was originally understood as simply a measurement issue. The paradigm set by Newton's *Principia Mathematica* was that the observations one has are simply the truth, which was to be captured by a set of deterministic laws, corrupted by an imperfect measuring process (noise / error). If one could model the noise, one could test agreement between a theory and observations to some degree of accuracy. Further, one could reduce the noise by improving the measurement process. This reductionist approach of deterministic truth and stochastic measurement error soon became insufficient to deal with the many demands made of science: before long the study of large ensemble systems gave birth to statistical mechanics, building a statistical bridge from microscopic classical mechanics to macroscopic laws; quantum mechanics, the most accurate microscopic description we have of matter to date, is a probabilistic model tested using statistics; and finally, modern physics, biology, and social science only have access to a multitude of distant effects (and hence observations corrupted by other factors) of causal models being examined, heavily relying on statistics to guide them to truth. Statistics and probability theory has entrenched its position in modern science, and in doing so has empowered us to study even more non-trivial systems and test more precise models.

As usual, the power comes at a price. It is no longer trivial to assess whether a hypothesis is supported by observation. The mainstream mode of ideal scientific progress, introduced by Karl Popper partly as a solution to the PoI, is that researchers propose theories that make *empirically refutable* predictions. If the prediction does not agree with empirical observations the theory can be cast aside as 'false', and the cycle starts anew. Combined with an *Occam's razor* argument (the more powerful and simpler theory should be tested before competing ones), the theories are refined through this process in a manner similar to Descartes's discarding of 'bad apples' in a basket of ideas, slowly producing a more accurate model of the world. This seems reasonable, but if the predictions are non-deterministic, a statistical framework is needed to assess consistency in finite datasets. The proposal-rejection cycle bears some resemblance to frequentist inference¹⁵, whereby implications of hypotheses are tested against empirical evidence, and the former rejected if found inconsistent. Things to note are that all hypotheses are assumed to eventually be proven false, and that we only have the power to reject hypotheses, not to accept them.

¹⁵Popper himself was an advocate of the *propensity* interpretation of probability, with implications on how one can treat model parameters.

Thomas Kuhn, on the other hand, described the progress of scientific research in a less idealised manner. He suggested that a set of *exemplars* (problems-solutions) serve as ideal examples to follow in the course of *normal science*; the collection of concepts and practices that dominates an era of research is the prevailing *paradigm*. This goes on with minor adjustment, until enough problems arise that are not (sufficiently) addressed by the paradigm. A new paradigm emerges and the cycle goes on. In this *paradigm shift*, all former evidence is re-interpreted in light of the new concepts that form the new paradigm, signifying a *scientific revolution*. Subject to debate, there is claim (Salmon, 1990) that Kuhn’s account better aligns with the Bayesian framework of belief updating.

Regardless of one’s preference between philosophies of science, it is apparent that statistical theory has a seminal function in the process of *learning* (or *inference*) from observation, and hence in the process of science. We look at the two prevailing statistical ideologies for inference here, frequentism and Bayesianism, and motivate their usage in this work. Other attempts have been made to interpret probability, some also aiming to provide a solution to the PoI, but they are mostly marginal.

2.3.2 Frequentism

The frequentist approach to probability is that there is a *true* set of parameters θ^* , which fully characterises the stochastic process generating the data. The researcher then constructs unbiased estimators $\hat{\theta}$ of these parameters (usually maximum likelihood estimators of moments) potentially associated with confidence intervals, and asks questions of the nature “are certain parameter values consistent with my data?”, or on a higher level “is a certain model (with a certain set of parameters) consistent with my data?”. This kind of inquiry is known as *hypothesis testing* and the answer is usually given in terms of *p*-values, or goodness-of-fit tests. The researcher rests easy upon the knowledge that, in the limit of infinite data, the unbiased estimators will converge on the true parameters ($\hat{\theta} \rightarrow \theta^*$) and false models will appear inconsistent with the data. The frequentist methods to parameter inference were advocated by Neyman, Pearson, Fisher, and others, as a response to the Bayesian framework that preceded them. Venn, Reichenbach and von Mises developed an interpretation of probability consistent with these methods and by making great use of frequency arguments. Frequentists wanted to lift inference out of the domain of ‘subjective’ beliefs upon which the Bayesian framework is based, and into an ‘objective’ basis, agnostic of the particular researcher’s or community’s beliefs. It has been argued (Colquhoun, 2014) that their efforts, coupled with publication incentives, have instead driven many fields of science to the ‘reproducibility crisis’ that plague them today.

2.3.3 Bayesianism

The Bayesian framework began with the work of an English clergyman and scholar, the Reverend Thomas Bayes. In 1764, an essay by Bayes (likely written to address the PoI as Zabell (1989) suggests) was published posthumously by his friend Richard Price. The essay treated a special case of the inference problem: having observed a series of binary values, random outcomes of successive Bernoulli trials with the same success probability p , what is the value of p ? The novelty was that Bayes treated the uncertain quantity p as a random variable with an associated density function over the $[0, 1]$ domain (uniform in Bayes' case) before observing the outcomes. Hence, the rules of probability were applied to yield a conditional distribution for p after observing the trials. Laplace later generalised this to arbitrary generative distributions and used the *Principle of Indifference* to influence his choice of prior distributions. Considerations of how to construct adequate priors set different schools of Bayesians apart.

In the Bayesian paradigm, probabilities reflect the *degrees of belief* that the researcher holds about the world. Ideally, the Bayesian agent will have a *prior* probability distribution over a domain of all possible realities Θ . The agent is required to be *rational*, which in this context implies that it obeys the axioms of probability, and additionally that it constructs appropriate priors in the objective school. Therefore, after observing some data \mathcal{D} , it updates its beliefs and calculates a *posterior* distribution according to Bayes' theorem. This concept is expressed in the following equation:

$$p(\theta | \mathcal{D}) = \frac{p(\mathcal{D} | \theta) p(\theta)}{p(\mathcal{D})}, \quad (2.23)$$

where $p(\theta)$ is the *prior*, $p(\mathcal{D} | \theta)$ is the *likelihood* (i.e. given some value of $\theta \in \Theta$, the probability of generating the observations under the model), and $p(\theta | \mathcal{D})$ is the *posterior* distribution. The denominator $p(\mathcal{D})$ has many aliases, including *evidence*, *marginal likelihood*, or in the parlance of statistical mechanics, *partition function*. There are two interesting things about this object: (i) it does not depend on θ , and so acts as a normalisation constant to the numerator so that $\int_{\Theta} d\theta p(\theta | \mathcal{D}) = 1$; (ii) perhaps counter-intuitively, its value is not 1. The latter point warrants unpacking. One would expect the probability of observing the data to be 1 — after all, we have observed the data and no one can argue with that. However, this quantity expresses the (subtly different) probability of observing the data *under the model considered*. That is to say, under all the possible realities the Bayesian agent considers, it gives the probability of

observing such data. Mathematically,

$$p(\mathcal{D}) = \int_{\Theta} d\theta p(\mathcal{D} | \theta) p(\theta)$$

which ensures that the posterior is a normalised probability distribution. The evidence is often the problematic part of Bayesian inference, as calculation of this constant for most complicated likelihoods is analytically intractable, and a slew of approximation methods exist to estimate posteriors without explicitly calculating it.

As in the frequentist setting, we can extend the Bayesian framework to include various models considered and guide us to the right one. Where we had p -values and rejection of inconsistent hypotheses at certain significance thresholds, we now have posterior distributions that ascribe more probability mass to the more likely hypotheses. When one wishes to make a prediction for unobserved variables in the Bayesian framework, instead of picking a single value of θ or a single model, one takes a weighted sum from all possible realities considered according to their posterior distribution. After observing data \mathcal{D} , with the likelihood $p(\mathcal{D} | \theta)$ and prior $p(\theta)$, the Bayesian constructs a predictive distribution for unobserved data \mathcal{D}' ,

$$p(\mathcal{D}' | \mathcal{D}) = \int_{\Theta} d\theta p(\mathcal{D}' | \theta) p(\theta | \mathcal{D}).$$

Common criticisms levied against frequentist inference include the susceptibility of maximum likelihood estimators (MLEs) to outliers, the arbitrary nature of regularisation procedures to deal with the former, the dependence of any conclusions on the sampling scheme, and others. Criticisms on Bayesian inference include the prior specification problem (often the prior is a convenient mathematical object that serves more as a regularisation device than an accurate representation of the beliefs of the researcher), the intransigence of the agent in the face of data originating from a model outside its belief domain (the agent will never decide that none of its beliefs are probable), and the fact that objective beliefs do not translate to invariant uninformative priors.

We recognise the shortcomings of both paradigms, and opt for the following in most of the thesis. First, we posit our assumptions about the generating process in a Bayesian framework, which keeps track of regularisation choices and allows us to leverage the many powerful and flexible inference techniques available. After inference, we attempt to validate what we learned by generating synthetic data from the inferred model and comparing their statistics to the evidence. This is a kind of *goodness-of-fit* test and the entire methodology follows the suggestions of [Gelman and Shalizi \(2013\)](#) in some sense.

2.3.4 Gaussian processes

Gaussian processes (GPs) as a de-noising tool were first proposed in a rudimentary form by Wiener and Kolmogorov independently in the 1940s. Initially used for time-series filtering and smoothing, they were extended in the 1960s by Matheron and Krige for geostatistics (*kriging* as spatial field estimation) and later further generalised as a regression tool within the realm of statistics and machine learning by O’Hagan, Neal, Williams, and Rasmussen. The history is traced in (Cressie, 1990). The treatment below follows the established text by Rasmussen and Williams (2006) to provide an adequate background for GPs and their usage in this thesis. No explicit algebraic results are derived here, since they can be found in (Rasmussen and Williams, 2006).

A Gaussian process is a stochastic process $\{f(t)\}_{t \in T}$, where any finite subset of the collection of random variables in the process $(f(s))_{s_1, \dots, s_n \in T} = (f(t_1), f(t_2), \dots, f(t_n))$ is a multivariate normal variable. A GP can therefore be thought of as a normal distribution over an infinite-dimensional separable Hilbert space \mathcal{H} , where each dimension is a point in the domain of the process. A sample of the Gaussian process is a sample from that normal distribution over \mathcal{H} . The mean and covariance of the GP can be defined to represent a particular distribution over functions, which makes the GP a valuable tool in Bayesian inference. We write

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$$

where $m : T \rightarrow \mathbb{R}$ is the mean function, and $k : T \times T \rightarrow \mathbb{R}$ is the covariance kernel of the normal distribution over \mathcal{H} . The support space of the distribution, \mathcal{H} , can be constructed to be a dense subspace of $L^2(T)$ by choosing an appropriate kernel $k(\cdot, \cdot)$ to be the inner product on \mathcal{H} (e.g. the squared exponential ‘Gaussian’ kernel). For such cases we say that the GP is a *universal approximator* — it can approximate any function in $L^2(T)$ arbitrarily well.

Remark. A GP with a covariance kernel $k(t, t') = \delta_{t, t'}$ is the Gaussian ‘white’ noise, and its time integral is the standard Wiener process which is also a (non-stationary) GP.

An isometry between infinite-dimensional separable Hilbert spaces allows us to view the GP as a linear combination of a (countable) orthonormal basis. As such, a continuous function $f(t)$ over a compact domain T can be composed as

$$f(t) = \sum_i \alpha_i \phi_i(t)$$

where $\{\phi_i\}_{i \in \mathbb{N}}$ is an arbitrary complete orthonormal basis of \mathcal{H} (e.g. the Fourier basis). This shift in perspective conveniently transforms GPs to a tool for Bayesian linear regression with basis functions, where the prior over the coefficients is a normal distribution $\alpha \sim \mathcal{N}(0, \Sigma_p)$. The choice of kernel encodes this prior as well as the choice of basis functions, since $k(t, t') = \phi(t)^\top \Sigma_p \phi(t')$ where $\phi(t)$ is an eigenfunction basis for \mathcal{H} . Note that the number of basis functions is unlimited, and so the prior distribution may be over almost all continuous functions in a compact domain, in contrast to Bayesian linear regression. In practice we can represent the result of the GP regression as a linear combination of as many basis functions as there are observations; the representation is either finite and changes with observations, or infinite and fixed.

Therefore, for a set of stochastic observations $\mathcal{D} = \{(t_i, y_i)\}_{i=1}^N$ where $t_i \in T$ is the input value and $y_i \in \mathcal{Y}$ the output, we construct a Bayesian model with latent variable the function $f : T \rightarrow \mathbb{R}$. A Gaussian process $f \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$ serves as a prior probability distribution over the possible latent functions and we can use Bayes' theorem to recover a posterior distribution for the output function $g : T \rightarrow \mathcal{Y}$. In practice we only evaluate the distribution of the output at some particular point(s) in the domain, $y_\star = g(t_\star)$, $t_\star \in T$. With $\mathbf{t} = (t_i)_{i=1}^N$, $\mathbf{y} = (y_i)_{i=1}^N$, and $\mathbf{f} = (f(t_i))_{i=1}^N$, we write this as

$$p(y_\star | t_\star, \mathcal{D}) = \int p(y_\star | f_\star) p(f_\star | t_\star, \mathcal{D}) df_\star,$$

where

$$p(f_\star | t_\star, \mathcal{D}) = \int p(f_\star | t_\star, \mathbf{t}, \mathbf{f}) \frac{p(\mathbf{y} | \mathbf{f}) p(\mathbf{f} | \mathbf{t})}{p(\mathbf{y} | \mathbf{t})} d\mathbf{f}.$$

The integration may appear daunting, but for certain cases it becomes quite manageable. For instance, a Gaussian likelihood $p(y | f) = \mathcal{N}(y; f, \sigma^2)$ allows the use of standard results for integrating products of Gaussian distributions and makes all operations analytically tractable; the output distribution remains Gaussian. A delta likelihood implies *noiseless* observation of the latent function and is also tractable and results in a Gaussian output distribution. Other likelihoods (e.g. a probit likelihood for classification) require approximation methods to produce output distributions, most commonly Laplace's method, or expectation-propagation (EP), which often present accuracy or computational challenges. Another issue is that this method scales poorly with the number of observations considered¹⁶ and so a number of approximation methods exist to deal with a large dataset.

¹⁶For N observations, the standard GP regression with Gaussian likelihood incurs a memory cost of $\mathcal{O}(N^2)$, and a complexity of $\mathcal{O}(N^3)$ for the required Cholesky decomposition, with an additional cost of $\mathcal{O}(N^2)$ for the predictive mean and variance at a test point respectively.

2.4 Logic and modelling

As noted by Hodges (2018), there may not appear to be a direct link between modelling and logic. Logic is a manner by which various relations can be determined to hold or not between various objects. When speaking of formal logic, the logic consists of a formal language, in which relations or formulae are defined; a deductive system, which dictates how one is to reason and draw conclusions; and a semantics, which interprets arbitrary elements of the logic as elements of a particular domain, and so allows one to determine satisfaction of formulae for a given interpretation.

The theory of logic runs deep, originating from Aristotle and branching through the ages into many refinements and flavours (Shapiro, 2009). A major branch is that of mathematical logic, stemming from some early ideas of Leibniz and heavily developed during the 19th and 20th century. The main development consisted of the efforts of Bolzano, Boole, Cantor, Carnap, Dedekind, Frege, and Peano, to name but a few, to ground known mathematics on logical principles. By 1920, Whitehead and Russell brought together most of these results in the expansive volumes of *Principia Mathematica*, in which they derived many areas of mathematics from elementary logical notions; *Logicism*, as it became known, was popularised in the English speaking world largely due to them. The work of Carnap followed after a couple of decades, containing an interpretation of probability as a logical concept — the *degree of confirmation* of a hypothesis by a piece of evidence — and so provided much of the philosophical grounding for alternatives to frequentism to emerge (including the Bayesianism account discussed above, a more subjective interpretation of logical probability). Problems with Logicism arise and are circumvented time and again by variations of the main theme, but the practical outcome is that logic has now earned a reputable place in mathematics, from logical probability to Tarski's work setting the standard for truth definition in most model-theoretic languages.

A link between Logic and Modelling may be discerned, when we consider how Logicism succeeds in its endeavours. The mathematical objects we take for granted (e.g. numbers, lines, etc.) are derived as logical objects that satisfy a set of relations between them (Russell, 1993). Whether the object has an interpretation in the real world is irrelevant — it exists only as a logical abstraction, relevant to all objects that may satisfy the relations that define it. Similarly, modelling is a logical abstraction for natural phenomena. A model should satisfy certain relations as the object being modelled would satisfy them, but agreement is only guaranteed for a subset of all possible relations: there may be conditions in which the model fails to retain all relationship status. If the model is perfect, then every possible relation will be satisfied in agreement with the modelled object.

We are here reminded of Hume’s empiricism: any form of knowledge from Nature stems from a collection of sensory experiences (phenomena); and the more radical form of the argument in *phenomenalism*: Nature is nothing further than that collection of phenomena. Such phenomena, or relations, include measurements and sensory experiences (perceptions). In this thesis, we embrace the notion of modelling as the agreement in satisfaction of a partial set of relations of the original object, and produce abstractions that agree with respect to a set of logical formulae pertaining to a phenomenon of interest. The title of the thesis reflects this — albeit somewhat redundantly, for what else could modelling be but phenomenological?

2.4.1 Formal modelling and verification for CTMCs

Reasoning about behavioural properties of dynamical systems is a central goal of formal modelling. Recent years have witnessed considerable progress in this direction, with the definition of formal languages (Ciocchetta and Hillston, 2009; Danos et al., 2007) and logics (Donzé and Maler, 2010) which enable compact representations of dynamical systems, and mature reasoning tools to model-check properties in an exact (Kwiatkowska et al., 2011) or statistical way (Jha et al., 2009; Younes and Simmons, 2006).

Metric interval temporal logic

In much of this thesis we will make use of *Metric interval Temporal Logic* (MITL) to examine the behaviour of pCTMCs, as introduced by Maler and Nickovic (2004). We are particularly interested in properties that can be verified on single trajectories, and assume metric bounds on the trajectories, so that they are observed only for a finite amount of time. MITL offers a convenient way to formalise such specifications and we therefore revise it here. Its syntax is described as follows using common grammar notation:

$$\phi ::= \mathbf{tt} \mid \mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathbf{U}_{[T_1, T_2]} \phi_2,$$

where \mathbf{tt} is the true formula, conjunction and negation are the standard boolean connectives, and the time-bounded until $\mathbf{U}_{[T_1, T_2]}$ is the only temporal modality, with the variables T_i taking values from the temporal domain $[0, T]$. Atomic propositions μ are boolean predicates, extended in time.

Since our domain of interpretation is (population) CTMCs, we let the variables in the formulae take values either from the temporal domain, or from the domain of trajectories that the chain draws from (Maler and Nickovic, 2004). A MITL formula is interpreted over a function of time \mathbf{x} , and atomic propositions μ are boolean predicate

transformers: they take a real valued function $\mathbf{x}(t)$, $\mathbf{x} : [0, T] \rightarrow \mathbb{R}^n$, as input, and produce a boolean signal $s(t) = \mu(\mathbf{x}(t))$ as output, where $s : [0, T] \rightarrow \{\mathbf{tt}, \mathbf{ff}\}$. As customary, boolean predicates μ are (non-linear) inequalities on population variables, that are extended point-wise to the time domain. More temporal modalities, such as the time-bounded eventually and always, can be defined in terms of the until operator: $\mathbf{F}_{[T_1, T_2]}\phi \equiv \mathbf{ttU}_{[T_1, T_2]}\phi$ and $\mathbf{G}_{[T_1, T_2]}\phi \equiv \neg\mathbf{F}_{[T_1, T_2]}\neg\phi$.

MITL formulae evaluate as true or false on individual trajectories; when trajectories are sampled from a stochastic process, the truth value of a MITL formula is a Bernoulli random variable. Computing the probability of such a random variable is a model checking problem. Model checking for MITL properties evaluated on trajectories from a CTMC requires the computation of transient probabilities; despite major computational efforts (Kwiatkowska et al., 2011), this is seldom possible exactly due to state-space explosion. Statistical model checking (SMC) methods circumvent such problems by adopting a Monte Carlo perspective: by drawing repeatedly and independently sample trajectories, one may obtain an unbiased estimate of the truth probability, and statistical error bounds can be obtained by employing either frequentist or Bayesian statistical approaches (Jha et al., 2009; Younes and Simmons, 2006). It should be pointed out that such bounds do not carry the same guarantees as numerical results obtained say by transient analysis; however, simply by drawing more samples one may reduce the uncertainty in the bounds arbitrarily.

Armed with such tools for examining relations (logical properties) of a CTMC, we may now attempt to emulate satisfaction of a certain set of properties pertaining to a behaviour of interest that the system manifests. We can thus construct statistically adequate surrogate models which: are better interpretable in terms of satisfaction of logical properties, are often less demanding of resources, and retain the relevant behaviour in a specified range of conditions.

Chapter 3

Coarsening discrete systems

When we examine a system we often wish to assess particular elements of the system behaviour; this is especially the case in quantitative fields, where we have come to characterise the behaviour of a system by a set of measurable properties. As a result, the questions we ask when examining a system can commonly be expressed in terms of satisfaction probabilities of logical properties $p(\phi)$. For instance, a positive Lyapunov exponent indicates that a system is chaotic, low entropy implies a concentrated probability distribution, and infection rates determine how sustainable diseases are within a population.

This makes it possible to substitute the original system with one which will give similar answers to similar questions of property satisfaction, but which is cheaper to query. The result is that experiments probing the behaviour of the system become less costly, and the approximation might substitute the original system in situations where only a coarse measure of $p(\phi)$ is needed. With this in mind we propose a methodological framework to achieve the construction of such substitutes in the case of continuous-time Markov chain (CTMC) systems.

Many issues arise in the attempt to coarsen the state-space of a Markovian system, most notably the loss of its defining characteristic, the Markov property. To illustrate, consider a CTMC which describes a dynamical system. An attempt to partition the states of such a CTMC to construct macro-states for a new CTMC which is exactly consistent with the original will fail, unless certain conditions for lumpability are met. If the system is not lumpable, transition rates from a macro-state to another will be dependent on sojourn times. Further, the macro-state sequence and time spent in past macro-states will contain additional information about the future, since they will have implications about the parts of the macro-states visited. In summary the dynamics describing the evolution of the system through macro-states will no longer be Markovian,

since the original states (in terms of which a Markovian dynamics describes system evolution) will be inaccessible after coarsening.

Broadly speaking, state-space reduction can be achieved by either model simplification, usually by abstracting some system behaviours into a simpler system, or state aggregation, often by exploiting symmetries or approximate invariances. A prime example of model simplification is the technique of time-scale separation, which replaces a large system with multiple weakly dependent sub-systems (Bortolussi et al., 2015a; Gunawardena, 2014; Jacobi, 2012; Rödenbeck et al., 2001). Most aggregation methods are instead based on grouping different states which are similar in terms of their transition probabilities. This idea is at the core of *approximate lumpability*, which extends the exact lumpability relationship by aggregating states based on a pre-defined metric on the outgoing exit rates (Abate et al., 2015; Buchholz and Kriege, 2014; Deng et al., 2011; Milios and Gilmore, 2015; Tschaiowski and Tribastone, 2015).

However, if we are only interested in certain aspects of the state (e.g. whether it satisfies a particular property $\phi(\mathbf{x})$), we might sacrifice some accuracy in the macro-scale dynamics for a coarser, more efficient model (Hoel, 2017; Wolpert et al., 2014).

In this chapter we propose a novel state-space reduction paradigm by shifting the focus from the infinitesimal properties of states (i.e. their transition rates) to the global properties of trajectories. Namely, we seek to aggregate states that yield *behaviourally similar* trajectories according to a set of pre-defined logical specifications. Intuitively, two states will be aggregated if trajectories starting from either state exhibit similar probabilities of satisfying the logical specifications. We define a statistical algorithm based on statistical model checking and Gaussian Process emulation to define this behavioural similarity across the whole state-space of the system. We then propose a dimensionality reduction and clustering pipeline to aggregate states and define reduced (non-Markovian) dynamics. To illustrate our approach, we give a running example of model reduction for the Susceptible-Infected-Recovered-Susceptible (SIRS) model, a non-linear stochastic system widely used in epidemiology. This work was published in *Proceedings of Quantitative Evaluation of Systems (QEST) 2016* (Michaelides et al., 2016).¹

¹Dimitrios Milios, Jane Hillston and Guido Sanguinetti provided general feedback and advice in the development of the material, and edited the manuscript. Dimitrios Milios also contributed Figures 3.5 and 3.6.

3.1 Population models

We will consider *population* models here, formalised by population CTMCs (pCTMCs) as defined previously (Definition 2.2.3). Such models have the convenient trait of a naturally ordered state-space, which we can leverage to efficiently estimate the satisfaction probability of logical statements, or properties, comprised of (temporal) conditions on species counts. Because of the highly structured nature of pCTMCs, where transition rates scale linearly within each ordering dimension (species concentration), we expect that regression methods from machine learning such as a Gaussian process (GP) will competently approximate a function over the state-space. This function should map initial states to property satisfaction probabilities, with relatively few observations (sample trajectories of the original system from simulation). In principle, the ordered state-space condition could be lifted, and the GP emulation replaced by complete statistical model checking (SMC) for estimating satisfaction probabilities for each initial state of the system. However, this would become extremely costly for systems with larger state-spaces, or for a larger set of properties which would require more samples to accurately estimate satisfaction probabilities. We therefore retain our attention on pCTMC models here, and in particular an SIRS model.

Example 1.1 We introduce our running example, the Susceptible-Infected-Recovered-Susceptible (SIRS) model of epidemic spreading. The SIRS model is a discrete stochastic model of disease spread in a population, where individuals in the population can be in one of three states, Susceptible, Infected and Recovered. There are different variations of the model, some open (individuals can enter and exit the system), others with individuals relapsing to a susceptible state after having recovered. Here, we consider a relapsing, closed system, which evolves in a discrete, 2-dimensional state-space, where dimensions are the number of Susceptible and Infected individuals in the population (Recovered numbers are uniquely determined since the total population is constant). We also introduce a spontaneous infection of a susceptible individual with constant rate, independent of the number of infected individuals, to eliminate absorbing states.

With a population size of N , states in the 2D space can be represented by $\mathbf{x} = (S, I)$, $S \in \{0, \dots, N\}$, $I \in \{0, \dots, N - S\}$ for a total of $(N + 1)(N + 2)/2$ states. The chemical reactions for this system are:

- **infection** $S + I \xrightarrow{\alpha} 2I$;
- **spontaneous infection** $S \xrightarrow{\beta/5} I$;

- **recovery** $I \xrightarrow{\beta} R$;
- **relapsing** $R \xrightarrow{\beta} S$.

We set the infection rate $\alpha = 0.005$, recovery rate $\beta = 0.01$, and population size $N = S + I + R = 100$, for a total of 5151 states in this SIRS system. Sample trajectories of the system were simulated using the Gillespie algorithm.

3.2 Behaviour through logical properties

We formally specify trajectory behaviours by using temporal logic properties. We are particularly interested in properties that can be verified on single trajectories, and assume metric bounds on the trajectories, so that they are observed only for a finite amount of time. Metric Interval Temporal logic (MITL) (Maler and Nickovic, 2004) as introduced in Section 2.4.1 offers a convenient way to formalise such specifications.

Example 1.2 MITL formulae can be used effectively to obtain behavioural characterisations of the system’s trajectory. We turn again to the SIRS model to illustrate this concept.

Assume one may want to express a global bound on the virulence of the infection, so that the fraction of infected population never exceeds λ . This can be done by considering the formula ϕ_1 , defined as

$$\phi_1 ::= \mathbf{G}_{[0,100]}(I < \lambda N) \quad (3.1)$$

which translates to:

$$\phi_1(\mathbf{x}) = \begin{cases} \mathbf{tt} & \text{if } I_t < \lambda N \ \forall t \in [0, 100], \\ \neg \mathbf{tt} & \text{otherwise.} \end{cases}$$

Statistical model checking of this formula is trivial: one simply draws a trajectory using Gillespie’s algorithm, and monitors that the maximal number of infected does not exceed the specified threshold in the $[0, 100]$ interval.

3.3 High level method description

We first present a high-level description of the proposed methodology; the technical ingredients will be introduced in the following subsections. Figure 3.1 provides an intuitive

roadmap of the approach. The overarching idea is to provide a state-space aggregation algorithm which uses behavioural similarities as an aggregation criterion.

The input to the approach is a CTMC model and a set of MITL formulae ϕ_1, \dots, ϕ_n which define the behavioural traits we are interested in. We formalise some of the key concepts through the following definitions.

Definition 3.3.1. A *coarsening map* \mathcal{C} for a CTMC \mathcal{M} is a surjective map

$$\mathcal{M} : \mathcal{S} \rightarrow \mathcal{R}, \quad (3.2)$$

from the state-space \mathcal{S} of \mathcal{M} to a finite set \mathcal{R} , such that $\text{card}(\mathcal{S}) \geq \text{card}(\mathcal{R})$.

Definition 3.3.2. The *macro-states* of the coarsened system are the elements of the image of the coarsening map \mathcal{C} .

Therefore, the set of all macro-states is a partition of the set of initial states \mathcal{S} , where each element in the partition is a macro-state. In general, there is no way to retrieve the initial state configuration of the system only from information of the macro-state configuration, i.e., the coarsening entails an information loss.

We illustrate the various steps of the proposed procedure in Figure 3.1. The first step is to take a sample of possible initial states; we then evaluate the joint satisfaction of the n formulae, given a particular state as initial condition. This implicitly defines a map

$$\Phi : \mathcal{S} \rightarrow \left\{ \mathbf{r} \in \mathbb{R}^{2^n} : \sum_i r_i = 1, r_i \geq 0 \forall i \right\} \quad (3.3)$$

which associates each initial state with the probability of each possible satisfaction pattern of the n formulae (the standard probability simplex for 2^n possible outcomes). Notice that all of the 2^n possible truth values are needed to ensure correlations between properties are captured. Constructing such a *property map* by exhaustive exploration of the state-space is clearly computationally infeasible; we therefore evaluate it (by SMC) on a subset of possible initial states, and then extend it using a statistical surrogate, a Gaussian Process (Figure 3.1 top).

The property representation contains the full information over the dependence of the properties of interest on the initial state. It can be endowed with an information-theoretic metric by using the Jensen-Shannon divergence (JSD) between the resulting probability distributions. However, the high dimensionality and likely very non-trivial structure of the property representation may make this unwieldy. We therefore propose a dimensionality reduction strategy which maintains approximately the metric structure of the property

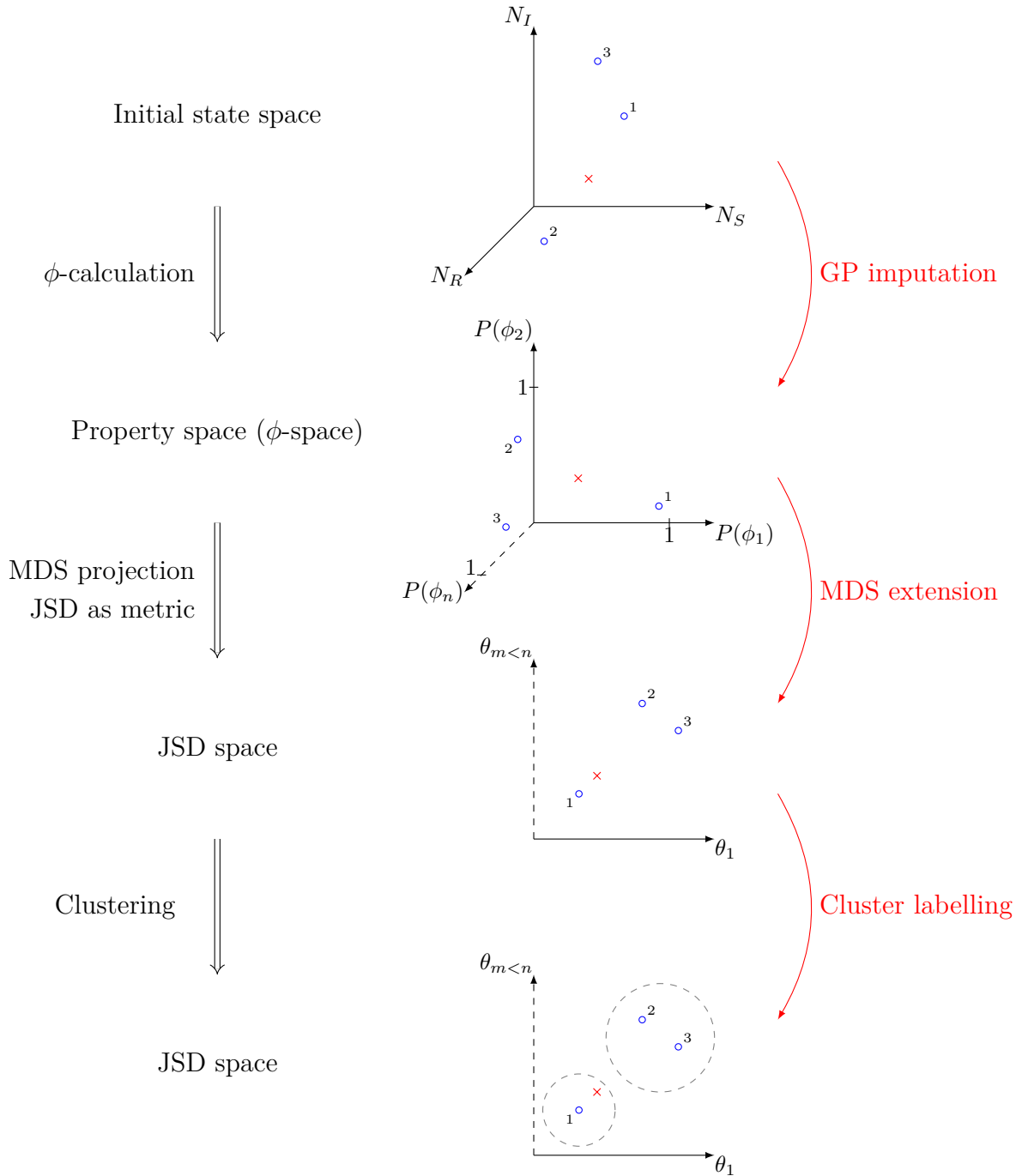


Figure 3.1 The sequence of transformations from space to space are shown in the figure. States from the original state-space (blue circles 1-3) are projected to ϕ -space according to satisfaction rate of set properties (found via simulation of the system). MDS is used to project from ϕ -space to a space where the square root of the Jensen-Shannon divergence (JSD) of ϕ satisfaction probability distributions between states is preserved as Euclidean distance (in the figure, $\text{JSD}[P_\phi(2) \parallel P_\phi(3)] < \text{JSD}[P_\phi(1) \parallel P_\phi(2)]$, $\text{JSD}[P_\phi(1) \parallel P_\phi(3)]$ so states 2, 3 are placed closer together than 1). The states are then clustered to produce macro-states. Out-of-sample states (red cross) can be projected to ϕ -space, using GP imputation to estimate satisfaction probabilities. MDS extension allows projecting from ϕ -space to the JSD space without moving the sampled states. The most likely cluster for the state to belong to (nearest centroid) is the macro-state it belongs to.

representation using Multi-Dimensional Scaling (MDS; Figure 3.1 middle). MDS will also have the advantage of automatically identifying potentially redundant characterisations, as implied for example by logically dependent formulae.

