Developing hybrid continuum-discrete models for the study of environment-metabolism feedback in cellular systems

by

Connah Griffith Michael Johnson

Thesis

Submitted to the University of Warwick

for the degree

Doctor of Philosophy

Centre for Complexity Science

April 2022
## Contents

List of Tables iv

List of Figures v

Acknowledgments viii

Declarations ix

Abstract x

Abbreviations xi

### Chapter 1  Bacterial biofilms and niche construction 1

1.1 Introduction .................................................. 1

1.2 Spatial organisation at the cell level: Niche construction in bacterial biofilms ............................................. 3

1.3 Cell behaviour affects population organisation .............. 4

1.4 Spatial population organisation affects cell behaviour .... 8

1.5 Outlook ...................................................... 11

### Chapter 2  The mathematics of chemical reaction systems 13

2.1 Introduction .................................................. 13

2.2 Reaction-diffusion dynamics describe the dynamic evolution of chemical systems ........................................... 14

2.3 The analysis of cell-environment feedback in cellular systems .... 23

2.4 Outlook ...................................................... 29

### Chapter 3  Agent based software for modelling cell-environment feedback 31

3.1 Introduction .................................................. 31
List of Tables

3.1 Summary of a selection of agent-based simulation software focusing on the cell descriptions. ........................................ 47
3.2 Summary of a selection of agent-based simulation software focusing on the environment descriptions. ........................ 48

4.1 Reaction kinetic laws deployed in the current version of ChemChaste. 61
4.2 Transport laws for cell-environment molecule exchange provided in ChemChaste. ..................................................... 69
4.3 Reaction rate laws for reactions occurring at the membrane of the cell ................................................................. 70
4.4 ChemChaste simulation parameter options available in the current release ................................................................. 74
4.5 Global summary statistics describe the whole simulation output. . . 83
4.6 Local summary statistics describe each cell in the simulation. . . . 84
4.7 Parameters used in the Sel’kov-Schnakenberg reaction diffusion simulation. ............................................................... 96
4.8 Parameter values for the cell internal dynamics in the cooperator-cheater model. ....................................................... 101
4.9 Environment parameters for the cooperator-cheater simulation. . . 102

5.1 Summary of the parameters in the Ratzke and Gore (2018) ODE model.112

A.1 PDE coefficients for 2nd-order PDEs are related to the type of PDE which require different solving procedures. .................. 142
# List of Figures

1.1 Cellular cross-feeding and reductive metabolism. .................................. 6  
1.2 Growth oscillations in biofilms may form due to nutrient supply and inhibitor buildup. .................................................. 10  
2.1 Bifurcation diagram for the Sel’kov-Schnakenberg reaction system. . 27  
2.2 Coupling a cell to a reaction system changes the bifurcation diagram and admissible kinetics. ........................................... 29  
3.1 Chaste provides a range of options to describe a cell. ......................... 38  
4.1 A 2D schematic showing the biofilm problem to be simulated. .... 54  
4.2 ChemChaste builds upon the Chaste framework. .......................... 55  
4.3 Schematic for ChemChaste algorithm. ................................................ 58  
4.4 The PDE source term depends on the presence of a cell. .......... 63  
4.5 Schematic of the cell division process ................................................ 71  
4.6 The command file for a ChemChaste simulation. ......................... 73  
4.7 The configuration file for a ChemChaste simulation. ................... 74  
4.8 An example of an individual cell directory. ................................. 76  
4.9 An example environment domain configuration where the different regions of a heterogeneous domain may be defined. .......... 77  
4.10 The ChemChaste reaction and transport law classes follow an inheritance structure. .......................................................... 79  
4.11 Parameter sweeping is performed through value substitution in the ChemChaste configuration and model directory files. .............. 81  
4.12 Example output for the Paraview software for the heat equation with a strip initial condition. .................................................. 88  
4.13 ChemChaste utilises the finite element method to solve PDE problems. .............................................................. 90  
4.14 The direct outputs of Chaste were tested to bypass Paraview analysis. 92
4.15 ChemChaste simulations of the Fisher-KPP equation. .................... 94
4.16 The Sel’kov-Schnakenberg reaction diffusion system showing both oscillatory and patterning dynamics. ................................. 96
4.17 The Sel’kov-Schnakenberg reaction diffusion system simulated through ChemChaste aligned with a separate bifurcation analysis for the homogeneous system. ................................. 97
4.18 Cartoon representing the simulation set-up for the coupled cooperator-cheater system. .......................................................... 100
4.19 Heatmaps for the enzyme, $E$, and substrate, $S$, concentrations across the domain at $t = 10000$. ............................................ 102
4.20 Snapshots showing the growth of a cell community with cooperator-cheater dynamics. ......................................................... 104
4.21 Plot of the chemical concentrations within the progenitor cooperator cells and cheater cells. .................................................... 105
4.22 Parameter sweeping may be used to explore the dynamics of the cooperator-cheater model. ..................................................... 106
4.23 The ChemChaste summary statistics can be used to explore parameter sweeping results. ........................................................... 107

5.1 The Ratzke and Gore (2018) ODE model may be used to determine the population of each species in a community connected through a shared environment. ............................................. 114
5.2 Bifurcation diagram for the Ratzke and Gore (2018) ODE model for a community connected through a shared environment. ............ 115
5.3 ChemChaste simulations demonstrating the effect of cell-environment coupling on the ratio of two species in cell population starting at different population ratios. ........................................... 120
5.4 Plots showing the number of cells of each type in the environment-cell model comparing the coupled to the uncoupled case. .......... 121
5.5 Plot showing the initial and final cell populations for a cell-environment coupled simulation with different initial cell ratios. ....... 121
5.6 Snapshots showing the evolution of a cell population with an environment dependent growth rate. ............................................ 122
5.7 Snapshots of the evolution of the cell community and environment $pH$ with low environment diffusivity. ................................ 124
5.8 Snapshots of the evolution of the cell community and environment $pH$ with high environment diffusivity. ................................. 125
5.9 The growth rate of cells displaying selfish coupling affects the fixation of the cell types. 

5.10 The population ratios vary depending on the growth rate of the individual species. 

5.11 The diffusion of pH through a cell community and changes in the growth rates effect the heterogeneity of cell systems. 

5.12 The introduction of resource competition on the spatial distribution of a selfishly coupled 50-50 mixed inoculation. 

5.13 Schematic showing the impact that resource competition may have on radial stratification in cell communities. 

C.1 Agent based models come in two forms, on-lattice and off-lattice, which defined how the cell agent is described. 

E.1 The coupled cell-environment simulation configuration file system. 

E.2 Cell properties and reactions for Cell A in the coupled cell-environment model. 

E.3 The domain files for the coupled cell-environment model. 

F.1 The resource competition model configuration file system. 

F.2 Cell properties and reactions for Cell A in the resource competition model. 

F.3 The domain files for the resource competition model.
Acknowledgments

This thesis would not be possible without the advice of my supervisor Prof. Orkun Soyer and collaborator Dr. Alexander Fletcher who were of profound help with my understanding the mathematical and computational aspects contained within this thesis. In addition to this the wider Chaste developer community greatly contributed to my understanding of both the inner workings of the Chaste software and my wider knowledge of agent based modelling that so much of my work is built upon. I have been fortunate to interact with such a wide array of founts of knowledge during my PhD and I have learnt a great deal from them.

I would also like to thank the members (both past and present) of the OSS group in the School of Life Sciences for being so welcoming and supportive during my time in the room next to the lab and beyond. I have learnt a great deal about biology and experimental work, even if I still can’t use a pipette.

To my family and friends; thank you for your continued support and understanding. I hope this document in part illustrates what I’ve been up to these last few years.
Declarations

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree. The work contained within chapters 5 and 6 has been performed in collaboration with Dr. Alexander G. Fletcher at the University of Sheffield whom advised on the development of ChemChaste and suggested application avenues. The work contained within this thesis has not been submitted for a degree at any other university.

Parts of this thesis have been published by the author:

Chapter 1 has been influenced by;


Chapter 4 has built upon;

Abstract

Cellular systems are ubiquitous in nature. They form tissues and multicellular organisms where the cell interactions form complex biological structures. Understanding cell system dynamics is of vital importance to a range of fields, such as in medical infection studies and natural product production in industrial bioreactors. Cells change their environment to form ecological niches through the excretion and uptake of chemicals which form complex chemical gradients that influence organisation and signalling between cells. However, how cell growth rate changes and the organisation of different species emerge from these gradients is unclear. Here, we are concerned with developing methods to understand the impact that these feedback cycles may have on the development of cell systems.

Mathematical modelling can help us improve our understanding of such systems with the aim to facilitate the design of informative experiments. However, no general simulation framework is widely available to study these problems. Therefore we introduce ChemChaste, a software framework that couples an extensive agent based model to reaction-diffusion partial differential equation solvers. ChemChaste builds and expands the capabilities of the Chaste software and we introduce a powerful method to solve the complex cell-environment problems necessary to elucidate the complex chemical interactions occurring within cell systems. Using ChemChaste we explore and demonstrate some illustrative cases to show how cell systems can change their environment and how environment feedback can change the species proliferation through simple chemical excretions and uptake. Therein we motivate the importance of considering the cell-environment feedback in modelling cell systems.
Abbreviations

ODE  Ordinary differential equation (i.e $dy/dt$)

PDE  Partial differential equation (i.e $\partial y/\partial t$)

BCs  Boundary Conditions (for a simulation or PDE system)

ICs  Initial Conditions (for a simulation or PDE system)

KPP  Kolmogorov–Petrovsky–Piskunov

DDI  Diffusion-driven instabilities (Turing patterning)

BZ  Belousov-Zhabotinsky (reaction system)

LSA  Linear Stability Analysis

ABM  Agent based model

CA  Cellular automaton

CPM  Cellular Potts model

OS  Overlapping Sphere

HPC  High performance computing

XML  Extensible Markup Language

CSV  Comma Separated Values

FBA  Flux balance analysis
GA Genetic algorithm
AL Active learning
EMEWS Extreme-scale Model Exploration with Swift
CC3D CompuCell3D
HAL Hybrid Automata Library
CFL Courant–Friedrichs–Lewy
GUI Graphical User Interface
CPU Central processing unit
GPU Graphics processing unit
MPI Message Passing interface
FE Finite element
SBML Systems Biology Markup Language
LUT Lookup table
ECM Extracellular matrix
EPS Extracellular polymeric substances
ATP Adenosine triphosphate
ADP Adenosine diphosphate
NAD Nicotinamide adenine dinucleotide
Chapter 1

Bacterial biofilms and niche construction

1.1 Introduction

Cells are considered as the basic units of life. In single-celled organisms, like bacteria, individual cells represent the species and can form communities composed of a range of different species. In multi-cellular organisms, cells differentiate to undertake different roles. The interactions of individual cells can produce complex effects that govern the dynamics of the biological systems.

The emergence of spatial heterogeneity is however vital to develop structured cell communities providing a range of benefits to the cell systems, such as resource management and protection from environmental fluctuations. However, how the cell mechanisms that promote heterogeneity are formed from simple cell interactions are not fully understood. Chemical signalling and resource availability can affect cell proliferation and support heterogeneous cell populations. The cells affect the chemical concentrations in the environment which in turn affects cell proliferation producing feedback dynamics. Cells exchange chemical concentrations with their environment through leakage across a semipermeable membrane and so metabolic processes can indirectly alter the environment chemistry. These changes may promote or inhibit their own proliferation and the proliferation of neighbouring cells. Without a clear understanding and simulation of the chemical reaction, gradients, and cell effects, the level to which the environment affects cell proliferation in a heterogeneous communities or the degree to which the cells control the robustness of the environment remains unknown.

The cell can be seen as a chemical factory (Berlanga and Guerrero, 2016).
It encloses a region of space to form a bounded compartment with distinct chemical conditions that are separated from the surrounding environment. The cell-environment boundary is often semi-permeable and controls the flows of chemical into, or out of, the cell. A set of chemicals can be selectively exchanged with the environment to drive chemical reactions. The internal network of coupled reactions is referred to as the cellular metabolism and produces the cellular building blocks vital for life. These metabolic reactions produce cell bound chemicals, or metabolites, that are used to perform cell functions, such as cell growth and replication. However, unused metabolic products may be excreted into the environment through passage across the cell-environment boundary during metabolic overflow. These excretions change the local concentrations and may further react to produce complex chemical gradients and couple the two reaction systems.