The low-dimensional output of the MDS projection can then be visually inspected for groups of initial states (*macro-states*) with similar behaviours with respect to the properties. This operation is a *coarsening map*, which can also be automated by using a variety of clustering algorithms.

The model dynamics induce, in principle, a dynamics on this reduced space \mathcal{R} . In practice, such dynamics will be non-Markovian and not easily expressible in a compact form; we propose a simple, simulation-based alternative definition which re-uses some of the computation performed in the previous steps to define an empirical, coarse-grained dynamics on the macro-states.

3.4 Satisfaction probability as a function of initial conditions

The starting point for our approach consists of embedding the initial state-space into the property space, ϕ -space. This is achieved by computing satisfaction probabilities for the 2^n possible truth patterns of the n properties we consider. As in general these satisfaction probabilities can only be computed via SMC, this is potentially a tremendous computational bottleneck. To obviate this problem, we turn the computation of the property map into a machine learning problem: we evaluate the 2^n functions on a (sparse) subset of initial states, and predict their values on the remaining initial states using a Gaussian process (GP).

We previously introduced Gaussian processes in Section 2.3.4 as a powerful non-parametric universal approximator for smooth functions. In the present setting, the input-output relationship is the property map from initial states to satisfaction probabilities of the properties. This function is defined over a discrete space, but we can use the population structure of the pCTMC to embed the state-space \mathcal{S} in a (subset) of \mathbb{R}^D for some D . We can then treat the problem as a standard classification problem for 2^n classes over a continuous input space, learning a function $f_\phi: \mathbb{R}^D \rightarrow \{\mathbf{r} \in \mathbb{R}^{2^n} : \sum_i r_i = 1, r_i \geq 0 \forall i\}$.

Remark GPs have previously been used to explore the dependence of the satisfaction probability of a formula on model parameters in the so-called Smoothed Model Checking approach (Bortolussi et al., 2016). There, the authors proved a smoothness result which justified the use of smoothness-inducing GPs for the problem. It is easy to see that such

smoothness does not hold in general for the function f_ϕ ; for example, the probability of satisfying the formula $x(0) > N$ has a discontinuity at $x = N$. However, since we only ever evaluate f_ϕ on a discrete set of points, the lack of smoothness is not an issue, as a continuous function can approximate arbitrarily well a discontinuous function when restricted to a discrete set.

Example 1.3 We exemplify this procedure on the SIRS example. We consider here three properties of interest: the global bound encoded in formula ϕ_1 defined in equation (3.1), and two further properties encoded as

$$\phi_2 ::= \mathbf{F}_{[0,60]}\mathbf{G}_{[0,40]}(0.05N \leq I \leq 0.2N), \quad (3.4)$$

$$\phi_3 ::= \mathbf{F}_{[30,50]}(I > 0.3N). \quad (3.5)$$

Satisfaction of ϕ_2 requires that the infection has remained within 5 to 20% of the total population for 40 consecutive time units, starting anytime in the first 60 time units; satisfaction of ϕ_3 requires that the infection peaks at above 30% between time 30 and time 50.

The property map in this case would have an 8-dimensional co-domain, representing the probability of satisfaction for each of the 2^3 possible truth values of the three formulae. Figure 3.2 plots the probability of satisfaction for the three formulae individually, as we vary the initial state. In this case, 10% of all possible initial states were randomly selected and numerically mapped to the property space via SMC, while the satisfaction probabilities for the remaining 90% were imputed using GPs. We see that throughout most of the state-space the second property has low probability. Also it is of interest to observe the strong anti-correlation between the first and third properties: intuitively, if there is very high probability that the infection will be globally bounded below 40% of individuals, it becomes more difficult to reach a peak at above 30%.

3.5 Dimensionality reduction of behaviours

Once states are mapped onto ϕ -space, reducing dimensionality of this space is useful to remove correlations and redundancies in the properties tracked. Properties may often capture similar behaviour, leading to strong correlations in their satisfaction probability. Reducing the dimensionality of the property space mostly retains the information of how behaviour differs from state to state, eliminating redundancies. Moreover, reduced

dimensional mappings can aid practitioners to visually identify structures within the state-space of the system.

In order to quantify the similarity of different initial states with respect to property satisfaction, the square root of the Jensen–Shannon divergence (root-JSD) between the probability distributions of property satisfaction is used as a metric. JSD is an information theoretic symmetric measure of similarity between probability distributions — the higher the difference between the distributions, the higher JSD is. Between two distributions, P, Q , JSD is defined as

$$\text{JSD}[P \parallel Q] = \frac{1}{2}(\text{KL}[P \parallel M] + \text{KL}[Q \parallel M]),$$

where $M = (P + Q)/2$ the average of the distributions, and $\text{KL}[P \parallel Q] = \sum_i P(i) \log \frac{P(i)}{Q(i)}$, the Kullback-Leibler divergence.

The JSD enables us to derive a matrix of pairwise distances in property space between different initial states. Such a distance is not Euclidean, and is defined in the high-dimensional property space. To map the initial states in a more convenient, low-dimensional space, we employ a dimensionality reduction technique known as Multi-Dimensional Scaling (MDS) (Borg and Groenen, 2005).

MDS has its roots in the social science literature; it is a valuable and widely used tool in psychology and similar fields where data is collected by assessing similarity between pairs.

Given some points $X = (x_1, \dots, x_I)$ in an n -dimensional space, metric MDS finds the position of corresponding points $Z = (z_1, \dots, z_I)$ in an m -dimensional space, where usually $m < n$, such that a given metric is optimally preserved between the points. In the most common case, (also known as Torgerson–Gower scaling or Principal Component Analysis), the metric to be preserved is the Euclidean distance, and is preserved by minimisation of a loss function. The loss function \mathcal{L} (generally called *stress*) is optimised over point locations Z , $z_i \in \mathbb{R}^m$, where

$$\mathcal{L}(z_1, \dots, z_I) = \left[\sum_{ij} \frac{(d(x_i, x_j) - \|z_i - z_j\|)^2}{\sum_{lk} d^2(x_l, x_k)} \right]^{1/2},$$

and $d(z_i, z_j)$ is the dissimilarity measure between points x_i, x_j ; the norm $\|\cdot\|$ denotes the Euclidean norm.

For the classical MDS case, the chosen dissimilarity is the Euclidean distance in the original space. The minimisation can then be achieved by eigenvalue decomposition of a distance matrix of the (normalised) points XX^\top , and subsequently reconstructing the

points from the m largest (eigenvector, eigenvalue) pairs. This results in Z , a projection of the points to an m -dimensional space, where Euclidean distance is optimally preserved. In general and especially for a large number of points, a numerical optimisation algorithm is usually employed to find a solution.

In vanilla MDS, the projection is defined statically for the available data points and needs ab initio re-computation if new points become available. In (Bengio et al., 2004), the method is extended to new points by constructing a new dissimilarity matrix of new points to old ones, by which the projection of new points will be consistent to that of the old points. The kernel for this new matrix achieves this by replacing the means required for centring with expectations over the old points; such that for points $x, y \in X$

$$\tilde{K}(x, y) = -\frac{1}{2} \left(d^2(x, y) - \frac{1}{I} \sum_{x'} d^2(x', y) - \frac{1}{I} \sum_{y'} d^2(x, y') + \frac{1}{I^2} \sum_{x', y'} d^2(x', y') \right),$$

where $\tilde{K}(x, y)$ is the kernel used for the dissimilarity matrix, is replaced by

$$\tilde{K}(a, b) = -\frac{1}{2} \left(d^2(a, b) - E_x[d^2(x, a)] - E_{x'}[d^2(b, x')] + E_{x, x'}[d^2(x, x')] \right),$$

where a can be an out-of-sample point ($a \notin X, b \in X$).

This reconstructs the dissimilarity matrix for the original points exactly, and allows us to generalise to out-of-sample points and find their positions in the embedding learned, as described in Bengio et al. (2004). Extending MDS allows us to create macro-states based on samples of points, and then project new points on the space created by MDS to find in which clusters they belong.

Example 1.4 We have introduced three properties in Equations (3.1), (3.4) and (3.5), and the associated property map. This has an eight-dimensional co-domain, but already some of its properties can be gleaned by the three-dimensional plot of the single-formula probabilities shown in Figure 3.2. Particularly, these reveal strong negative correlations, indicating that MDS may prove fruitful.

Figure 3.3 shows the states projected to a 2D space where proximity implies similar probability distribution over property satisfaction. This was achieved using MDS to project the states, with the square root of the JSD (root-JSD) used as the metric to be preserved as Euclidean distance in the new 2D space. Aspects of the state distribution in ϕ -space (Figure 3.2) are preserved, with the states of high probability satisfaction for property ϕ_2 appearing further from the connected outline (bottom left group in Figure 3.3).

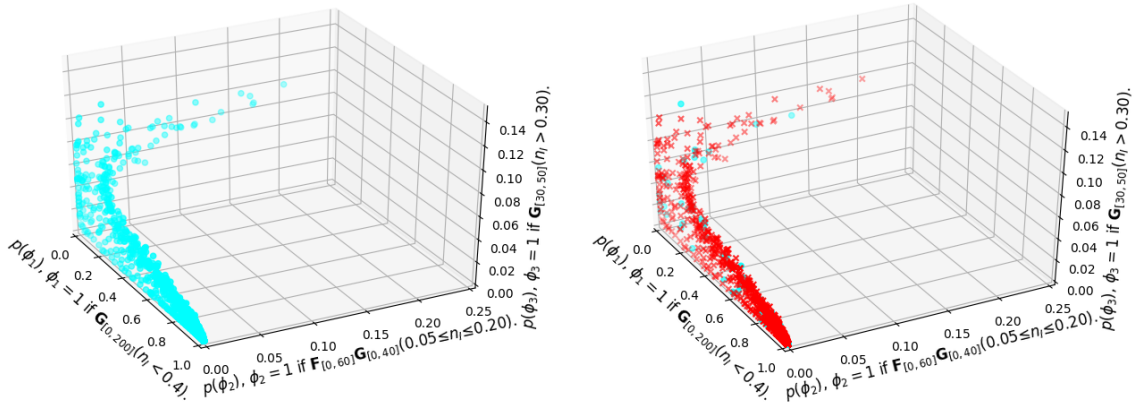


Figure 3.2 Left: Projection of states in ϕ -space via SMC (trajectory simulations for each initial state). Notice the non-trivial state distribution structure. Right: Projection of states in ϕ -space using SMC for 10% of the states, and GP regression to estimate $P(\phi)$ for the rest 90% of states (red crosses).

3.6 Clustering and structure discovery

The MDS projection enables us to visually appreciate the existence of non-trivial structures within the state-space, such as clusters of initial states that produce similar behaviours with respect to the property specification. Our intuition is that such structures should form the basis to define macro-states of the system, groups of states that will exhibit similar satisfaction probabilities for the properties defined. To automate this process, we propose to use a clustering algorithm to define macro-states. Since our goal is to group states with similar behaviours, we adopt k -means clustering (Bishop, 2006a), which is based on the Euclidean distance of the states in the MDS space (representative of the root-JSD between the probability satisfaction distributions). k -means requires specification of the desired number of clusters (the k parameter); this allows the user to select the level of coarsening required. Figure 3.4 shows the clusters produced in the reduced MDS space for the running SIRS model example, where we set the number of clusters $k = 10$.

3.7 Constructing coarse dynamics

Once states have been grouped into macro-states, a major question is how to construct dynamics for the now coarsened system. The coarsened system naturally inherits dynamics from the original (fine-grained) system; however, such dynamics are non-Markovian, and

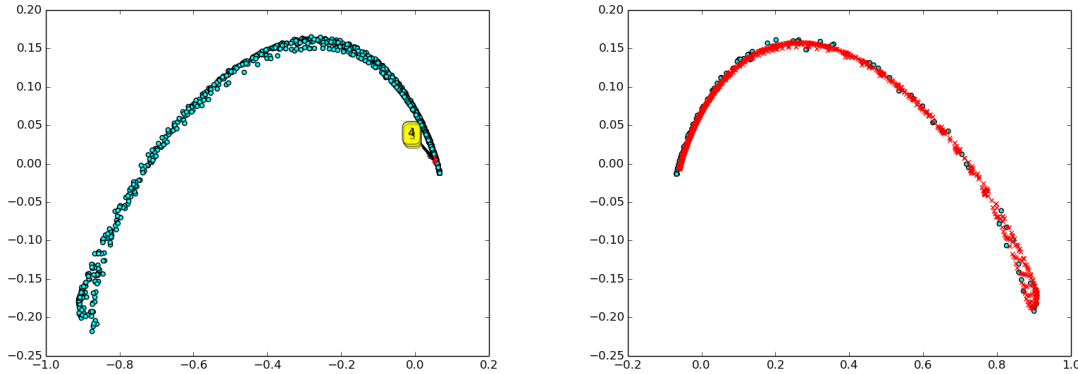


Figure 3.3 Left: $P(\phi_1, \phi_2, \phi_3)$ estimated via SMC for each state. MDS was then used to project them from an 8D to a 2D space. Right: GP estimates of $P(\phi_1, \phi_2, \phi_3)$ for 90% of states (red crosses) produce an almost identical MDS projection.

in general fully history dependent so that transition probabilities would have the form

$$p(k', t | k, h) = p(k' | k, t, h) p(t | k, h), \quad (3.6)$$

where h denotes the history of the process. Simulating such a non-Markovian system is very difficult and likely to be much more computationally expensive than simulating the original system.

We therefore seek to define approximate dynamics which are amenable to efficient simulation, but still capture aspects of the non-Markovian dynamics. The most natural approximation is to replace the system with a semi-Markov system: transitions are still history-independent, but the distribution of sojourn times is non-exponential. We write

$$p(k', t | k, h) \approx p(k' | k, t) p(t | k), \quad (3.7)$$

where we dropped the dependence on h . To evaluate the sojourn-time distribution, we resort to an empirical strategy, and construct a distribution of sojourn times by re-using the simulated trajectories of the fine system that were drawn to define the coarsening. In other words, once a clustering is defined, we retrospectively inspect the trajectories to construct a histogram distribution of sojourn times, approximating $p(t | k)$.

A possible drawback of this semi-Markov approximation is that it may introduce transitions which are actually impossible in the original state-space. This is because states were clustered based on behaviour rather than transition rates, and therefore states that are actually quite far in the original state-space may end up being clustered together.

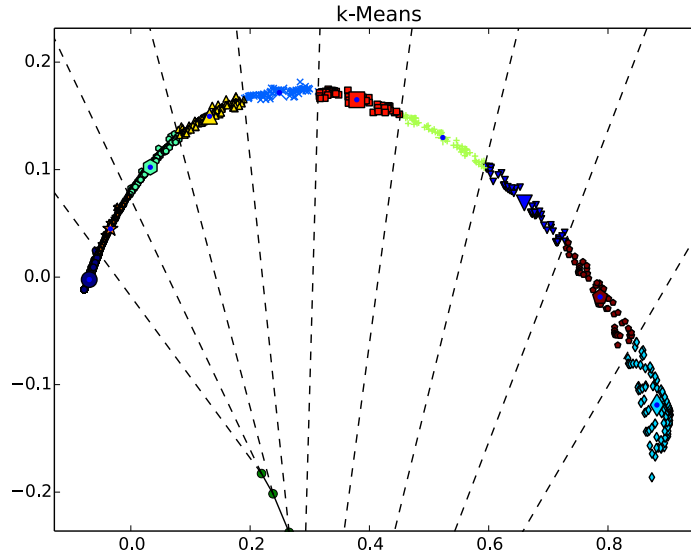


Figure 3.4 The states were clustered in the space created by the MDS projection and coloured accordingly, using k -means (10 clusters). Since the Euclidean distance in this space is representative of distance in probability distributions over properties, states with different behaviour should be in different clusters.

Since the identity of the original states is lost after the coarse graining, impossible transitions may be introduced.

Retrospectively inspecting whole system trajectories, rather than agnostically examining cluster transitions of the original system with a uniform initial state distribution within the cluster, ameliorates this problem. Similarly, estimates of $p(k' | t, k)$ are produced from the same trajectories; these are the macro-state transition frequencies in each bin of the sojourn time probability histogram. This method avoids a lot of impossible trajectories one might generate, if the above probabilities were estimated by sampling randomly from initial states in a macro-state and looking at when the macro-state is exited and to which macro-state the system transitions. Assuming the original system has a steady state, the empirical dynamics constructed here capture this steady state macro-state distribution; however, accuracy of transient dynamics suffers, and the coarsened system enters the steady state faster than the original system.

Steady state consistency We show here how the coarsened dynamics has a steady state consistent with the original dynamics. Conditional probability distributions for the

original CTMC obey the solution to the dCKE:

$$p(X(t+s) = X_j | X(t) = X_i) = [e^{sQ}]_{ij} \quad \forall t,$$

where $X(t)$ is the state of the process at time t , and $[e^{sQ}]_{ij}$ is the ij entry of the matrix exponential of sQ . Note that transition probabilities for elapsed time s are invariant with respect to t , implying stationary dynamics. When we coarsen, the state alphabet $\{X_i \mid i = 1, \dots, N\}$ is surjectively mapped to $\{Y_j \mid j = 1, \dots, M\}$, with $M < N$. The mapped probability distribution at any time t is

$$p(Y(t) = Y_j) = \sum_{i: X_i \mapsto Y_j} p(X(t) = X_i),$$

with $Y(t)$ the macro-state of the coarsened system at time t , and where the sum runs through all states X_i that map to macro-state Y_j .

Exact transition probabilities in the macro-scale are given by

$$\begin{aligned} p(Y(t+s) = Y_l \mid Y(t) = Y_k) &= \sum_{j: X_j \mapsto Y_l} \sum_{i: X_i \mapsto Y_k} p(X(t+s) = X_j \mid X(t) = X_i) \\ &\quad p(X(t) = X_i \mid Y(t) = Y_k) \\ &= \sum_{j: X_j \mapsto Y_l} \sum_{i: X_i \mapsto Y_k} [e^{sQ}]_{ij} \left(\frac{p(X(t) = i)}{\sum_{m: X_m \mapsto Y_k} p(X(t) = X_m)} \right), \end{aligned}$$

which is a sum of all transitions from states of one macro-state to states of another, weighted by the relative probability of being in each state of the macro-state at time t . We observe that the necessary weighting by $p(X(t) = X_i \mid Y(t) = Y_k)$, given by the last term in brackets, introduces a dependence on t in the dynamics so that they are no longer stationary in the macro-scale.

However, consider the limit $t \rightarrow \infty$ when the system is in steady state, $\lim_{t \rightarrow \infty} p(X(t) = X_i) = [\pi_\infty]_i$. Then the transition probability between macro-states is

$$\lim_{t \rightarrow \infty} p(Y(t+s) = Y_l \mid Y(t) = Y_k) = \sum_{j: X_j \mapsto Y_l} \sum_{i: X_i \mapsto Y_k} [e^{sQ}]_{ij} \left(\frac{[\pi_\infty]_i}{\sum_{m: X_m \mapsto Y_k} [\pi_\infty]_m} \right),$$

which is stationary in t . Further, the steady state $\tilde{\pi}_\infty$ over the macro-states, is given by

$$\begin{aligned}
[\tilde{\pi}_\infty]_l &= \lim_{t \rightarrow \infty} p(Y(t+s) = Y_l) = \lim_{t \rightarrow \infty} \sum_k p(Y(t+s) = Y_l | Y(t) = Y_k) p(Y(t) = Y_k) \\
&= \sum_k \sum_{j: X_j \mapsto Y_l} \sum_{i: X_i \mapsto Y_k} [e^{sQ}]_{ij} \left(\frac{[\pi_\infty]_i}{\sum_{m: X_m \mapsto Y_k} [\pi_\infty]_m} \right) \sum_{n: X_n \mapsto Y_k} p(X(t) = X_n) \\
&= \sum_k \sum_{j: X_j \mapsto Y_l} \sum_{i: X_i \mapsto Y_k} [e^{sQ}]_{ij} [\pi_\infty]_i = \sum_{j: X_j \mapsto Y_l} [e^{sQ^\top} \pi_\infty]_j \\
&= \sum_{j: X_j \mapsto Y_l} [\pi_\infty]_j \quad ,
\end{aligned}$$

which is consistent to the steady state of the original CTMC mapped to the macro-states.

Example 1.5 We illustrate and evaluate the quality of the coarsened trajectories with respect to the original ones on the SIRS example. In particular, we examine the probability distribution over the macro-states at different times in the evolution of the system. The macro-state distribution has been estimated empirically by sampling trajectories using the Gillespie algorithm for the fine system, and our coarse simulation scheme for the coarsened system. We have then constructed histograms to capture the distribution of the categorical random variables that represent the macro-state. Finally, we measure the histogram distance between histograms obtained from the fine and the coarse systems. Figure 3.5 depicts the evolution of the macro-state histograms over time.

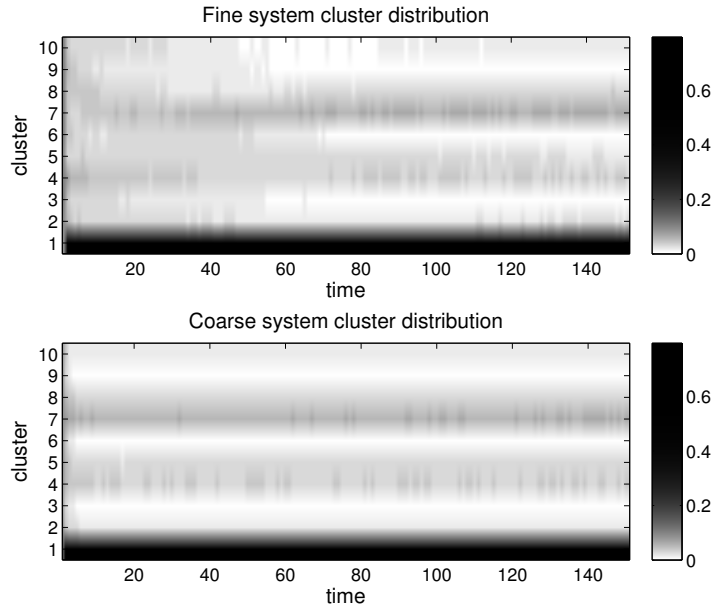


Figure 3.5 Evolution of the macro-state histograms over time. Credit to Dimitrios Milios.

Quality of approximation In order to put any distance between empirical distributions into context, this has to be compared with the corresponding average self-distance, which is the expected distance value when we compare two samples from the same distribution. In this work, we estimate the self-distance using the result in [Cao and Petzold \(2006\)](#): given N samples and K bins in the histogram, an upper bound for the average histogram self-distance is given by $\sqrt{(4K)/(\pi N)}$. In our example, we have $K = 10$ histogram bins, which are as many as the macro-states. In practice, a distance value smaller than the self-distance implies that the distributions compared are virtually identical for a given number of samples. In [Figure 3.6](#), we see the estimated distances for $N = 10000$ simulation runs for times $t \in [0, 150]$. It can be seen that the steady-state behaviour of the system is captured accurately, as the majority of the distances recorded after time $t = 60$ lie below the self-distance threshold. However, the transient behaviour of the system is not captured as accurately. Upon a more careful inspection of the shape of the histograms in [Figure 3.5](#), we see that the coarsened system simply converges more quickly to steady-state.

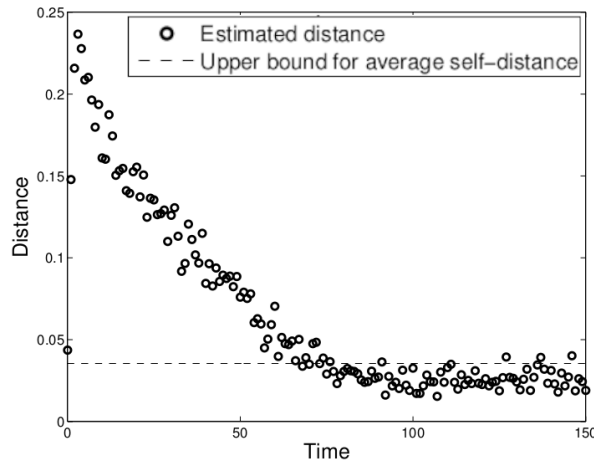


Figure 3.6 Evolution of the macro-state histogram distances over time. Credit to Dimitrios Milios.

Computational savings State-space coarsening results in a more efficient simulation process, since the coarse system is characterised by lower complexity as opposed to the fine system. We demonstrate these computational savings empirically in terms of the average number of state transitions invoked during simulation. More specifically, we consider a sample of 5000 trajectories of the fine and the coarse system. We have recorded 320 ± 25 initial state transitions on average in each trajectory of the fine system, compared to 56 ± 31 macro-state transitions in trajectories of the coarsened system. The

number of transitions in the coarse system is an order of magnitude lower than in the fine one, owing to the reduction of states in the system from a total of 5151 to 10 (the number of macro-states). Clearly, our procedure, particularly the GP imputation, incurs some computational overheads. Table 3.1 presents the computational savings of using GPs to estimate satisfaction probability distributions for most states, instead of exhaustively exploring the state-space. All simulations were performed using a Gillespie algorithm implementation, taking 1000 trajectories starting at each examined state, running on 10 cores.

Table 3.1 Real running times for simulations of varying sample size (percentage of state-space) and GP estimation of remaining states.

| Sample size | GP & MDS time (s) | Simulation time (s) | Total time (s) | Percentage of exhaustive total time (Total time/8516s) |
|-------------|-------------------|---------------------|----------------|--|
| 100% | 1616* | 6900 | 8516 | 100% |
| 50% | 1133 | 3450 | 4583 | 54% |
| 40% | 884 | 2760 | 3644 | 43% |
| 30% | 595 | 2070 | 2665 | 31% |
| 20% | 354 | 1380 | 1734 | 20% |
| 10% | 170 | 690 | 860 | 10% |

* No GP was performed here, just the MDS.

3.8 Discussion

We presented a novel approach to the coarsening of a CTMC, in order to gain a stochastic process with a much smaller state-space. Unlike previous approaches to CTMC aggregation, which are based on structural properties of the state-space, our approach is based on property satisfaction, allowing the coarse-grained system to focus on abstracting the dynamics in terms of aspects of behaviour that are important in the modelling study. The further steps are to identify key clusters of states in property space, or a lower-dimensional representation of it, and approximate the transition dynamics between them. For example, this approach might be used within multi-scale modelling to reduce the state-space of a lower level model before embedding in a higher-level representation (see Chapter 4).

Common aggregation techniques, such as (approximate) lumpability, often impose stringent conditions on the symmetries and transition rates within the original state-space. Moreover, the macro-states produced can be difficult to interpret when the reduction is

applied directly at the state-space level (i.e. without a corresponding bisimulation over transition labels). In contrast, the property-based approach allows macro-states to be defined by high-level behaviour, rather than them emerging from an algorithm applied to low-level structure.

The GP regression we employed for estimating satisfaction probability of properties for out-of-sample states proved quite accurate; simulation estimates for 10% of the states were sufficient to reconstruct the state distribution in the space defined by the probability of property satisfaction, ϕ -space, without substantial loss of structure. Therefore, the proposed approach may be helpful in effectively understanding the behavioural structure of large and complex Markovian systems, with implications for design and verification.

Initial experiments on a simple system show that our approach can be practically deployed, with considerable computational savings. The approach induces coarsened dynamics which empirically match the original system's dynamics in terms of steady-state behaviour. However, the recovery of transient coarse-grained dynamics poses more of a challenge and this will provide a focus for future work. In particular, we will seek to explore the possibility of quantifying the information lost through the coarsening approach, at least asymptotically, for systems which admit a steady state. Exploring the scalability of the approach on more complex, higher dimensional examples will also be an important priority. In general, we expect our approach to be beneficial when simulation costs dominate the overheads incurred by the GP regression approach. This condition will be mostly met for systems with moderately large state spaces but complex (e.g. stiff) dynamics. For extremely large state spaces, the cubic complexity (in the number of retained states) of GP regression may force users to adopt excessively sparse sub-sampling schemes, and it may be preferable to replace the GP regression step with alternative schemes with better scalability. Exploration of these computational trade-offs would likely prove insightful for the methodology.

Chapter 4

Statistical abstraction for multi-scale spatio-temporal systems

Science is often tasked with examining natural or artificial systems characterised by spatial dependence and complex dynamics. The complexities that these characteristics induce on the emergent system behaviour mean that detailed models are often constructed in order to study them through simulation. This approach has been used extensively in applications ranging from cyber-physical systems to collective adaptive systems of human behaviour and to cellular systems. Nevertheless there is still room for advancement through automating the ability to recover simpler models that still capture the dynamics with sufficient faithfulness, but which may have a lower computation cost. This is especially true for systems involving onerous stochastic simulations ([Dada and Mendes, 2011](#); [Gilbert et al., 2015](#)).

We consider here a general framework which encompasses a large class of spatio-temporal systems. In this framework, multiple identical agents are distributed in space over an external field. The agents perceive the field locally and perform internal stochastic computations to determine their subsequent behaviour, such that their actions are influenced by their environment. We also allow the agents to act locally upon the external field, enabling the latter to become a medium for signals between agents. This framework subsumes a wide range of systems, from swarm robots performing a task in space, to bacteria exploring a nutrient field, or agents responding to distress signals.

In this chapter, we propose a method to replace expensive stochastic parts of the model with input-output maps estimated via a machine learning procedure. We focus on a particular macro-scale behaviour as output from the model, and devise a statistical abstraction of the system in order to produce a simpler system which preserves the macro-scale behaviour. Crucially, we do not care for the detailed internal state of the model, but

only an abstracted version sufficient to capture its qualitative behaviour. The abstracted state is formalised as the satisfaction output of a set of logical properties evaluated on the original state. We estimate the necessary input-output relation by learning a parameters-to-behaviours regression map using Gaussian Process (GP) regression. Our work is motivated by earlier work on using GPs to learn effective characterisations of system behaviour (Bortolussi et al., 2015b, 2016; Michaelides et al., 2016).