The excretion of metabolic byproducts provides a means for a cell to change its surroundings to produce a customised microenvironment, a so-called environmental niche. Chemicals diffuse from the wider environment replenishing nutrients or removing the build up of excreted metabolites which in turn can alter the niche conditions. These niche conditions may be beneficial or antagonistic to cell growth, either forming a beneficial ecological space to enhance the cell’s manufacturing capabilities or inhibit the growth of the cell species.

The combination of metabolic reaction processing of the nutrient species and environmental nutrient diffusion can produce heterogeneous distributions of nutrient molecules. As the proliferation of a cell is dependent on these nutrients the cell’s metabolic reactions would be effected as the starvation of a particular chemical subset would inhibit metabolic pathways and the associated byproducts (Stewart and Franklin, 2008). Changes in cell proliferation can lead to spatial organisation of the cell species in multi-species communities. Therefore investigating the construction and dynamics of an environment niche can elucidate interactions between the metabolic reactions of a population of cells, where changes in each chemical system may feedback upon the other systems.

In this chapter we discuss the construction of environmental niches and the effects environmental variability has on the cell behaviours and functions. We focus on growth driving nutrients and consider the construction of chemical gradients in biofilms in section 1.2. The spatial organisation of cell populations within a niche can affect the cell behaviours through the uptake and excretion of molecules as considered in section 1.3. We use bacterial biofilms as a model system where the cell-environment interactions produce complex chemical systems. These environments allow different cell species to become coupled through the diffusive
interchange of metabolites. We consider the benefits that metabolite exchange provide to cell populations and consider the capabilities of bacterial cells to construct an environmental niche and motivate the further study of these cell-environment interaction processes.

1.2 Spatial organisation at the cell level: Niche construction in bacterial biofilms

The metabolism is not solely shaped by the genetic code. Changes in the physical and biochemical conditions of the environment can affect the cellular metabolic flux and the degree to which the individual pathways and cell functions are utilised. The availability of the nutrient sources for metabolic reactions is dependent on the local cell conditions and the ability of a cell to take in nutrients. Additionally, nutrient supplies in the environment may be exhausted by extracellular reactions, depleted by competing cells, or removed through diffusion. These effects collectively maintain an out of equilibrium system and act as a driving force behind the metabolism (Johnson et al., 2021). Cells in a population may organise into distinct regions that characterise different environmental niches and this organisation may be driven by these conditions.

An environmental niche is an ecological space. A niche combines a set of spatial locations with the associated environmental conditions (chemical concentrations, temperature, pH etc.) and is used to describe the spatial distribution of resources utilised by the inhabiting species (Slagsvold and Wiebe, 2007; Chaloner et al., 2020). A niche is constructed and influenced by the inhabitants (Hutchinson, 1957) and have been used to study the invasion of a new species into an ecological system (Tingley et al., 2014; Leitão and Santos, 2019). They play a role in both supporting the inhabitant species and accelerating or inhibiting the development of new species (Pili et al., 2020). Niches that are under utilised by the inhabitants, and where an invasive species may be better suited to the niche conditions, will support the invasion of a new species through providing favourable conditions. Conversely, when niche and inhabitants are strongly coupled, a change in the environment may be required for an invasive species to gain a foothold.

A biofilm may be considered as a organised collection of multiple spatial niches which in turn may possess multiple bacterial species and support a heterogeneous community. A niche develops when a subset of the cell species inhabiting the spatial location have an enhanced growth rate due to the niche conditions protecting or providing beneficial nutrients to the favoured species, or
inhibiting the growth of competing species. The environment-niche interaction has been considered as a tailorable system whose conditions may be altered to affect the development of complex communities (Deines et al., 2020; Koza et al., 2011).

1.3 Cell behaviour affects population organisation

The organised structure of a cell colony is linked to niche formation and chemical cross-feeding dynamics between cells. Two possible motivations for the emergence of structure in cell populations are heterogeneous cell motility rates where cells spread in space based on their migration speed, and the formation of chemical gradients via processes such as diffusion inhibition or spatially distributed chemical reactions (Evans et al., 2020). In well mixed scenarios, where the distribution of the chemical concentrations is uniform, spatially distributed phenotypic heterogeneity may produce local niches. These localities support cell function specialisation and metabolic cross-feeding interactions can utilise nutrient resources to increase the metabolic efficiency of the whole population. The distribution of cell functions and cooperation across a population requires robust community interactions that are protected from exploitation or environment destruction.

Local niches present a wider range of physiological conditions in which cells may cluster, presenting opportunities for metabolic and phenotypic diversification. The structure of the colony therefore imparts a role for the formation of chemical gradients where cell species can capitalise on the varying conditions; for example outer layers of a biofilm inhibiting the diffusion of nutrients into the core (Nadell et al., 2006). Even in simple environments where there is a supply of a single carbon source (e.g. glucose), these local conditions are accentuated through the excretion of metabolic by-products. These products can then support complex multispecies communities where the constituents do not all readily process the carbon source. This community support would not be possible without the construction of a niche (San Roman and Wagner, 2018) and has been viewed as a general property of metabolic systems suggesting the important role that niches play in the maintenance of metabolic diversity.

1.3.1 Cellular cross-feeding

Cellular cross-feeding is a community process in which the metabolic excretion of one cell may be taken in and utilised by a second cell. The recipient cell may utilise the excreted product as an additional energy source, a complex metabolite for downstream pathways, or provide a biological function such as the protection from
chemical attack. The cross-feeding interactions may be classified by: whether the cross feeding molecule was fully metabolised into waste by the original cell; whether partially metabolised by the cell and therefore a result of overflow; or whether a metabolic cost for the benefactor was induced during the metabolite production (Smith et al., 2019). The development of cross-feeding systems may originate from the removal of redundancies in metabolically coupled communities, see figure 1.1.

The black queen hypothesis (BQH) is a theory of reductive evolution where a cell’s metabolic pathways may become deactivated as a method of growth cost reduction (Morris et al., 2012; Mas et al., 2016; Smith et al., 2019). If a cell can fulfil a metabolic function without the need to perform a costly enzyme synthesise then to reduced production would confer a fitness advantage. This function may be fulfilled by bacterial communities where cross-feeding facilitate the biological functions in a distributed sense. If the community as a whole can perform the complete set of functions, and the excretion, diffusion, and uptake of metabolites are possible, then each cell may survive and proliferate without the need to produce the metabolites itself.
Figure 1.1: Cellular cross-feeding and reductive metabolism. A cell (cell A) can support a range of different metabolic systems through the conversion of a nutrient resource into a succession of metabolic precursors (red, orange) towards vital metabolites (black). The reactions performing these conversions can have a high cost (dashed arrow) and therefore be advantageous to bypass (cell B). Additionally, a cell may lack the ability to grow on a particular resource and utilise secondary carbon sources from metabolic excretions (cell C). Community interactions are needed to support these optimisations.

This process is however a balancing act. The loss of metabolic functions amongst sub-populations can introduce vulnerabilities into the community. An over-reliance on the generalist species, so-called keystone species, can leave the community at risk of collapse should this species be removed (Smith et al., 2019). The loss of the keystone species would severely inhibit the community’s ability to proliferate, as the reintroduction of removed pathways is a costly process that is unlikely to occur through random genetic mutations alone. Therefore only complex multispecies systems, where secondary pathways exist, can compensate for keystone loss and survive.

1.3.2 Metabolic division of labour

The collection of metabolic reactions within a cell perform a set of functions necessary for the cell to live, grow, and proliferate. The metabolic pathways
utilised to implement these functions may be costly if the synthesis of complex or highly energetic reaction precursors is required. However, these precursors could conceivably be produced in an extracellular environment and taken up by the cell for use in the downstream pathway reactions. In dense biofilms, metabolites may readily diffuse between cells through the extracellular medium and therein metabolic products produced by one cell can be utilised as reaction precursors in another. Here, the metabolic cost of the precursor production has been burdened by a benefactor cell. This phenomena of precursor sharing is seen in heterogeneous biofilms through the so-called metabolic division of labour (DOL) and can increase the population metabolic efficiency.

For pathways producing a final beneficial product, the pathway payoff occurs at the end when the full pathway function is complete. In DOL communities, the cells are segmented into those performing the preliminary metabolic steps and those performing the final functional steps. That is, the preliminary cells bear the resource burden in order to confer a fitness advantage on the functional cells (West and Cooper, 2021). For the community to persist, the metabolic cost that the preliminary cells experience must be outweighed by a cooperation growth advantage (Wang et al., 2021a).

For the community of cooperating cells, the "preliminary" cells act as to support the "functional" cells which gain the benefit of a reduced metabolic cost. This support can therefore perpetuate a biofilm structure within which the functional cells would otherwise become extinct and may be beneficial for the community as a whole through conferring a resistance to chemical/antimicrobial attack or contain a novel metabolic byproduct that may be utilised as a secondary carbon source. Wang et al. (2021a) described the assembly of such a DOL system and the frequency of each cell type through considering the metabolic benefits of the preliminary and functional behaviours.

Whether and when a strain performs the functional or preliminary roles can be viewed as a plastic trait of the cell’s phenotype. The role may change over the course of the cell’s lifetime through altering the cell cycle details encoded within the genome or by changing metabolic phenotype using pathway switching. Therefore evolutionary and game theoretic treatments may be used to explore the role behaviour switching can play in the formation of heterogeneous populations. This switching may be considered as a response to the environment.

The phenomena of phenotypic heterogeneity provides a means for a given cell to respond to variations in the local environment. This response is performed by changing the active metabolic pathways and changing the primary carbon source,
metabolic precursors produced, and the excreted byproducts. These input and output adjustments change the local environment themselves leading to feedback cycles, where changing the environment affects the cell uptake profile and further altering the metabolic byproducts and excretions. As cells proliferate the system complexity increases, potentially leading to multistability or bifurcations in the cell type composition for heterogeneous biofilms (Svardal et al., 2014; Doebeli and Dieckmann, 2000).

1.4 Spatial population organisation affects cell behaviour

The organisation of cells within a niche can change the behaviours that the cells exhibit. These behavioural changes are implemented by switching fluxes through the different pathways of the metabolism. An exemplar case for metabolic pathway switching is the Warburg-Crabtree effect found to occur in yeast and mammalian cells (De Deken, 1966; Hagman et al., 2014). The Warburg-Crabtree effect sees a switch between respiration and fermentation pathways based on aerobic (high oxygen) or anaerobic (low oxygen) environment conditions. In high oxygen conditions, the respiration pathways utilises oxygen as an electron acceptor to drive the redox reactions in the aerobic pathway. However under low oxygen conditions, the fermentation pathway uses organic acids, such as acetaldehyde for the yeast Saccharomyces cerevisiae (Hagman and Piškur, 2015), to accept the electrons through the anaerobic pathway (Gottschalk, 1986). The yields of energy currency metabolite, ATP, and the pathway reaction rates are not equal for both pathways with the fermentation pathway producing 1/9-th the ATP yield per Glucose nutrient molecule when compared to respiration (Pfeiffer and Morley, 2014). These pathways therefore present an high pathway yield versus high pathway rate trade off when may be beneficial for the cell in different environments.

The environmental availability of these electron acceptors or the pathway carbon source can impact the metabolic flux through each pathway. Not all species display the Warburg-Crabtree effect with some being solely respiring or fermenting and whose growth rate may be strongly inhibited under unfavourable environments, i.e respirators in starved anaerobic environments (De Deken, 1966; Colwell and Rangel, 2009; Meyer et al., 2013; Nanchen et al., 2006; Postma et al., 1989; Rieger et al., 1983; Dauner et al., 2001). The ability to switch between the two behaviours may confer a selection advantage to the cell due to the maintenance of a moderate growth rate under either conditions whereas non-switching cells experience starvation. When considering multispecies biofilms, pathway switching can impact
the structure of the heterogeneous cell populations. Cells that exhibit the switchable
behaviours can access a greater range of viable environment conditions and may
support neighbouring non-switching cells through altruistic processes such as cellular
cross-feeding or DOL (Pfeiffer and Morley, 2014).

1.4.1 Examples of spatial dynamics driven by cell-environment
interactions
The spatial organisation of cells is driven by the cell-environment dynamics. The
formation of bacterial biofilms is one example which demonstrates the
cell-environment interactions forming an organised niche in spatial settings. The
formation of biofilms is considered to be involved in a large proportion of bacterial
infections as a response to environmental stimuli that drives genetic expressions
which favour the formation of the protective aggregate structure (Jefferson, 2004;
Hall-Stoodley et al., 2004). The structure of these films is dependent on the chemical
dynamics and may drive patterning in cell species or oscillating expansions of the
film envelope.

An exemplar dynamic for cell-environment interactions is the onset of
oscillatory growth dynamics in bacterial biofilms, shown in figure 1.2. Growth
rate oscillations have been investigated by linking the growth dynamics to the
environmental conditions through the diffusion of signalling molecules through the
environment. These oscillations have been seen experimentally by Liu et al. (2015)
with mathematical model support from a simple 1-dimensional reaction-diffusion
model for the expansion by Mikami et al. (2019). In both experiment and
mathematical model, the oscillations were described by following the concentration
of a nutrient and a growth inhibiting factor, such as ammonium in the experiments or
electrical signals in the modelling. The nutrient was supplied into the environment at
the edge of the biofilm and therefore needed to diffuse into the biofilm. This nutrient
was utilised to drive growth of the film. For small films, the growth inhibiting factor
was low, such as when ammonium concentrations are low due to only a few cells
producing the byproduct, therefore growth was high. This inhibition increased as
the film grew and nutrient penetration through the larger film was reduced. Here,
growth was curtailed and the envelope began to shrink. A later model attempted to
capture some aspects of electric potential signalling and produced similar oscillatory
dynamics (Mikami et al., 2020). These models do not however explicitly consider
the chemical dynamics occurring within the film.
Figure 1.2: Growth oscillations in biofilms may form due to nutrient supply and inhibitor buildup. a) The cells grow through the conversion of a nutrient into biomass. This reaction may produce a byproduct that may inhibit further growth. For small films (red) the nutrient can readily diffuse into the film and the inhibitor diffuse out. For large cells (blue) this diffusion is inhibited leading to an accumulation of inhibitor and a reduction of nutrients. b) When the size of the biofilm envelope is low the nutrients can readily diffuse through the film and growth is promoted. For a large film the nutrients do not readily progress into the interior, inhibiting factors can accumulate and growth is inhibited and cell death leads to a retraction of the envelope, shrinking the cell.

A chemical dynamics view for the biofilm oscillations was investigated by considering the formation of heterogeneous biofilm through metabolic heterogeneity and resource consumption (Bocci et al., 2018). The authors introduced a model for biofilm expansion by initially modelling a biofilm envelope with a linear radial growth before differences in metabolic phenotypes and the changing environmental conditions resulted in an oscillating film radius. A heterogeneous biofilm was simulated where a single bacterial species is separated into two different metabolic phenotypes each possessing different cellular reaction pathways. These cell types formed two phenotypic regions, one cell type in the interior and one in the exterior of the film. The interior phenotype used glutamate as a fuel source for an active glutamate pathway and produced ammonium as a byproduct. The exterior phenotype did not produce ammonium as any resulting product would be lost due to diffusion into the bulk (a so-called futile product). When the cells experienced a high growth rate the film envelope expanded which reduced the glutamate penetration.
through to the film centre and equally reduced the amount of ammonium reaching the film exterior.

As both ammonium and glutamate are required for the cells to grow, an imbalance inhibits the biofilm growth. Cells die when starved of nutrients and the film shrinks, permitting the nutrients to penetrate through the film and restoring a concentration balance for future cell growth. These colony oscillations are not the result of metabolic switching but are due to two static phenotypes which are spatially assorted. It is the diffusion of metabolites drive the oscillatory dynamics.

Different motility rates in multi-species biofilms can also produce complex spatial structures on the macroscopic scale. When the cell population features species with similar growth rates and different cell motility rates, a high motility species tends to push the biofilm envelope and increase biofilm population proportion of the high motility cell (Gude et al., 2020). However, when the more motile species has a lower growth rate the species may be expelled from the core of the colony and left to colonise the periphery. High motility species can reach fresh territory as the biofilm expands and may inhibit the diffusion of nutrients from the bulk environment which reduces the nutrient availability for the low motility cell species. While motility was the focus of Gude et al. (2020), high growth rate and proliferation may provide similar segregation affects in higher density clusters by increasing the population pressure and resulting in frequent cell-cell shunting. Here, the colony expands without cells being actively motile and the composition of species inhabiting the expanding front would depend on the cell growth rates.

1.5 Outlook

In this chapter we have considered cells as analogous to chemical factories that change the environment by selectively controlling uptake and excretion rates and used this as a simplified model of cell systems. Complex cell systems may be modelled through changing which reactions are present in the cell and making the reaction rates dependent on the environment conditions. The population of cells introduce cell-environment feedbacks into the system. The physical properties of the cells affect this feedback as the permeability of the cell boundary facilitates the direct coupling between the two chemical systems. The indirect cell-cell chemical communication through the excretion and uptake of metabolites couples the two metabolic networks providing an opportunity for non-linear interactions to develop. These processes act to increase the complexity of the environment and further the biofilm development through the establishment of a shared environmental niche.
The complex system arising from the feedback dynamics complicates the analysis of bacterial biofilms and an understanding of the spatial dynamics remains underdeveloped. An improvement of the theory behind the chemical dynamics in bacterial biofilms would provide a framework to understand the formation, proliferation, and survival of bacterial biofilms in a wide range of biotechnological and clinical settings. We aim to understand how chemical gradients and spatial patterning of the environment niche form due to the presence of cell systems, with the admissible dynamic phenomena being dependent on the kinetic model and where generally nonlinear equations are used to describe the reaction networks. Here, we develop a hybrid continuum-discrete simulation framework that couples the cell behaviours produced through agent based cell models to biochemical reaction-diffusion systems. We discuss the formation of chemical niche environments in biofilms and the benefits this may introduce for the growth of the cell population. However, a review of current software shows that there lacks a capability to effectively and generally investigate such systems which places a wealth of biological systems out of current simulation reach. In this work we bridge the gap between computational methods and the biological theory, while highlighting the potential chemical complexity of cellular systems.

We develop the hybrid continuum-discrete model in chapter 2 in which discrete cells are coupled to a continuous environment that contains the reaction and diffuse transport of chemical nutrients. We motivate the possible impact of cells on a reaction system by considering the change in bifurcation dynamics. We review a range of software frameworks developed to model cellular systems in chapter 3 and construct a set of modelling aims for the development of a novel software that may fully capture the emergent dynamics of biochemical systems.

In chapter 4 we detail the development of our software, ChemChaste. ChemChaste is an extensive hybrid continuum-discrete simulation suite developed within this thesis to fill a vital gap in research simulation capabilities. We utilise our software in chapter 5 to consider cell-environment feedback cycles in disperse cellular systems. We show how the cell-environment coupling can form chemical gradients in the environment and how these gradients can impact cell proliferation rates. We progressively increase the complexity of the simulation to illustrate the onset of cell species segregation and stratification under competition for nutrient resources. Conclusions are drawn in chapter 6 where we also discuss possible future expansions of ChemChaste.
Chapter 2

The mathematics of chemical reaction systems

2.1 Introduction

The cell may be described as a chemical container which can host a wide range of complex chemical interactions. These interactions are typically described by a system of chemical reaction ordinary differential equations (ODEs) that track how the amount of each chemical constituent change over the interaction process. These ODE systems are constructed through the product of a stoichiometric matrix and a reaction rate vector, where the stoichiometry encodes the proportional changes of each species in a chemical reaction and the reaction rate vector describes the flux through the chemical reaction. The reaction rates are described by a kinetic rate law which models how the reaction progresses; such as modelling the involvement of catalytic enzymes, or capture the influence of the environment on the reaction conditions. The selected kinetic model describes the dynamics of a reaction system and can introduce nonlinearities into the model leading to complex dynamical phenomena (see section 3.2).

The spatially homogeneous cell bound chemical system may be modelled as an ODE system. Under this formulation, the concentrations are uniform across a domain and any spatial effects are negligible. However, in many biological systems the spatial effects are significant and chemical gradients or concentration perturbations prevent a uniform concentration distribution. In cell populations and bacterial biofilms, cells are spatially distributed within a chemically active domain; therefore the effects of heterogeneous reaction systems are significant. The cells are coupled to the domain environment through nutrient uptake and excretion which
affect the local chemical concentrations, perturb the environment concentrations and form chemical gradients. Therefore the well-mixed assumption may no longer be appropriate for biofilm investigations and considering the affects of chemical transport is required.

To consider these spatial chemical models, a set of partial differential equations (PDEs) requires development, which expands the ODE system to account for spatial concentration variation. A pertinent set of PDEs are of the reaction-transport form where the PDEs couple the environment reaction ODEs and cell uptake/excretion into a spatially dependent coupled source term. These sources are used to drive the chemical dynamics while each PDE tracks the transport and reactions of chemical concentrations across the domain. Transport across a biological domain can be complex. These transport forms may be due to passive diffusion, in reaction-diffusion PDEs, or forced through the environment conditions such as driven by advection within a fluid environment forming reaction-advection PDEs, or a combination thereof. These spatial conditions increase the set of dynamic phenomena that may manifest adding another layer of complexity to the chemical dynamics of cell populations and biofilms (Page et al., 2003; Alpkvist and Klapper, 2007; Kondo and Miura, 2010; Halatek and Frey, 2018).

To model the formation of chemical gradients in cell populations, the chemical exchange between cells and their environment is considered with the cell contributing to the source term in the reaction-diffusion PDE. This models each cell as a spatially localised contribution to the concentration fields. The dynamism of the cell and their complex chemical machinery combine to provide a non-static chemical source or sink within the spatial reaction system, which complicates modelling efforts as discussed in this chapter. In this chapter, we develop a hybrid continuum-discrete model of a cell population, construct a reaction-diffusion PDE system (see section 3.2), and present a selection of the possible dynamic phenomena when spatial affects are considered.

2.2 Reaction-diffusion dynamics describe the dynamic evolution of chemical systems

Biochemical systems may be modelled by considering the movement and interactions of chemicals across a spatial domain. However, this distribution of chemicals is dependent on the chemical properties, the set of admissible reactions, and the environment conditions. To simplify the model development we consider a static environment (in the absence of advection flow) and construct the reaction-diffusion
PDE system by considering the physical system where the mechanistic interactions between chemicals drive the reaction dynamics. Finally, we explore the dynamics of chemical waves and the onset of spatial patterning in reaction-diffusion system.

2.2.1 Physical basis for the reaction-diffusion equation

The reaction-diffusion equation may be derived by considering the reaction and diffusion components separately. We consider an N-dimensional domain, \( \Omega \subset \mathbb{R}^N \), containing a continuum scalar concentration field, \( u(x, t) \in \mathbb{R}_{\geq 0} \), that captures the concentration of a chemical at a point \( x \in \Omega \) at a time \( t \in \mathbb{R} \). We wish to describe how this field changes in the presence of chemical reactions (temporal changes) and diffusion (spatial changes).

When developing the appropriate PDE system both the changes due to temporal and spatial effects need capturing. The reaction component considers the concentration change at a point in the domain due to a chemical process. To derive the diffusion component, we consider the conditions where the molecules exist within a surrounding medium, such as within an aqueous fluid environment, and describe the molecular diffusion through this environment. In these systems the diffusive flux can be derived via Fick’s law (Carslaw and Jaegar, 1959).

Fick’s law considers the passive transport of molecules through a region of space. The law is built upon phenomenological observations of the formation of chemical gradients across a region. That is, the diffusion flux through the region is driven by an imbalance in concentration between an area of high chemical concentration to an area of low concentration (Fick, 1855). The concentration gradient through an infinitesimal region;

\[
F(u, x, t) = -D(u, x) \cdot \nabla u(x, t)
\]  

where \( D(u, x) \in \mathbb{R}_{\geq 0}^N \times \mathbb{R}_{\geq 0}^N \) is the general anisotropic diffusion rate tensor describing the diffusion rate in each of the \( N \) spatial dimensions.