This work was published in the span of two papers: an initial publication (Michaelides et al., 2017) appeared in *Proceedings of Quantitative Evaluation of Systems (QEST) 2017*, followed by an invited extension to appear in *Transactions on Modeling and Computer Simulation (TOMACS)*.¹ The extended version as presented here adapts the statistical framework to also handle agent-environment interactions, thereby closing the information loop and allowing for environment-mediated agent-agent interactions. To illustrate our framework we construct abstractions for two biological systems which exhibit chemotaxis in the macro-scale: the bacterium *Escherichia coli* and the social amoeba *Dictyostelium discoideum*. The former follows positive chemical concentrations in search of nutrient-rich environments, and the latter responds to signals emitted by other amoebae to aggregate into clusters; both of which are forms of chemotaxis.

The rest of the chapter is organised as follows: we start with some background on spatio-temporal systems (Section 4.1). The general framework for our statistical abstraction methodology is presented in Section 4.2, followed by a brief discussion of related work in Section 4.3. We then present two case studies describing applications of the abstraction on a model of *E. coli* chemotaxis and a model of *D. discoideum* aggregation (Sections 4.4 and 4.5 respectively), which exemplify the methodology and provide results assessing the quality and efficiency of the abstraction. We conclude with a discussion on the utility of the method and closing remarks about prospective expansion of the work (Section 4.6).

4.1 Background

4.1.1 Spatio-temporal agent models

We start by defining the class of spatio-temporal agent models we will consider in this paper. Let \mathcal{D} be a spatial domain (usually a compact subset of \mathbb{R}^n with $n = 2, 3$), and let $[0, T]$ be the temporal interval of interest. We define the *spatio-temporal field*

¹Jane Hillston and Guido Sanguinetti provided feedback and advice in the development of the material, and edited the manuscript.

$f: \mathcal{D} \times [0, T] \rightarrow \mathbb{R}$ to be a real-valued function defined on the spatial and temporal domains of interest. A spatio-temporal agent model is a triple $(\mathcal{D}, f, \mathcal{A})$ where \mathcal{A} is a collection of point *agents* whose location follows a stochastic process which depends on the spatio-temporal field. Note that even though we realise that this is not the most general case, as agents may be spatially extended, or directly interact with each other, a form of agent-agent interaction is feasible within this framework. As illustrated through the case study of a *D. discoideum* model in Section 4.5, the agents may affect the evolution of the spatio-temporal field — this allows the field to transmit signals from agent to agent, enabling interaction.

4.1.2 Multi-scale models

In many practical situations, one is interested in modelling not only the movement of the agents, but also the mechanism through which sensing and decision making is carried out within each agent. This naturally leads to structured models with distinct layers of organisation, with behaviour in each layer informing the simulation that takes place at the layer above or below. We will assume that the internal workings of the agent are also stochastic, and we model them here as a Markov chain with a discrete state-space.

In the first case-study presented (Section 4.4), the internal workings of an agent are modelled by a *population Continuous Time Markov Chain* (pCTMC). Note that the pCTMC is the internal model for a *single* agent here, not for multiple agents. In the second case-study (Section 4.5), the internal workings of a cell are modelled by a Discrete Time Markov Chain (DTMC).

4.1.3 Simulating multi-scale systems

Multi-scale spatio-temporal systems are in general amenable to analytical techniques only in the simplest of cases. For the vast majority of real-world models, simulation-based analysis is the only option to gain behavioural insights.

Simulation of spatio-temporal systems typically employs nested algorithms: having chosen a time discretisation for the spatial motion (which is assumed to have the slower time-scale), a spatial step is taken. Then, the value of the external field is updated, and the internal model is run for the duration of a given time-step with the new rates (corresponding to the updated value of the external field). A sample from the resulting state distribution then determines the velocity of the agent for the next time-step.

Clearly, this iterative procedure, while asymptotically exact (in the limit of small time discretisation), is computationally very demanding. This has motivated several

lines of research in recent years (Bortolussi et al., 2015b; Goutsias, 2005; Haseltine and Rawlings, 2002; Rao and Arkin, 2003).

4.2 Methodology for statistical abstraction

In a multi-scale system, output from a set of processes in one layer in the system is passed as input to another layer; these processes are often computationally expensive. We present a methodology to abstract away such a set of processes and replace them with a more efficient stochastic map from the input to the output, governed by an underlying probability function. We approximate this probability function using Gaussian processes after observing many input-output pairs from the processes to be abstracted. The output consists of truth evaluations of properties expressed in logical formulae, which capture some behaviour of the system that is to be preserved by the abstraction.

4.2.1 Statistical abstraction framework

Consider a Markov chain S , which given an initial state \mathbf{s}_0 , running time Δt , and input $\mathbf{q} \in \mathbb{R}^D$ which completely determines transition rates, generates a trajectory $\mathbf{s}_{[0,\Delta t]}$. At each time step n in the simulation of a multi-scale system, the trajectory $\mathbf{s}_{[0,\Delta t]}^{(n)}$ is checked for satisfaction of a logical property resulting in output $y^{(n)} = f(\mathbf{s}_{[0,\Delta t]}^{(n)})$, $y^{(n)} \in \{\top, \perp\}$. For the next time step, the last state in the Markov chain is kept as the new initial state (i.e. $\mathbf{s}_0^{(n)} = \mathbf{s}_{\Delta t}^{(n-1)}$), and with new input $\mathbf{q}^{(n)} \in \mathbb{R}^D$ to determine transition rates the process is repeated. This layer of the multi-scale system can therefore be described as a set of operations at each time step n :

$$S(\mathbf{s}_0^{(n)} = \mathbf{s}_{\Delta t}^{(n-1)}, \Delta t, \mathbf{q}^{(n)}) = \mathbf{s}_{[0,\Delta t]}^{(n)}; \quad (4.1)$$

$$f(\mathbf{s}_{[0,\Delta t]}^{(n)}) = y^{(n)}. \quad (4.2)$$

Note that we consider a single property here for simplicity (so a single binary value), but one could generalise to multiple properties, and hence, to a multi-valued output. This output then becomes input to a higher layer in the multi-scale system.

Our goal is to construct a system \tilde{S} that is cheaper to simulate, whose output will be consistent with the original system S . Since the system is stochastic, *consistent* refers to having the same probability distribution for the output random variable $y^{(n)}$ given the same input $\mathbf{q}^{(n)}$ and following previous output $y^{(n-1)}$. Fundamentally, we seek to approximate the (generally) non-Markov process of outputs $\{y^{(n)}\}$ with a Markov one.

To describe the abstracted system, we write:

$$\tilde{S}(y^{(n-1)}, \mathbf{q}^{(n)}) = y^{(n)}. \quad (4.3)$$

Replacing the initial state $\mathbf{s}_0^{(n)} = \mathbf{s}_{\Delta t}^{(n-1)}$ input with the previous output $y^{(n-1)}$ allows us to substitute the whole layer of fine operations (4.1, 4.2) with the cheaper abstracted system \tilde{S} (4.3), unburdening the multi-scale system. We regard this abstracted system to be a stochastic map from the internal state of the system, now abstracted to the last output $y^{(n-1)}$, and some external input $\mathbf{q}^{(n)}$, to a new output $y^{(n)}$. The latter being a discrete random variable, the task is to estimate a probability distribution over the output domain from which to sample the output. Since we expect this distribution to depend upon the previous output $y^{(n-1)}$ and external input $\mathbf{q}^{(n)}$, we use Gaussian process regression with an appropriate observation likelihood to estimate an underlying probability function $\Psi(y, \mathbf{q})$ which governs the output of \tilde{S} .

It follows that the abstraction will become more accurate the faster the Markov chain S mixes, since dependence on the initial state of the chain will no longer matter — in fact, for fast enough mixing times relative to Δt , one could even drop the output feedback and produce the stochastic output y only given the input \mathbf{q} . Thus, we expect the abstraction to work particularly well for components of a system which equilibrate faster than others. For the cases we present below, notice that the internal agent dynamics which determine motility are faster than the changes in the environmental input that affect them.

In our general construction, the output of the system is taken to be a combination of boolean satisfaction values for a set of properties. Owing to its discrete nature, the resulting abstraction could be interpreted as a discrete-time Markov chain (DTMC) whose state-space is comprised of every output combination. Our task is then to determine transition rates for this DTMC to make its paths consistent with output of the original system. If one wishes to increase accuracy, the DTMC can be made to be of a higher order. A higher-order DTMC means that a longer output history is retained and affects the next output, and can therefore be expected to better approximate the original output dynamics. In the two examples presented here, we construct a first-order DTMC for the first case and a second-order DTMC for the second case.

4.2.2 Approximating the underlying probability function

There are many approaches one could take to infer the probability function $\Psi(y, \mathbf{q})$, necessary for the abstraction. Here we make use of Gaussian processes (GPs), a non-parametric regression method introduced in Section 2.3.4.

GPs are universal function approximators. The choice of covariance kernel determines the prior over the space of functions considered, and thus affects how many observations are required to get a good estimate of the underlying function.² However, given enough observations, a GP with an appropriate kernel will approximate any function within a particular family arbitrarily well. Here we make use of the squared exponential, or Gaussian, kernel

$$k(x, x') = \sigma^2 \exp \left[-\frac{1}{2} (x - x')^\top M (x - x') \right],$$

where σ is a scalar *amplitude* hyperparameter which indicates the magnitude of variation in the function, and $M = \text{diag}(\ell)^{-2}$ is a diagonal matrix which scales each input dimension by a *characteristic length-scale*, indicative of how correlated the output is along that dimension. Intuitively, functions that exhibit more frequent variations along a dimension i are more probable when ℓ_i is smaller, and functions with larger amplitudes of variation are more probable when σ^2 is larger. We refer to (Rasmussen and Williams, 2006) for a comprehensive account of GPs.

Since training observations are binary samples of a Bernoulli distribution (satisfaction *true* or *false* of logical properties) or samples of a categorical distribution (in the case of multiple properties), but GPs regress over a continuous unbounded variable, some adjustments to the standard GP regression must be made for correct evaluation of the underlying probability function Ψ . GP regression with its many variations for different problem tasks is well described in (Rasmussen and Williams, 2006). The necessary adjustments which we adopt here are found in the Gaussian process classification (GPC) section of the book, and essentially amount to identifying that the class probability function is Ψ , where the class is the property satisfaction outcome. A detailed explanation can be found in Appendix A.1.

As discussed, GP regression is a statistical approach to approximate an unknown function based on a finite set of input-output instantiations. The quality of this approximation is affected by a number of factors, including model choices such as the covariance function, but naturally the prime determinant of approximation quality is the amount of data available. In this application, since the data is generated by model simulations, we have a degree of control on how much data is available. In practice, however, it is extremely difficult to estimate *a priori* the size of the data-set required for a certain

²If an oracle allowed observation of the function at any point in the input domain, we would be able to actively reduce our uncertainty over it as desired (i.e. we could take observations where we deem lower uncertainty necessary). In the application cases presented here the system is observed through entire simulations, and so we do not have this power.

accuracy; in the case study in Section 4.4 we present a practical empirical strategy to address this problem.

4.3 Related work

When it comes to *abstracting* stochastic systems, there is a wide literature of methods to consider. For the cases where the system is solely defined in terms of a population continuous-time Markov chain (pCTMC) found normally at large counts, the chemical Langevin equation provides an approximation for the whole process, while systematic approximation methods for the moments of the distribution of the process existed since the 60s (Gillespie, 2000; Kurtz, 1971; van Kampen, 1961). Both approaches have seen considerable improvement over the last decades, increasing their range of applicability (Schnoerr et al., 2017b). Other approaches to the problem of efficiently solving biochemical systems attempt to graduate species' concentration levels to discrete intervals (Ciocchetta and Hillston, 2009; Palaniappan et al., 2017), thereby reducing the state-space of the underlying CTMC to be solved, or employ time-scale separation if possible (Bortolussi et al., 2015b; Goutsias, 2005; Haseltine and Rawlings, 2002; Rao and Arkin, 2003).

All of the above are firmly situated in the domain of pCTMCs and are agnostic to the demands made of the process downstream — whether the pCTMC is checked for reaching a particular value, or having maintained a value for a particular duration in some time interval, does not affect the approximation these methods will yield. We take a more holistic view in this work and consider the system to be abstracted as a component of a larger multi-scale system. As such, only a particular aspect of the component is relevant to the multi-scale system, and it is this aspect which our abstraction attempts to preserve.

We take the relevant component output to be the evaluation of a logical property on a stochastic trajectory drawn from a Markov chain, which comprises the internal process of the component to be abstracted. The transition rates of the chain depend on some input the component receives, and this enables us to utilise results in (Bortolussi et al., 2016) to estimate the output given the input via the method of Gaussian process regression.

4.4 The case of *E. coli* chemotaxis

Foraging is a central problem for microbial populations. The bacterium *Escherichia coli* will normally perform a random walk within a spatial domain where nutrient concentration

is constant (e.g. a Petri dish). When presented with a spatially varying nutrient field, a phenomenon known as *chemotaxis* arises. As the bacterium performs a random walk in the nutrient field encountering changing nutrient levels, its sensory pathway effectively evaluates a temporal gradient of the nutrients (or ligands) it experiences; the walk is biased so that the bacterium experiences a positive temporal gradient more often than not (Berg et al., 1972; Sourjik and Wingreen, 2012; Vladimirov et al., 2008). Since the bacterium is moving in the field, the temporal gradient is implicitly translated into a spatial one, so the bacterium drifts toward advantageous concentrations. Implicitly translating a temporal gradient to a spatial one through motion is necessary for the bacterium cell, because its body size is too small to allow for effective calculation of the spatial gradient of a chemical field at its location. As a result, we can safely regard the bacteria as point-like agents.

Motor control in *E. coli* An *E. coli* cell achieves motility by operating *multiple* flagellum/motor pairs (F/M), which can either drive it straight (subject to small Brownian perturbation), or rotate it in place. Thus, the cell can either be ‘tumbling’ (re-orienting itself while stationary) or ‘running’ (propelling itself forward while maintaining direction) at any time (Figure 4.1: left, centre). The motility state, RUN/TUMBLE, of the cell is determined by the number of flagella found in particular conformations. The model in (Sneddon et al., 2012) suggests three possible conformations for a flagellum: *curly* (C), *semicoiled* (S) and *normal* (N). The associated motor is modelled as a stochastic bistable system, which rotates either clockwise (CW) or counter-clockwise (CCW). Changes in motor rotation induce conformational changes on the associated flagellum. Transition rates between motor states are given by rate parameters k_+ and k_- for transitions $CW \rightarrow CCW$ and $CCW \rightarrow CW$, respectively. The possible transitions between flagellum/motor states are summarised in the schematic diagram in Figure 4.1 (right). *E. coli* normally has of the order of ten flagella and associated motors; the dynamics of the pair flagellum/motor population therefore lends itself to be easily described as a pCTMC. The k_{\pm} transition rates depend on the temporal gradient evaluated by the chemotaxis pathway, and represent the functional interface of the bacterium with its external environment.

The classical mathematical model for the sensory response of the cell to external ligand concentration changes is provided by the Monod-Wyman-Changeux (MWC) model (Hansen et al., 2008; Sneddon et al., 2012; Sourjik and Berg, 2004). The model considers sensor clusters which signal information about ligand concentration changes to the motors, by triggering a biochemical response in the cell (phosphorylation of the CheY protein which binds to the motors) affecting the switching rates of rotation direction, k_{\pm} .

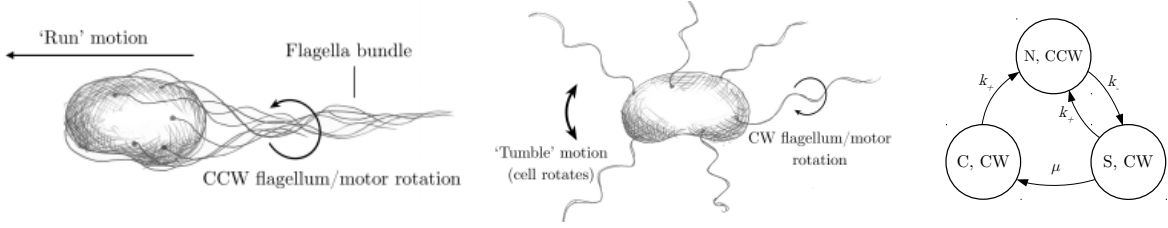


Figure 4.1 The two motility modes of an *E. coli* cell. Left: the F/M are in CCW conformations, forming a helical bundle and propelling the cell. Centre: the F/M are in CW conformations, breaking the bundle apart and causing the cell to re-orient in place. Right: CTMC for a single F/M, with three conformation states and transition rates $k_{\pm}(m, L)$ and fixed $\mu = 5\text{s}^{-1}$.

The full MWC model is still highly complex; in practice, we follow (Sneddon et al., 2012) and adopt a simplified model of sensory response to describe the dependency of motor rates k_{\pm} on ligand concentrations. This involves resolving the CheY signalling pathway to the single variable m , which represents the methylation state of the ligand receptors and whose stochastic evolution is dependent on the ligand concentration L . Since m depends on past L concentrations the cell has been in, one may think of it as a *chemical memory* of sorts which encodes the value of L at previous times. The time comparison window is determined by how fast methylation happens — faster methylation leads to a shorter memory.

Sneddon et al. (2012) then resolve the entire dependency chain of the chemotaxis pathway to Equations (4.4) and (4.5). The motor switching rates $k_{\pm}(m, L)$ are given by the deterministic equation

$$k_{\pm} = \omega \cdot \exp \left\{ \pm \left[\frac{g_0}{4} - \frac{g_1}{2} \left(\frac{Y_p(m, L)}{Y_p(m, L) + K_D} \right) \right] \right\}, \quad (4.4)$$

where

$$Y_p(m, L) = \alpha \cdot \left[1 + e^{\epsilon_0 + \epsilon_1 m} \cdot \left(\frac{1 + L/K_{\text{TAR}}^{\text{off}}}{1 + L/K_{\text{TAR}}^{\text{on}}} \right)^{n_{\text{TAR}}} \cdot \left(\frac{1 + L/K_{\text{TSR}}^{\text{off}}}{1 + L/K_{\text{TAR}}^{\text{on}}} \right)^{n_{\text{TSR}}} \right]^{-1}.$$

The methylation process can be naturally modelled as a birth / death process with rates depending on ligand concentration; again following (Sneddon et al., 2012) we take a fluid approximation of this, yielding the Ornstein-Uhlenbeck (OU) process:

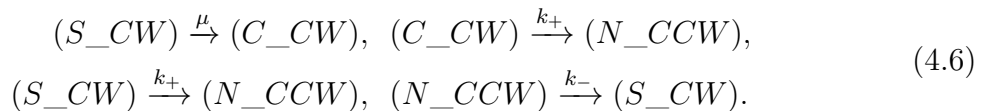
$$\frac{dm}{dt} = -\frac{1}{\tau}(m - m_0(L)) + \eta_m(t). \quad (4.5)$$

In the above stochastic differential equation (SDE), $\eta_m = \sigma_m \sqrt{2/\tau} \Gamma(t)$, $\Gamma(t)$ is the normally distributed random process with 0 mean and unit variance, σ_m is the standard deviation of fluctuations in the methylation level, and $m_0(L)$ is an empirically derived function whose output is the methylation level required for full adaptation at the current external ligand concentration L . The adaptation rate τ , determines how fast methylation occurs and so, how long the ‘chemical memory’ of previous L values is in the system. The constants τ , along with mb_0 and α involved in the $m_0(L)$ function (see (Sneddon et al., 2012)), fully parametrise the methylation evolution. See (Vladimirov et al., 2010) for reported values of constants used in Equation (4.4) and (Frankel et al., 2014; Sneddon et al., 2012) for a detailed derivation of the results. Equations (4.4) and (4.5) couple the transition rates of the pCTMC in Figure 4.1: Right, with the external ligand concentrations, and therefore fully describe the internal model of the *E. coli* chemotactic response.

4.4.1 Simulating chemotaxis in *E. coli*

Simulations of the *E. coli* model outlined proceed along the general lines discussed above. Given a value of the ligand field and a characteristic time-step Δt , we draw samples of the SDE (4.5) using the Euler-Maruyama method, a standard method for simulating SDEs.

In the F/M pCTMC system and following the reaction equation style, each species represents a different F/M conformation for a total of three species ($(N_C W)$, $(S_C W)$, $(C_C W)$) and ten agents (10 F/M pairs). States of the pCTMC count how many F/M (agents) there are of each conformation (species). As visualised in Figure 4.1 (right), the following transitions (reactions) occur:



Note that in the above rate transitions there are dependencies on both external (L) and internal (m) states: $k_{\pm}(m, L)$, where L is an external input to the system (the external chemoattractant concentration at the time) and m is the current methylation level (sampled from the OU process in Equation 4.5 every Δt). Instead, the rate transition for $(S_CW) \rightarrow (C_CW)$ is fixed, $\mu = 5\text{s}^{-1}$.

Using the exact Gillespie algorithm (Gillespie, 1977), we then simulate the internal pCTMC for a length of time Δt to draw a sample configuration of the flagella/motors system. Formally, trajectories of length Δt are checked against a property specifying the

motility state for the cell (RUN/TUMBLE),

$$\phi_{\text{RUN}}(\mathbf{s}) = (N \geq 2) \wedge (S = 0), \quad (4.7)$$

where $\mathbf{s} = (S, C, N)$ is the last state of the flagellum/motor pairs in the CTMC trajectory (each element of the vector counts how many F/M pairs there are in each conformation). The logical property above evaluates to *true* (1) if a given cell state \mathbf{s} has more than 2 F/M in the *normal* conformation and none in the *semicoiled* conformation; otherwise, it evaluates to *false* (0).

The spatial location of the bacterium is then updated according to a simple rule: if the sampled internal state corresponds to RUN, the agent moves rectilinearly and updates its position $\vec{r} \leftarrow \vec{r} + \vec{v} \cdot \Delta t$, where $v = 20\mu\text{m/s}$, the speed of the bacterium. Otherwise, if the internal state corresponds to TUMBLE, the agent remains still and its velocity is updated $\vec{v} \leftarrow R(\theta) \cdot \vec{v}$, where $R(\theta)$ is the standard 2D unitary rotation matrix through an angle θ , and θ is a tumbling angle sampled from a Gamma distribution as reported in [Sneddon et al. \(2012\)](#).

The above simulation scheme, outlined in Algorithm 1 (Appendix A.2), produces a chemotactic response to a ligand gradient. It takes $\sim 270\text{s}$ to simulate a single cell trajectory of $t_{\text{end}} = 500\text{s}$ with a time-step $\Delta t = 0.05$.

4.4.2 Abstracting the *E. coli* chemotaxis pathway

Following the abstraction framework put forth in Section 4.2.1, we associate the original system S with the pCTMC system of F/M conformations (Equations 4.6), along with the OU methylation process in Equation 4.5. The input starting state \mathbf{s}_0 is the last F/M state of the pCTMC, and the last methylation level m . The simulation time T is the variable Δt introduced earlier, which is also used for the integration step-size of the OU in the Euler-Maruyama scheme. The transition rates k_{\pm} are calculated using the variables m and L , the last methylation level and external ligand concentration at the position of the cell, respectively. The output of this system, \mathbf{s}_t , is then a sampled pCTMC trajectory and new methylation level. Finally, the run property (4.7) is evaluated on (the last state of) the drawn pCTMC trajectory and the output determines whether the cell ‘runs’ or ‘tumbles’.

In observing the truth value of property ϕ_{RUN} for the state of the pCTMC at regular intervals of Δt , we cast the original pCTMC model (S) into a DTMC (Figure 4.2). This DTMC has only two states, $\phi_{\text{RUN}} \in \{\top, \perp\}$, and transition probabilities depending on the transition rates k_{\pm} , μ , of the original pCTMC.

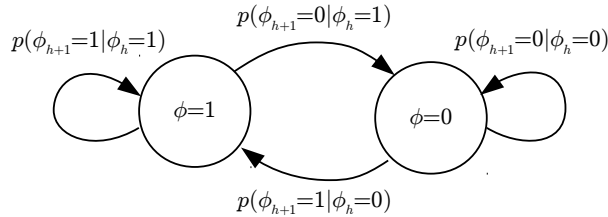


Figure 4.2 DTMC with two states, $\phi_{\text{RUN}} \in \{\top, \perp\}$. The transition probabilities depend on internal methylation level m and external ligand concentration L .

Since this is only a two-state DTMC, the state at the next time-step conditioned on the current one can be modelled as a Bernoulli random variable:

$$\phi' \mid \phi \sim \text{Bernoulli}(p = p_{\phi'=1|\phi}(m, L)), \quad (4.8)$$

where ϕ , ϕ' are the ϕ_{RUN} DTMC states at time-steps h , $h + 1$ respectively. Also, the boolean $\{\perp, \top\}$ truth values of the properties have been mapped to the standard corresponding integers $\{0, 1\}$ for mathematical ease.

We recognise that a *single step* transition of this DTMC ($\phi' \mid \phi, m, L$) is the output $y' \mid y, \mathbf{q}$ produced by the abstracted layer $\tilde{S}(y, \mathbf{q})$. Identifying the corresponding probability function $p_{\phi'=1|\phi}(m, L)$ as the underlying governing function $\Psi(y, \mathbf{q})$ completes the setting of *E. coli* chemotaxis model abstraction to the methodology framework given above (Section 4.2.1). Note that the OU process for methylation is retained in the abstracted model as a parallel running process in the same layer of the multi-scale system. The OU process output m , together with the ligand concentration L (output of a different layer in the multi-scale system), constitute the input \mathbf{q} . The altered simulation scheme for this abstracted model is outlined in Algorithm 2 (Appendix A.2). Notice how Steps 5, 6 there replace the more expensive Steps 22, 23 in Algorithm 1 (Appendix A.2).

Philosophical remark One may observe that our method requires choosing which parts of the model to abstract using our framework, and this is at the modeller's discretion. In this case, for instance, asking of the method to abstract a large pCTMC modelling the methylation process might be feasible, but redundant, as we already know of a very efficient abstraction for it: the Langevin SDE for the OU process. It is therefore beneficial and desirable to aid the method where possible because we have particular insight. This agency reflects our focus on inquiring whether a particular interpretation of an accurate micro-scale model may provide a useful mechanism for observed macro-scale behaviour,

especially in areas where domain knowledge is lacking. In this sense, the nature of the attempted abstraction puts different questions to the model.

Constructing Ψ in *E. coli* chemotaxis A central part of our abstraction methodology is estimating an underlying probability function Ψ , which is used to produce a stochastic output. In our *E. coli* example model, a single DTMC transition ($\phi' \mid \phi, m, L$) corresponds to the output $y' \mid y, \mathbf{q}$ produced by the stochastic mapping $\tilde{S}(y, \mathbf{q})$. Therefore, $\tilde{S}(\phi, (m, L))$ consists of sampling from a Bernoulli distribution $Bernoulli(p = p_{\phi'=1|\phi}(m, L))$ where $p_{\phi'=1|\phi}(m, L)$ is the underlying probability function $\Psi(y = \phi, \mathbf{q} = (m, L))$ in the general formalism.

We approximate $\Psi(y, \mathbf{q}) = p_{\phi'=1|\phi}(m, L)$, using GPs trained on observations from *micro-trajectories*, i.e. trajectories of the fine F/M pCTMC system which are then mapped onto the property space, $\phi \in \{0, 1\}$, to serve as training data. The nature and training of the GPs is described in Section 4.2.2. Note that the Bernoulli distribution likelihood, used here for Gaussian process classification (GPC), is a special case result because of both the binary $y = \phi$ output and the single observation of transitions at a particular (m, L) parametrisation.³ Lifting these restrictions would result in the more general multinomial distribution likelihood.

Observations are gathered from simulation of the original system, and are therefore generated as follows. At a given (m, L) the pCTMC with transition rates $k_{\pm}(m, L)$ is at a state \mathbf{s}_0 which maps onto $\phi(\mathbf{s}_0)$. After a time Δt , the same CTMC is found at a state $\mathbf{s}_{\Delta t}$, which maps onto $\phi(\mathbf{s}_{\Delta t})$. An observation $\phi(\mathbf{s}_{\Delta t}) \mid \phi(\mathbf{s}_0), m, L$ is in this way recorded for every parametrisation (m, L) the bacterium has visited in the micro-trajectories.

Since the output of \tilde{S} is binary ($y = \phi \in \{0, 1\}$) we construct two probability functions $\Psi_{\phi}(m, L) = p_{\phi'=1|\phi}(m, L)$. Each is approximated with a separate GPC function, where $\Psi_0(m, L)$ is trained on observations of transitions originating from the ‘TUMBLE’ state ($p_{\phi'=1|\phi=0}(m, L)$) and $\Psi_1(m, L)$ using transitions from the ‘RUN’ state ($p_{\phi'=1|\phi=1}(m, L)$). Notice that we need not estimate separate functions for $\phi' = \{0, 1\}$, since $p_{\phi'=1|\phi}(m, L) = 1 - p_{\phi'=0|\phi}(m, L)$. Having access to these underlying probability functions we are now able to sample the DTMC at any parametrisation (m, L) the bacterium finds itself in, by using the function estimate for $p_{\phi}(m, L)$ despite not having observations at that m, L .

The function $p_{\phi'=1|\phi}(m, L)$ is particularly challenging for GPs. This is due to a sharp boundary in the m, L domain, where there is a transition from $p_{\phi'=1|\phi}(m, L) \approx 0$ to $p_{\phi'=1|\phi}(m, L) \approx 1$. The bacterium has a steady state very close to this boundary,

³It is highly unlikely to have more than a single transition since (m, L) are continuous values that constantly change for the bacterium.

determined by the motor bias mb_0 , and that is where they are most often found. Therefore, accurate estimation of this boundary is crucial for this problem. Furthermore, the low probability of finding bacteria away from the boundary (in a relatively smooth ligand field) gives a very narrow window of where the function is observed. To get a better overall estimate, we sporadically perturb the position of bacteria in the micro-trajectory phase of collecting observations, such that the bacterium finds itself producing observations away from the boundary for a while, before the system returns close to steady state again. Despite these difficulties, we produce a good reconstruction of the underlying functions $p_{\phi'=1|\phi=0}$ and $p_{\phi'=1|\phi=1}$ over the m, L domain (see Figure 4.3).

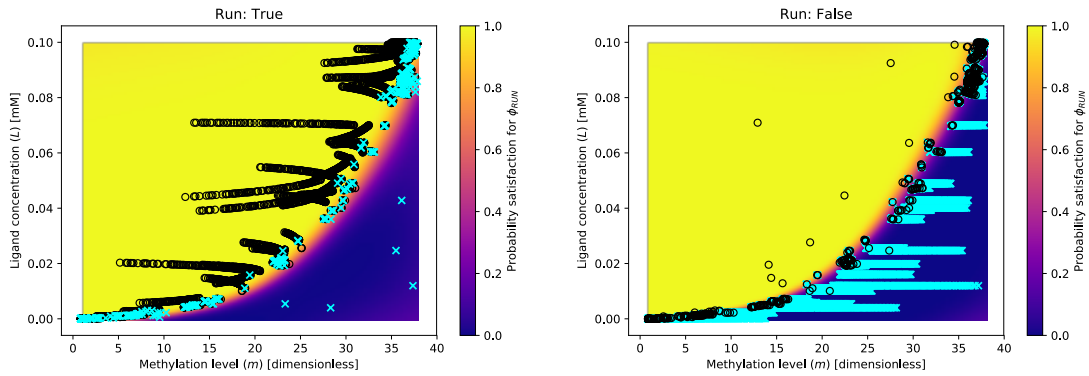


Figure 4.3 The probability functions $p_{\phi'=1|\phi}(m, L)$ (left: $\phi = \top$, right: $\phi = \perp$) produced by the GP with hyperparameters $\ln(\ell) = (3.5, -2.5)$ and $\ln(\sigma) = 5, 100$ inducing points (FITC approximation), and 10,000 observations (black circles for $\phi' = \top$, cyan crosses for $\phi' = \perp$). The steep boundary is accurately captured, producing a sharp, switch-like transition from the run domain to the tumble domain.

The function estimates shown in Figure 4.3 were obtained by using information from 10,000 simulation runs to construct a data set for GP regression. It is an interesting question how robust our results are against changes in the choice of number of training simulations. To assess this, we repeated the full analysis (including the one to be discussed in the next section) with 5,000, 2,000 and 1,000 training points. This exercise resulted in very similar overall results for 5,000 training simulations; however, for 2,000 and 1,000 simulations a significant deterioration of the approximation quality became apparent.