The dynamic evolution of the chemical concentration at a point, \( x \in \Omega \) in the domain is described by the continuity equation;

\[
\frac{\partial u}{\partial t} + \nabla \cdot F(u, x, t) = 0
\]

where the temporal change in concentration, \( \partial u/\partial t \) is described by the divergence of the flux through the point. When integrated over the whole domain the continuity ensures that the total concentration in the system, without reactions or flow at
Chemical reactions break the conservation conditions by changing the chemical concentration at a location $\mathbf{x}$ in space. Therefore the reactions act as sources/sinks for the chemicals. Considering both the flux and the chemical reactions in a region of space, the continuity equation can be modified with a source/sink term,

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{F}(u, \mathbf{x}, t) = R(u, \mathbf{x}, t),$$  \hspace{1cm} (2.3)

as the reaction term may be dependent on the location. Different reaction systems may be implemented in different sub-regions of the domain in order to model spatially heterogeneous reaction systems. This heterogeneous capability is particularly relevant for biochemical systems as reactions may occur in spatial compartments where the environment conditions can be controlled, such as forming niche conditions.

The reaction term, $R(u, \mathbf{x}, t)$, describes the change in chemical concentration per unit time therefore the units are given by $[a/\text{length}^N \text{time}]$. After collecting the diffusive and reaction contributions the general reaction-diffusion equation is given by,

$$\frac{\partial u}{\partial t} = \nabla \cdot \mathbf{D}(u, \mathbf{x}) \cdot \nabla u(\mathbf{x}, t) + R(u, \mathbf{x}, t)$$ \hspace{1cm} (2.4)

for the concentration, $u$, of a chemical represented as a scalar field over an $N$-dimensional fluid domain with the passive transport of concentration through diffusion. The dynamics of a particular instance of the equation (2.4) are driven by the reaction model.

### 2.2.2 Reaction kinetics

For each reaction we consider a set of chemicals $\mathcal{C}$ which inhabit a fluid domain and let the set of concentrations for these chemicals be denoted $\mathbf{u} \in \mathcal{R}_{\geq 0}^{\mid \mathcal{C} \mid}$. Each interaction process between a subset of these chemicals is described by a different kinetic mechanism where we introduce a reaction rate term to describe the rate at which these reactions are performed; $\mathbf{R}(\mathbf{u}, \mathbf{x}, t) : \mathcal{R}_{\geq 0}^{\mid \mathcal{C} \mid} \times \mathcal{R}_{\geq 0}^N \times \mathcal{R}_{\geq 0} \rightarrow \mathcal{R}_{\geq 0}^{\mid \mathcal{C} \mid}$. The reaction rates are generally dependent on the concentrations of chemicals in the system, their location in spatial settings (which may affect the system properties) and the time of the reaction, in the case of a temporal change in the system properties. The set of chemicals contains all the chemicals present in the system, including those not involved directly in a particular reaction mechanism. In complex
reaction systems these chemicals may however be involved in further reactions which can introduce a temporal aspect to the reaction dynamics, such as the oscillatory dynamics of activator-inhibitor reaction systems (see appendix B).

In general, chemical reactions are reversible. When reactions involve the binding between two chemicals there is a probability of unbinding which leads to both association, \( k_f \), and dissociation, \( k_r \), rates for the process. A reaction involving a species set \( \{A, B, C, D\} \) is given by

\[
\alpha_A A + \alpha_B B \xrightleftharpoons[k_r]{k_f} \beta_C C + \beta_D D
\]  

(2.5)

where the stoichiometric coefficients, \( \{\alpha_A, \alpha_B, \beta_C, \beta_D\} \), provide the multiplicities that the chemical species are utilised in the reaction. The chemical reaction may be reduced into a subset of substrates \( C_s = \{A, B\} \subset C \), a subset of products \( C_p = \{C, D\} \subset C \), a stoichiometry vector \( S = [-\alpha_A, -\alpha_B, \beta_C, \beta_D] \), and a rate vector \( k = [k_f, k_r] \). The stoichiometric vector shows the proportion that the chemical consumed, \( S < 1 \), or produced \( S > 1 \), during the reaction occurring in the forward sense, i.e the reverse reaction utilises the stoichiometric vector \( S' \) where \( S' = -S \). The kinetic rate law, \( R(u, S; k) \), and the stoichiometric vector are combined to produce the ODE system describing the change in concentration

\[
R(u, x, t) = \frac{du}{dt} = S \cdot R(u, S; k),
\]  

(2.6)

where the functional form of the reaction kinetics, \( R(u, S; k) \), is dependent on the kinetic model selected to describe the reaction process.

Chemical reactions are dependent on the concentration of chemicals in the system. They require the chemical substrates to be physically present for the reaction to progress and therefore the rate at which the reaction can occur is dependent on the concentrations. The law of Mass Action provides a general kinetic law by considering the likelihood of having all of the reaction substrates in one place in sufficient number to perform a one step reaction. This presents a combinatorial problem with the greater the number of substrates the lower the probability of having the sufficient quantities and consequently the lower the reaction flux. The Mass Action kinetic rate is dependent on the product of substrate concentrations while the stoichiometric ratios denote the needed quantities of each substrate. That is;

\[
R^{MA}(u, S; k_f, k_r) = k_f \prod_{c \in S} u_c^{\alpha_c} - k_r \prod_{c \in P} u_c^{\beta_c}
\]  

(2.7)
where the reaction is generally reversible such that the kinetic rate is also dependent on the accumulation of reaction products. Here, the forward reaction rate coefficient $k_f$ describes a rate constant for the reaction involving the substrate set $S \subset C$ and the equivalent reverse reaction rate coefficient $k_r$ for the product set $P \subset C$. Overall, Mass Action kinetics present a standard rate law for biochemical studies which may be tied to experimental systems and the environment conditions through careful selections for the rate coefficients.

### 2.2.3 Developing a hybrid continuum-discrete model for cellular systems

Understanding the impact that cells have on the formation of chemical gradients and dynamics in a biofilm environment requires the development of appropriate reaction-diffusion PDE systems. Cells act as a spatially localised concentration sources and sinks and the amount that the cells excrete or consume can be capture in a transport law. Combining the cell sources with the environment reaction-diffusion equation describes the evolution of the concentration distributions $u = \mathbb{R}^{[C]}$, which is described by the parabolic PDE system coupled to the discrete sources:

$$\frac{\partial u}{\partial t} - \nabla \cdot [D(x) \cdot \nabla u] = R(x, u, t) + \sum_{p \in P} T_p(u, t) \delta(x - x_p), \quad (2.8)$$

where $T_p(u, t)$ describe the cell transport laws for cell $p$ centred at the position $x_p$. The environment reaction system is described by $R(x, u, t) : \Omega \times \mathbb{R}^{[C]} \times [0, t) \rightarrow \mathbb{R}^{[C]}$ formed from a reaction ODE system defined over the domain $\Omega$. The locations of the cells are selected in the continuous PDE through defining $\delta$ as the Dirac delta distribution. Using these delta distributions to model the area over which the cells and environment exchange molecules views the cells as point sources. This approximation loses information about the size of the cell and where along the cell membrane molecule are exchanged however point sources are easier to implement within a simulation.

When a single static point source is considered, a closed form solution may be possible to describe the concentration distribution over the domain. However, when multiple point sources are considered a closed form solution becomes impossible (Gillespie, 2020). Furthermore, when cell processes, such as movement or birth/death, are considered the point sources both change their location and change in number through cell birth and death processes. Therefore a closed form solution for the point source approximation is not readily determined and developing
simulations for the biochemical system becomes a necessity.

When simulations are implemented the locations of the cells in the PDE system may be identified with the location of cells in an agent based model. The ABM can provide a framework to model the complex cell-cell interactions. A natural description for describing the cells would be to couple the domain location to the centre of a cell agent and apply an off-lattice method such as the overlapping sphere model. Therefore an OS model may describe the dynamics and interactions of a population of cells containing internal reaction systems and coupled to their respective locations in the PDE system through the transport laws.

2.2.4 Exemplar dynamical phenomena in reaction-diffusion systems

Many biochemical studies consider ODEs to model the convoluted reaction systems inherent to biological systems (see equation (2.6)). Without the inclusion of chemical diffusion, complex dynamics; such as chemical waves and spatial patterning, cannot be modelled. In general the diffusion of chemical through a space is dependent on the state conditions of the system (Lau and Lubensky, 2007), however these phenomena can be demonstrated with simple diffusion rates where chemical diffusion is isotropic and homogeneous. Therefore a simplification to the general reaction-diffusion equation (see equation (2.4)) is performed;

\[ \frac{\partial u^c}{\partial t} - D^c \nabla^2 u^c = R^c(u) \]  (2.9)

where the scalar concentration field for a chemical species \( c \in C \) evolves due to the isotropic diffusion rate, \( D \in \mathbb{R}_{\geq 0} \) and coupled to the other chemical fields, \( u \in \{ u^c | c \in C \} \), through the source term for species \( c \), \( R^c(u) \). In this section, we consider the analysis of two exemplar reaction systems that demonstrate diffusion drive dynamics in chemical systems; chemical waves in the Fisher-Kolmogorov-Petrovsky-Piskunov (Fisher-KKP) system and spatial patterning in the Sel’kov-Schnakenberg reaction system. These systems will be revisited as test cases for the hybrid continuous-discrete simulations in Chapter 4.

2.2.5 Chemical waves

The progression of a chemical wave through a domain is a well studied phenomena that is both a diffusion dependent mode of transport and is dependent on the conditions of the chemical reactions (Ross et al., 1988; Laughlin, 2015; Deneke and Di Talia, 2018). These wave solutions are present in a range of biochemical settings;
including the spreading of Amoeba cells (Höfer et al., 1995; Pálsson et al., 1997; Vanag and Epstein, 2003) or calcium ion spikes in neuronal cells (Balaji et al., 2017). In biochemical systems chemical waves represent a transport of chemicals from regions of high concentration to regions of low concentration. One well known example of a system that posits travelling wave solution is the Fisher-KPP equation (Fisher, 1937; Avanzini et al., 2019). The PDE,

$$\frac{\partial U}{\partial t} - D \nabla^2 U = r U \left(1 - \frac{U}{\kappa}\right),$$

admits travelling waves through a single scalar field, $U$, where the source term is motivated by the dynamics of a logistic growth form. The dynamics of the logistic growth function, $F(U, t) = r U \left(1 - \frac{U}{\kappa}\right)$, are such that the growth of $U$ is promoted for concentrations above a system carrying capacity $\kappa \in \mathbb{R} \geq 0$ (i.e $U < \kappa$) and inhibited for concentrations above $U \geq \kappa$ where the concentration of the system tends to $\kappa$. This is due to the fixed points of the system being $U^* = 0$ and $U^* = \kappa$. Therefore a wavefront solution for a chemical wave will pass from an area of high $U$ to an area of low $U$ as the introduction of a small amount (or perturbation) of the chemical will be promoted by the logistic growth (Storch et al., 2017). For a general solution through the system is $u(x, t) = U(z)$ where $z = k \cdot x - \sigma t$ were $k$ is the wavenumber, $c = \frac{\sigma}{k}$ the wave velocity and $c = |c|$ the wave speed. For the Fisher-KPP equation the minimum wave velocity may be calculated and seen to be dependent on the linear growth rate and the diffusion coefficient;

$$c_{\text{min}} = 2 \sqrt{rD},$$

however the observed wave velocity, $c \geq c_{\text{min}}$, is in general greater than the minimum due to fluctuations in the initial concentration (El-Hachem et al., 2019). The derived expression for the progression of the wave through the domain may be used as a test for the simulation output, see chapter 4.

### 2.2.6 Spatial Patterning: Diffusion driven Turing instabilities and tipping

In biochemical systems spatial patterning provides a distributed chemical gradient that is fixed in time. The asymptotic steady state for the spatially distributed chemical concentrations form separate regions of high and low concentrations become stable in time. These separate regions develop patterns whose formation is driven by the diffusion rates of the set of chemicals in the space and heterogeneous reaction
environment.