While it is difficult to *a priori* decide on the number of training points, a useful empirical diagnostic can be obtained by observing plots of the probability function $p_{\phi'=1|\phi}(m, L)$. Figure 4.4 shows clearly that, while the left-hand plot (corresponding to 5,000 training points) still retains a good approximation of the probability function, the middle and particularly right-hand panels (corresponding to 2,000 and 1,000 points respectively) give an inaccurate reconstruction where the areas far from the boundary

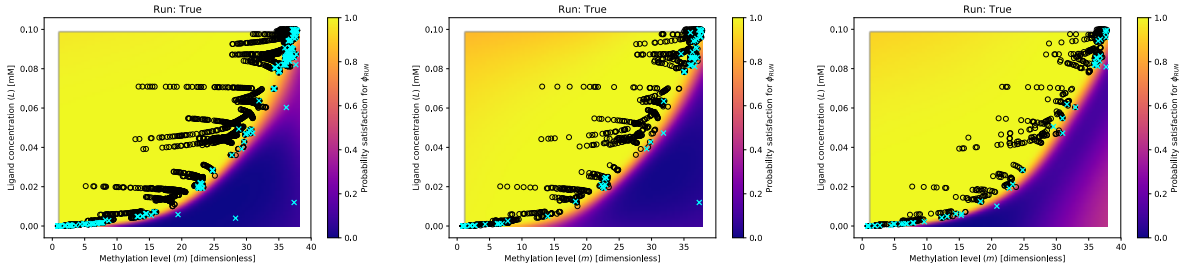


Figure 4.4 Sensitivity of the estimate of the probability function $p_{\phi'=1|\phi=1}(m, L)$ when varying the number of training points (left: 5,000, middle: 2,000, right: 1,000). The middle and left figures clearly show insufficient exploration of the property space, resulting in an inaccurate estimation of the probability function. Particularly and especially away from the transition boundary, there are not enough cyan observations causing the function to revert to the mean prior (observe colour difference near the edges).

region reverts heavily to the prior GP mean. The latter is due to a lack of sufficient observations belonging to the minority class (these are the cyan crosses for the plotted $p_{\phi'=1|\phi=1}(m, L)$ in Figure 4.4); the $p_{\phi'=1|\phi=1}(m, L) \approx 1$ domain, mostly populated by positive observations (black circles), is affected to a lesser degree.

4.4.3 Results

When assessing performance of our method for statistical abstraction, there are two things of interest: accuracy and computational savings. Accuracy refers to how similar behaviour of the abstracted system is to the behaviour of the original system. In our case of chemotaxis in *E. coli*, this is seen by comparing population distributions in a ligand field, resulting from simulations using the original fine system and the abstracted one. We also compare run and tumble duration distributions as another metric of how closely we approximate the output and the behaviour of the original model.

Learning the transition probability functions for the dual-state DTMC enabled us to simulate bacteria using our abstracted model on a host of different ligand field profiles. Beyond comparing bacteria population distributions under the original Gaussian ligand field used for learning (see L_1 below), we did the same for a linear and dynamic field (L_2 , L_3 below), using the same learned functions $p_{\phi'=1|\phi}(m, L)$, $\phi \in \{0, 1\}$.

The ligand fields tested were:

$$L_1(\vec{r}) = 0.1 \cdot \exp \left[-0.5(\vec{r}^\top \Sigma^{-1} \vec{r}) \right], \quad \Sigma = 3 \cdot \mathbf{I}_2; \quad (4.9)$$

$$L_2(\vec{r}) = \max \left(10^{-5}, 0.1 - 0.05 \sqrt{(\mathbf{A} \vec{r})^\top \mathbf{A} \vec{r}} \right), \quad \mathbf{A} = \begin{pmatrix} 1/5 & 0 \\ 0 & 1/2 \end{pmatrix}; \quad (4.10)$$

$$L_3(\vec{r}, t) = 0.1 \cdot \exp \left[-0.5(\vec{r}^\top \Sigma(t)^{-1} \vec{r}) \right], \quad \Sigma(t) = 3(t/50 + 1) \cdot \mathbf{I}_2. \quad (4.11)$$

In the fields above, the maximum value is 0.1 (units are mM) and this peak concentration is at $\vec{r} = (0, 0)$. The field L_2 is a static, non-isotropic, linear field, whereas L_3 is a dynamic field: a Gaussian spreading out over time, similar to what one might expect to be produced by a diffusing drop of nutrients. As expected, as long as the stimulus concentrations and their spatial gradients are within the region observed in training, the population distributions show consistency with those produced when simulating using the original full model (see the discussion on *accuracy evaluation* below, as well as Table 4.1 and Figure 4.5).

Computational cost savings Computational savings are given empirically here by comparing running times of simulations for both systems. A hundred (100) cells are simulated in each of the ligand fields, for a time $t_{\text{end}} = 500\text{s}$ and a time-step of $\Delta t = 0.05$. Therefore, one million (1000000) iterations of the main while loop in Algorithms 1, 2 (Appendix A.2) are compared in the reported speed-up factor (Table 4.1). We observe a speed-up factor of ~ 8 , reducing running times from $\sim 460\text{m}$ to $\sim 60\text{m}$. Table 4.1 reports speed-up factors for each ligand field experiment.

The reported factor values do not include the costs paid for training the GP and producing the training data. It takes $\sim 4\text{min}$ to train GPs for both Ψ_ϕ functions, and $\sim 10\text{min}$ for producing 20000 observations of pCTMC transitions from the original fine system (10000 training points for each Ψ_ϕ function). The relatively low times compared to simulation times, combined with the fact that one only pays this once, upfront, make these costs negligible.

Accuracy evaluation To evaluate how closely results from the abstracted model are compared to the original one, we applied the Kolmogorov-Smirnov (KS) two-sample test (Chakravarty et al., 1967) to the population distributions of the two models at several time-points in the simulation, as well as to the distributions of running and tumbling duration. We have 100 samples from each population distribution since we simulated 100 cells. However, in the case of ‘Run’ and ‘Tumble’ duration distributions we have ~ 60000

observations from each, because we aggregate observations from the entire trajectory; we choose a random 1000 sample of these to perform the KS test.⁴ In light of these difficulties, a different test which quantifies the distance between the two distributions (e.g. Jensen-Shannon divergence) might be more useful here, but that requires analytic forms of the distributions.

Inspecting Table 4.1 we find no KS distance higher than 0.2 indicating very similar distributions, as supported by the associated high p-values. The latter do not allow rejecting the null hypothesis with the current sample, which is that the samples originate from the same distribution. An exception is the ‘Tumble’ duration distributions in the L_1 ligand field, where the somewhat higher KS distance of the large sample sizes gives an exaggerated p-value (see footnote 4).

We note how even in the case of the dynamic L_3 field, the resulting population behaviour of the abstracted model is preserved without any additional training necessary. The fact that the original training occurred in a static field does not affect the ability of the abstract model to cope with a dynamic one.

Table 4.1 KS two-sample test statistics, where the first (top) value reports KS distance and the second (in brackets, bottom) the associated p-value. One sample came from 100 trajectories of fine *E. coli* system simulations, and the other from 100 abstracted system simulations. The first four columns show KS test results of original and abstracted bacterial population distances from peak concentration at various times t (shown in Figure 4.5). ‘Run’ and ‘Tumble’ columns compare the distributions of run and tumble durations respectively for 1000 samples from each system. The last column reports the observed speed-up factor based on running times and normalising for core utilisation.

| Field | $t = 125s$ | $t = 250s$ | $t = 375s$ | $t = 500s$ | Run | Tumble | Speed-up \times |
|--|------------------|------------------|------------------|------------------|------------------|--------------------------------|-------------------|
| Gaussian: $L_1(\vec{r})$ | 0.110 (0.556) | 0.160 (0.140) | 0.170 (0.099) | 0.160 (0.140) | 0.039 (0.425) | 0.101 ($7 \cdot 10^{-5}$) | 7.8 |
| Linear: $L_2(\vec{r})$ | 0.010 (0.677) | 0.150 (0.193) | 0.170 (0.100) | 0.130 (0.344) | 0.022 (0.967) | 0.014 (0.100) | 9.4 |
| Dynamic Gaussian: $L_3(\vec{r}, t)$ | 0.140 (0.261) | 0.070 (0.961) | 0.140 (0.261) | 0.080 (0.894) | 0.047 (0.214) | 0.039 (0.425) | 8.9 |

⁴ We sub-sample because the KS test p-value depends heavily on sample size. Even if two distributions generating samples might be very close, in the limit of an infinite sample size one approaches the true distributions. In such a case, the KS test will reject that the two samples were produced by *the same* distribution, returning lower p-values as sample size increases (for the same KS distance). We do not expect to produce the same distributions here since we are making approximations, so comparing p-values for very large sample sizes is not of interest.

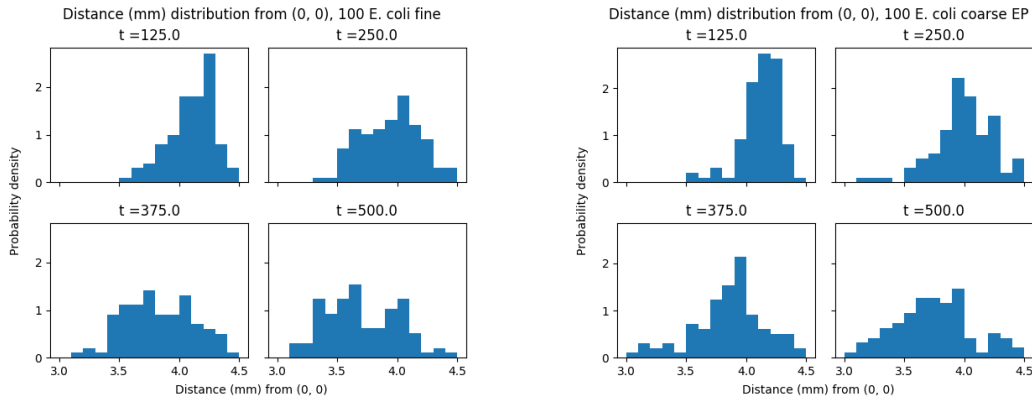


Figure 4.5 Empirical distributions for the distance of 100 bacteria from peak of L_1 ligand field at different times t of the simulation. Left: original full system simulations. Right: abstracted system simulations.

4.5 Closing the information loop

The *E. coli* model we abstracted had a one-way flow of information: the environment affected the agent state, but the agents had no effect on the state of the environment layer. Here, we demonstrate the method on a model which closes this information loop by having agents influencing their immediate environment (local interactions) which additionally allows for agent-agent interaction with the environment layer acting as a conduit for signalling. *Dictyostelium discoideum* (commonly *slime mould*) is a species of amoeba which naturally displays such behaviour, making models thereof a suitable choice for our abstraction framework. These single-cell organisms live in colonies, where they are in close proximity to many other members of the species. They exhibit agent-agent interaction through emission and response to a particular chemical. This triggers a chemotactic response which enables the amoebae to aggregate into multi-cellular groups, which is beneficial to their survival under starvation conditions, as well as a natural part of their reproductive cycle.

The chemotactic behaviour of the amoebae exhibits all the hallmarks of a multi-level spatio-temporal system where agents can also communicate with each other through environmental signals. In keeping with our general philosophy, our angle of attack is to choose a detailed model of the individual cell’s motility response and abstract it, so that the emergent aggregation behaviour in the population is still observed. Extensive observations at cell level have shown that *D. discoideum* move by extending tentacle-like cell wall deformities called *pseudopodia*, and shifting themselves in that direction (Bosgraaf and Haastert, 2009; Haastert, 2010). Based on this general principle, there are various approaches to model the trajectories produced by such a mechanism. Here,

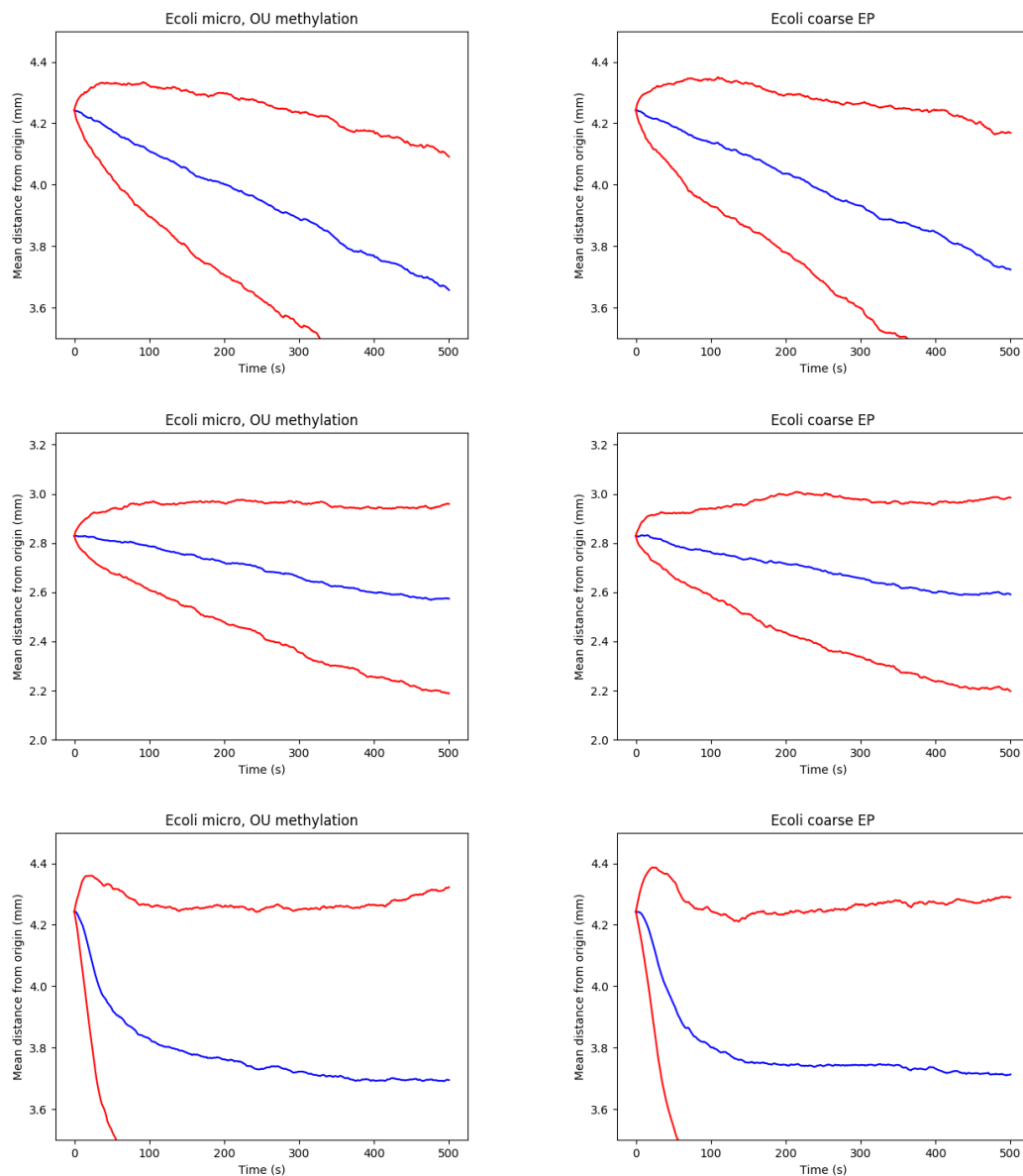


Figure 4.6 Average (blue) and standard deviation (red) of distance from peak ligand concentration for a population of 100 *E. coli* over a time of 500s. Left: original full system simulations. Right: abstracted system simulations. Rows (top to bottom): L_1, L_2, L_3 ligand fields respectively.

we opt to work with discrete models where the cells take indivisible steps towards a particular direction. Other approaches exist where the cell is taken to perform continuous stochastic motion with correlations, as in (Li et al., 2011).

It has been further well observed that in the absence of environmental stimuli, the angle at which the next pseudopodium is extended relative to the current one follows a

symmetric probability distribution over $(-\pi, \pi)$, centred at 0. This causes the amoebae to explore their environment by performing isotropic random walks in the long run. However, the angle probability distribution shifts in the presence of a cyclic Adenosine Mono Phosphate (cAMP) chemical gradient to induce a drift in the random walk aligned to the gradient. When *D. discoideum* cells fall into starvation they emit this chemical which diffuses through space and reaches other nearby amoebae (Haastert and Bosgraaf, 2009; Robertson and Grutsch, 1981). In response, the sensing amoebae emit more cAMP which relays the signal further, and also begin to chemotax towards the cAMP source. The amoebae eventually congregate into a *multicellular pseudoplasmodium* — a slug-like structure which behaves as a single organism, giving the population under the starvation conditions a better chance of survival. The slug collective moves as one to find environmentally favourable conditions, where it settles and begins a reproductive process (Robertson and Grutsch, 1981).

The model we abstract is an amalgamation of two other models, one focusing on the internal workings of the amoeba to induce motility by Eidi (2017), and the other focusing on cAMP emission and response cycles by Calovi et al. (2010). The former provides a discrete-time Markov chain (DTMC) formalism where directions in which the cell moves are states of the chain, and the transition probabilities depend on the sensed cAMP gradient. The latter provides an efficient method to compute the cAMP chemical field at a point in space and time $\gamma(\mathbf{x}, t)$, by utilising a Green’s function (GF) method to solve the diffusion equation with degradation. Having melded the two models, we verify that the aggregation behaviour between the agents under cAMP emission is consistently observed and show that it is preserved in the abstraction.

We postulated that the motion of the amoeba cells can be well approximated by a simple Markovian process: a Bernoulli trial which determines the alignment of the next pseudopodium to be extended with respect to the cAMP gradient, based on the latest kind of pseudopodium (split/ *de novo*) and its alignment to the direction of the cAMP gradient. Pseudopodia can either be a result of *splitting* the current pseudopodium, or a *de novo* extension unrelated to the current one. The two kinds induce probability distributions over the possible extension angles of the next pseudopod with respect to the cAMP gradient. Since this dynamics is more complex than the simple *run/tumble* dynamics of the *E. coli* model, and the agent layer model in (Eidi, 2017) is already considerably simplified, we do not expect the abstraction to yield significant computational gains. Nevertheless, it is still a useful proof of principle of our methodology, and illustrates how additional insights can be gained through identifying the necessary properties for preserving the qualitative behaviour of this model, which rests upon interaction between

agents. We find that we retain the influence of agents on the environment and observe that the agent-environment-agent communication results in the macro-scale behaviour of aggregating amoebae for the abstracted model as well, albeit with some loss in accuracy.

4.5.1 Original model

The original model consists of two layers, the environment layer and the internal agent (*D. Discoideum* cell) layer. The two are coupled such that output from one layer is the input to the other layer and vice versa. The environment layer takes as input the cAMP emission history from each amoeba cell, and evaluates cAMP concentration and its gradient at all agent positions. This serves as input to the internal agent layer, which picks a direction for the cell to move and updates its position accordingly; it also updates the cAMP emission history for the cell with the latest cAMP emission value.

Environment layer The model laid out in (Calovi et al., 2010) for cAMP diffusion in space and the corresponding methodology is taken as the environment layer we use. Following the authors, we describe cAMP diffusion with degradation and emitting sources through the coupled differential equations of Martiel and Goldbeter (1987) (MG equations):

$$\partial_t \gamma(\mathbf{x}, t) = \frac{k_t}{h} \beta(\mathbf{x}, t) - k_e \gamma(\mathbf{x}, t) + D \nabla^2 \gamma(\mathbf{x}, t), \quad (4.12)$$

$$\beta(\mathbf{x}, t) = \sum_{j=1}^N \beta_j(t) \exp \left[-\frac{4}{\sigma^2} (\mathbf{x} - \mathbf{x}_j)^2 \right], \quad (4.13)$$

$$\frac{d\beta_j}{dt} = \phi(\rho_j, \gamma_j) - (k_i - k_t) \beta_j, \quad \frac{d\rho_j}{dt} = f_2(\gamma_j)(1 - \rho_j) - f_1(\gamma_j) \rho_j, \quad (4.14)$$

where f_1 , f_2 , ϕ , and Y_j are defined as:

$$f_1(\gamma_j) = \frac{k_1 + k_2 \gamma_j}{1 + \gamma_j}, \quad f_2(\gamma_j) = \frac{k_{-1} + \lambda_1 k_{-2} \gamma_j}{1 + \lambda_1 \gamma_j}, \quad (4.15)$$

$$\phi(\rho_j, \gamma_j) = \frac{\lambda_2 + Y_j^2}{\lambda_3 + Y_j^2} \times 1800, \quad Y_j = \frac{\rho_j \gamma_j}{1 + \gamma_j}, \quad (4.16)$$

and \mathbf{x}_j is the location of the j th amoeba in Cartesian coordinates. The cAMP concentration field is given by $\gamma(\mathbf{x}, t)$, which evolves according to the partial differential equation (PDE) (4.12). The term $\beta(\mathbf{x}, t)$ describes the emission of cAMP from every cell, and $-k_e \gamma(\mathbf{x}, t)$ models the chemical's natural degradation. Constants used in the above equations are fixed (Table A.1 Appendix A.3, as given in (Calovi et al., 2010)).

The last couple of differential equations (Eqs. (4.14)) describe intracellular concentration and the ratio of active cAMP receptors of the j th amoeba respectively. These are intracellular processes independent within each cell, and so we implement them in the internal agent layer of our model.

To solve Equation (4.12), we implement a Green's function method⁵ as in (Calovi et al., 2010), and produce the solution

$$\gamma(\mathbf{x}, t) = \sum_{j=1}^N \int_0^t \frac{c_j(s)}{2^{2d}} \exp[-k_e(t-s)] \prod_{k=1}^d \left[\operatorname{erf} \left(\frac{l_{j,k} + R_\sigma}{\sqrt{4D(t-s)}} \right) - \operatorname{erf} \left(\frac{l_{j,k} - R_\sigma}{\sqrt{4D(t-s)}} \right) \right] ds, \quad (4.17)$$

where $c_j(s) = \frac{k_x}{2h} \beta_j(s)$ is the amount of cAMP created by the j th amoeba at time $t-s$, and $l_{j,k}$ is the distance between j th amoeba and \mathbf{x} on the k Cartesian coordinate.

Despite the integration in time in Equation (4.17) which necessitates the use of numerical integration techniques, it is still a more efficient solution to calculate than alternative finite element techniques for integrating the PDE (4.12). The latter require discretisation of space to a fine resolution (comparable to amoeba cell size) and would scale accordingly, whereas Equation (4.17) scales linearly with the number of amoebae. Additionally, the natural degradation of cAMP allows us to only keep a finite history of the emission from every agent, and limits the integration time required for achieving a good enough value for the concentration field.

Agent layer The intracellular set of processes takes as input the concentration and gradient of cAMP at the cell's location and determines the cell's cAMP emission rate and its direction for the next Δt time step. The MG Equations 4.14 model the emission rate evolution and are solved in a forward Euler manner as shown below.

Formally, the agent layer comprises of a set of equations:

$$\mathbf{s}^t = S(\mathbf{s}^{t-1}, \nabla \gamma_j^{t-1}), \quad (4.18)$$

$$\beta_j^t = \beta_j^{t-1} + \Delta t \left[\phi(\rho_j^{t-1}, \gamma_j^{t-1}) - (k_i - k_t) \beta_j^{t-1} \right], \quad (4.19)$$

$$\rho_j^t = \rho_j^{t-1} + \Delta t \left[f_2(\gamma_j^{t-1})(1 - \rho_j^{t-1}) - f_1(\gamma_j^{t-1}) \rho_j^{t-1} \right], \quad (4.20)$$

⁵A well-known method for solving a partial differential equation (PDE), the GF is the inverse of the linear differential operator in the PDE; it is therefore the solution $G(x, y, t, s)$ of the PDE at (x, t) when driven by a point source $\delta(x-y, t-s)$. In the case of the diffusion equation with degradation, the solution for a point source is analytically known. Since the operator is linear, one can then reconstruct the solution to the PDE with the actual source $f(x, t)$ by superposition of the GF solutions for every point source in the domain of space and time — i.e. computing $\int dy ds G(x, y, t, s) f(y, s)$. See (Butkov, 1995) for a comprehensive exposition.

where the last two equations are forward Euler methods for Eqs. 4.14, the superscripts $t \in \mathbb{N}$ denote time steps, and S is a step on a Markov chain. The Markov chain has state space $I = \{(s_1, s_2)\}$ where $s_i \in \{0, 1, 2, 3, 4, 5\}$ signifies the angle $\theta = s_i(2\pi/6)$ of a step taken by the cell, with respect to the horizontal axis in a 2D space. The state space consists of tuples of step directions since both the latest step and the one taken before it are necessary to correctly describe the cell's motion. The transition probabilities satisfy the condition $s_2^t = s_1^{t-1}$ that the second element of the tuple is the previous step direction. They also shift according to $\theta_{rel}(s_i, \nabla\gamma)$, the angle between a potential step direction and the cAMP gradient $\nabla\gamma$, in order to bias the next step direction to align with the cAMP gradient. As modelled in (Eidi, 2017), the bias is a linear superposition of the term $\epsilon \cos(\theta_{rel}(s_i, \nabla\gamma))$ on the four s_i directions other than the latest step direction of the cell and the one directly opposite it. Picking an appropriate $\epsilon = 0.04$, we retain the probability conditions for the transition probabilities of the Markov chain, $0 \leq p_{ik} \leq 1$ and $\sum_j p_{ik} = 1, \forall ik$. The result is transition probabilities $p_{ik} = p_{ik}^0 + \epsilon \cos(\theta_{rel}(s_k, \nabla\gamma))$ where p_{ik}^0 are the unbiased transition probabilities. Note that these states are tuples of step directions, and so the difference between consecutive steps is relevant in determining probabilities for the next step direction.

Simulation scheme We perform simulations of the original model presented above with the following set-up: we initialise 250 agents at random positions drawn from a uniform distribution over a 0.0625mm^2 square with centre $(0, 0)$. We set $\Delta t = 0.3\text{m}$ and iterate between executing processes in the environment layer (evaluating the cAMP concentration and gradient at the agents' positions), and executing processes in the internal agent layer (updating the agents' positions and cAMP emissions). The simulation ends at $t_{end} = 30\text{m}$, at which point the agents have congregated into one or more clusters. Initial values for the internal agent parameters are $\beta_j^0 = 0$, and ρ_j^0 drawn from a uniform distribution $U[0, 1]$ for each j agent.

4.5.2 Abstracted model

Our aim is to find appropriate logical properties evaluated on the states such that their satisfaction probabilities can be used to correctly recreate the motility characteristics of *D. discoideum* cells. Guided by the biological theory used to craft the model by Eidi, we

formalise the abstracted internal agent model with the following equations:

$$\mathbf{s}^t = \tilde{S}(\mathbf{s}^{t-1}, \mathbf{s}^{t-2}, \nabla\gamma_j^{t-1}), \quad (4.21)$$

$$\text{where } \tilde{S}(\mathbf{s}^{t-1}, \mathbf{s}^{t-2}, \nabla\gamma_j^{t-1}) = \text{Categorical}(p_\phi); \quad (4.22)$$

$$\phi \sim \text{Bernoulli}(\Psi_y(\theta_{rel}(s_1^{t-1}, \nabla\gamma_j^{t-1}))), \quad (4.23)$$

with Bernoulli outcomes 1 for $|\theta_{rel}(s_1^{t-1}, \nabla\gamma_j^{t-1})| < \pi/2$ (extension will be aligned to cAMP gradient), or 0 otherwise (not aligned); and

$$y = \begin{cases} 1 & \text{for } |\theta_{rel}(s_1^{t-1}, s_2^{t-1})| < \pi/2 \text{ (split extension),} \\ 0 & \text{otherwise (de novo extension).} \end{cases} \quad (4.24)$$

The above describes the process of receiving an input $(\mathbf{s}^{t-1}, \nabla\gamma_j^{t-1})$ and going through the steps of: (1) evaluating whether the current pseudopod extension is of a split or *de novo* nature ($y \in \{1, 0\}$); (2) using the appropriate learned stochastic function Ψ_y to sample whether the next pseudopod will be within $\pi/2$ rad from the cAMP gradient direction (ϕ); and finally (3) determining the direction of the next pseudopod by picking from a categorical distribution of the possible directions with event probability vectors p_ϕ . The vectors p_ϕ are constructed as follows:

$$p_\phi = [r_1, \dots, r_6], \quad (4.25)$$

with elements

$$r_i = \begin{cases} 1/Z & \text{if } (i \neq s_1^{t-1}) \wedge \\ & \left((\phi = 1 \wedge |\theta_{rel}(i, \nabla\gamma_j^{t-1})| \leq \pi/2) \vee (\phi = 0 \wedge |\theta_{rel}(i, \nabla\gamma_j^{t-1})| \geq \pi/2) \right), \\ 0 & \text{otherwise,} \end{cases}$$

where Z is a normalisation constant such that $\sum_i r_i = 1$. The conditions for the elements of p_ϕ imply that two consecutive steps cannot be taken in the same direction ($i \neq s_1^{t-1}$), and that for $\phi = 1$ the step must be in an angle within $\pi/2$ rad relative to the gradient $\nabla\gamma_j^{t-1}$ and vice versa for $\phi = 0$. If these conditions are not met for a step direction i , no probability mass is given ($r_i = 0$). Note also that the probability mass is distributed equally amongst allowed directions ($1/Z$), violating the original model's rule that a step directly opposite the last one has a constant probability of occurring ($1/21$) which is unaffected by the perceived cAMP gradient. The abstracted system is essentially a

second-order DTMC with transition rates dependent on an external input, the cAMP gradient.

Finally, we retain the forward Euler method for solving the MG differential equations for cAMP emission (Equations 4.19, 4.20):

$$\beta_j^t = \beta_j^{t-1} + \Delta t \left[\phi(\rho_j^{t-1}, \gamma_j^{t-1}) - (k_i - k_t)\beta_j^{t-1} \right], \quad (4.26)$$

$$\rho_j^t = \rho_j^{t-1} + \Delta t \left[f_2(\gamma_j^{t-1})(1 - \rho_j^{t-1}) - f_1(\gamma_j^{t-1})\rho_j^{t-1} \right]. \quad (4.27)$$

As before, the probability functions $\Psi_y(\theta_{rel}(s_1^{t-1}, \nabla\gamma_j^{t-1}))$ for $y \in \{1, 0\}$ are approximated by GPs trained on original model observations (Figure 4.7). However, the two functions here do not correspond to the satisfaction value ϕ of the property being approximated by the GP, but a different one. Effectively we define two separate properties to achieve a good abstraction: one for whether the current pseudopod was a split or *de novo* extension ($y \in \{1, 0\}$), used to differentiate which stochastic function Ψ_y is used; and another of whether the next extension will be aligned to the cAMP gradient ($\phi = 1$ if s_1^t within $\pi/2$ rad of $\nabla\gamma_j^{t-1}$, 0 otherwise). Because of the two different properties, this abstraction is not simply a reduction of the original Markov chain to one with less states as the one for *E. coli* was, but rather a re-defined set of processes making use of learned stochastic functions to estimate needed probabilities for preserving the motility behaviour. As in our previous abstraction of the *E. coli* model, the same input to the layer is needed to produce the same output, so we can replace the original internal model with the abstracted one.

This *D. discoideum* model abstraction involves re-interpreting the original 30 state Markov chain producing 6 discrete possible outcomes for the internal agent layer, as chained functions. The first function takes as input the nature of the current pseudopod (split / *de novo*) and its angle relative to the cAMP gradient $\theta_{rel}(s_1^{t-1}, \nabla\gamma_j^{t-1})$, and stochastically determines whether the next step will be within $\pi/2$ rad of the cAMP gradient; the second function takes this decision and (stochastically) converts it to one of 6 possible directions for the next pseudopod which are consistent with the decision. The conversion to direction in the latter function happens through categorical distributions with fixed probability vectors which do not depend on the angle θ_{rel} — i.e. the first function removes the dependence of the layer output on the relative angle and split / *de novo* nature of pseudopod. Given ϕ , the satisfaction of the alignment to cAMP gradient for the next step, we have all the necessary information to produce the internal layer's output.

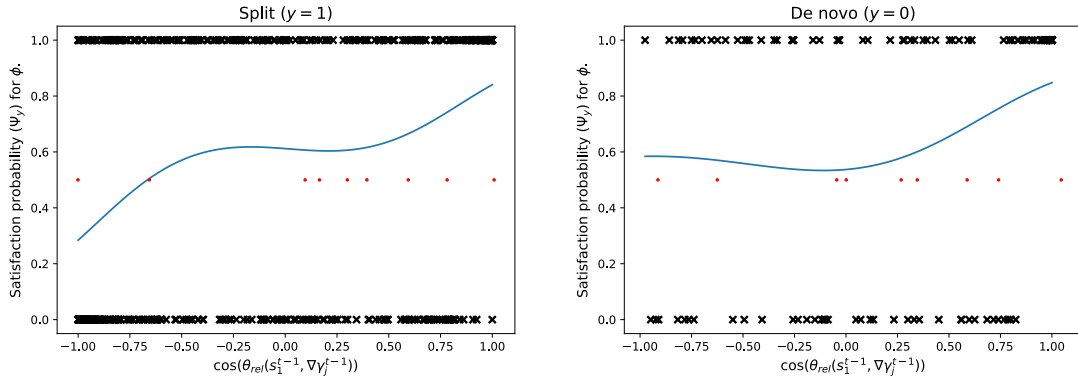


Figure 4.7 $\Psi_{y=1}$ left, $\Psi_{y=0}$ right. Black crosses are observed outcomes from simulation of the original system. Red dots are the (nine) inducing points used for the FITC approximation. The curves show how the satisfaction probability of ϕ (whether the next step will be aligned to the cAMP gradient), is inferred to respond to the relative angle between the cosine of current step and the gradient: $\cos(\theta_{rel}(s_1^{t-1}, \nabla \gamma_j^{t-1}))$. As can be seen, the nature of the current pseudopod (split $y = 1$, or *de novo* $y = 0$) produces significantly different curves.

4.5.3 Results

As before, we present results of both the accuracy and computational costs of the abstracted model with respect to the original one, after running twenty simulations of each model. In the original model, the internal agent layer amounted to sampling a single transition in a discrete-time Markov chain of effectively 30 states but with only 5 possible transitions from each state. Transition probabilities had to be re-evaluated before each sampling according to environmental input, but the whole layer has low computational costs. The abstracted model instead takes as input a binary satisfaction for a logical property and a continuous value $\in [-1, 1)$ and outputs one of 5 possible values (corresponding to the 5 permitted transitions from each state of the DTMC) stochastically. Due to the low computational cost of the original layer, we do not expect significant gains from the abstraction; we therefore focus on recovery of the emergent behaviour dependent on non-linear interaction effects between parts of the model which we are abstracting.

The macro-scale phenomenon we wish to retain here is the aggregation of agents, which we attempt to quantify for comparison purposes. To that end, we construct an empirical distribution of the agents' locations over space through the use of Gaussian kernel density estimators (KDE) (Bishop, 2006b), at various times in the evolution of both systems, as seen in Figures 4.8, 4.9. An inspection of Figures 4.8, 4.9 shows that

qualitatively both the original and the abstracted model show an aggregation behaviour into a major cluster within the time-frame of the experiment. Since there is a single cluster forming and the uniform distribution was centred at $(0, 0)$, we expect that the distribution of the agents' distance from the centre is a good proxy for quantifying aggregation of the population.

To get a quantitative measure of the agreement between original and abstracted model, we pool the results of 20 simulations and compare the histogram distributions of agent distance from the origin at different time points. Figure 4.10, left panel, shows the first two statistics (mean and standard deviation) of the distance of the agents from the origin as a function of time for both models. This figure shows an excellent statistical agreement between the two models, even though the rate of aggregation seems slightly higher in the abstracted model than the original one. We can gain some insights into the origin of this discrepancy by looking at the histogram of agent distance from origin at the end of the simulation time, shown in Figure 4.10, right panel. This shows that, while the bulk and the mode of the distribution are well matched between the two models, the original model distribution has a heavier tail than the abstracted one. A potential cause of these deviations is that the choice of property matched (Eq. 4.23) does not contain sufficient information to precisely match the macro-scale behaviour, particularly in terms of rarer events. This points to a need for an automatic property construction method, which is something to be explored in future work.

Running times for a simulation were 219 ± 6 min and 249 ± 8 min for the original and abstracted system respectively. Experiments were run on a server machine with 48 CPU cores working in parallel. Note that most resources went towards evaluating the integral in Eq. (4.17) for each agent at every step. As expected, the abstraction does not yield computational gains for this model. We can attribute this to the small size of the original model and the overhead of producing predictions from a trained GP. This illustrates a fundamental trade-off within our abstraction approach: while the approach is generally applicable, the wisdom of applying such an abstraction depends on the specific model. In particular, when the internal agent layer is straightforward to simulate, the additional overheads of the GP abstraction can nullify the computational advantages stemming from using a simpler system.

4.6 Discussion

In many domains, ranging from cyber-physical systems to biological and medical processes, consideration of spatio-temporal aspects of behaviour is essential. However, this comes

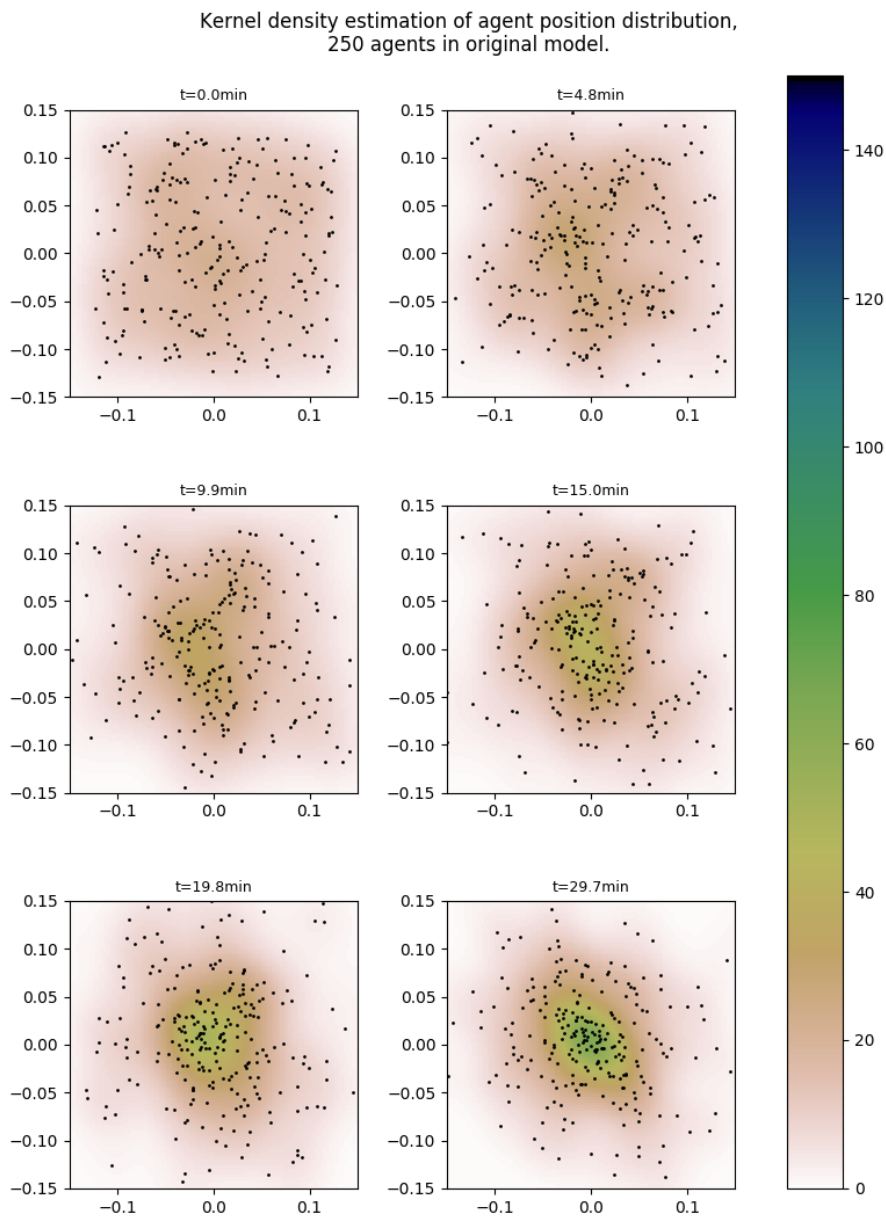


Figure 4.8 Original model, 250 agents (black dots) in a 0.0625mm^2 space at different times. Background intensity field shows the KDE estimation of population distribution density. Note how the agents begin to aggregate to one major cluster by $t = 15\text{min}$.

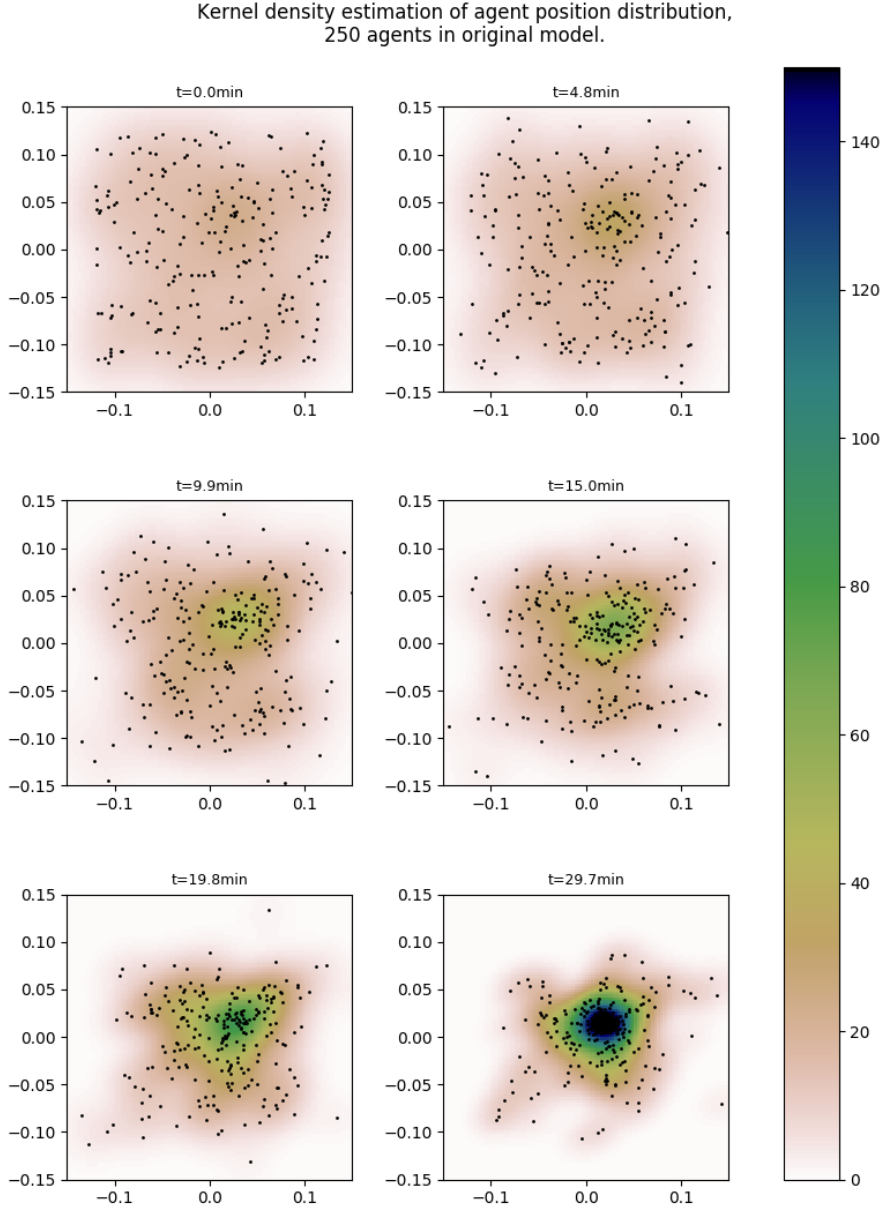


Figure 4.9 Same as Figure 4.8 for abstracted model. Note here that the agents begin to aggregate sooner than the original model. By $t = 15\text{min}$, what will be the final major cluster has already formed.

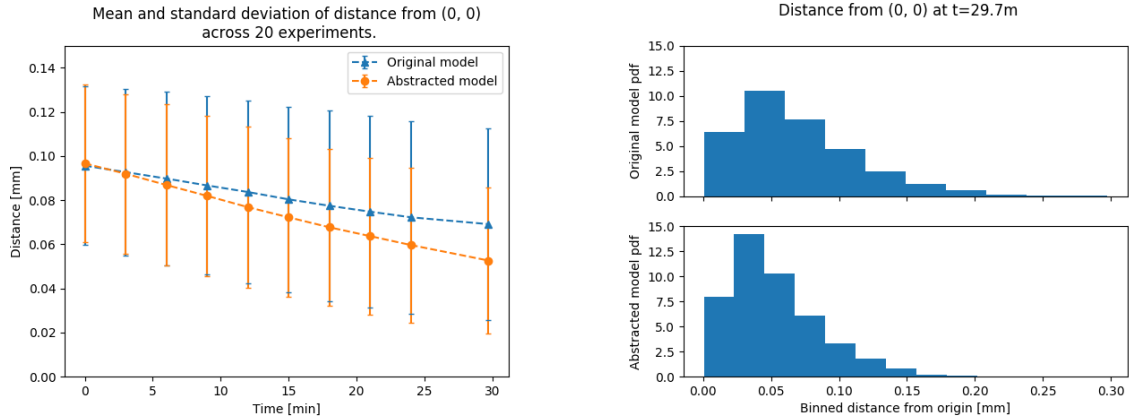


Figure 4.10 Results from all 20 experiments were pooled together and treated as a single population for this analysis to increase sample size. Individual experiments are consistent with the pooled results. Left: original and abstracted model mean distance of *D. discoideum* agents from origin (blue triangles and orange circles respectively) and standard deviation (error bars) over time. It can be seen that the population moves towards the origin as the system evolves, which reflects the aggregating tendencies of both models. Deviations exist (abstracted model aggregates faster), but general behaviour is retained. Right: density histogram of the pooled populations at the last time step of the simulation. One should note that despite the closeness in median values, the heavier tail of the original model distribution is much diminished in the abstraction.

at great computational expense. We have presented a methodology that allows layers of a computationally intensive multi-scale model to be replaced by more efficient abstract representations. This is a stochastic map, constructed based on some exploratory simulations of the full model and GP regression. Our results show that we are able to achieve significant speed-up without sacrificing accuracy. This establishes a framework for such statistical abstraction on which we plan to elaborate in future work.

It should be noted that the specifics of the abstraction are not automatically determined by this framework, but are left to the researcher. Having to manually specify the abstraction introduces an element of flexibility, since different abstractions may be tested and so one can see which are suitable and produce accurate approximations, indicating that pertinent elements of the original model are preserved in the coarsening. Additionally, there may be various valid ways to coarsen a model, depending on what the focus of the inquiry is. On the other hand, it shifts some of the burden of abstracting the model to the researcher, who has to find a suitable set of properties which capture the output behaviour of the layer to be abstracted.

Complete automation is commonly embraced by other methodologies, where every abstraction choice may be driven by accuracy metrics. We sacrifice complete automation

in this work for the ability to examine different kinds of fundamental models and assess various abstraction mechanisms. Since the era of modern science, the task has largely been to determine a minimal set of laws that concisely describe the range of phenomena observed. As such, fundamental equations at different levels of nature abound and mechanisms to bridge the various gaps, accompanied by ways to validate them, are in high demand. Our framework allows validation of an existing mechanism, but does not generate the mechanism itself — an abstraction framework which automatically produced interpretable mechanisms would solve many problems.

Our choice of application case studies illustrates well the importance that such an automated abstraction framework would have. While the *E. coli* chemotaxis model presents an obvious avenue for accurate abstraction, the *D. discoideum* model considered here does not have an obvious set of properties to act as a conduit from micro- to macro-scale behaviour. Specifically, the *D. discoideum* model consisted of a DTMC with states corresponding to directions of agent motion. Abstracting it required re-interpreting the model as motion dictated by a set of relevant properties; namely, the property of whether the cell produced a split or *de novo* pseudopod last, determined whether or not the next pseudopod direction would be aligned to the cAMP gradient sensed by the cell. The process of re-interpretation might not produce a particularly useful abstraction in terms of computational efficiency, but it is valuable in understanding the processes in the layer and the relevant information flow through them.

Because the properties necessary for an accurate abstraction have to be manually defined, we can also use this abstraction framework to evaluate consistency of the original model with other higher-level models which have different assumptions. For instance, we first attempted to replace the agent layer of the *D. discoideum* model with a stochastic function which receives as input whether the current pseudopod was a split or *de novo* nature and produces output of whether the next one will be a result of split or *de novo*, akin to the *E. coli* case. This is a simpler abstracted model than the one we presented, and is supported by the literature (Haastert and Bosgraaf, 2009; Li et al., 2008). The result was a population of agents which did not display aggregation and instead retained their original uniform distribution in space, or even slightly diffused. We can therefore assert that the original model by Eidi used here cannot be cast down to a simple first order two-state discrete Markov chain (split / *de novo* being the two states) which has transition probabilities dependent on the output. The property of whether the next step is aligned to the cAMP gradient (beyond split / *de novo*) *is relevant* and cannot be discarded. This raises the question of which is the better model for *D. discoideum* motility, since other models claim to achieve similar aggregation behaviour with simple

split / *de novo* models. Failure to reconcile models in this manner is indicative of inherent differences in the models, which may prove useful in assessing their veracity with respect to reality.

Future work avenues include, for example, allowing more properties to be expressed and using them to guide the abstraction to capture more complex behaviours. Additionally, we could infer abstracted model parameters or underlying functions from real data, instead of synthetic ones. Finally, one would ideally like to have a way to infer suitable properties for preserving a particular macro-scale behaviour. As seen in the case of the *D. discoideum* model, this is not trivial to achieve and often the properties fall short of accurately reproducing the behaviour. An automatic way to construct these properties would relieve the researcher from having to make the choice, and might reveal further insights to the models abstracted.

Chapter 5

Continuous approximations to discrete systems: the geometric fluid approximation

Continuous-time Markov chains (CTMCs) have emerged over the last two decades as an especially powerful class of models to capture the intrinsic discreteness and stochasticity of a range of systems, from artificial processes to biochemical reactions at the single cell level. The ensuing cross-fertilisation of methods originating from various application fields has produced great advancements in the analysis of such discrete stochastic systems.

Despite the unquestionable success of these efforts, scaling formal analysis techniques to larger systems remains a major challenge, since such systems usually result in very large state-spaces, making subsequent analysis particularly onerous. In most cases, retrieving the evolution of the state distribution, while theoretically possible (by solving the Chapman-Kolmogorov equations), in practice is prohibitively expensive. These hurdles also affect statistical techniques based on Monte Carlo sampling, since trajectories from CTMCs with a large state-space typically exhibit very frequent transitions and therefore require the generation of a very large number of random numbers. A popular alternative is therefore to rely on *model approximations*, by constructing alternative models which in some sense approximate the system's behaviour. In the special case of CTMCs with a population structure (pCTMCs), *fluid approximations* replacing the original dynamics with a deterministic set of ordinary differential equations (ODEs) have seen great success, due to their scalability and their well understood convergence properties (Darling and Norris, 2008; Hillston, 2005; Kurtz, 1971). Such approximations rely on the particular structure of the state-space of pCTMCs; to the best of our knowledge, fluid approximations for general CTMCs have not been developed.

In this chapter of the thesis, we propose a general strategy to obtain a fluid approximation for any CTMC. Our approach utilises manifold learning approaches, popular in machine learning, to embed the transition graph of a general CTMC in a Euclidean space. A powerful Bayesian non-parametric regression method complements the embedding, by inferring a drift vector field over the Euclidean space to yield a continuous process, which we term the *geometric fluid approximation*. Crucially, we show that in a simple pCTMC case our geometric fluid approximation is consistent with the standard fluid approximation. Empirical results on a range of examples of CTMCs without a population structure show that our approach captures well the average behaviour of the CTMC trajectories, and can be useful to solve efficiently approximate reachability problems. This work is published on arXiv (Michaelides et al., 2019) and is currently under review for publication in *Proceedings of the Royal Society A: Mathematical, physical and engineering sciences*.¹

5.1 Background theory and related work

Bearing in mind Definitions 2.2.2 and 2.2.3 given in Chapter 2 for a CTMC and population CTMC (pCTMC) respectively, we briefly review here the foundations of the fluid approximation for pCTMCs (Darling and Norris, 2008; Gardiner, 2009; Norris, 1998), and highlight the specific aspects that render them amenable to such a construction.

5.1.1 Continuous relaxation and the *fluid limit*

As discussed in Chapter 2, the Markovian nature of CTMCs naturally provides an exact sampling algorithm for drawing CTMC trajectories: the *stochastic simulation algorithm* (SSA), or *Gillespie's algorithm* (Gillespie, 1977). The same Markovian nature also leads to a set of ordinary differential equations (ODEs) governing the evolution of the single-time marginal state probability, the celebrated *Chapman-Kolmogorov equations* (CKE), which in the case of pCTMCs go under the name of *Master equation*, also reviewed in Chapter 2. Unfortunately, such equations are rarely practically solvable, and analysis of CTMCs is often reliant on computationally intensive simulations.

In the case of pCTMCs, a more concise description in terms of the collective dynamics of population averages is however available. Starting with the seminal work of van Kampen (1961), and motivated by the interpretation of pCTMCs as chemical reaction systems, several approximation schemes have been developed which relax the original

¹Jane Hillston and Guido Sanguinetti provided feedback and advice during the development process, and edited the manuscript.

pCTMC to a continuous stochastic process (introduced in Section 2.2.4); see (Schnoerr et al., 2017b) for a recent review of the most prominent approximations.

In this work, we are primarily interested in the so-called *fluid approximation*, which replaces the pCTMC with a set of ODEs, which capture the average behaviour of the system. Fluid approximations have been intensely studied and their limiting behaviour is well understood, providing specific error guarantees. There are two characteristics of a pCTMC which are instrumental to enabling the fluid approximation. Firstly, there is a natural interpretation of the states as points in a vector space, where each dimension represents a species. Secondly, a *drift vector field* can be naturally defined by extending the propensity function to be defined on the whole vector space, which is a polynomial function of the number of agents of each type (i.e. polynomial function of the elements of the system state vector).

Established guarantees Following Darling and Norris (2008), we examine and formalise the aspects of pCTMCs which render them especially amenable to the fluid approximation. As mentioned, the first is that pCTMC state-spaces are countable and there exists an obvious ordering. We can therefore write a trivial linear mapping from the discrete, countable state-space I to a continuous Euclidean space $\mathbf{x} : I \rightarrow \mathbb{R}^d$, where d is the number of agent types in the system.

The second aspect is that rates of transition from each state to all others (i.e. elements of the Q -matrix) can be expressed as a function of the state vector \mathbf{x} . A *drift vector* $\beta(\xi)$ can be defined as

$$\beta(\xi) = \sum_{\xi' \neq \xi} (\mathbf{x}(\xi') - \mathbf{x}(\xi)) q(\xi, \xi'),$$

for each $\xi \in I$. Since $q(\xi, \xi')$ is some parametric function of ξ, ξ' in pCTMCs (due to the indistinguishable nature of the agents) the definition of the *drift vector* can be extended over the entire Euclidean space \mathbb{R}^d to produce the *drift vector field* $b(\mathbf{x}) : U \rightarrow \mathbb{R}^d$, where $U \subseteq \mathbb{R}^d$. There is then a set of conditions given in (Darling and Norris, 2008) that must be satisfied by these elements to bind the error of the fluid approximation to the Markov process. The conditions ensure that:

The first exit time from a suitably selected domain of the Euclidean mapping of the Markov chain state-space U , converges in probability to the first exit time of the fluid limit.

Canonical embedding of pCTMCs In the canonical embedding for continuous relaxation of pCTMCs, we construct an $E \subset \mathbb{R}^d$ Euclidean space, where each dimension corresponds to the concentration of each species in the system, $i \in \{1, \dots, m\}$. The

states are then uniformly embedded in continuous space $[0, 1]^m \in E$ at intervals $1/n_i$ by $\mathbf{x}(\xi) = u_i/n_i$, where ξ represents the population $\sum_i u_i \hat{\mathbf{e}}_i$. Further, $N = |I| = \prod_i n_i$, is a scale parameter which defines $\mathbf{x}_N(\xi)$, $q_N(\xi, \xi')$ and $\beta_N(\xi)$ for any such pCTMC of size N . The motivation is that in the limit of $N \rightarrow \infty$, the distance between neighbouring states will vanish in the embedding, and jump sizes will similarly vanish, producing an approximately continuous trajectory of the system in the continuous concentration space.

In (Darling, 2002), we find how the canonical embedding above satisfies the conditions given in (Darling and Norris, 2008), and that the approximation error shrinks as the scale parameter N grows. Specifically, the authors show that there exists a fluid approximation (deterministic trajectory) to the \mathbf{x} -mapped pCTMC, whose error diminishes in N , under the conditions that:

- *initial conditions* converge, i.e. $\exists a \in U$, $a \neq \mathbf{x}(\xi_0)$ such that

$$\Pr [\|\mathbf{x}_N(\xi_0) - a\| > \delta] \leq \kappa_1(\delta)/N, \quad \forall \delta > 0;$$

- *mean dynamics* converge as $N \rightarrow \infty$, i.e. $\tilde{b} : U \rightarrow \mathbb{R}^d$ is a Lipschitz field independent of N , such that

$$\sup_{\xi} \|\beta_N(\xi) - \tilde{b}(\mathbf{x}_N(\xi))\| \rightarrow 0 \quad \text{as } N \rightarrow \infty;$$

- *noise* converges to zero as $N \rightarrow \infty$, i.e. that,

$$\begin{aligned} \sup_{\xi} \left\{ \sum_{\xi' \neq \xi} q_N(\xi, \xi') \right\} &\leq \kappa_2 N, \quad \text{and} \\ \sup_{\xi} \left\{ \|\beta_N(\xi)/q_N(\xi)\|^2 + \sum_{\xi' \neq \xi} \|\mathbf{x}_N(\xi') - \mathbf{x}_N(\xi)\|^2 q_N(\xi, \xi')/q_N(\xi) \right\} &\leq \kappa_3 N^{-2}, \end{aligned}$$

where $\kappa_1(\delta)$, κ_2 , κ_3 are positive constants, $q(\xi) = \sum_{\xi' \neq \xi} q(\xi, \xi')$, and the inequalities hold uniformly in N .

There are many ways to satisfy the above criteria, but a common one (used in pCTMCs) is “hydrodynamic scaling”, where the increments of the N -state Markov process mapped to the Euclidean space are $\mathcal{O}(N^{-1})$ and the jump rate is $\mathcal{O}(N)$.

5.2 Methodology

As discussed in the previous section, the fluid approximation of pCTMCs is critically reliant on the structure of the state-space of pCTMCs being isomorphic to a lattice in \mathbb{R}^n . It further relies on transition rates being a function of the state's position in the lattice. This enables the definition of a drift vector field, which can be then naturally extended to the whole ambient space and, under mild assumption, leads to convergence under suitable scaling. Neither of these ingredients are obviously available in the general case of CTMCs lacking a population structure.

In this section, we describe the proposed methodology for a geometric fluid approximation for CTMCs. We motivate our approach by describing an exact, if trivial, general embedding of a CTMC's state-space into a very high-dimensional space. Such an embedding however affords the non-trivial insight that suitable approximate embeddings may be obtained considering the spectral geometry of the generator matrix. This provides an unexpected link with a set of techniques from machine learning, *diffusion maps*, which embed graphs into Euclidean spaces. The geometry of diffusion maps is well studied, and their distance preservation property is particularly useful for our purpose of obtaining a fluid approximation.

Diffusion maps however provide only one ingredient to a fluid approximation; they do not define an ODE flow over the ambient Euclidean space. To do so, we use Gaussian Process regression: this provides a smooth interpolation of the dynamic field between embedded states. Smoothness guarantees that nearby states in the CTMC (which are embedded to nearby points in Euclidean space by virtue of the distance preservation property of diffusion maps) will have nearby drift values, somewhat enforcing the pCTMC property that the transition rates are a function of the state vector.

This two-step strategy provides a general approach to associate a deterministic flow on a vector space to a CTMC. We empirically validate that such flow indeed approximates the mean behaviour of the CTMCs on a range of examples in the next section. Prior to the empirical section we prove a theorem showing that, in the special case of pCTMCs of birth/ death type, a related construction to our geometric fluid approximation is consistent to the canonical fluid limit construction.

5.2.1 Eigen-embeddings of CTMCs

Trivial embedding of CTMCs Consider a CTMC with initial distribution π and generator matrix Q , on countable state-space $\Xi \subset \mathbb{N}$. The single time marginal p_t over Ξ

at time t of the process obeys the CKE:

$$\partial_t p_t = Q^\top p_t, \quad (5.1)$$

where p_t is a column vector. Given an arbitrary embedding of the states in some continuous space, $x : \Xi \rightarrow \mathbb{R}^d$, the projected mean $\langle x_t \rangle = X^\top p_t$, obeys:

$$\begin{aligned} \partial_t \langle x_t \rangle &= X^\top \partial_t p_t = X^\top Q^\top p_t \\ &= X^\top Q^\top X^{-\top} X^\top p_t \end{aligned} \quad (5.2)$$

$$= X^\top Q^\top X^{-\top} \langle x_t \rangle, \quad (5.3)$$

where X_{ij} refers to the $j \in \{1, \dots, d\}$ coordinate of state $i \in \Xi$. In general, step (5.2) is only possible for $XX^{-1} = I$, with I the $|\Xi| \times |\Xi|$ identity matrix (i.e. with $d = |\Xi|$).

We note that choosing the trivial embedding $X = I$ (i.e. each state mapped to the vertex of the probability $(|\Xi| - 1)$ -simplex), equates the fluid process to the original CKE:

$$\partial_t \langle x_t \rangle = Q^\top \langle x_t \rangle. \quad (5.4)$$

The fluid approximation For any embedding $y : \Xi \rightarrow \mathbb{R}^d$, the standard fluid approximation defines the drift at any state $y(i) \equiv y_i$, $i \in \Xi$, to be:

$$\begin{aligned} \beta(y_i) &= \sum_{j \neq i} (y_j - y_i) Q_{ij} \\ &= \sum_{j \neq i} y_j Q_{ij} - y_i \sum_{j \neq i} Q_{ij} \\ &= \sum_{j \neq i} y_j Q_{ij} + y_i Q_{ii} \\ &= [QY]_i, \end{aligned} \quad (5.5)$$

where Y is a $|\Xi| \times d$ matrix, and $\beta(y_i)$ is the i th row of QY .

In order to extend the drift over the entire space \mathbb{R}^d , we let $q(y_t, y_j) = \sum_i Q_{ij} y_i^\top y_t$ be the transition kernel between any point $y_t \in \mathbb{R}^d$ and any state y_j , $j \in \Xi$. Then we

naturally define the continuous vector field b to be

$$\begin{aligned}
 b(y_t) &= \sum_j y_j q(y_t, y_j) \\
 &= \sum_j y_j \sum_i Q_{ij} y_i^\top y_t \\
 &= Y^\top Q^\top Y y_t.
 \end{aligned} \tag{5.6}$$

Trivially, $Y = I$ yields the original CKE as shown above,

$$\partial_t y_t = b(y_t) = Y^\top Q^\top Y y_t = Q^\top y_t. \tag{5.7}$$

If embedded states U are eigenvectors of $Q = U \Lambda U^{-1}$, then the mean of the mapped process $\langle u_t \rangle$ is given by

$$\begin{aligned}
 \langle u_t \rangle &= U^\top p_t \\
 &= U^\top U^{-\top} e^{t\Lambda} U^\top \pi \\
 &= e^{t\Lambda} U^\top \pi,
 \end{aligned} \tag{5.8}$$

where $\pi = p_0$, the initial state distribution. Similarly, in the fluid approximation with $q(y_t, y_j) = \sum_i Q_{ij} y_i^\top y_t$, and $y_0 = U^\top \pi$, one has

$$\begin{aligned}
 y_t &= e^{t\Lambda U^\top U} y_0 \\
 &= e^{t\Lambda U^\top U} U^\top \pi.
 \end{aligned} \tag{5.9}$$

We can therefore claim the following: *in the case of $U^\top U = I$, or for a symmetric $Q = U \Lambda U^\top$, the fluid approximation process is exactly equivalent to the projected mean.* This suggests that an approximate, low dimensional representation might be obtained by truncating the spectral expansion of the generator matrix of the CTMC. Spectral analysis of a transport operator is also the approach taken by *diffusion maps*, a method which is part of the burgeoning field of manifold learning for finding low-dimensional representations of high-dimensional data.

Markov chain as a random walk on a graph Another avenue to reach the same conclusion is to consider the CTMC as a random walk on an abstract graph, where vertices represent states of the chain, and weighted directed edges represent possible transitions. From this perspective, it is natural to seek an embedding of the graph in a suitable low-dimensional vector space; it is well known in the machine learning community

that an optimal (in a sense specified in Section 5.2.2) embedding can be obtained by spectral analysis of the transport operator linked to the random walk on the graph. We expect that if the graph geometry is a discrete approximation of some continuous space, the latter will serve well as a continuous state-space for a fluid limit approximation, when endowed with an appropriate drift vector field to capture the non-geometric dynamics in the graph.

5.2.2 Diffusion maps

A natural method to embed the CTMC states in continuous space for our purposes is *diffusion maps* (Coifman et al., 2008; Coifman and Lafon, 2006; Coifman et al., 2005; Nadler et al., 2006a,b). This is a manifold learning method, where the authors consider a network defined by a symmetric adjacency matrix, with the goal of finding coordinates for the network vertices on a continuous manifold (as is usually the case with similarities of points in high-dimensional spaces).