These so-called diffusion-driven instabilities (DDI) depend on the difference in the diffusion rates for the reacting chemical species. While driven by the consideration of diffusion rates, DDI requires particular reaction conditions and the reaction parameters at the onset of patterning are linked to the growth rate of patterning perturbations through a dispersal relation (Guin et al., 2012). Patterning in chemical systems represents stable spatial heterogeneity in the scalar concentration fields capable of supporting long-range order in the system. Patterning due to DDI are also referred to as Turing patterning (Turing, 1952) and chemical patterning can be seen in a variety of chemical systems; such as developmental systems (Barrio et al., 1999) or protein motif models (Glock et al., 2019).

The Sel’kov-Schnakenberg system is one such set of coupled chemical reactions that present both Hopf bifurcation oscillations and Turing patterning (Schnakenberg, 1979; Goldbeter, 1996; Han and Bao, 2009; Liu et al., 2013). The set of 3 chemical reactions describing the interactions between 2 chemical species was proposed as a model for the oscillatory behaviour of the Glycolysis metabolic pathway that metabolises glucose into pyruvic acid with the release of metabolic energy. The Sel’kov-Schnakenberg model has become a standard model for the investigation of dynamical behaviours; such as discrete time chaos control (Din and Haider, 2020), production of semi-analytic solutions (Noufaey, 2021), time delay induced bifurcations (Jiang et al., 2019), and skipping of the onset of Turing patterning (Gaffney and Monk, 2006). The model is a spatial case of the Gierer-Meinhardt activator-inhibitor model (Gierer and Meinhardt, 1972) and similar activator-inhibitor chemical reaction systems have been shown to produce Turing patterns under experimental conditions (Lee and Swinney, 1997; Dúzs et al., 2019).

The Sel’kov-Schnakenberg reaction system involves two chemical species, \(U\) and \(V\), each of which are produced, inter-converted, and removed via the reactions:

\[
\begin{align*}
0 & \xrightarrow[k_1]{k_{-1}} U \\
0 & \xrightarrow[k_2]{k_3} V \\
2U + V & \xrightarrow[k_3]{k_3} 3U
\end{align*}
\]

where the parameters \(k_1, k_{-1}, k_2, k_3 \in \mathbb{R}_{\geq 0}\) denote the reaction rate constants for each process. By applying mass action kinetics to these reactions we obtain the ODE
system model

\[
\frac{dU}{dt} = k_1 - k_{-1} U + k_3 U^2 V \equiv R_U(U, V), \tag{2.12}
\]

\[
\frac{dV}{dt} = k_2 - k_3 U^2 V \equiv R_V(U, V), \tag{2.13}
\]

for the concentrations of \(U, V \in \mathbb{R}_{\geq 0}\) described by the reaction set. Equations (2.12) and (2.13) describe the spatially homogeneous system that can be used to model the reaction system under well-mixed conditions. The homogeneous system exhibits the oscillations and necessary stable states for spatial patterning. To model the onset of patterning we extend this ODE model to include diffusion for chemicals \(U\) and \(V\) with constant diffusion coefficients \(D_U\) and \(D_V\), respectively, leading to a set of coupled PDEs:

\[
\frac{\partial U}{\partial t} - D_U \nabla^2 U = R_U(U, V), \tag{2.14}
\]

\[
\frac{\partial V}{\partial t} - D_V \nabla^2 V = R_V(U, V), \tag{2.15}
\]

where \(R_U(U, V)\) and \(R_V(U, V)\) represent the reaction rate ODEs for the concentrations of \(U\) and \(V\) respectively. The homogeneous system is used to determine the equilibrium states, or fixed points, in the reaction system and the resultant reaction dynamics with the heterogeneous system, equations (2.14) and (2.15), determining the spatial behaviour (see appendix B).

Turing patterns occur for perturbations about concentrations at the Hopf bifurcation point. Subsequently, the transitions between the Hopf oscillatory dynamics and steady state Turing patterns are present in a range of biochemical systems. Hopf bifurcations are categorised by the onset of oscillatory dynamics from a fixed equilibrium concentration, with oscillations being common in metabolic systems (Goldbeter, 1996) and cell excretions themselves may be oscillatory (Adams and Stefanovska, 2021). These bifurcations may be evident when the kinetics parameter of the system support oscillations in chemical concentrations which are common in biology (Guckenheimer and Labouriau, 1993). These kinetics may change due to a variation in the environment condition which may affect reaction kinetics, such as temperature or pH, or the introduction of a new enzyme catalysing the reaction at a different rate and exploring a different point in kinetic parameter space. Under these kinetic changes a previously stable equilibrium concentration may lose stability, which may potentially cause a dramatic change in system dynamics.

The stable and oscillating states are present and may be linked for a range
of chemical systems. The reaction conditions supporting a Hopf bifurcation may be determined through considering linear stability analysis (LSA) (Strogatz, 2015) and stability conditions can be used to determine the necessary conditions for pattern formation via DDI (Murray, 2003; Korvasová et al., 2015; Guin et al., 2012; Gambino et al., 2013). The transition between these two phenomena has been considered by Ledesma-Durán and Aragón (2019). The mixed mode is formed from a linear superposition of a Turing pattern solution and an oscillatory solution both of which describe the concentration vector $u$;

$$u(x, t) \approx T(x, t) \exp (ik \cdot x)u_T + H(x, t) \exp (i\omega t)u_H + CC,$$  

(2.16)

where $T$ and $H$ are the superposition weights of the Turing and Hopf modes respectively, describing their contributions for the particular parameter set at a point in the domain $x$. In equation (2.16), patterning is included as a set of static spatial waves with wave vector $k$ and Hopf oscillations are included as the temporal oscillations with frequency $\omega$. Each regime contributes a corresponding concentration vectors $u_T$ and $u_H$ to the system and $CC$ is included to consider complex conjugate forms for the two modes (due to the complex values).

Oscillations are prevalent in cell physiology, such as flux through metabolic pathways or cell behaviours being tied to circadian rhythms, therefore cell excretions into the environment can also exhibit oscillations. Higher order structures in bacterial biofilms may be described as emergent Turing patterns (Xavier et al., 2009; Karig et al., 2018). Therefore the potential for dynamic transitions between oscillation and patterning is significant. Subsequently, simulations built to model these biological systems need to support modelling of the full range of ODE and PDE phenomena.

### 2.3 The analysis of cell-environment feedback in cellular systems

The coupling between cell and environment can influence cell behaviours. The environment can influence the cells through the formation of chemical gradients or mediating cell-cell interactions through diffusion of chemical signals and metabolite cross-feeding. Additionally, chemical reactions in the environment can lead to the generation of spatial inhomogeneities in the chemical concentrations. These changes in concentrations can in turn alter the spatial transport of signals and metabolites through the environment and further feedback on the cells.
The dynamic evolution of the environment concentrations is described through systems of reaction-diffusion PDEs, as considered in section 2.2, and agent based models describe the mechanical properties of discrete cell agents, as considered in appendix C. For these systems to be coupled a simulation is required to model a reaction-diffusion PDE system with a separate ABM model and couple the separate models through the cell locations. We perform this coupling by considering an exchange of chemical concentrations at a location associated with both the environment and the cell.

2.3.1 Cell inducing environment perturbation and tipping

Cell-environment interactions can influence both the cell behaviours and the environment composition. Cells contain complex internal reaction system which can be perturbed through coupling to an external chemical environment. Cell bound biochemical reaction networks form web-like structures of non-linear interactions capable of supporting multiple stationary states. The number of steady states may be determined from the reaction network by applying theorems of the so-called Chemical Reaction Network Theory (CRNT) (Feinberg, 2019), or for simple systems through inspecting the systems bifurcation diagrams.

Multistability of chemical concentration systems may be seen in a range of reaction networks and can provide a mechanisms to produce switch-like behaviours (Johnson et al., 2021; Sandefur et al., 2012; Kaufman and Soulé, 2019). These behaviour switches may produce a diversion of chemical flux through different reaction pathways. The diversions can be produced through perturbations in the chemical concentrations (Laurent and Kellershohn, 1999) or through changes in the reaction kinetic rates (Otero-Muras et al., 2012), both of which could favour a different dynamical configuration. These multistable states have also been shown to occur in the reactions used in cellular signalling systems and, as such, can be seen as a key component of the interactions of cells with their environment (Angeli et al., 2004; Kothamachu et al., 2015). Hence the formation of chemical gradients within the environment can readily alter the state of the reaction system which in turn influence the local environment concentrations.

A small perturbation of the environment system from changes in cell behaviour can lead to a flip in the long-term concentrations, referred to as tipping. These tipping events can be catastrophic for cell populations as a state in which the cells happily proliferate can rapidly crash towards a detrimental state unable to support the population. A biologically relevant example of this was the early Earth oxygenation which lead to a sharp decrease in oxygen-phobic species.
and the favouring of aerobic species (Jiang et al., 2012). We hypothesise that microbial communities may develop a defence from tipping by controlling their local environment and enhancing their ability to survive.

One possible mechanism for a biological system to resist critical tipping in their environment is the implementation of diffusion driven patterning (Turing patterns) (Rietkerk et al., 2021). This patterning can slow the tipping transition and permit the evolution of biological survival mechanisms and adaptation. The local concentrations formed through the patterning mechanism may settle into different equilibrium states of multistable systems. The resulting spatially organisation can produce regions possessing different concentrations away from the critical states (i.e the fixed points of the environment system affected by the tipping event). This "Turing before tipping" process may be seen through regions of bifurcation parameter space where stable Turing patterns are possible for small changes in the system parameters, a region of parameter space referred to a a Busse balloon (Doelman et al., 2012).

While (Rietkerk et al., 2021) considered tipping in Savannah ecosystems, we may investigate whether the same principles can drive patterning in cellular systems. In biochemical reaction systems environment concentration perturbations occur when chemical source or sinks are present, such as via chemical transport across a cell membrane. These changes affect the rate of change of chemical production and therein act to traverse the bifurcation space. Changes in the bifurcation space alter the admissible equilibrium concentration profiles potentially switching from one stable equilibrium state to another and tipping the state of the system. It remains unclear how waves or perturbations would propagate through these systems of heterogeneous stable states. Therefore the development of simulations to explore this process are required.

2.3.2 Cells as a driver of system state change

The chemical reactions present in an environment can exhibit a range of dynamic phenomena. These possible environment dynamics include settling into global steady states or temporal oscillations in the chemical concentrations. Cells coupled to this environment can change these admissible dynamics present in the environment. For a reaction ODE system these dynamics can be classified through continuation analysis and explored via plotting a bifurcation diagram.

Bifurcation analysis is used to explore the fixed points in the reactions, where the rate of change in concentration is zero (Auchmuty and Nicolis, 1976; Olsen and Epstein, 1991). The behaviours of the system when acted on by small linear
concentration perturbations about these fixed point is analysed. If the perturbation dissipates and the system returns to the fixed point concentrations then the point is classified as stable and represented by a thick solid line in the bifurcation diagram (see figure 2.1). If the perturbation leads to a change in system concentration away from the fixed point then the point is classified as unstable and represented by a thin solid line in the bifurcation diagram.

The presence of oscillatory dynamics in a bifurcation diagram is determined through the presence of a Hopf bifurcation point. The concentrations will oscillate between two different states shown as solid circles in the diagram. These dynamics for the ODE system affect the possible spatial phenomena with oscillations leading to waves and stable states being a necessary condition for spatial patterning (see appendix B). The bifurcation diagram for the Sel’kov-Schnakenberg reaction system is shown in figure 2.1).
Figure 2.1: Bifurcation diagram for the Sel’kov-Schnakenberg reaction system. The Sel’kov-Schnakenberg reaction system equations (2.12)–(2.13), were used to produce the bifurcation diagram through continuation analysis using XPPAUT (Ermentrout, 2002). The fixed points for the concentration of chemical species $U$ are presented when the bifurcation parameter used is $k_1$ and where the other reaction parameters are given by $(k_{-1}, k_2, k_3) = (2.2, 1.5, 1.0)$. The bifurcation plot shows the stable fixed points (black solid line), unstable fixed points (thin black dashed line), Hopf branches (solid circles) and the directions of dynamic evolution (arrows). Two lines of constant kinetic rates are presented (red dashed line and green dashed line) to show the possible long term concentration states for a given reaction kinetic profile.