Diffusion on a manifold The method follows from taking the normalised similarities to be the transition kernel of a diffusion process, evolving on a hidden manifold $\mathcal{M} \subset \mathbb{R}^p$ where the network vertices lie and with a smooth boundary $\partial\mathcal{M}$. Resting on this, a family of diffusion operators are constructed which can be spectrally analysed to yield coordinates for each vertex on the manifold. The continuous operators which are theoretically constructed are assumed to be approximated by the analogous discrete operators which are constructed from data. The method can be thought to optimally preserve the normalised *diffusion distance* of the diffusion process on the high-dimensional manifold, as Euclidean distance in the embedding. *Diffusion distance* between two vertices $\mathbf{x}_0, \mathbf{x}_1$ at time t is defined to be the distance between the probability densities over the state-space, each initialised at $\mathbf{x}_0, \mathbf{x}_1$ respectively, and after a time t has passed:

$$D_t^2(\mathbf{x}_0, \mathbf{x}_1) = \|p(\mathbf{x}, t|\mathbf{x}_0) - p(\mathbf{x}, t|\mathbf{x}_1)\|_{L_2(\mathcal{M}, w)}^2,$$

where $L_2(\mathcal{M}, w)$ is a Hilbert space in which the distance is defined, with $w(\mathbf{x}) = 1/\phi_0(\mathbf{x})$, the inverse of the steady-state distribution $\phi_0(\mathbf{x}) = \lim_{t \rightarrow \infty} p(\mathbf{x}, t|\mathbf{x}_0, 0)$.

Diffusion with drift for asymmetric networks The methodology of diffusion maps has been extended in (Perrault-Joncas and Meilă, 2011) to deal with learning manifold embeddings for directed weighted networks. Given an asymmetric adjacency matrix, the symmetric part is extracted and serves as a discrete approximation to a geometric

operator on the manifold. Spectral analysis of the relevant matrix can then yield embedding coordinates for the nodes of the network. In the same manner as for the original formulation of diffusion maps a set of backward evolution operators are derived, the two relevant ones being:

$$- \partial_t = \mathcal{H}_{aa}^{(\alpha)} = \Delta + (\mathbf{r} - 2(1 - \alpha)\nabla U) \cdot \nabla, \quad (5.10)$$

$$\text{and} \quad - \partial_t = \mathcal{H}_{ss}^{(\alpha)} = \Delta - 2(1 - \alpha)\nabla U \cdot \nabla. \quad (5.11)$$

The operators are parametrised by α , which determines how affected the diffusion process on the manifold is by a sampling potential, U . Choosing $\alpha = 1$ allows us to spectrally analyse a discrete approximation to the Laplace-Beltrami operator $\Delta = \mathcal{H}_{ss}^{(\alpha=1)}$, separating it from the density dependent term $-2(1 - \alpha)\nabla U \cdot \nabla$ in the diffusion operator $\mathcal{H}_{ss}^{(\alpha)}$.

Diffusion maps for CTMCs For an arbitrary CTMC(π, Q), we regard $Q \in \mathbb{R}^{N \times N}$ to be a discrete approximation of the operator $\mathcal{H}_{aa}^{(\alpha)}$. However, it is unclear how one can extract the geometrically relevant component Δ under a hidden potential U and parameter α . In practice, therefore, we assume a uniform measure on the manifold, i.e. constant U , which renders Q a discrete approximation of $\mathcal{H}_{aa} = \Delta + \mathbf{r} \cdot \nabla$ (the choice of α no longer matters); further, we take the sampling transition kernel corresponding to this operator to be composed of a symmetric and anti-symmetric part (without loss of generality) which renders $\lim_{N \rightarrow \infty} (Q + Q^T)/2 = \Delta$. This contains the relevant geometric information about the network, with the first $k + 1$ eigenvectors of the operator used as embedding coordinates in a k -dimensional Euclidean space (ignoring the first eigenvector which is trivial by construction). A detailed exposition of the method as it relates to our purposes of embedding a Markov chain network can be found in Appendix B.1.

It should be noted that, while diffusion maps have been used to construct low-dimensional approximations of high-dimensional SDEs (Coifman et al., 2008), and to embed a discrete-time Markov chain in continuous space with an accompanying *advective field* (Perrault-Joncas and Meilă, 2011), doing the same for a continuous-time Markov chain has not been attempted. Distinctively, the focus of that work was not to clear a path between discrete and continuous state Markov processes, but rather the low-dimensional embedding of processes or sample points. In terms of the convenient table presented in (Nadler et al., 2006b) and restated here in Table 5.1, we seek to examine the omitted entry that completes the set of Markov models; this is the third entry added here to the original table, taking $N < \infty$ and the limit $\epsilon \rightarrow 0$ to be the case of a CTMC with finite generator matrix Q .

Table 5.1 Resulting random walk (RW) or processes from the limiting cases of number of vertices N and time step parameter ϵ in the diffusion maps literature (Nadler et al., 2006b). We highlight the addition of the third entry for CTMCs to complete the set.

| Case | Operator | Stochastic Process |
|--|---|---|
| $\epsilon > 0$ $N < \infty$ | finite $N \times N$ matrix P | RW in discrete space discrete in time (DTMC) |
| $\epsilon > 0$ $N \rightarrow \infty$ | operators T_f, T_b | RW in continuous space discrete in time |
| $\epsilon \rightarrow 0$ $N < \infty$ | infinitesimal generator matrix $Q \in \mathbb{R}^{N \times N}$ | Markov jump process; discrete in space, continuous in time |
| $\epsilon \rightarrow 0$ $N \rightarrow \infty$ | infinitesimal generator \mathcal{H}_f | diffusion process continuous in space & time |

5.2.3 Gaussian processes for inferring the drift vector field

Diffusion maps provide a convenient way to embed the CTMC graph into a Euclidean space E ; however, the push-forward CTMC dynamics is only defined on the image of the embedding, i.e. where the embedded states are. In order to define a fluid approximation, we require a continuous drift vector field to be defined everywhere in E . A natural approach is to treat this extension problem as a regression problem, where we use the push-forward dynamics at the isolated state embeddings as observations. We therefore use Gaussian processes (GPs) to infer a smooth function $b : E \rightarrow \mathbb{R}^d$ that has the appropriate drift vectors where states lie.

Gaussian process regression Suppose we observe evaluations $\mathbf{f} = (f(t_1), \dots, f(t_n))$ of the (otherwise hidden) drift vector field, at points $\mathbf{t} = (t_1, \dots, t_n)$ of the continuous state-space. Given the diffusion maps embedding $\mathbf{x} : \Xi \rightarrow U \subset \mathbb{R}^d$, we construct the drift vectors

$$\beta(\xi) = \sum_{\xi' \neq \xi} (\mathbf{x}(\xi') - \mathbf{x}(\xi)) q(\xi, \xi'),$$

$\forall \xi \in \Xi$. We then take $\mathbf{x}(\xi_i) \equiv t_i$ and $\beta(\xi_i) \equiv b(\mathbf{x}(\xi_i)) \equiv f(t_i)$ to be observations of the drift vector field $b(\mathbf{x}) : U \rightarrow \mathbb{R}^d$ which is to be inferred over the whole continuous space.

By imposing an appropriate prior distribution over a family of vector fields, we are able to perform Bayesian inference to obtain a posterior distribution over that family of fields, consistent with our observations. The problem is addressed via Gaussian process regression, where the prior is the distribution given by the kernel. Prediction of the function value f_* at an unobserved domain point t_* , conditioning on observations \mathbf{f} at

points \mathbf{t} , is given in terms of the distribution:

$$f_\star | \mathbf{f}, \mathbf{t} \sim \int p(f_\star | t_\star, \mathbf{t}, \mathbf{f}) p(\mathbf{f} | \mathbf{t}) d\mathbf{f}.$$

Since the integral involves only normal distributions, it is tractable and has a closed form solution, which is again a normal distribution. The observations may also be regarded as noisy, which will allow the function to deviate from the observed value in order to avoid extreme fluctuations. Use of an appropriate noise model (Gaussian noise), retains the tractability and normality properties. The mean of the predictive distribution is used as a point estimate of the function value, as customary. This is the classical Gaussian process regression setting introduced in Chapter 2, Section 2.3.4; we refer the reader to (Rasmussen and Williams, 2006) for a comprehensive account of the theory and implementation details.

In our case, the choice of kernel and its hyperparameters is critical, especially when the density of states is low. In the limit of infinite observations of the function, the Gaussian process will converge to the true function over T , if the function is in the space defined by the kernel, regardless of the hyperparameters chosen (Rasmussen and Williams, 2006). However, the number of states we embed is finite and so the choice of an appropriate prior can greatly aid the Gaussian process in inferring a good drift vector field. Here, we use the standard squared exponential kernel with a different lengthscale for each dimension, and select hyperparameters which optimise the likelihood of the observations². The optimisation is performed via gradient descent since the gradient for the marginal likelihood is available.

5.2.4 Consistency result

The geometric fluid approximation scheme is applicable in general to all CTMCs; it is therefore natural to ask whether it reduces to the standard fluid approximation on pCTMCs. We have the following result for a related construction, the *unweighted Laplacian fluid approximation*.

Theorem 5.2.1. *Let \mathcal{C} be a pCTMC, whose underlying transition graph is a multi-dimensional grid graph. The unweighted Laplacian fluid approximation of \mathcal{C} coincides with the canonical fluid approximation in the hydrodynamic scaling limit.*

The proof (see Appendix B.1.2) relies on the explicit computation of the spectral decomposition of the Laplacian operator of an unweighted grid graph (Kłopotek, 2017),

²This procedure is in the family of point-approximations known as *empirical Bayes methods*.

and appeals to the universal approximation property of Gaussian Processes (Rasmussen and Williams, 2006). We conjecture that the conditions for fluid approximation for such a pCTMC will also be satisfied by our *geometric fluid approximation*.

Intuitively, away from the boundaries of the network, the coordinates of the embedded states approach the classical concentration embedding, where each dimension corresponds to a measure of concentration for each species. As the network grows (i.e. allowing larger maximum species numbers in the state-space of the chain) states are mapped closer together, reducing jump size, but preserving the ordering. The spacing of states near the centre of the population size is almost regular, approaching the classical density embedding, and the GP smoothing will therefore converge to the classical extended drift field.

5.3 Empirical observations

Experimental evidence of our geometric fluid approximation is necessary to give an indication of the method's validity, and a better intuition for its domain of effectiveness. We apply the geometric fluid approximation to a range of CTMCs with differing structure, and present the experimental results in this section. The CTMC models we used are defined in Section 5.3.1.

There is no absolute way to assess whether the method produces a good approximation to the true probability density evolution; we therefore focus on two comparisons: how close the geometric fluid trajectory over time is to the empirical mean of the original CTMC, mapped on the same state-space (Section 5.3.2); and how close the first-passage time (FPT) estimate from the fluid approximation is to the true FPT cumulative density function (estimated by computationally intensive Monte Carlo sampling; Section 5.3.4).

Further, we demonstrate in Section 5.3.3 how the method is applicable to a subset of the CTMC graph, such that only a connected region of the state-space is embedded. This may result in fluid approximations for graphs whose global structure is not particularly amenable to embedding in a low-dimensional Euclidean space, and so is useful for gauging the behavioural characteristics of the system near a section of the state-space.

In all figures in this section, red lines are solutions of our geometric fluid approximation, obtained via numerical integration of the drift vector field as inferred by GP regression, and blue lines are the mean of CTMC trajectories mapped to the embedding space, which were obtained via Gillespie's exact stochastic simulation algorithm (SSA) (Gillespie, 1977). Finally, in figures showing trajectories on the diffusion maps (DM) manifold, grey

line intersections are embedded states (the grey lines being the possible transitions, or edges of the network).

5.3.1 Models

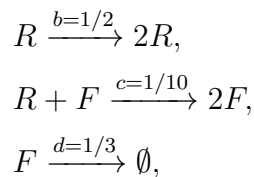
We examine an array of models to assess the applicability domain of our method. The models are defined below and empirical comparisons for each are presented throughout this section.

Two species birth-death processes This model describes two independent birth-death processes for two species, and serves as a basic sanity check. The CTMC graph has a 2D grid structure and in this sense resembles the system in Theorem 5.2.1. In the usual chemical reaction network notation, we write:



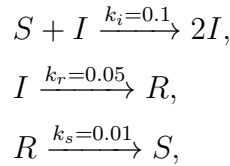
for the two species A, B , and note that there is a system size variable $N = 30$ such that the count for each species n_A, n_B cannot exceed N ; this produces a finite-state CTMC that can be spectrally decomposed and embedded. Note further that the birth process involves no particles here, and so transitioning from state $s = (n_A, n_B)$ to state $s' = (n_A + 1, n_B)$ (or from $s = (n_A, n_B)$ to $s' = (n_A, n_B + 1)$) occurs at the same rate of $10/N$ per second $\forall n_A, n_B$. Conversely, death processes are uni-molecular reactions, such that transitioning from $s = (n_A, n_B)$ to $s' = (n_A - 1, n_B)$ occurs at a rate of $(1/2)n_A$ per second $\forall n_A, n_B$, as the chemical reaction network interpretation dictates.

Two species Lotka-Volterra model This is a Lotka-Volterra model of a predator-prey system. Allowed interactions are prey birth, predators consuming prey and reproducing, and predator death. The interactions with associated reaction rates are defined below in the usual chemical reaction network notation:



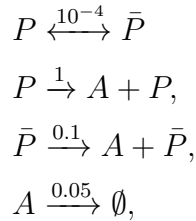
where prey is represented by species R (rabbits) and predators by species F (foxes), with maximum predator and prey numbers of $N = 30$.

SIRS model We describe a widely used stochastic model of disease spread in a fixed population, wherein agents can be in three states: susceptible, infected, and recovered (S , I , R) and a contagious disease spreads from infected individuals to susceptible ones. After some time, infected individuals recover and are immune to the disease, before losing the immunity and re-entering the susceptible state. We define a pCTMC for the process as follows:



where the constants (k_i , k_r , k_s) have been chosen such that the ODE steady state is reached some time after $t = 100$ s. The state of the pCTMC at time t is $X(t) = (S(t), I(t), R(t))$, where $S(t)$ refers to the number of agents in state S at time t , and so on for all species.

Genetic switch model This is a popular model for the expression of a gene, when the latter switches between two activation modes: *active* and *inactive* (Larsson et al., 2019; Vu et al., 2016). While active, the gene is transcribed into mRNA at a much faster rate than while inactive (factor of ~ 10). The gene switches between the two modes stochastically with a slow rate. We have the following reactions:



where species A represents the mRNA, and the active and inactive modes of the gene are represented by species P and \bar{P} respectively, with a maximum count of 1. Despite being able to express this model in the usual chemical reaction network language, we emphasize that the binary nature of the switch prohibits usual scaling arguments for reaching the fluid limit.

5.3.2 Assessing fluid solution and mean trajectory in embedding space

In our geometric fluid approximation, we create a map using *directed diffusion maps* to embed the CTMC states into a Euclidean space of small dimensionality, and use Gaussian process regression to infer a drift vector field over the space. The resulting continuous trajectories, which we refer to as the *geometric fluid approximation*, are in this section compared to average trajectories of the CTMC systems, projected on the same space. The latter are obtained by drawing 1000 trajectories of the CTMC using the SSA algorithm, and taking a weighted average of the state positions in the embedding space.

Our geometric fluid approximation does well for pCTMC models, where we know that the state-space can be naturally embedded in a Euclidean manifold. This is especially true for systems like the independent birth-death processes of two species, which do not involve heavy asymmetries in the graph structure. The more the structure deviates from a pCTMC and the more asymmetries in the structure, the larger the deviations we expect from the mean SSA trajectory. Additionally, we expect large deviations in the case of bi-modal distributions over the state-space, as is the general case for fluid approximations. This is because the latter are point-mass approximations of a distribution, and so are naturally more suited to approximate uni-modal, concentrated densities.

Two species birth-death processes As a sanity check, we examine how our method approximates the mean trajectory of the trivial system of two independent birth-death processes described above. The true distribution for such a system is uni-modal in the usual concentration space, and the graph has the structure of a 2D grid lattice. As shown in Figure 5.1, the geometric fluid approximation is very close to the empirical mean trajectory, which supports our consistency theorem.

Lotka-Volterra model We perform our geometric fluid approximation for the non-trivial case of a Lotka-Volterra system, which models a closed predator-prey system as described above. The asymmetric consumption reaction distorts the grid structure representative of the Euclidean square two species space. Therefore, the manifold recovered is the Euclidean square with shrinkage along the consumption dimension — more shrinkage is observed where predators and prey numbers are higher, since this implies faster consumption reactions. We observe in Figure 5.2 that the fluid estimate keeps close to the mean initially and slowly diverges; however, the qualitative characteristics of the trajectory remain similar.

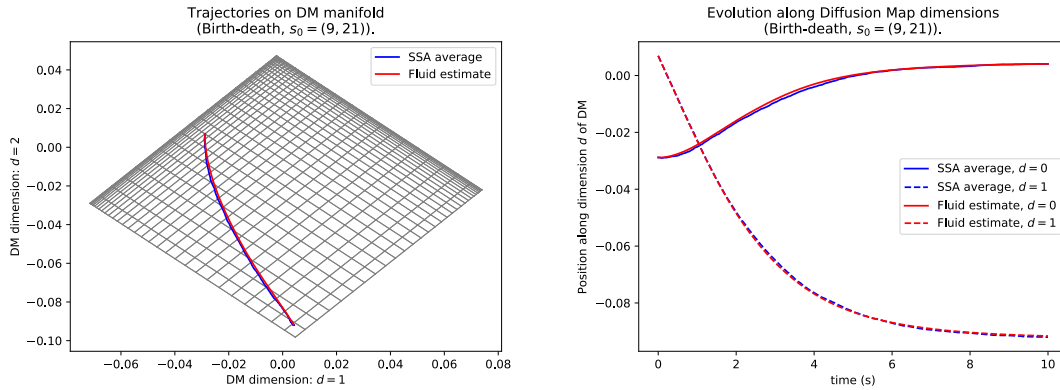


Figure 5.1 Independent birth-death process for two species, showing the fluid solution (red) and the projected mean evolution (blue). Left: embedded state-space and trajectories in \mathbb{R}^2 , where grid structure is preserved and species counts are in orthogonal directions. Right: fluid and mean SSA trajectories along embedded dimensions over time.

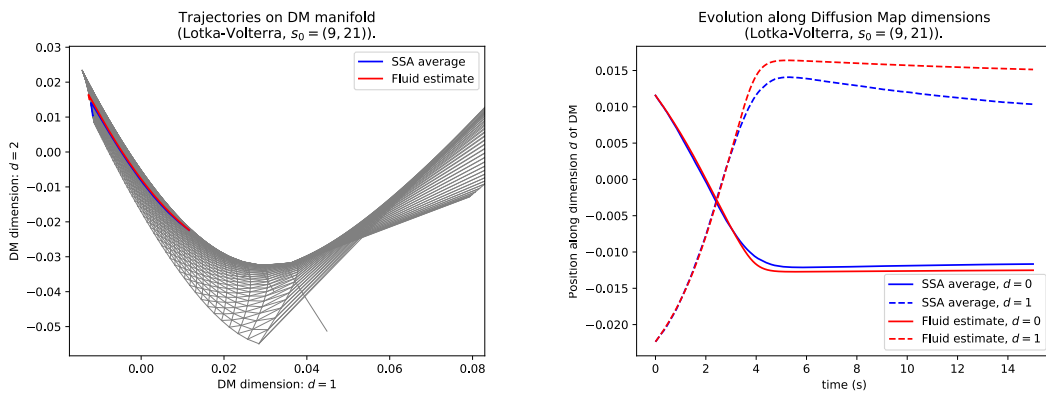


Figure 5.2 A two species Lotka-Volterra model, showing the fluid solution (red) and the projected mean evolution (blue) slowly diverging from each other. The qualitative behaviour of both is similar as they begin to perform the oscillations typical of this system.

SIRS model The SIRS model gives us the opportunity to compare trajectories in the embedding space of the geometric fluid, with trajectories in the concentration space used by the standard fluid approximation. We observe in Figure 5.3 good agreement with the empirical mean trajectory for both fluid methods.

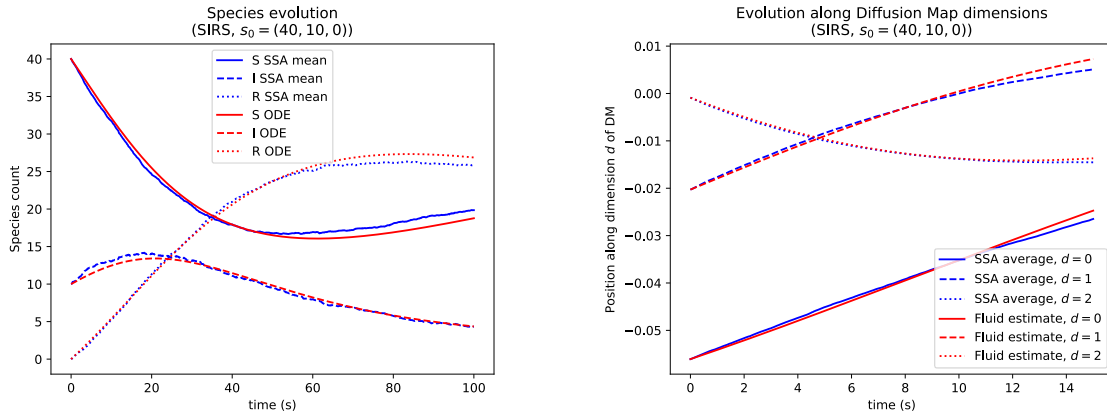
The classical fluid trajectory (Figure 5.3a) is attainable in terms of the concentration of each species; it evolves according to coupled ODEs:

$$\frac{ds}{dt} = k_s r(t) - \frac{k_i}{N} i(t) s(t), \quad (5.12)$$

$$\frac{di}{dt} = \frac{k_i}{N} i(t) s(t) - k_r i(t), \quad (5.13)$$

$$\frac{dr}{dt} = k_r i(t) - k_s r(t), \quad (5.14)$$

where $x(t) = (s(t), i(t), r(t)) = (S(t), I(t), R(t))/N$, and $N \in \mathbb{N}_{>0}$ is the total population. Increasing N linearly scales the ODE solution without affecting the dynamics; the SSA average converges to the ODE solution as $N \rightarrow \infty$. Similarly, Figure 5.3b shows the fluid solution in \mathbb{R}^3 obtained by our geometric fluid approximation.



(a) Trajectories of the SIRS model in the space of species counts, each line tracks the count of a species in the system. The classical ODE solution (red) for the three species closely follows the simulation average trajectory (blue).

(b) Trajectories of the SIRS model in \mathbb{R}^3 , where each line tracks the evolution along a dimension d of the DM embedding. The fluid solution (red) closely follows the simulation average trajectory (blue), as in 5.3a. Note that these dimensions are no longer interpretable as counts of each species.

Figure 5.3 Trajectories of the SIRS model with states embedded in continuous space $\subset \mathbb{R}^3$: 5.3a the classical embedding to concentration space; 5.3b our embedding with diffusion maps and Gaussian process regression for estimating the drift.

Genetic switch model The model of a genetic switch is a departure from the usual pCTMC structure, since the binary switch introduces very slow mixing between two birth-death processes each with a different fixed point. The bi-modality of the resulting steady-state distribution is problematic to capture for any point-mass trajectory, and quickly leads to divergence of the fluid trajectory from the mean. With the particularly slow switching rate of 10^{-4}s^{-1} , our method produces fluid trajectories close to the mean trajectory for up to 100s, mostly because the mixing is very slow and the distribution remains relatively concentrated for a long time (Figure 5.4). However, with the faster rate of $5 \cdot 10^{-3}\text{s}^{-1}$, our fluid approximation quickly diverges from the mean trajectory (Figure 5.5), as the expected result of faster mixing.

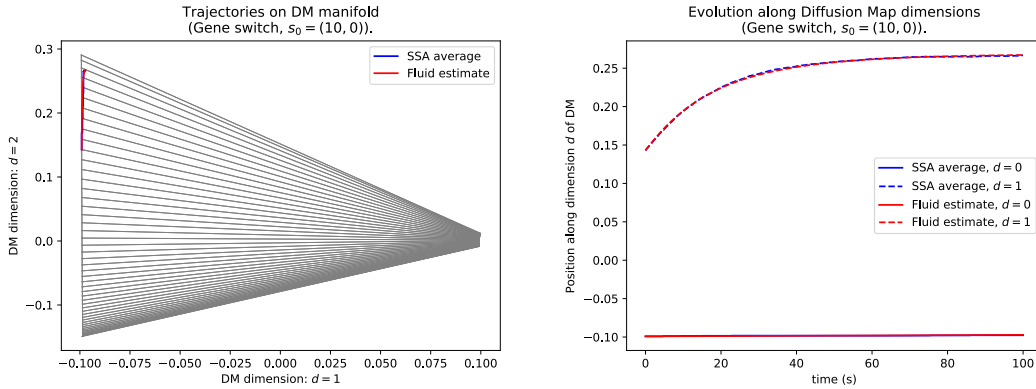


Figure 5.4 The genetic switch model with switching rate 10^{-4}s^{-1} , showing the fluid solution (red) and the projected mean evolution (blue) keeping close to each other. Transitions from the set of states at $d_1 = -0.1$ (inactive mode) to the set of states at $d_1 = 0.1$ happen very rarely, which is reflected by the mean SSA trajectory.

pCTMC perturbations It is expected that the method will perform well for CTMCs that are in some sense similar to a pCTMC, but cannot be exactly described by a chemical reaction network. We therefore demonstrate how the method performs for perturbations of a Lotka-Volterra system. To achieve the perturbation, we add noise to every existing transition rate (non-zero element of Q) of the Lotka-Volterra system we had above. The perturbed transition matrix Q_{per} is described in terms of the Lotka-Volterra matrix Q_{LV} by

$$[Q_{\text{per}}]_{ij} = \begin{cases} [Q_{LV}]_{ij} + |\eta_{ij}|, & \text{if } [Q_{LV}]_{ij} > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (5.15)$$

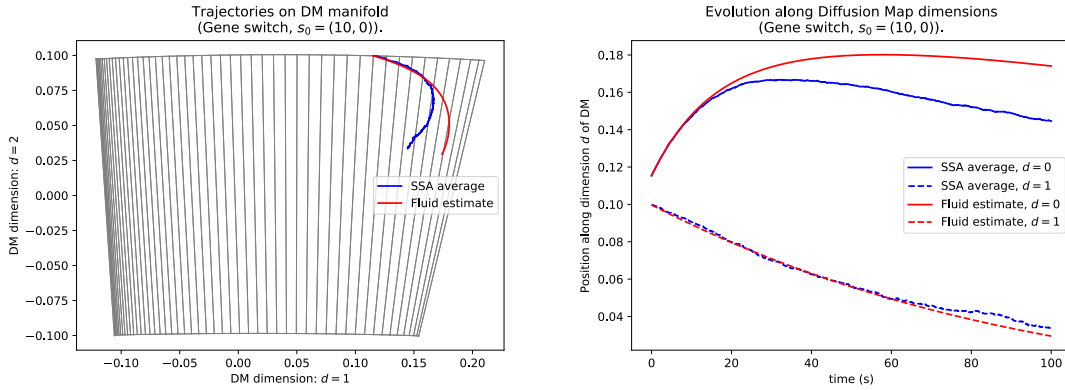


Figure 5.5 The genetic switch model with a faster switching rate ($5 \cdot 10^{-3} \text{s}^{-1}$), showing how the fluid solution (red) diverges from the projected mean evolution (blue) after $t \approx 20$ s; the qualitative aspects of the trajectory remain similar.

for all $i \neq j$, where $\eta_{ij} \sim \mathcal{N}(0, 0.5^2)$, and $[Q_{\text{per}}]_{ii} = \sum_j [Q_{\text{per}}]_{ij}$ as usual. The projection in Figure 5.6 shows that our method performs reasonably well near the pCTMC regime, where no classical continuous state-space approximation method exists.

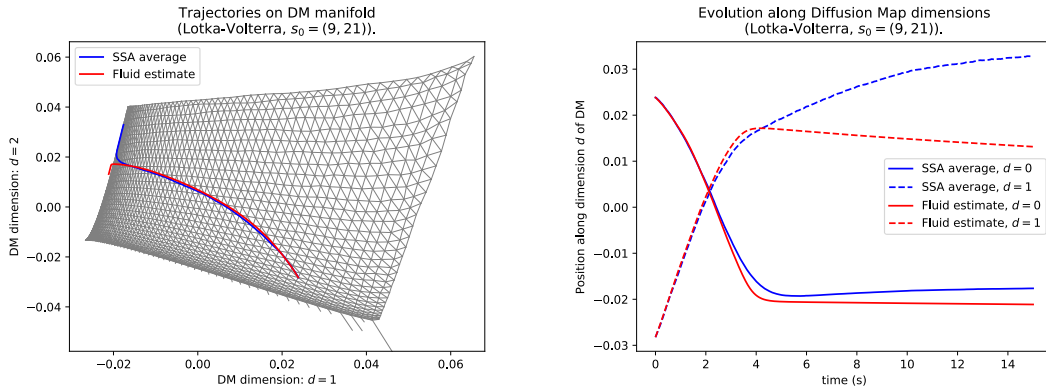


Figure 5.6 The two species Lotka-Volterra model, with noise added on all transition rates. This is a perturbed pCTMC that is not amenable to classical continuous approximation methods. The fluid solution (red) is close to the projected mean trajectory (blue) away from the boundary.

A different kind of perturbation is achieved by randomly removing possible transitions of the original pCTMC. This amounts to setting some off-diagonal elements of the Q matrix to 0, and re-adjusting the diagonal so that all rows sum to 0. In order to avoid creating absorbing states or isolated states, we remove transitions randomly with a probability of 0.1. Our method performs reasonably under both kinds of perturbations, as seen in Figure 5.7.

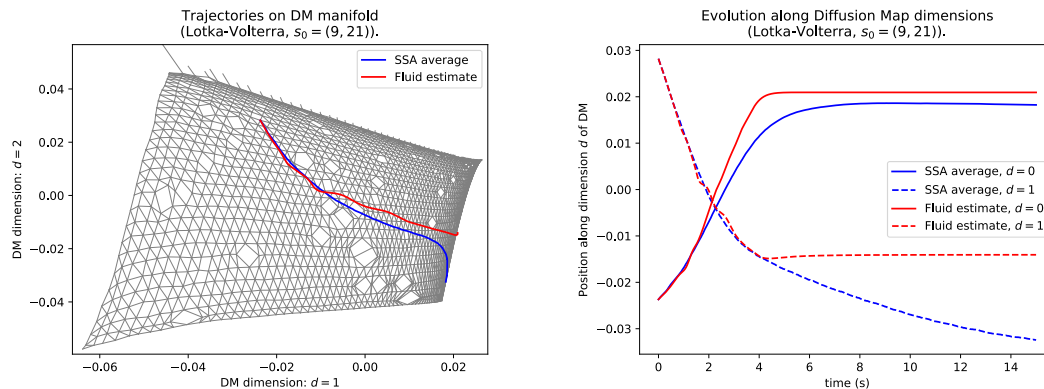


Figure 5.7 A two species Lotka-Volterra model, perturbed by both noisy transition rates and random removal of transitions. The fluid solution (red) remains similar to the projected mean trajectory (blue) away from the boundary.

5.3.3 Embedding a subset of the system

The empirical success of the method on perturbed pCTMC systems encouraged further exploration in cases where there is no global continuous approximation method, but the CTMC graph has regions which resemble a pCTMC structure, or are otherwise suitable for embedding in a continuous space. Consequently, we sought to embed only a subset of the state-space of a CTMC. Embedding state-space subsets can be useful for CTMCs that have a particularly disordered global structure (e.g. require many dimensions, or have areas on the manifold with low density), but which may contain a (connected) region of the state-space that better admits a natural embedding. Additionally, one could introduce coffin states near the boundary of a pCTMC to apply the method on reachability problems.

A subset includes every reachable state within r transitions from a selected *root state* s_r , denoted as $\Delta(s_r, r)$. Transitions from or to states outside the selected subset are ignored, and the remaining Q matrix is embedded in \mathbb{R}^2 . The drift vectors on boundary states lack all components of transitions outside the subset, and so the probability flux is inaccurate on the boundary. Figure 5.8 shows the Lotka-Volterra model subset $\Delta(s_r = (R = 5, F = 9), r = 8)$, embedded in \mathbb{R}^2 . We can see that the behaviour near the root state is close to the projected sample mean evolution, despite the boundary issues.