When cells are added to an environment reaction system they can act as a concentration source for the reaction system and couple their internal concentrations with the environment. We consider the addition of a transport term to the Sel’kov-Schnakenberg reaction ODE system (see equations (2.12)–(2.13)) to
represent the source/sink dynamics of the cell,

\[
\frac{dU}{dt} = k_1 - k_{-1}U + k_3 U^2 V + T_U, \quad \text{(2.17)}
\]

\[
\frac{dV}{dt} = k_2 - k_3 U^2 V + T_V, \quad \text{(2.18)}
\]

where transport terms \( T_U \) and \( T_V \) are added to the ODEs for \( U \) and \( V \). These transport term describes the effect that the cell has on the environment reaction dynamics and may perturb the system away from the dynamics found in the absence of cells. Here, we assume that the cells have a preferred internal concentration for chemicals \( \{U, V\} \) which is regulated by an internal reaction network. Therefore the internal concentrations are held to fixed values \( \{U^*, V^*\} \);

\[
T_U = a(U^* - U)^2, \quad \text{(2.19)}
\]

\[
T_V = b(V^* - V)^2, \quad \text{(2.20)}
\]

where \( a, b \in \mathbb{R} \) are coupling coefficients that control the strength that the cell transport affect has on the environment. For \( a, b > 0 \) the cells act as sources to the reaction system and for \( a, b < 0 \) the cells act as sinks for the chemical concentrations. The form of equations (2.19) and (2.20) was chosen to provide a sufficient demonstration of a change in bifurcation diagram and is dependent on the difference between the cell internal concentration \( \{U^*, V^*\} \) and the environment concentrations \( \{U, V\} \). Different forms may be considered for analysis in future work but here the simple polynomial form was chosen as a simple non-linear system and introduce a second (unstable) steady state into the system. The resulting bifurcation diagrams for varying coupling strengths are shown in figure 2.2.

The bifurcation diagrams show that as the coupling strength is increased, the range of admissible kinetic rates is decreased. Furthermore, for a given reaction rate \( k_1 \) the resulting dynamic phenomena can change when the coupling is varied. For example, the red dashed line exhibits Hopf oscillations changing to unstable fixed points with diverging dynamics as the coupling strength is increased (see figures 2.2-b to 2.2-d). Meanwhile the green dotted line changes from a stable concentration that may support spatial patterning in the heterogeneous PDE system to Hopf oscillations as the coupling is increased.
Figure 2.2: Coupling a cell to a reaction system changes the bifurcation diagram and admissible kinetics. a) The cell is coupled to the environment reaction system through the permeability of the membrane producing coupling parameters \((a, b)\). b), c), d) present the bifurcation diagrams produced using XPPAUT for three different coupling strengths. The bifurcation diagrams show the stability change for equations (2.19)–(2.20) varying over the bifurcation parameter \(k_1\) where the other parameters are \((k_1, k_2, k_3) = (2.2, 1.5, 1.0)\) and changing the coupling strengths \(a\) and \(b\). Two lines of constant kinetic values are present (red dashed line and green dashed line) to show the variation in fixed point concentrations and system dynamics for the different coupling regimes. b) weak coupling \((a, b) = (0.1, 0.1)\). c) moderate coupling \((a, b) = (0.5, 0.5)\). d) strong coupling \((a, b) = (0.1, 0.1)\).

2.4 Outlook

When considering chemical niches within a microbial community, the onset of pattern formation could cement the affects of growing cell heterogeneity or force the system to adapt and adopt new spatial structures. In section 2.2, it was shown that a range of dynamic phenomena was possible depending on the properties of the reaction systems, such as concentration patterning. We can investigate the
admissible temporal phenomena through bifurcation analysis and produce associated bifurcation diagrams. These results will be revisited in chapters 4 and 5 as test cases to demonstrate the ChemChaste software.

Cell division and movement may be described through the cell-cell interactions in a agent based model (see appendix C). Therefore the cells may change their local chemical environments and act as source/sinks to areas previously devoid of cells. The introduction of new or different sources can act to perturb the concentration systems leading to the production of a sequence of metastable patterning. Therefore understanding the coupled cell-environment dynamics in bacterial biofilms requires simulations using an appropriate simulation suite. To this end a selection of simulation packages will be considered in chapter 3.
Chapter 3

Agent based software for modelling cell-environment feedback

3.1 Introduction

Agent based models (ABMs) have found a wide range of applications; from modelling human interactions in economics, to active matter studies, and animal behaviours in ecology. Each field has introduced bespoke versions of ABMs which has led to a range of model formulations and a wealth of software implementations, where each software focuses on different aspects of ABMs. Here we investigate the ability that different software frameworks have for the modelling chemical environment-cell interactions in cellular systems. One issue with having such a wealth of software is that disparate fields build their packages with different foci and features. When a new model question is introduced, navigating the available software to make an appropriate selection can be nontrivial. Introducing new bespoke software would add to the saturation problem unless a particular modelling capability is missing from the current software inventory. As the aim of this work is to model the chemical dynamics in cell systems such as bacterial biofilms, a software needs selecting. Therefore the following section will explore some commonly used software implementations and consider the application of these software to investigate the chemical dynamics of environment-cell feedback.
3.2 Modelling cellular systems using agent based models

Biofilms are composed of a large population of cells with multiple cell types and with complex chemical interactions occurring throughout the space. One method to model biofilms is to build systems of partial differential equations (PDEs) (Byrne and Drasdo, 2009). These PDE systems follow a continuum of cell densities where each cell species is described by a separate PDE. Consequently, these PDE systems lose the fine grained dynamics and interactions of the individual cells. A second method to model these systems is to consider the biofilm as a population of individual agents and apply agent-based modelling methods as discussed in the appendix C.

Modelling the interactions of a set of individuals is a mainstay in investigating complex systems. These so-called "agent-based" models consider groups of autonomous "agents" each determining their own behaviours as specified by a set of "rules". Considering individuals provides a more natural description of a biological system describing the "activities" of a set of agents rather than some abstract dynamical PDE. This viewpoint provides a range of benefits for modelling biological systems over coarse grained PDE models with the view to study the emergent population dynamics (Bonabeau, 2002). This modelling paradigm may be found under a variety of names; agent-based modelling (ABM), individual-based modelling, multi-agent system, or, specifically in biology, cell-based models. With advances in experimental capabilities producing a wealth of single cell observations, the finer resolution of individuals becomes more profitable and hence consideration of such low level models is important.

There are two main methods of representing the agents; as on-lattice objects or off-lattice objects. For on-lattice based models the Cellular Automata (CA) model and the Cellular Potts model (CPM) are two of the most common modalities, while off-lattice models include centre-based approaches such as the overlapping sphere (OS) model. These models differ on their interpretation of what constitutes a single agent within the simulation and therein emphasise different agent properties.

In contrast with PDE based models, agents may have both continuous properties and discrete properties. Continuous properties are modelled by a set of ODEs which may depend on the state of the cell and metabolite concentrations. Discrete properties may be rule based and occur when a set of conditions are satisfied, such as an agent’s division or death being triggered when an internal metabolite concentration exceeds a threshold value or the cell’s age reaches a limit. These properties and rules allow the cell to be more adaptive to the system state than can be readily included in a set of PDEs. These internal cell properties and the properties
of the neighbouring cells can alter the cell-cell interactions, requiring a finer model resolution than PDEs to fully capture possible interactions. These dynamics may contribute to emergent system dynamics that may be lacking in continuum based descriptions.

Coupled cell-environment systems have found wide application in cancer biology and cancer treatment studies with a focus on modelling the tumour microenvironment (Metz car et al., 2019; Norton et al., 2019). While these models have begun the process of coupling ABMs with continuum chemical fields, these models are focused on cancer systems and not on representing the dynamics of chemical systems. Understanding the invasion and dynamics of tumour cells with diffusion of anti-tumour drugs is the target for these simulations, the explicit modelling of environmental chemical reactions falls outside the scope of these studies. Subsequently, modifications to the software are often necessary to provide the flexibility to effectively model the reactive environment and heterogeneity in the environment chemistry.

There is a wide array of software developed to implement ABMs originating from the Sugarscape model originally developed for studies in the social sciences (Epstein and Axtell, 1996). These models considered a set of agents tracking their behaviours and their interactions are encoded within "social" relationships with nearby agents with individuals "reacting" to their environment (Macal and North, 2010). Overtime these original models were used to study biological problems e.g Conway’s Game of life (Gardner, 1970). There is a long history of applying ABMs to investigate a wide range of microbial phenomena, in particular when some spatial heterogeneity is present; for example, ecological models used in water treatment and the formation of biofilms (Hellweger and Bucci, 2008) These models have also found uses in healthcare settings (Silverman et al., 2021).

The development of agent based biological models has been motivated by the improvement in single-cell experiments which have provided a wealth of data on the level of individuals in a population (Hellwege r et al., 2016). Using this data to drive the modelling would provide an understanding of the influence of spatial effects (such as diffusion and boundary conditions) and the emergence of physical community structure. The potential disconnect between in-silico and in-situ experiments can be traversed through the construction of synthetic communities, enhancing single cell measurements, and the utilisation of chemical probes mapping the chemical composition of growth media (Herschend et al., 2018; Bonnin and Rizzoli, 2020).

A chief concern when developing agent based models is the computational complexity of the simulation. This costly problem is further exacerbated when
considering fitting simulations to experimental data or exploring models through parameter sweeping or Bayesian inference methods which require a simulation run for each data or parameter set. Therefore the agent complexity and simulation size needs to be taken into account when constructing a particular model. For anything other than the simplest agent descriptions facilitating the use of high performance computing (HPC) tools are a necessity (Montagud et al., 2021). So-called "Giga-scale" simulations, where the number of cells are on the order of $10^9$, have motivated the need to develop software that may be readily deployed to computer clusters. Therefore there is a push for code to be parallelisable and computationally efficient and the continuous development of previous software is necessary.

ABMs have been developed in a range of programming languages and have aimed to capture different aspects of the physics, chemistry, and biology seen in a range of biophysical systems (Koshy-Chenthittayil et al., 2021). Collections of microbial agents acting under diverse rules present a complex challenge to simulate efficiently and attempts to simulate these problems have spawned a large range of simulation packages. Each package has a range of benefits and constraints and may focus on a particular level of detail and biological realism; have different degrees of support for high performance computing; or different reliance on a user’s programming experience. Here, we consider a cross-section of some of the more popular and general packages and discuss their applicability to study the chemical reactions that drive the complex dynamics in biofilms.

In addition to the software considered in detail a non-exhaustive list of software not comparable to the environment-feedback aims of this work include; FLAME-GPU (Coakley et al., 2012; Richmond and Chimeh, 2017), BacArena (Bauer et al., 2017), LBIBCell (Tanaka et al., 2015), VirtualLeaf (Merks et al., 2011), CellSys (Hoehme and Drasdo, 2010), Timothy (Cytowski et al., 2014), COMETS (Dukovski et al., 2021). These software packages may focus on a simplified cell model suitable for a particular biological domain (e.g VirtualLeaf) or consider a different length scale such as the molecular dynamics simulations of COMETS. The proliferation of these packages demonstrates the crowded domain for agent-based simulation software and the necessity to consider previous offering in detail before embarking on further software development.
For the wide spread adoption of a particular package certain aims have to be achieved:

- A wide range of software applications and user customisability.
- Transparency in modelling assumptions and software capabilities/constraints.
- The software needs to be readily accessible to users of varying coding ability, for example an experimentalist who may have data to understand but potentially limited prior coding experience.
- Support for parameter sweeping and model exploration with parallelisation and/or high performance computing support.
- An adaptable structure facilitating extensions and add-ons to increase the available modelling space.
- A reliable and portable code that runs on a maximal range of computer hardware architectures and operating systems.

With these aims in mind we investigated a range of ABM software to select a base package within which to develop the cell-environment simulations. For full details on a selection of software platforms considered see appendix D. One commonly used software for ABM simulations of particular interest for this work is the Chaste suite.

### 3.3 The Chaste software platform

Cancer, Heart And Soft Tissue Environment (Chaste) is an open source simulation suite written in C++ and is available under the BSD license (Cooper et al., 2020). Chaste, [https://www.cs.ox.ac.uk/chaste/](https://www.cs.ox.ac.uk/chaste/) (accessed on 14/04/2022), was originally developed to study a range of biological problems in biology using 1-, 2-, and 3-dimensional simulations. The multi-scale cell-based chaste simulation consists of a cell simulation with simulation modifiers; including the support for PDE system modifiers. These PDE systems are solved using Chaste’s inbuilt finite element solvers and domain meshing routines. One advantage of the Chaste system is the ability to readily implement models in multiple ABM types, both on- and off-lattice, to reduce the impact of potential simulation artefacts (Osborne et al., 2010, 2017).

Chaste has been developed to act as a toolkit for users to write their own simulation. This is shown with the implemented "test" system where individual simulations are composed of unit tests utilising the CTest framework. This ensures that a user’s additions to the coding environment remain isolated from the core
trunk code and that simulations retain high reproducibility, reliability, and software security. These security measures are complemented by regular automatic testing of the trunk code.

Compartmentalising functions and classes into tests has lead to a modular design structure for the library. Chaste leverages both compile time and run time polymorphism with heavy use of C++ class templating to develop a tree structure that may be adapted to suit the addition of new functionality. Additions inherit from previous trunk code classes which readily incorporate user classes into the underlying simulations. A user may tailor simulations by including simulation modifiers, inheriting from abstract classes, and tailor the capabilities to the user’s model.

Chaste has benefited from a wide development community, with groups focusing and pushing on different trunk code aspect using the agile programming technique (Pitt-Francis et al., 2008). This development is supported by Chaste coding standards and a development "trac" ticketing system and a user support mailing list. The software may be deployed on a wide selection of computer architectures and HPC structures but suffers from multiple dependencies and environment variables that may complicate installation. To remedy this Chaste may be deployed via the Docker system (Merkel, 2014).

The Chaste library supports many ABM modalities including CA, CPM and overlapping spheres and the modelling of both parabolic and elliptic PDEs (see appendix A). The CPM routines have been used to model colorectal cancer (Osborne, 2015), the onset of tissue hypoxia and oxygen-dependent proliferation have been simulated using CA (Figueredo et al., 2013), and cell vertex switching in epithelial tissues has been studied by Fletcher et al. (2013).

3.3.1 Cell model

For simulations where cells are represented by on-lattice models; such as CA and CPM, the populations are constructed with either the Moore or von Neumann neighbourhoods (see appendix C for details). When on-lattice populations are used a "cut-off" distance is defined and the cell neighbourhood is determined through implementing a rectilinear or honeycomb cell packing pattern. In both models, cells are provided with a range of cell property modifiers and cell states which alter the cell behaviours (see figure 3.1). Internal cell state variables are stored using the CellData classes which may store metabolite concentrations or cell phases.

Cells within Chaste are each provided with a cell cycle model. These models may either track the progress of a cell through a series of phases or perform a system
of ODEs to control the cell properties and behaviours. The cell cycle model uses the cell properties to control the cell behaviour. This model may be rule based, such as using the cell ages to trigger cell death or contact inhibited growth, or implement a time-dependent ODE system. A selection of ODE based cell models are provided as template ODE models; such as oxygen dependent growth or using the environment oxygen concentration to trigger hypoxia related cell death. These ODE systems are solved using the Forward Euler method as part of the cell simulation. A random number generator is provided which may be used to trigger stochastic cell cycle models. A cell can make decisions during the cell cycle based on the cell properties of their neighbours, such as the CellData for adhesion molecule or junction concentration for differential adhesion investigations.

To complement the cell cycle models, the internal dynamics of the cell may be modelled using a so-called "subcellular reaction network" (SRN). The SRN implements an ODE system which may track the chemical concentrations of a metabolic network or implement an abstract mathematical model. These chemical reactions require the user to define a CellData property entry for each chemical and explicitly implement an ODE model for the reactions.

The cell cycle model can be used to trigger cell division or cell death events. Cell division occurs through threshold arguments declaring that the cell is in a dividing state, then checking the cell’s neighbourhood for space to divide. If there is room then an offspring cell is placed at a random location in the parent’s neighbourhood and inherits the cell type of the parent. The standard behaviour for cell division is to copy over the cell properties and then call the reset methods on the cell cycle and SRN models. Cell death is triggered when cells are provided with an ApoptoticCellProperty. The property may be added when a so-called "cell killer" event is triggered. These events are added as cell properties from the AbstractCellKiller type which provide a condition for the cell to marked for death. The apoptotic property triggers the removal of the cell at the beginning of the next simulation loop, deleting the cell and the cell properties.
Figure 3.1: Chaste provides a range of options to describe a cell. a) Chaste supports both on-lattice and off-lattice cell descriptions. b) Each cell in the simulation, whether on-lattice or off-lattice, contains a cell model to describe the state of the cell. Chaste supports both phase-based and ODE-based models. In a phase-based model the state exists in a defined state, \( \{M, G_1, S, G_2, M\} \), which changes based on internal checkpoint criteria. In the ODE system the cell state is described by a chemical reaction network. c) Depending on the state, the cell can undergo birth or death processes.

3.3.2 Environment-Cell coupling

Multiscale models are implemented in Chaste by providing reaction-diffusion PDEs and coupling the environment PDEs to the cell models (Figueredo et al., 2013). These PDEs can account for both linear and nonlinear source terms and can also be coupled to ODE systems. To solve these PDEs, Chaste contains finite-element methods with a suite of different meshing routines. Underlying PDE solvers may be used to solve general elliptic or parabolic equations covering both spatial and temporal concentration changes over a domain. When the PDEs are coupled to a cell-based simulation, the value of these scalar fields at the location of the cell agents may be used to alter agent properties, such as triggering cell death or division events. For cells coupled to the environment, cell death may be attributed to reduced nutrient availability, as the environment concentrations falls below a threshold, and cell division events may be linked to environment thresholds. Within the Chaste trunk code the only supported environment-cell coupling occurs for "averaged-source" or "uniform-source" PDEs.
The Chaste user writes their own PDE class deriving from a set of base classes, for each PDE system. For coupled cell-environment simulations, these base classes are AveragedSourceEllipticPde and AveragedSourceParabolicPde for scenarios where the cell source contribution is average over the element, or UniformSourceEllipticPde and UniformSourceParabolicPde when each cell acts as a uniform source. The user may also implement cell independent simulations by deriving their PDE class from the AbstractLinearParabolicPdeSystemForCoupledOdeSystem class. These PDE classes act upon a mesh of finite element nodes and initial PDE conditions are implemented by assigning a concentration vector for each node in the mesh. Therefore heterogeneous initial conditions may be implemented by changing this initial value on a nodal basis.

PDEs that couple cells to the environment implement a reduced form of the reaction-diffusion equation and the cells act as source/sinks to the environment PDE. Let \( \rho(x) \in \mathbb{R}^N \) denote the averaged cell-centre density within an element of the \( N \)-dimensional PDE mesh. A reaction system where the concentrations depend only on the spatial distribution of cells and are independent of time a class deriving from the AveragedSourceEllipticPde abstract class may be used. This class implements an elliptic PDE system,

\[
\nabla \cdot (D(x) \circ \nabla u) + S(x)\rho(x)u = 0,
\]

where \( D(x) \) is the chemical diffusion coefficient for position \( x \) and \( S(x) \) are the cell-source value at position \( x \). For parabolic PDE systems where the concentrations vary over time, the AveragedSourceParabolicPde class is provided,

\[
C \frac{\partial u}{\partial t} = \nabla \cdot (D(x) \circ \nabla u) + S(X)\rho(x)u,
\]

where \( C \in \mathbb{R} \) is a scaling term for the time dependent term.

For a simulation where the direct effect of the cells may be replaced with a mean-field decay term for the chemical dynamics, a user's PDE may derive from the classes UniformSourceEllipticPde and UniformSourceParabolicPde. These abstract classes implement environment PDEs without the cell density source dependency (i.e \( S(X)\rho(x)u \rightarrow S(X) \) in equations (3.1) and (3.2)). Conversely, when the overlapping sphere cell model is implemented and the cells are provided with a radius/volume the VolumeDependentAveragedSourceParabolicPde abstract class may be used to weight the average cell density, \( \rho(x) \), by the radius of the cell. As hyperbolic PDEs are not solvable using Chaste, advection equations cannot be
implemented natively, however fluid flow has been applied by using the HemeLB package (Groen et al., 2013).

Environment parabolic PDEs may also be coupled to an ODE system inheriting from the AbstractLinearParabolicPdeSystemForCoupledOdeSystem class. These PDEs may be used to simulate a chemical reaction system in the environment;

$$\frac{\partial u}{\partial t} = \nabla \cdot (D(x) \cdot \nabla u) + f(x, u) \quad (3.3)$$

where the chemical concentrations vary according to the ODE system $f(x, u) \in \mathbb{R}^{|u|}$. The ODE component provides an effective means to simulate a chemical reaction network but the PDE system is however not coupled to a cell based simulation. Furthermore no symbolic chemical reaction system is implemented in Chaste and a user has to manually code their own reaction ODE system into a C++ class file, inheriting from the appropriate abstract PDE class and following the Chaste code structure.

Chaste supports Dirichlet, Neumann, and periodic boundary conditions (BCs) on the different edges of the domain boundary. BCs are only supported to 3 variables, or 3 chemical concentrations, limited by the underlying boundary conditions container code classes. A user can write their own BCs, such as to implement mixed Robin conditions, deriving from the AbstractBoundaryCondition class. These BCs are implemented on the boundary nodes of the finite element mesh and subsequently different conditions may be provided on different elements of the mesh and for different state variables.

Chaste does not fully consider the role that the environment plays on the growth of the cell population. Reaction-diffusion systems are limited to be separate from the cells, classes AbstractLinearParabolicPdeSystemForCoupledOdeSystem (equation (3.3)) for ODEs or constant decay/production in the environment classes UniformSourceEllipticPde and UniformSourceParabolicPde, or coupled to cells as a constant magnitude source AveragedSourceEllipticPde and AveragedSourceParabolicPde (equations (3.1) and (3.2) respectively). The effect of the environment on the cells has little effect on the cell cycle from the environment. This is limited to a constant influx/efflux linked to the source magnitude and may affect the cell ODE system or trigger rule based events in the appropriate cell models.

3.3.3 Cell mechanics

Chaste supports the development of a wide range of cell mechanic methods. The cell mechanics are composed of a summation of forces from the pairwise interactions
between the cells and their neighbours and forces exerted onto a single cell from the environment. The implementation of force laws are dependent on cell population choice and the force acts upon the location of a node representing the cell; the cell centre for overlapping sphere or pixel of a CPM or CA model. Each force is added as a simulation modifier and considered before updating the cell location. The location of the whole cell population, spreading of a tissue or community, evolves from the environment force contribution and/or shoving of cells due to cell division.