5.3.4 First passage times

Another common quest of such approximation techniques is estimating the *first passage time* (FPT) distribution for a target subset of states of the Markov chain. Literature on

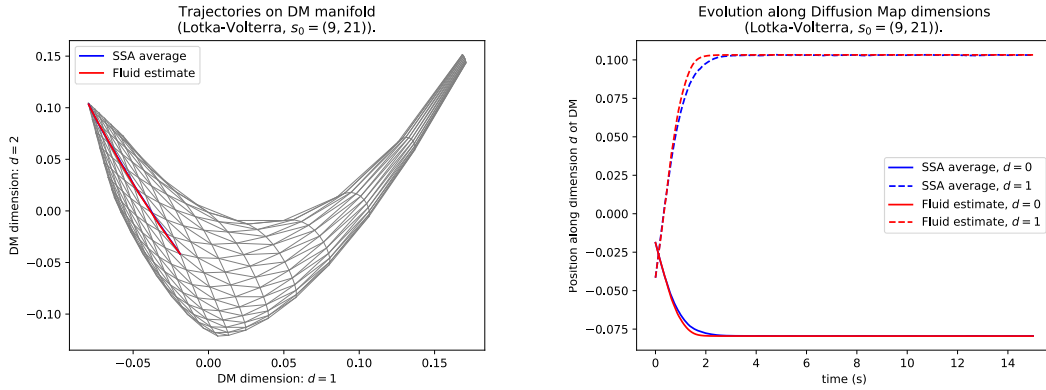


Figure 5.8 Embedding the subset $\Delta(s_r = (R = 5, F = 9), r = 8)$ of the two species Lotka-Volterra model. The fluid solution (red) remains similar to the projected mean trajectory (blue) away from the boundary, despite the boundary inaccuracies of the probability flux.

this is rich — there has been significant effort in this direction, utilising both established probability evolution methods and constructing new theoretical methods tailored to this problem (Darling and Siegert, 1953; Hayden et al., 2012; Schnoerr et al., 2017a). The former is possible since FPT estimation can be formulated as the classical problem of estimating how the probability distribution over the state-space evolves for a modified version of the Markov chain in question.

Specifically, consider a Markov chain with rate matrix Q for the state-space I . Let $B \subseteq I$ be a set of target states for which we want to estimate the distribution for the FPT τ , given some initial state $\xi_0 \in I \setminus B$. The FPT cumulative density function (cdf) is equivalent to the probability mass on the set B at time τ , if every state in B is made absorbing. In this manner, many methods for approximating probability density evolution over the state-space of a CTMC can also be used to approximate FPT distributions.

The fluid proximity approach A natural avenue to estimate the FPT when a fluid approximation to the CTMC exists, is to consider how close the fluid solution is to the target set B . The classical fluid approximation usually relies on population structured CTMCs, where the target set is often a result of some population ratio threshold (e.g. all states where more than 30% of the total population is of species A : $N_A/N > 0.3$). Since the set is defined in terms of population ratios, it is trivial to map threshold ratios to the continuous concentration space where the pCTMC is embedded, and hence define corresponding concentration regions. The time at which the fluid ODE solution enters that region of concentration space is then an approximation for the FPT cdf. The latter will of course be a step function (from 0 to 1) since the solution is the trajectory of

a point mass. Keeping the same threshold ratios for the target set, and scaling the population size N should drive the true FPT cdf towards the fluid approximation. If more moments of the probability distribution are approximated (for instance in moment closure methods) one can derive bounds for the FPT cdf; these can be made tighter as higher order moments are considered, as shown in (Hayden et al., 2012).

In our case, the fluid ODE solution only tracks the first moment of the distribution which implies a point mass approximation. Additionally, we have done away with the population structure requirement, such that thresholds for defining target sets are no longer trivially projected to the continuous space where we embed the chain. The latter challenge is overcome by considering the Voronoi tessellation of the continuous space, where each embedded state serves as the *seed* for a Voronoi cell. We then say that the fluid solution has entered the target region if it has entered a cell whose seed state belongs in the target set B . Equivalently, the solution is in the target region when it is closer (with Euclidean distance as the metric) to any target state than to any non-target state.

Checking which is the closest state is computationally cheap, and so we can produce FPT estimates at little further cost from the fluid construction. Results for the SIRS model, the Lotka-Volterra and perturbed Lotka-Volterra models follow.

FPT in the SIRS model We define a set of *barrier states* in the SIRS model, $B = \{(S, I, R) \mid R/N \geq 1/10\}$, and examine the FPT distribution of the system into the set B , with initial state $X(0) \notin B$. Note that the trivial scaling laws for this model, owing to the fixed population size, makes it simple to identify corresponding barrier regions in concentration space: $b = \{(s, i, r) \mid r \geq 1/10\}$. We can therefore compare the fluid solution FPT estimate to the empirical CDF (trajectories drawn by the SSA), as well as to our own fluid construction with an embedding given by diffusion maps and a drift vector field estimated via a Gaussian process. Figure 5.9 shows that our approach is in good agreement with both the empirical mean FPT and the classical fluid result.

FPT in the Lotka-Volterra model Here we embed the Lotka-Volterra model, and define the barrier set of states $B = \{(R, F) \mid 0.6N > F \geq 0.2N\}$ for which we estimate FPT cdfs, with initial state $X(0) = (0.3, 0.7)N$, for various system sizes $N = \{30, 40, 50\}$.

We show in Figure 5.10 (left) how our fluid construction estimates an FPT close to the SSA cdf. This is expected when embedding a structured model such as the Lotka-Volterra, where two dimensions are adequate to preserve the network topology and the Gaussian process can well approximate the continuous drift vector field. Finally,

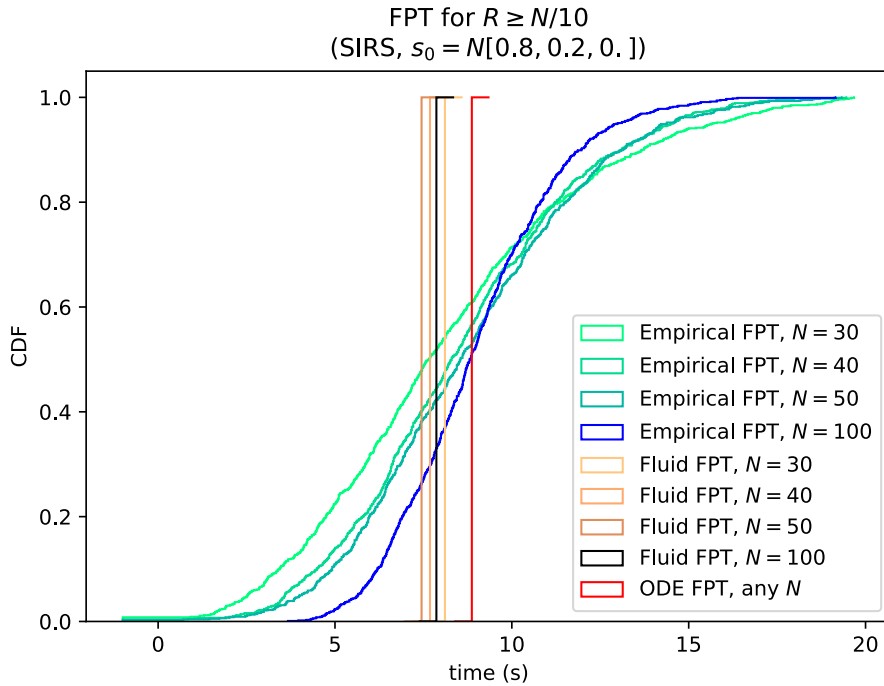


Figure 5.9 First passage time cdfs for the SIRS model with different populations. The classical solution gives the same estimate for all N , to which the SSA estimates converge as $N \rightarrow \infty$. Naturally, both the classical and our estimates are single step functions, since we approximate the probability distribution evolution by a point mass. We are consistently close to both the SSA and classical fluid CDFs.

we show in Figure 5.10 (right) that a good estimate of the FPT is recovered for the perturbed Lotka-Volterra, which is no longer a chemical reaction network.

5.4 Conclusion

CTMCs retain a central role as models of stochastic behaviour across a number of scientific and engineering disciplines. For pCTMCs, model approximation techniques such as fluid approximations have played a central role in enabling scalable analysis of such methods. These approximations, however, critically rely on structural features of pCTMCs which are not shared by general CTMCs. In this chapter, we presented a novel construction based on machine learning which extends fluid approximation techniques to general CTMCs. Our new construction, the *geometric fluid approximation*, is (with certain hyperparameters) equivalent to classical fluid approximations for a class of pCTMCs; empirically, the geometric fluid approximation provides good quality approximations in a number of non-trivial case studies from epidemiology, ecology and systems biology.

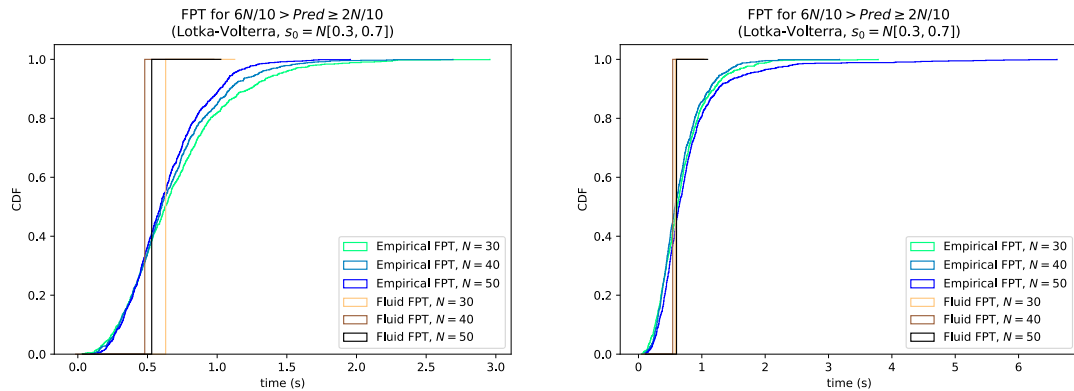


Figure 5.10 First passage time cdfs for the Lotka-Volterra model, with $B = \{(R, F) \mid 0.6N > F \geq 0.2N\}$. Left: unperturbed LV model; the fluid CDF step function crosses the SSA cdf at $\sim 0.4 - 0.5$, which is a reasonable estimate for a point mass approximation. Right: LV model perturbed by both noisy transition rates and random removal of transitions; the fluid CDF estimate is consistently close to the SSA CDF as the system size N increases.

Some potential paths forward become apparent under the lens of this work. Firstly, our method might be optimised to accommodate particular structures of CTMCs, for example by designing specific kernels for the GP regression part. This might be an effective way to incorporate domain knowledge and further improve the quality of the geometric approximation.

Secondly, we can extend this methodology by approximating the diffusion matrix field as well as the drift vector field. This would enable us to define a diffusion process on the manifold and so construct an approximating pdf rather than a point mass. An evolving pdf will be comparable to solutions produced by Van Kampen's *system size expansion*, *moment closure* methods, and the chemical Langevin equation for the case of CTMCs representing chemical reaction networks.

Finally, we have shown how the geometric fluid approximation can be used for estimating first passage times. In general, it would be interesting to extend this component to define methodologies to approximate more complex path properties, such as temporal logic formulae which are often encountered in computer science applications (Bortolussi et al., 2016; Milios et al., 2018).

Chapter 6

Conclusions and future directions

Science is an ongoing endeavour to construct logically cohesive and concise interpretations for observations of the world. To this end, scientific models which predict these phenomena act as the necessary interpretations, providing insight and enabling behavioural control. As such, Markovian processes are the scientific models *par excellence*, since they have the desirable property of encoding all information necessary to predict the future in the current state, which is often what is observed of the world.

In the quest of ever more accurate mechanistic models, the *reductionist* motif of examining the elements that constitute a system has been exceedingly successful. That is mostly because these smaller elements, when isolated, exhibit behaviour which admits a minimal mathematical description for a much wider range of conditions. Despite the unquestionable effectiveness of reductionism, analysis of large systems becomes untenable when the system comprises of many parts. Coarser models are therefore valuable to predict larger scale phenomena at sufficient accuracy; and indeed many statistical bridges have been cast to connect microscopic models to macroscopic models which (approximately) retain the Markov property.

Resisting these efforts are systems like CTMCs with unstructured state-spaces and transition rates, where properties of macro-scale behaviour are highly non-linear with respect to any state representation; *complex systems*, where system elements interact creating powerful feedback loops; or even structured CTMCs like chemical reaction networks, which are small enough to evade the usual statistical approximations that take effect at larger sizes.

In this thesis, we have developed novel frameworks in an effort to bridge the various scales of such systems. Particularly, we focused on coarsening discrete systems according to satisfaction of a set of logical properties which pertain to an emergent behaviour of interest. Further, we used statistical tools to infer dynamics on the coarsened state-space

which are approximately Markovian or semi-Markovian. While our method produces a consistent abstraction in the steady state, we have seen that the transient behaviour suffers. A valuable extension would be to develop corrections for the abstracted dynamics to better capture the transient dynamics.

Using similar principles, we abstracted layers of multi-scale systems which rely on CTMCs. Owing to a non-parametric regression method for inference of the abstracted dynamics, the abstracted layers have a consistent output to the original layers over a wide range of layer inputs, while retaining a clear interpretation for the abstraction mechanism. We have established that abstractions work well if the underlying CTMC equilibrates fast with respect to the abstraction mechanism, and expect that inaccuracies in the output can be minimised by introducing a higher order Markov abstraction. It would therefore be worthwhile to further investigate and clarify the relationship between the order of the Markov chain used in the abstraction mechanism, and the output accuracy.

Finally, we shifted our gaze to the deterministic continuous approximation for discrete chemical reaction networks, known as the *fluid limit*. We introduced here the *geometric fluid approximation* which is applicable to arbitrary CTMCs, without the need of an *a priori* interpretation for the states: a requirement for the classical fluid approximation to embed states in a continuous space and define dynamics. Our construction allows us to obtain a point-mass approximation in continuous space to the probability distribution of the CTMC evolving in time. There are many paths to pursue further from this work. One could derive dynamics for higher moments of the distribution in continuous space, in order to make the approximation more informative and better assess its accuracy. Also, our construction relates transition rate similarity to state similarity, by encoding distances between states as diffusion distances. This enables estimation techniques over the parameter space of the CTMC to be applied over the continuous state-space in which we embed, since trajectories with neighbouring initial states are expected to be similar.

Bibliography

- Abate, A., Brim, L., Ceska, M., and Kwiatkowska, M. Z. (2015). Adaptive aggregation of Markov chains: Quantitative analysis of chemical reaction networks. In *Proc. of CAV*, pages 195–213. (pages 2 and 32)
- Basharin, G. P., Langville, A. N., and Naumov, V. A. (2004). The life and work of A.A. Markov. *Linear Algebra and its Applications*, 386:3–26. (page 6)
- Baxendale, P. (2011). T. E. Harris’s contributions to recurrent Markov processes and stochastic flows. *The Annals of Probability*, 39(2):417–428. (page 8)
- Belkin, M. and Niyogi, P. (2003). Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396. (pages 123 and 124)
- Bengio, Y., Paiement, J.-F., Vincent, P., Delalleau, O., Le Roux, N., and Ouimet, M. (2004). Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Proc. of NIPS*, pages 177–184. (page 40)
- Berg, H. C., Brown, D. A., and others (1972). Chemotaxis in *Escherichia coli* analysed by three-dimensional tracking. *Nature*, 239(5374):500–504. (page 56)
- Bernoulli, J. (1713). *Ars conjectandi*. Impensis Thurnisiorum, fratrum. (page 7)
- Bishop, C. M. (2006a). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. (page 41)
- Bishop, C. M. (2006b). *Pattern recognition and machine learning*. Information science and statistics. Springer, New York. (page 74)
- Borg, I. and Groenen, P. (2005). *Modern Multidimensional Scaling: Theory and Applications*. Springer. (page 39)
- Bortolussi, L., Milios, D., and Sanguinetti, G. (2015a). Efficient stochastic simulation of systems with multiple time scales via statistical abstraction. In *Proc. of CMSB*, pages 40–51. (page 32)

- Bortolussi, L., Milios, D., and Sanguinetti, G. (2015b). Efficient Stochastic Simulation of Systems with Multiple Time Scales via Statistical Abstraction. In *Computational Methods in Systems Biology*, pages 40–51. Springer, Cham. (pages 50, 52, and 55)
- Bortolussi, L., Milios, D., and Sanguinetti, G. (2016). Smoothed model checking for uncertain Continuous-Time Markov Chains. *Information and Computation*, 247:235–253. (pages 37, 50, 55, and 104)
- Bosgraaf, L. and Haastert, P. J. M. V. (2009). The Ordered Extension of Pseudopodia by Amoeboid Cells in the Absence of External Cues. *PLOS ONE*, 4(4):e5253. (page 66)
- Buchholz, P. and Kriege, J. (2014). Approximate aggregation of Markovian models using alternating least squares. *Perform. Eval.*, 73:73–90. (pages 2 and 32)
- Butkov, E. (1995). *Mathematical physics*. Addison-Wesley series in advanced physics. Addison-Wesley, Reading, Mass., 32. print edition. OCLC: 258506311. (page 70)
- Calovi, D. S., Brunnet, L. G., and de Almeida, R. M. C. (2010). camp diffusion in ‘*Dictyostelium discoideum*’: A green’s function method. *Phys. Rev. E*, 82(1):011909. (pages 68, 69, 70, and 117)
- Cao, Y. and Petzold, L. (2006). Accuracy limitations and the measurement of errors in the stochastic simulation of chemically reacting systems. *J. Comput. Phys.*, 212(1):6–24. (page 46)
- Chakravarty, I. M., Laha, R. G., and Roy, J. D. (1967). *Handbook of methods of applied statistics*. McGraw-Hill, New York, NY. (page 64)
- Ciocchetta, F. and Hillston, J. (2009). Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33-34):3065–3084. (pages 28 and 55)
- Coifman, R. R., Kevrekidis, I. G., Lafon, S., Maggioni, M., and Nadler, B. (2008). Diffusion Maps, Reduction Coordinates, and Low Dimensional Representation of Stochastic Systems. *Multiscale Modeling & Simulation*, 7(2):842–864. (pages 88, 89, and 121)
- Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30. (pages 88, 121, and 123)
- Coifman, R. R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F., and Zucker, S. W. (2005). Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences*, 102(21):7426–7431. (pages 88 and 121)

- Colquhoun, D. (2014). An investigation of the false discovery rate and the misinterpretation of p-values. *Royal Society Open Science*, 1(3):140216–140216. (page 22)
- Cressie, N. (1990). The origins of kriging. *Mathematical Geology*, 22(3):239–252. (page 25)
- Dada, J. O. and Mendes, P. (2011). Multi-scale modelling and simulation in systems biology. *Integrative Biology*, 3(2):86. (page 49)
- Danos, V., Feret, J., Fontana, W., Harmer, R., and Krivine, J. (2007). Rule-based modelling of cellular signalling. In *In Proc. of CONCUR*, pages 17–41. (page 28)
- Darling, D. A. and Siegert, A. J. F. (1953). The First Passage Problem for a Continuous Markov Process. *The Annals of Mathematical Statistics*, 24(4):624–639. (page 101)
- Darling, R. and Norris, J. (2008). Differential equation approximations for Markov chains. *Probability Surveys*, 5(0):37–79. (pages 81, 82, 83, and 84)
- Darling, R. W. R. (2002). Fluid Limits of Pure Jump Markov Processes: a Practical Guide. *arXiv:math/0210109*. arXiv: math/0210109. (page 84)
- Dayar, T. and Stewart, W. (1997). Quasi Lumpability, Lower-Bounding Coupling Matrices, and Nearly Completely Decomposable Markov Chains. *SIAM Journal on Matrix Analysis and Applications*, 18(2):482–498. (page 2)
- Deng, K., Mehta, P. G., and Meyn, S. P. (2011). Optimal Kullback-Leibler aggregation via spectral theory of Markov chains. *Automatic Control, IEEE Transactions on*, 56(12):2793–2808. (page 32)
- Donzé, A. and Maler, O. (2010). *Robust satisfaction of temporal logic over real-valued signals*. Springer. (page 28)
- Eidi, Z. (2017). Discrete Modeling of Amoeboid Locomotion and Chemotaxis in *Dictyostelium discoideum* by Tracking Pseudopodium Growth Direction. *Scientific Reports*, 7(1):12675. (pages 68 and 71)
- Franceschinis, G. and Muntz, R. R. (1994). Bounds for quasi-lumpable Markov chains. *Performance Evaluation*, 20(1):223–243. (page 2)
- Frankel, N. W., Pontius, W., Dufour, Y. S., Long, J., Hernandez-Nunez, L., and Emonet, T. (2014). Adaptability of non-genetic diversity in bacterial chemotaxis. *eLife*, 3:e03526. (pages 58 and 118)
- Gardiner, C. W. (2009). *Stochastic methods: a handbook for the natural and social sciences*. Springer series in synergetics. Springer, Berlin, 4th ed edition. (pages 7, 82, and 123)

- Gelman, A. and Shalizi, C. R. (2013). Philosophy and the practice of Bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 66(1):8–38. (page 24)
- Gilbert, D., Heiner, M., Takahashi, K., and Uhrmacher, A. M. (2015). Multiscale Spatial Computational Systems Biology (Dagstuhl Seminar 14481). *Dagstuhl Reports*, 4(11):138–226. (page 49)
- Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361. (pages 9, 58, 82, and 92)
- Gillespie, D. T. (2000). The chemical Langevin equation. *The Journal of Chemical Physics*, 113(1):297–306. (page 55)
- Goutsias, J. (2005). Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems. *The Journal of Chemical Physics*, 122(18):184102. (pages 52 and 55)
- Gunawardena, J. (2014). Time-scale separation - Michaelis and Menten’s old idea, still bearing fruit. *FEBS Journal*, 281(2):473–488. (page 32)
- Haastert, P. J. M. V. (2010). Chemotaxis: insights from the extending pseudopod. *J Cell Sci*, 123(18):3031–3037. (page 66)
- Haastert, P. J. M. V. and Bosgraaf, L. (2009). Food Searching Strategy of Amoeboid Cells by Starvation Induced Run Length Extension. *PLOS ONE*, 4(8):e6814. (pages 68 and 79)
- Hansen, C. H., Endres, R., and Wingreen, N. (2008). Chemotaxis in Escherichia coli: A Molecular Model for Robust Precise Adaptation. *PLOS Comp Bio*, 4(1):e1. (page 56)
- Haseltine, E. L. and Rawlings, J. B. (2002). Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *The Journal of Chemical Physics*, 117(15):6959–6969. (pages 52 and 55)
- Hayden, R. A., Stefanek, A., and Bradley, J. T. (2012). Fluid computation of passage-time distributions in large Markov models. *Theoretical Computer Science*, 413(1):106–141. (pages 101 and 102)
- Hillston, J. (2005). Fluid flow approximation of PEPA models. In *Second International Conference on the Quantitative Evaluation of Systems (QEST’05)*, pages 33–42. (page 81)
- Hodges, W. (2018). Model Theory. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2018 edition. (page 27)
- Hoel, E. (2017). When the Map Is Better Than the Territory. *Entropy*, 19(5):188. (page 32)

- Jacobi, M. N. (2012). Hierarchical Dynamics. In Meyers, R. A., editor, *Computational Complexity*, pages 1514–1534. Springer New York. (page 32)
- Jha, S. K., Clarke, E. M., Langmead, C. J., Legay, A., Platzer, A., and Zuliani, P. (2009). A Bayesian approach to model checking biological systems. In *Proc. of CMSB*, pages 218–234. (pages 28 and 29)
- Kemeny, J. G. and Snell, J. L. (1960). *Finite Markov chains*. van Nostrand Princeton, NJ. (page 2)
- Kurtz, T. G. (1971). Limit Theorems for Sequences of Jump Markov Processes Approximating Ordinary Differential Processes. *Journal of Applied Probability*, 8(2):344–356. (pages 2, 55, and 81)
- Kwiatkowska, M., Norman, G., and Parker, D. (2011). PRISM 4.0: Verification of probabilistic real-time systems. In *Proc. of CAV*, pages 585–591. (pages 28 and 29)
- Kłopotek, M. A. (2017). Spectral Analysis of Laplacian of a Multidimensional Grid Graph. *arXiv:1707.05210 [math]*. arXiv: 1707.05210. (pages 91 and 125)
- Larsson, A. J. M., Johnsson, P., Hagemann-Jensen, M., Hartmanis, L., Faridani, O. R., Reinius, B., Segerstolpe, a., Rivera, C. M., Ren, B., and Sandberg, R. (2019). Genomic encoding of transcriptional burst kinetics. *Nature*, 565(7738):251–254. (page 94)
- Lawrence, N. D., Girolami, M., Rattray, M., and Sanguinetti, G. (2010). *Learning and inference in computational systems biology*. MIT press Cambridge, MA. (page 21)
- Li, L., Cox, E. C., and Flyvbjerg, H. (2011). ‘Dicty dynamics’: Dictyostelium motility as persistent random motion. *Phys. Biol.*, 8(4):046006. (page 67)
- Li, L., Nørrelykke, S. F., and Cox, E. C. (2008). Persistent Cell Motion in the Absence of External Signals: A Search Strategy for Eukaryotic Cells. *PLOS ONE*, 3(5):e2093. (page 79)
- Maler, O. and Nickovic, D. (2004). Monitoring temporal properties of continuous signals. In *Proc. of FORMATS*, pages 152–166. (pages 28 and 34)
- Markov, A. A. (1906). Extension of the law of large numbers to dependent quantities. *Izv. Fiz.-Matem. Obsch. Kazan Univ.(2nd Ser)*, 15:135–156. (page 7)
- Martiel, J.-L. and Goldbeter, A. (1987). A Model Based on Receptor Desensitization for Cyclic AMP Signaling in Dictyostelium Cells. *Biophysical Journal*, 52(5):807–828. (page 69)

- Michaelides, M., Hillston, J., and Sanguinetti, G. (2017). Statistical Abstraction for Multi-scale Spatio-Temporal Systems. In Bertrand, N. and Bortolussi, L., editors, *Quantitative Evaluation of Systems*, pages 243–258. Springer International Publishing. (page 50)
- Michaelides, M., Hillston, J., and Sanguinetti, G. (2019). Geometric fluid approximation for general continuous-time Markov chains. *arXiv:1901.11417 [cs, stat]*. arXiv: 1901.11417. (page 82)
- Michaelides, M., Milios, D., Hillston, J., and Sanguinetti, G. (2016). Property-Driven State-Space Coarsening for Continuous Time Markov Chains. In *Quantitative Evaluation of Systems*, pages 3–18. Springer, Cham. (pages 32 and 50)
- Milios, D. and Gilmore, S. (2015). Component aggregation for pepa models: An approach based on approximate strong equivalence. *Perform. Eval.*, 94:43–71. (page 32)
- Milios, D., Sanguinetti, G., and Schnoerr, D. (2018). Probabilistic Model Checking for Continuous-Time Markov Chains via Sequential Bayesian Inference. In McIver, A. and Horvath, A., editors, *Quantitative Evaluation of Systems*, volume 11024, pages 289–305. Springer International Publishing, Cham. (page 104)
- Nadler, B., Lafon, S., Coifman, R. R., and Kevrekidis, I. G. (2006a). Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127. (pages 88 and 121)
- Nadler, B., Lafon, S., Kevrekidis, I., and Coifman, R. R. (2006b). Diffusion Maps, Spectral Clustering and Eigenfunctions of Fokker-Planck Operators. In Weiss, Y., Schölkopf, B., and Platt, J. C., editors, *Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press. (pages 88, 89, 90, and 121)
- Naqvi, K. R. (2005). The origin of the Langevin equation and the calculation of the mean squared displacement: Let’s set the record straight. *arXiv:physics/0502141*. arXiv: physics/0502141. (page 16)
- Norris, J. R. (1998). *Markov Chains*. Cambridge University Press. (pages 7, 9, and 82)
- Palaniappan, S. K., Bertaux, F., Pichené, M., Fabre, E., Batt, G., and Genest, B. (2017). Abstracting the dynamics of biological pathways using information theory: a case study of apoptosis pathway. *Bioinformatics*, 33(13):1980–1986. (page 55)
- Perrault-Joncas, D. C. and Meilă, M. (2011). Directed Graph Embedding: an Algorithm based on Continuous Limits of Laplacian-type Operators. In Shawe-Taylor, J., Zemel,

- R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 990–998. Curran Associates, Inc. (pages 88, 89, 121, and 127)
- Rao, C. V. and Arkin, A. P. (2003). Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm. *The Journal of Chemical Physics*, 118(11):4999–5010. (pages 52 and 55)
- Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass. OCLC: ocm61285753. (pages 25, 54, 91, 92, and 116)
- Robertson, A. D. and Grutsch, J. F. (1981). Aggregation in *Dictyostelium discoideum*. *Cell*, 24(3):603–611. (page 68)
- Russell, B. (1993). *Introduction to mathematical philosophy*. Courier Corporation. (page 27)
- Rödenbeck, C., Beck, C., and Kantz, H. (2001). Dynamical systems with time scale separation: averaging, stochastic modelling, and central limit theorems. In Imkeller, P. and von Storch, J.-S., editors, *Stochastic Climate Models*, pages 189–209. Birkhäuser Basel, Basel. (page 32)
- Salmon, W. C. (1990). The Appraisal of Theories: Kuhn Meets Bayes. *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 1990:325–332. (page 22)
- Schnoerr, D., Cseke, B., Grima, R., and Sanguinetti, G. (2017a). Efficient Low-Order Approximation of First-Passage Time Distributions. *Physical Review Letters*, 119(21):210601. (page 101)
- Schnoerr, D., Sanguinetti, G., and Grima, R. (2017b). Approximation and inference methods for stochastic biochemical kinetics—a tutorial review. *Journal of Physics A: Mathematical and Theoretical*, 50(9):093001. (pages 2, 55, and 83)
- Shapiro, S. (2009). *Philosophy of Mathematics and Its Logic: Introduction*, volume 1. Oxford University Press. (page 27)
- Sneddon, M. W., Pontius, W., and Emonet, T. (2012). Stochastic coordination of multiple actuators reduces latency and improves chemotactic response in bacteria. *PNAS*, 109(3):805–810. (pages 56, 57, 58, 59, and 118)
- Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian Processes using Pseudo-inputs. In Weiss, Y., Schölkopf, P. B., and Platt, J. C., editors, *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT Press. (page 116)

- Sourjik, V. and Berg, H. C. (2004). Functional interactions between receptors in bacterial chemotaxis. *Nature*, 428(6981):437–441. (page 56)
- Sourjik, V. and Wingreen, N. S. (2012). Responding to Chemical Gradients: Bacterial Chemotaxis. *Curr Opin Cell Biol*, 24(2):262–268. (page 56)
- Tschaikowski, M. and Tribastone, M. (2015). A unified framework for differential aggregations in Markovian process algebra. *J. Log. Algebr. Meth. Program.*, 84(2):238–258. (page 32)
- van Kampen, N. G. (1961). A POWER SERIES EXPANSION OF THE MASTER EQUATION. *Canadian Journal of Physics*, 39(4):551–567. (pages 55 and 82)
- Vladimirov, N., Lebedez, D., and Sourjik, V. (2010). Predicted Auxiliary Navigation Mechanism of Peritrichously Flagellated Chemotactic Bacteria. *PLOS Comp Bio*, 6(3):e1000717. (page 58)
- Vladimirov, N., Løvdok, L., Lebedez, D., and Sourjik, V. (2008). Dependence of Bacterial Chemotaxis on Gradient Shape and Adaptation Rate. *PLOS Comp Biol*, 4(12):e1000242. (page 56)
- Vu, T. N., Wills, Q. F., Kalari, K. R., Niu, N., Wang, L., Rantalainen, M., and Pawitan, Y. (2016). Beta-Poisson model for single-cell RNA-seq data analyses. *Bioinformatics*, 32(14):2128–2135. (page 94)
- Wolpert, D. H., Grochow, J. A., Libby, E., and DeDeo, S. (2014). Optimal high-level descriptions of dynamical systems. *arXiv:1409.7403 [cond-mat, q-bio]*. arXiv: 1409.7403. (page 32)
- Younes, H. L. S. and Simmons, R. G. (2006). Statistical probabilistic model checking with a focus on time-bounded properties. *Inf. Comput.*, 204(9):1368–1409. (pages 28 and 29)
- Zabell, S. L. (1989). The Rule of Succession. *Erkenntnis (1975-)*, 31(2/3):283–321. (page 23)

Appendix A

Supplementary material for the statistical abstraction framework

A.1 Gaussian process classification details

In mathematical language, we observe the mapping we wish to approximate at N points of its domain, $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^N$ where $X_i \in \mathcal{X}$ is the input value and y_i the output. In our case, the output is a binary value of the property evaluation $y_i \in \{\top = 1, \perp = 0\}$. We denote the collection of all $\{X_i\}_{i=1}^N = \mathbf{X}$ and $\{y_i\}_{i=1}^N = \mathbf{y}$. Given \mathcal{D} , we would like to infer the underlying probability function $\Psi : \mathcal{X} \rightarrow [0, 1]$ which we assume to give the probability parameter of the Bernoulli distribution generating the observations y :

$$y \mid X \sim \text{Bernoulli}(p = \Psi(X)). \quad (\text{A.1})$$

To learn Ψ from \mathcal{D} observations, the GP assumes the existence of a latent function $f : \mathcal{X} \rightarrow \mathbb{R}$ which we pass through an inverse-probit transformation to bring it within Bernoulli parameter domain range $[0, 1]$, such that $\Psi(X_i) = \Phi(f(X_i))$, where $\Phi : \mathbb{R} \rightarrow [0, 1]$ is the cumulative distribution function of the standard normal distribution $\mathcal{N}(0, 1)$. This forms the likelihood of the Bernoulli parameter being approximated ($\Psi(\cdot)$) given the latent function $f(\cdot)$. In fact, the GP assumes a whole distribution over possible latent functions; this is a multivariate normal defined by our choice of covariance structure (kernel $k(\cdot, \cdot)$) and mean function $m(\cdot)$, denoted as $f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$. Considering only the collection of function values $\mathbf{f} = [f(X_i)]_{i=1}^N$ where we have observations \mathbf{X} , our prior distribution becomes a finite-dimensional Gaussian $\mathbf{f} \mid \mathbf{X} \sim \mathcal{N}(\mathbf{m}, \mathbf{K})$, where bold letters denote the functions evaluated at each observation X_i or X_i, X_j pair. We proceed to condition on outputs \mathbf{y} to obtain a posterior distribution over the latent functions

through standard Gaussian distribution results.

$$p(y_i | X_i, f(\cdot)) = \text{Bernoulli}(\Psi(X_i)) = \Phi(f(X_i))^{y_i} (1 - \Phi(f(X_i)))^{1-y_i}, \quad (\text{A.2})$$

$$\text{giving posterior } p(\mathbf{f} | \mathcal{D} = \mathbf{X}, \mathbf{y}) \propto p(\mathbf{y} | \mathbf{f})p(\mathbf{f} | \mathbf{X}). \quad (\text{A.3})$$

To predict $\Psi(X_*)$ at an unobserved domain point X_* not in \mathcal{D} , we take a weighted average of all possible values of $\Phi(f(X_*))$ under the posterior distribution of latent function values $\Psi(X_*) = \langle \Phi(f(X_*)) \rangle_{f_* | \mathcal{D}, x_*}$, where for notational simplicity $f_* = f(X_*)$.

$$\Psi(X_*) = \int \Phi(f_*)p(f_* | \mathcal{D}, x_*)df_*, \quad (\text{A.4})$$

$$\text{where } p(f_* | \mathcal{D}, x_*) = \int p(f_* | X, \mathbf{f}, x_*)p(\mathbf{f} | \mathcal{D}) d\mathbf{f}. \quad (\text{A.5})$$

GP regression with its many variations for different problem tasks is well described in [Rasmussen and Williams \(2006\)](#). The necessary adjustments which we adopt here are found in the Gaussian process classification (GPC) section of the book, and essentially amount to identifying that the class probability function is Ψ , where the class is the property satisfaction outcome. Standard results make the integrals in [Eqs. A.4-A.5](#) analytically tractable if we approximate the posterior $p(\mathbf{f} | \mathcal{D})$ with a Gaussian. To do this, we use Minka's Expectation-Propagation (EP) technique because it is more accurate than the alternative Laplace approximation. Further, we use fully independent training conditional (FITC) approximation [Snelson and Ghahramani \(2006\)](#) to allow a large number of observations to be considered for learning the underlying function, while maintaining a low cost of predicting at any point of the domain. The approximation essentially replaces the original training data (input-output set) with a smaller set of *inducing points* (dummy input-output set) which is constructed from the former and used in prediction; these *inducing points* produce a conditional distribution over functions that is close to the one produced by conditioning on the real data. It relies on a rank-reduced approximation of the original covariance matrix.

A.2 Simulation schemes for *E. coli* model

A.2.1 Simulation scheme for original *E. coli* model

See Algorithm 1.

A.2.2 Simulation scheme for abstracted *E. coli* model

See Algorithm 2

A.3 Constants of *D. discoideum* model

Table A.1 Table of fixed constants for the *D. Discoideum* model. These are used in Equations 4.12-4.16 and taken from Calovi et al. (2010), where their physical interpretation is also examined.

| Parameter | Value | Parameter | Value |
|-------------|---------------------------|-----------|----------------------------|
| λ_1 | 10 | k_i | 1.7 min ⁻¹ |
| λ_2 | 0.18 | k_t | 0.9 min ⁻¹ |
| λ_3 | 463.5 | k_e | 5.4 min ⁻¹ |
| k_1 | 0.036 min ⁻¹ | σ | 0.01 mm |
| k_{-1} | 0.36 min ⁻¹ | D | 0.024 mm ² /min |
| k_2 | 0.666 min ⁻¹ | h | 0.025 |
| k_{-2} | 0.00333 min ⁻¹ | | |

Algorithm 1 Simulation scheme for the *E. coli* model, based on full simulation of the pCTMC describing F/M conformation changes. Below, τ , mb_0 , α are constants which parametrise the model (see [Sneddon et al. \(2012\)](#)), and Δt is the fixed simulation time-step.

```

1: function RUN( $\vec{r}$ ,  $\vec{v}$ ,  $\Delta t$ )
2:    $\vec{r} \leftarrow \vec{r} + \vec{v} \cdot \Delta t$ 
3:   return  $\vec{r}$ 
4: end function
5:
6: function TUMBLE( $\vec{v}$ ,  $\Delta t$ )
7:    $\theta \sim \Gamma(\text{shape} = 4, \text{scale} = 18.32)$             $\triangleright$  Sample tumbling angle from distribution
                                                         given in Sneddon et al. \(2012\).
8:    $\vec{v} \leftarrow R(\theta) \cdot \vec{v}$                   $\triangleright$   $R(\theta)$  is a 2D rotation matrix through
                                                         angle  $\theta$ .
9:   return  $\vec{v}$ 
10: end function
11:
12: function OU-EULER-MARUYAMA( $m$ ,  $L$ ,  $\Delta t$ )
13:    $\bar{m} \leftarrow \text{MEANMETH}(L, mb_0, \alpha)$         $\triangleright$  Mean methylation level  $\bar{m}(L, mb_0, \alpha)$ 
                                                         as in Frankel et al. \(2014\); Sneddon
                                                         et al. \(2012\).
14:    $m \leftarrow m + \left[ \Delta t / \tau (\bar{m} - m) + \sigma_m \sqrt{2/\tau} dW(\Delta t) \right]$ 
15:   return  $m$ 
16: end function
17:
18: procedure SIMULATEFINEECOLICELL( $t_{\text{end}}$ )
19:    $t \leftarrow 0$ 
20:   while  $t < t_{\text{end}}$  do
21:      $L \leftarrow L(\vec{r}, t)$                         $\triangleright$  The ligand field  $L$  value, at the cell's
                                                         location  $\vec{r}$ .
22:      $\mathbf{s} \leftarrow \text{PCTMC}(\mathbf{s}, m, L, \Delta t)$   $\triangleright$  Drawing F/M pCTMC trajectory of
                                                         length  $\Delta t$ , with parameters  $k_{\pm}(m, L)$ 
                                                         and initial state the last pCTMC state
                                                         of the cell.
23:      $\psi \leftarrow \phi_{\text{RUN}}(\mathbf{s})$                   $\triangleright$  Evaluating the  $\phi_{\text{RUN}}$  on (the final state
                                                         of) the pCTMC trajectory.
24:     if  $\psi$  then
25:        $\vec{r} \leftarrow \text{RUN}(\vec{r}, \vec{v}, \Delta t)$ 
26:     else
27:        $\vec{v} \leftarrow \text{TUMBLE}(\vec{v}, \Delta t)$ 
28:     end if
29:      $m \leftarrow \text{OU-EULER-MARUYAMA}(m, L, \Delta t)$   $\triangleright$  Evolving methylation.
30:      $t \leftarrow t + \Delta t$ 
31:   end while
32: end procedure

```

Algorithm 2 Simulation scheme for the abstracted *E. coli* model, based on GP approximation for the RUN/TUMBLE probability. Steps 5, 6 here replace the expensive Steps 22, 23 in Algorithm 1.

```

1: procedure SIMULATEABSTRACTEDECOLICELL( $t_{\text{end}}$ )
2:    $t \leftarrow 0$ 
3:   while  $t < t_{\text{end}}$  do
4:      $L \leftarrow L(\vec{r}, t)$ 
5:      $p \leftarrow \text{GP}_{\psi}(m, L)$ 
6:      $\psi \sim \text{Bernoulli}(p)$ 
7:     if  $\psi$  then
8:        $\vec{r} \leftarrow \text{RUN}(\vec{r}, \vec{v}, \Delta t)$ 
9:     else
10:       $\vec{v} \leftarrow \text{TUMBLE}(\vec{v}, \Delta t)$ 
11:    end if
12:     $m \leftarrow \text{OU-EULER-MARUYAMA}(m, L, \Delta t)$ 
13:     $t \leftarrow t + \Delta t$ 
14:  end while
15: end procedure

```

Appendix B

Supplementary material for the Geometric Fluid Approximation

B.1 Diffusion maps for Markov chains

There exists extensive literature examining the implications of diffusion maps, as well as their limitations and strengths (Coifman et al., 2008; Coifman and Lafon, 2006; Coifman et al., 2005; Nadler et al., 2006a,b). What follows is therefore not an attempt to re-derive these results or convince the reader of the validity of the method, but rather to set notation and highlight the aspects that are relevant to our purposes. The exposition below is also necessary to act as a foundation for the results of Perrault-Joncas and Meilă (2011) that build upon the original concept of diffusion maps as put forth by Coifman, Lafon, Nadler, and Kevrekidis.

B.1.1 Undirected graphs

In (Coifman et al., 2008; Nadler et al., 2006a), the authors consider a family of density-normalised (i.e. anisotropic) symmetric kernels

$$k_\epsilon^{(\alpha)}(\mathbf{x}, \mathbf{y}) = \frac{k_\epsilon(\mathbf{x}, \mathbf{y})}{p_\epsilon^\alpha(\mathbf{x})p_\epsilon^\alpha(\mathbf{y})}$$

characterising the distance between high-dimensional points $\mathbf{x}, \mathbf{y} \in \mathcal{M} \subseteq \mathbb{R}^p$, and with non-uniform density $p(\mathbf{x}) = e^{-U(\mathbf{x})}$. The kernel used here is the radial basis function $k_\epsilon(\mathbf{x}, \mathbf{y}) = \exp(-d(\mathbf{x}, \mathbf{y})^2/\epsilon)$, which provides a similarity between points based on the Euclidean distance d in the original space. The density-normalising factor $p_\epsilon^\alpha(\mathbf{x})$ depends on the manifold density, $p_\epsilon(\mathbf{x}) = \int k_\epsilon(\mathbf{x}, \mathbf{y})p(\mathbf{y})d\mathbf{y}$, and the choice of the power α leads

to transition kernels of different diffusion process operators (see below). Indeed, $p_\epsilon(\mathbf{x})$ is a local density estimate at \mathbf{x} , and for a unit normalised kernel we have

$$\lim_{\epsilon \rightarrow 0} p_\epsilon(\mathbf{x}) = p(\mathbf{x}) + \epsilon \Delta p(\mathbf{x}) + \mathcal{O}(\epsilon^{3/2}).$$

For a finite set of points we can construct an adjacency matrix whose elements are given by the kernel, for a network with points as nodes and weighted undirected edges.

Assuming that the points were sampled by observing a diffusion process in the space \mathcal{M} , the authors then take the forward Markov transition probability kernel to be

$$M_f^{(\alpha)}(\mathbf{x}|\mathbf{y}) = \Pr[\mathbf{x}(t + \epsilon) | \mathbf{x}(t) = \mathbf{y}] = \frac{k_\epsilon^{(\alpha)}(\mathbf{x}, \mathbf{y})}{d_\epsilon^{(\alpha)}(\mathbf{y})},$$

where $d_\epsilon^{(\alpha)}(\mathbf{y}) = \int_{\mathcal{M}} k_\epsilon^{(\alpha)}(\mathbf{x}, \mathbf{y}) p(\mathbf{x}) d\mathbf{x}$ is the graph Laplacian normalisation factor. Since this is the transition probability for the putative continuous diffusion process evolving in the space \mathcal{M} , the (forward) infinitesimal diffusion operator of the process is given by

$$\frac{\partial}{\partial t} = \mathcal{H}_f^{(\alpha)} = \lim_{\epsilon \rightarrow 0} \left[\frac{T_f^{(\alpha)} - I}{\epsilon} \right],$$

where I is the identity operator, and $T_f^{(\alpha)}$ is a (forward) transport operator defined as $T_f^{(\alpha)}[\phi](\mathbf{x}) = \int_{\mathcal{M}} M_f^{(\alpha)}(\mathbf{x}|\mathbf{y}) \phi(\mathbf{y}) p(\mathbf{y}) d\mathbf{y}$, which evolves a function $\phi : \mathcal{M} \rightarrow \mathbb{R}$ according to $M_f^{(\alpha)}$ and the manifold measure $p(\mathbf{y}) = e^{-U(\mathbf{x})}$.

By asymptotic expansion of the relevant integrals, they show that the forward and backward operator pair is

$$\mathcal{H}_f^{(\alpha)} = \Delta - 2\alpha \nabla U \cdot \nabla + (2\alpha - 1)(\|\nabla U\|^2 - \Delta U), \quad \text{and} \quad (\text{B.1})$$

$$\mathcal{H}_b^{(\alpha)} = \Delta - 2(1 - \alpha) \nabla U \cdot \nabla \quad (\text{B.2})$$

respectively.

We then regard the adjacency matrix W of a given network to be a discrete approximation of the transition kernel k_ϵ defined over continuous space. From that, we can construct discrete (in time and space) approximations to the diffusion operators \mathcal{H}^α above by performing the necessary normalisations. To retrieve the embedding coordinates for each network vertex one needs to spectrally analyse the approximation to the diffusion operator, taking the 1 to $k + 1$ eigenvectors $\{\psi_j\}_{j=1}^d$ ordered by the associated eigenvalues $\{-\lambda_j\}_{j=1}^d$ with $\lambda_0 = 0 > -\lambda_1 \geq -\lambda_2 \geq \dots \geq -\lambda_d$, to be the vertices' coordinates in the first $k < d$ dimensions of the embedding. The first eigenvector is discarded as a

trivial dimension where every vertex has the same coordinate by construction. Thus, the k -dimensional diffusion map at time t is defined as:

$$\Psi_k^t(\mathbf{x}) := \left(e^{-\lambda_1 t} \psi_1(\mathbf{x}), e^{-\lambda_2 t} \psi_2(\mathbf{x}), \dots, e^{-\lambda_k t} \psi_k(\mathbf{x}) \right),$$

where we have discarded ψ_0 associated with $\lambda_0 = 0$ as a trivial dimension. The time parameter t refers to the diffusion distance after time t which is preserved as Euclidean distance in the embedding space. Trivially, as $t \rightarrow \infty$ all network nodes are mapped to the same point since the diffusion distance vanishes.

The parameter α adjusts the effect that the manifold density has on the diffusion process. Choosing $\alpha = 1$ recovers the Laplace-Beltrami operator Δ as the backward diffusion operator, if the points approximately lie on a manifold $\mathcal{M} \subset \mathbb{R}^d$. Thus, the diffusion map corresponds to an embedding of the points unaffected by the manifold density (such that if two different networks were sampled from the same manifold \mathcal{M} but with different densities, we would recover consistent positions of the points on \mathcal{M}). Choosing $\alpha = 0$ is equivalent to the *Laplacian eigenmaps* method which preceded diffusion maps (Belkin and Niyogi, 2003). If the vertices are sampled uniformly from the hidden manifold, Laplacian eigenmap becomes equivalent to analysing the Laplace-Beltrami operator, and so constructing a diffusion map with $\alpha = 1$ and with $\alpha = 0$ will recover the same embedding (Coifman and Lafon, 2006).

Consider now an Itô stochastic differential equation (SDE) of the form

$$\dot{\mathbf{x}} = \boldsymbol{\mu}(\mathbf{x}) + \boldsymbol{\sigma} \dot{\mathbf{w}}, \quad (\text{B.3})$$

where \mathbf{w}_t is the d -dimensional Brownian motion. A probability distribution over the state-space of this system $\phi(\mathbf{x}, t)$ with condition $\phi(\mathbf{x}, 0) = \phi_0(\mathbf{x})$, evolves forward in time according to the Fokker-Planck (FP) equation, also known as the Kolmogorov forward equation (KFE):

$$\partial_t \phi(\mathbf{x}, t) = - \sum_i \partial_i [\mu_i(\mathbf{x}) \phi(\mathbf{x}, t)] + \sum_i \sum_j \partial_i \partial_j \left[\frac{1}{2} \sigma_i \sigma_j \phi(\mathbf{x}, t) \right], \quad (\text{B.4})$$

with the sums running over all d dimensions and ∂_i denoting partial derivatives with respect to the i th dimension ($\partial_i = \partial/\partial x_i$) (Gardiner, 2009). Similarly, the probability distribution $\psi(\mathbf{y}, s)$ for $s \leq t$ and condition $\psi(\mathbf{y}, t) = \psi_t(\mathbf{x})$ satisfies

$$-\partial_s \psi = \boldsymbol{\mu} \cdot \nabla \psi + \frac{1}{2} \boldsymbol{\sigma} \boldsymbol{\sigma}^\top \Delta \psi, \quad (\text{B.5})$$

where the differentiations are with respect to \mathbf{y} . Terms in the backward FPE become directly identifiable with the backward operator $\mathcal{H}_b^{(\alpha)}$ if we take $\boldsymbol{\sigma} = \sqrt{2}\mathbf{I}$ and $\boldsymbol{\mu} = 2(1 - \alpha)\nabla U$.

The original formulation of diffusion maps, as described above, assumes a symmetric kernel $k_\epsilon(\mathbf{x}, \mathbf{y}) = k_\epsilon(\mathbf{y}, \mathbf{x})$. Given a CTMC with a symmetric generator matrix Q , the methodology laid out so far would be sufficient to recover an embedding for the states on a continuous compact manifold \mathcal{M} , on which we can define an SDE approximation to the Markov jump process of the CTMC. Encouragingly, it has also been shown that the jump process would satisfy the reflecting (no flux) conditions on the manifold boundary $\partial\mathcal{M}$, as required by a diffusion FP operator defined on such a manifold — i.e. for a point $\mathbf{x} \in \partial\mathcal{M}$ where \mathbf{n} is a normal unit vector at \mathbf{x} , and a function $\psi : \mathcal{M} \rightarrow \mathbb{R}$,

$$\left. \frac{\partial\psi(\mathbf{x})}{\partial\mathbf{n}} \right|_{\partial\mathcal{M}} = 0.$$

B.1.2 Embedding unweighted, undirected, grid graphs

Taking the case of a pCTMC produced by a particular class of chemical reaction networks, we show that the embedding produced by *Laplacian eigenmaps* (Belkin and Niyogi, 2003) (equivalent to diffusion maps with $\alpha = 0$) for the unweighted, undirected transition matrix, is consistent in some respect to the canonical (manual) embedding for the fluid limit of chemical reaction systems. This implies that we ignore any density information of the vertices (states) on the manifold, and any directional component. We will later return to how this information affects our results.

Laplacian eigenmaps embedding Assume that we have symmetric similarity matrix W between n points. We construct the Laplacian matrix $L = D - W$, with $D_{ii} = \sum_j W_{ji}$. The Laplacian eigenmaps algorithm solves the minimisation problem

$$\operatorname{argmin}_{\Upsilon^\top D \Upsilon = I} \frac{1}{2} \sum_{i,j} \|\mathbf{y}^{(i)} - \mathbf{y}^{(j)}\|_2^2 W_{ij} \quad (\text{B.6})$$

$$= \operatorname{argmin}_{\Upsilon^\top D \Upsilon = I} \operatorname{Tr}(\Upsilon^\top L \Upsilon), \quad (\text{B.7})$$

where $\mathbf{y}^{(i)}$ is the i th row of Υ , and the constraint $\Upsilon^\top D \Upsilon = I$ serves to exclude the trivial solution of mapping everything to the same point. The solution $\Upsilon \in \mathbb{R}^{n \times m}$ is a matrix with each column vector corresponding to the m -dimensional coordinate embedding of each datum ($m < n$). It is shown that the solution to the problem is the eigenvector

matrix corresponding to the m lowest eigenvalues of $L\mathbf{y} = \lambda D\mathbf{y}$, excluding the $\lambda = 0$ solution.

This emphasis on preserving local information allows us to appropriate the algorithm for embedding the network of states without having to calculate global state separation. For a CTMC described by a transition matrix Q , we transform Q to be an adjacency matrix between the nodes (states) of the network (CTMC) by placing an undirected edge of weight 1 between states which are separated by a single transition and 0 otherwise:

$$W_{ij} = 1 - \delta_{0,Q_{ij}}\delta_{0,Q_{ji}}. \quad (\text{B.8})$$

If the network is connected and m (the dimensionality for the embedding space) is picked appropriately, the algorithm will attempt to preserve local dimensions and therefore global ones if the network fits in that m space. If m is chosen higher than necessary, some states which are far apart might be placed closer together in the embedding, but local distances will still be preserved.

The unweighted Laplacian fluid approximation The proof for Theorem 5.2.1 is laid out here. It involves the construction of an undirected, unweighted graph with adjacency matrix W from the Q matrix of a specific kind of pCTMCs, as shown above. Explicit eigenvectors of the Laplacian L of this graph give analytic coordinates for the vertices of Q in some space \mathbb{R}^d . A drift vector field is inferred on this space using Gaussian process regression, from Q and the embedding coordinates. We show from these how conditions for a fluid approximation are met, as stated in Chapter 5, Section 5.1.1. Specifically, we show how *initial conditions converge*, *mean dynamics converge*, and *noise converges to zero* (via Taylor expansion of the relevant analytic coordinates), in the same way as in the canonical embedding of such a pCTMC resulting from *hydrodynamic scaling*.

Theorem 5.2.1 *Let \mathcal{C} be a pCTMC, whose underlying transition graph is a multi-dimensional grid graph. The unweighted Laplacian fluid approximation of \mathcal{C} coincides with the canonical fluid approximation in the hydrodynamic scaling limit.*

Proof. We examine a particular case of pCTMCs, produced by allowing reactions that only change the count of a single species per reaction. This produces an adjacency matrix W for the network of states describing a grid network in d dimensions. Following the derivation for the eigenvectors of the Laplacian L of such a network presented in (Kłopotek, 2017), we find that the lowest eigenvalue λ_1 (excluding $\lambda_0 = 0$) is degenerate

($\lambda_1 = \lambda_{\{2, \dots, d\}}$), and associated with d eigenvectors \mathbf{v}_j , $j \in \{1, \dots, d\}$. Their elements are

$$\mathbf{v}_{j, [x_1, \dots, x_d]} = \cos\left(\frac{\pi}{n_j} \left(x_j - \frac{1}{2}\right)\right) \quad (\text{B.9})$$

where the index $[x_1, \dots, x_d]$ is the mapping of the node to its integer grid coordinates. Therefore, the embedded coordinate of a node in dimension j is $\cos(\pi/n_j(x_j - 1/2))$, where $x_j \in \{1, \dots, n_j\}$ is the integer grid position of the node in that j dimension. We observe that away from the boundaries (i.e. near the centre of the grid $x \approx n/2$) and for large n , the argument of \cos is near $\pi/2$, so we approach the linear part of \cos . This means that near the centre states are almost uniformly distributed, as in the canonical embedding.

We define the volume $\Omega_U([x_1, \dots, x_d])$ for a state with grid coordinates $[x_1, \dots, x_d]$ in the network, to be the volume of the polygon (n -orthotope) whose vertices are that state and the next state along each grid dimension:

$$\Omega_U([x_1, \dots, x_d]) = \prod_j \left(\mathbf{v}_{j, [x_1, \dots, x_j+1, \dots, x_d]} - \mathbf{v}_{j, [x_1, \dots, x_j, \dots, x_d]} \right) \quad (\text{B.10})$$

$$= \prod_j \left[\cos\left(\frac{\pi}{2n_j} (2x_j + 1)\right) - \cos\left(\frac{\pi}{2n_j} (2x_j - 1)\right) \right]. \quad (\text{B.11})$$

We then observe that $\lim_{n \rightarrow \infty} \Omega_U = 0$ for all states; this satisfies the convergence condition of initial states for a fluid approximation.

We define dynamics by means of a drift field $\langle b \rangle : U \rightarrow \mathbb{R}^d$. The function is inferred using Gaussian process regression, $b(\cdot) | Q \sim \mathcal{GP}(m(\cdot) | Q, k(\cdot, \cdot) | Q)$, such that it is a Lipschitz field. This satisfies the convergence condition of mean dynamics for a fluid approximation. In the canonical embedding of a pCTMC, the drift vector field is a polynomial function $f_p \in L^2(U)$ over the concentration space. Away from the boundaries, the Laplacian embedding approaches this canonical embedding. As $n \rightarrow \infty$, the inferred field in this region will tend to the same polynomial function:

$$\langle b \rangle \rightarrow f_p \quad ,$$

as the Gaussian process can approximate any function in $L^2(U)$ arbitrarily well.

Finally, the conditions for noise converging to zero are trivially met, since embedding distances γ are at most $\mathcal{O}(n^{-1})$:

$$\begin{aligned}\gamma &= \cos\left(\frac{\pi}{2n_j}(2x_j+1)\right) - \cos\left(\frac{\pi}{2n_j}(2x_j-1)\right) \\ &= 1 - \frac{1}{2!}\left(\frac{\pi}{2n_j}(2x_j+1)\right)^2 + \frac{1}{4!}\left(\frac{\pi}{2n_j}(2x_j+1)\right)^4 - \dots \\ &\quad - 1 + \frac{1}{2!}\left(\frac{\pi}{2n_j}(2x_j-1)\right)^2 - \frac{1}{4!}\left(\frac{\pi}{2n_j}(2x_j-1)\right)^4 + \dots \\ &= \mathcal{O}(n_j^{-1}),\end{aligned}$$

and $n = \sum_j n_j$, such that $\gamma^2 = \mathcal{O}(n^{-2})$.

Thus the criteria for *fluid approximation* of this pCTMC are satisfied. Further, for some region of the state-space and in the limit of infinite states, this construction is consistent with the embedding and dynamics recovered by *hydrodynamic scaling*, the canonical *fluid approximation* of a pCTMC. This concludes our proof. \square

B.1.3 Directed graphs

Our focus necessarily shifts to embedding an arbitrary CTMC with no symmetry condition on Q . Following [Perrault-Joncas and Meilă \(2011\)](#) assume that we observe a graph G , with nodes sampled from a diffusion process on a manifold \mathcal{M} with density $p = e^{-U}$ and edge weights given by the (non-symmetric) kernel k_ϵ . The directional component of the kernel is further assumed to be derived from a vector field \mathbf{r} on \mathcal{M} without loss of kernel generality. As the authors saliently put it: “The question is then as follows: can the generative process’ geometry \mathcal{M} , distribution $p = e^{-U}$, and directionality \mathbf{r} , be recovered from G ?”

In the same manner as for the original formulation of diffusion maps a set of backward evolution operators are derived, the two relevant ones being:

$$-\partial_t = \mathcal{H}_{aa}^{(\alpha)} = \Delta + (\mathbf{r} - 2(1-\alpha)\nabla U) \cdot \nabla, \quad \text{and} \quad (\text{B.12})$$

$$-\partial_t = \mathcal{H}_{ss}^{(\alpha)} = \Delta - 2(1-\alpha)\nabla U \cdot \nabla. \quad (\text{B.13})$$

To construct this family of operators, the kernel is first decomposed into its symmetric $h_\epsilon(\mathbf{x}, \mathbf{y}) = h_\epsilon(\mathbf{y}, \mathbf{x})$ and anti-symmetric $a_\epsilon(\mathbf{x}, \mathbf{y}) = -a_\epsilon(\mathbf{y}, \mathbf{x})$ parts,

$$k_\epsilon^{(\alpha)}(\mathbf{x}, \mathbf{y}) = \frac{k_\epsilon(\mathbf{x}, \mathbf{y})}{p_\epsilon^\alpha(\mathbf{x})p_\epsilon^\alpha(\mathbf{y})} = \frac{1}{p_\epsilon^\alpha(\mathbf{x})p_\epsilon^\alpha(\mathbf{y})} [h_\epsilon(\mathbf{x}, \mathbf{y}) + a_\epsilon(\mathbf{x}, \mathbf{y})],$$

and further normalised according to either the asymmetric¹ $d_\epsilon^{(\alpha)}(\mathbf{x}) = \int_{\mathcal{M}} k_\epsilon^{(\alpha)}(\mathbf{x}, \mathbf{y}) p(\mathbf{y}) d\mathbf{y}$, or symmetric out-degree distribution $\tilde{d}_\epsilon^{(\alpha)}(\mathbf{x}) = \int_{\mathcal{M}} h_\epsilon^{(\alpha)}(\mathbf{x}, \mathbf{y}) p(\mathbf{y}) d\mathbf{y}$ (where $h_\epsilon^{(\alpha)}(\mathbf{x}, \mathbf{y}) = h_\epsilon(\mathbf{x}, \mathbf{y}) / (p_\epsilon^\alpha(\mathbf{x}) p_\epsilon^\alpha(\mathbf{y}))$ as the notation implies). The subscript indices denote the type of kernel used to construct the operator and the out-degree distribution used to normalise it (such that \mathcal{H}_{aa} associates to the full asymmetric kernel $k_\epsilon^{(\alpha)}$ normalised with asymmetric degree distribution $d_\epsilon^{(\alpha)}(\mathbf{x})$, and so on).

Discrete approximations for these operators can be constructed for an asymmetric kernel matrix of distances between N high-dimensional points, $\mathbf{W} \in \mathbb{R}^{N \times N}$. The symmetric matrix $H_{ss}^{(1)} \in \mathbb{R}^{N \times N}$ can be extracted and the necessary eigen-decomposition carried out to yield an embedding, where $\lim_{N \rightarrow \infty} H_{ss}^{(1)} = \mathcal{H}_{ss}^{(1)} = \Delta$. However, given the infinitesimal generator of a CTMC Q , we do not have access to \mathbf{W} , but rather to the discrete approximation of the final evolution operator, $\lim_{N \rightarrow \infty} Q = \mathcal{H}_{aa}^{(\alpha)}$. In order to recover the initial kernel matrix \mathbf{W} that gave rise to Q , we take $\alpha = 0$, a uniform measure on the manifold $U(\mathbf{x}) = 0 \implies p(\mathbf{x}) = 1$, and a small value for ϵ . This makes the transformations from \mathbf{W}_ϵ to Q reversible, since

$$Q = \lim_{\epsilon \rightarrow 0} \left[\frac{T_\epsilon^{(\alpha=0)} - \mathbf{I}}{\epsilon} \right], \text{ and} \quad (\text{B.14})$$

$$T_\epsilon^{(\alpha=0)} = D^{-1} \mathbf{W}_\epsilon, \text{ such that} \quad (\text{B.15})$$

$$\mathbf{W}_\epsilon = D(\mathbf{I} + \epsilon Q) \text{ for } \epsilon \rightarrow 0. \quad (\text{B.16})$$

In the above, D is a diagonal matrix which forces the diagonal of \mathbf{W}_ϵ to be 1, as expected from a distance-based kernel matrix. The final step $(\mathbf{I} + \epsilon Q)$ is the familiar *uniformisation* procedure which approximates a CTMC with a DTMC. The choice of $\epsilon < (\max_i |Q_{ii}|)^{-1}$ determines the quality of approximation.²

Once the kernel matrix \mathbf{W}_ϵ is recovered we can proceed to construct the operators $\Delta = \mathcal{H}_{ss}^{(1)}$ and $(\mathcal{H}_{aa}^{(0)} - \mathcal{H}_{ss}^{(1)}) = (\mathbf{r} - 2\nabla U) \cdot \nabla$, which are used to embed the state-space on a manifold $\mathcal{M} \in \mathbb{R}^d$, and endow it with the advective field in the KBE $\boldsymbol{\mu} = (2\nabla U + \mathbf{r})$, respectively.

¹*Asymmetric* is used here to express lack of symmetry — compare with *anti-symmetric* used to express $a_\epsilon(\mathbf{x}, \mathbf{y}) = -a_\epsilon(\mathbf{y}, \mathbf{x})$.

²In theory the smaller ϵ is, the better approximation; in practice, we must make a choice of ϵ which will introduce an error: the kernel similarity will be 0 between some states, when it should be >0 for $\epsilon > 0$.