Exemplar forces implemented in the Chaste trunk code include forces derived from two-body interactions and environment mediated forces acting on a given cell agent. On-lattice cells apply force interactions between cells within the neighbourhood and in off-lattice cell models, forces are applied using a cut-off distance to define connectivity between two cells in the off-lattice model. Focusing on off-lattice overlapping sphere model, examples of pairwise interactions provided in the trunk code include generalised spring forces and Buske forces. Buske adhesive and deformation force laws were develop to study cell organisation in intestinal crypts (Buske et al., 2011). These complex force laws are examples of classes derived from the AbstractTwoBodyInteractionForce template class with the force contribution provided by editing the GetMagnitudeOfForce function. For the adhesive pairwise force the BuskeAdhesiveForce class implements;

\[
F_{ij} = 2A \left( \frac{(r^2_i - r^2_j + |x_i - x_j|^2)/2}{|x_i - x_j|} \right) \left( 1 - \frac{(r^2_i - r^2_j + |x_i - x_j|^2)/2}{|x_i - x_j|^2} \right) \frac{(x_i - x_j)}{|x_i - x_j|}
\]

where \( A \in \mathbb{R} \) is a cell-cell adhesion energy, \( r_i \) and \( x_i \) are the radius and position of cell \( i \) which acts along the line of separation between cell \( i \) and cell \( j \). For the deformation force BuskeElasticForce implements,

\[
F_{ij} = \frac{(r_i + r_j - |x_i - x_j|)}{\delta} \left( r_i r_j/(r_i + r_j) \right)^{3/2} \frac{(x_i - x_j)}{|x_i - x_j|},
\]

where \( \delta \in \mathbb{R} \) is a deformation energy parameter. Chaste includes a generalised spring force to model simple cell-cell mechanical interactions,

\[
F_{ij} = k_{ij} l_{ij} |x_i - x_j| \frac{(x_i - x_j)}{|x_i - x_j|},
\]

where \( k_{ij} \in \mathbb{R}_{\geq 0} \) is the spring stiffness constant for a spring of natural resting length \( l_{ij} \in \mathbb{R}_{\geq 0} \). These spring forces may be used to implement basic cell shoving within a community.
In addition to pairwise forces, general forces acting on a cell may also be modelled through editing the \texttt{AddForceContribution} function in inherited classes of the \texttt{AbstractForce} template class. These classes act on each cell pixel of the CA and CPM models or the cell centre of the overlapping sphere model. For example the application of a chemotactic force (\texttt{ChemotacticForce} class) acting on cell \( i \) due to a concentration gradient of an environment species, \( \nabla u \), is given by,

\[
F_i = f(u_c, \nabla u) \frac{\nabla u}{|\nabla u|} \nabla u, \quad (3.7)
\]

where \( f(u_c, \nabla u) \in \mathcal{R} \) is a force magnitude factor as a function of internal cell chemical concentration \( u_c \) and the environment gradient magnitude \( \nabla u \). For \( f(u_c, \nabla u) > 0 \) the cell will move towards higher concentrations, with \( f(u_c, \nabla u) < 0 \) the cell will move towards lower concentrations.

As forces may be applied on a cell by cell basis cell migration may be implemented by applying a force that is cell dependent. For example, a user may model cell diffusion by adding the \texttt{DiffusionForce} force class;

\[
F_i = \nu \eta \sqrt{2D_{cell}} \frac{dt}{dt} \quad (3.8)
\]

where \( \eta \in Uniform([-1, 1]) \) is a uniformly distributed random variable, \( \nu \in \mathcal{R} \) a motion damping constant, and where \( D_{cell} \in \mathcal{R} \) is the diffusion coefficient of the cell assuming the cell may be modelled as a spherical object and applying Stoke’s law. As the diffusion is dependent on the time step the cell simulation timestep, \( dt \), is used to calculate the force.

### 3.3.4 Model implementation and exploration

Chaste is provided as a library from which individual simulations may be constructed. The standard workflow for a Chaste simulation is for a user to build a new project which acts as a bolt-on to the core Chaste trunk code (Mirams \textit{et al.}, 2013). The design philosophy behind Chaste was to develop a flexible toolkit to implement a wide range of cell models therefore no GUI is provided. This lack of GUI coupled to a \texttt{C++} code heavy user interface may restrict the accessibility of Chaste. Furthermore Chaste can be unwieldy for a new user due a steep learning curve that needs to be traversed before models may be implemented. However, Chaste provides a wide scope for applications with adaptable packages.

Users of Chaste adapt the toolkit by writing their own cell properties, ODE, and PDE classes with the view to build upon the limited models available in the
trunk code. For environment-cell interactions the simulation of chemical reactions is required. Chaste does not supply a general chemical reaction input method or chemical equation interpreter so the equivalent ODE system needs to be hard-coded into an ODE C++ class. One method to input chemical reactions would be to incorporate an SBML model however the necessary interpreters have so far only been included on an ad-hoc basis in order to model cell migration in crypts (Romijn et al., 2020). These aspects combine to inhibit the adoption of Chaste in biological communities. However, for more general simulations a range of tutorials on the Chaste website are provided to act as examples on how to design and add new cell properties, ODEs, and cell mechanics to a simulation.

Chaste utilises the Docker container system for cross platform software deployment. This containerised deployment system ensures that the necessary dependencies and environment variables are set correctly through the standardised user file directory tree. This removes the need for direct installation of Chaste dependencies with only the need to install the Docker software. For high performance computing, Docker may be used as a method to deploy a user project on distributed systems. The project files can be edited either within the container or on the user’s local computer and symbolically linked to the local directories to the Docker directories. The same linking can be performed for the Chaste output directories for local post-processing and visualisation.

The simulation outputs may be tailored by the user. The general output for a cell based simulation includes the cell positions and environment PDE results using a standardised VTK format. The developers recommend visualising this output with the ParaView software. This output may be complimented by invoking cell output writer classes as modifications to the simulation. The output writers can act on the individual cell, inheriting from the AbstractCellWriter class, or as a summary value for the whole population, inheriting from AbstractCellPopulationWriter. The output data can include cell property data, including the cell ages, and cell forces. Users can implement more complex output values by writing custom cell and population writer classes.

Parameter sweeping is not directly implemented within Chaste. However a user may convert tests into executable files or applications and call these applications using a Python, or Bash, command script. These scripts can utilise coarse, or naive, parallel algorithms to distribute the executable across processors and compute clusters using MPI. The user is free to implement simulation optimisation algorithms or advanced parameter sweeping in their command scripts on the prior simulation output.
3.4 Model development in systems biology

System biological and cell based models can utilise a wide array of software packages and model paradigms. One requirement for a biologically relevant model is that the results need to be independent of the software implementation or mathematical description of the simulation method. Therefore a method of model description that aims for portability between simulation packages is required. These portable descriptions would ensure reproducible of results that are agnostic of the simulation methods employed (Sandve et al., 2013).

Within these model descriptions the aim for accessibility in the description writing experience is needed for wide adoption by users. The use of chemical reaction interpreters to translate chemical equations into ODE systems under a selection of kinetic laws simplifies the modelling process. Subsequently, model development software packages have been produced to help a user less familiar with chemical reaction ODEs construct such systems. These packages, such as Antimony and COPASI (see section 4.4), allow a user to develop chemical reaction systems and cell conditions using familiar chemical notation.

One consideration when selecting a modelling software is whether to use a general discipline-spanning software with a wide application base or use a more specialised software to implement a particular model within a particular field. General software can implement a wide array of models with contribution from a varied development community, whereas specialised software tend to have a smaller user base but can offer more tailored advice. Furthermore, using general software can improve reproducibility of the results as a larger user base can perform the simulation and simulation artefacts will be consistent over a range of models. For reproducibility to be tested models need to be readily shared between different users. Two methods of sharing biological models, COMBINE and SBML, will be considered alongside two applications to produce SBML models, Antimony and COPASI.

3.4.1 COMBINE

An agent based model may require a range of different file types and metadata to fully describe the in-silico experiments and allow full reproducibility of the work. COMBINE (Computational Modelling in Biology Network) has been proposed as an archiving standard to contain, document, and disseminate biological models in a reproducible manner (Bergmann et al., 2014; Walker and Southgate, 2020). The archive is distributed as a compressed "ZIP" file containing a hierarchy of files capturing aspects of the simulation parameters/setting using appropriate mark up.
languages.

For each model archived, COMBINE includes an extensible Markup Language (XML, (Bray et al., 1998)) file that lists the contents and parsing formats pulled from an externally hosted list of possible format specifications. These archives may then be parsed by appropriate software and implemented into a user’s model pipeline. The key benefit of such a system is that all files needed for reproducing the work are contained within the one archive folder and that these may be created and exchanged separately to the user’s architecture and simulation or distributed onto separate computing systems with no corruption of data or model.

3.4.2 The Systems Biology Markup Language (SBML)

A common method of incorporating parameters and model descriptions into systems biology software is the use of mark-up languages. The Systems Biology Markup Language (SBML) is a popular method of disseminating in silico models developed to aid reproducibility in biological simulations. The SBML models may then be shared by uploading the model to a repository, for example the BioModels repository (Glont et al., 2018; Malik-Sheriff et al., 2020), then downloaded by a user and used to implement a ABM using a compatible software.

SBML is an XML based markup language to encode biochemical reaction networks involved in systems biology. The standard is broken into levels which facilitate a gradual expansion of a stable standard. SBML level 1 encodes biochemical reactions (Hucka et al., 2003). A level 1 model breaks the model into a series of non-spatial compartments each with a specified volume. Chemical species are then provided for each compartment and provided with initial concentrations. These species are then used as products and/or reactants for a set of reactions where a simple kinetic law formulae may be provided. SBML level 3 provides the capabilities to consider spatial properties for the simulations (Hucka et al., 2018; Keating et al., 2020). SBML-spatial is a sub specification of the SBML level 3 specification and provides the standards to convey reaction-diffusion systems. Molecular transport via diffusion and advection is added to the level 1 biochemical reactions and the compartments in which the reactions are active are mapped to spatial domains referred to by instances of their DomainTypes in the specification.

3.4.3 Model development platforms (Tellurium and COPASI)

To support the introduction of a standard model specification the development of software to read and write the models are a necessity. Model development
platforms provide fully fledged analysis and simulation suites to develop and inspect biochemical models and then export the models to well defined model standards, such as COMBINE and SBML. Two example tools that support the development of SBML files following the Level 1 specification are Tellurium (Medley et al., 2018) and COPASI (Mendes et al., 2009).

Tellurium is a simulation environment that provides a suite of tools to create and edit SBML models wrapped into COMBINE archives. Tellurium may be used through a Jupyter notebook development environment that combines Python code blocks, in-application data visualisation, and equation mark up languages (Choi et al., 2018). Tellurium also supports the use of bifurcation tool to perform a bifurcation analysis of the chemical reaction system using the AUTO package.

To aid in the development of chemical reaction systems, Tellurium includes the Antimony code library (Hoops et al., 2006), a chemical reaction language that allows a user to write the chemical system which may then be converted into the SBML standard (Smith et al., 2009). Antimony uses a text based system for a user to input chemical reactions. For each reaction, a reaction name, chemical equation, are accompanied by a mathematically written reaction rate law and parameters values. More complex systems can be constructed by encapsulating reactions into modules which can be imported in future reaction models. This is particularly useful coupled with Antimony compartments, mirroring the SBML capabilities. Overall the text based system can readily produce SBML standard models and a GUI application has been included to aid model development.

COPASI (COmplex PAthway SImulator) is a software developed to simulate complex reaction networks for the development and simulation of biochemical reaction models. To implement models, users select from known kinetic laws with automatic rate constant unit determination or a user may implement general mathematical reaction rates. Both sets of reaction rates are converted into a system of ODEs for simulating by the inbuilt ODE solvers. This reaction modelling is complemented by parameter estimation and optimisation with in-application visualisation. Models can be built through a bespoke GUI which has been developed to be readily used by a wide range of users. These models may be exported in a range of standard specifications, including SBML Level 1.

3.5 The case for ChemChaste

The selection of software packages considered in the previous sections presents a wide range of functionalities and aspects benefiting cell-environment modelling and
<table>
<thead>
<tr>
<th>Model</th>
<th>Cell model</th>
<th>Cell bound reactions</th>
<th>Cell-cell interactions</th>
<th>Cell physiology</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;well-mixed&quot; ODE model</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>PhysCell</td>
<td>OS</td>
<td>Yes</td>
<td>No</td>
<td>Limited</td>
</tr>
<tr>
<td>CompuCell3D</td>
<td>CA, CPM</td>
<td>No</td>
<td>Yes</td>
<td>Limited</td>
</tr>
<tr>
<td>Chaste</td>
<td>CA, CPM, OS,</td>
<td>No</td>
<td>Yes</td>
<td>Limited</td>
</tr>
<tr>
<td>Morpheus</td>
<td>CPM</td>
<td>Yes</td>
<td>No</td>
<td>Limited</td>
</tr>
<tr>
<td>iDynomics</td>
<td>OS</td>
<td>Yes</td>
<td>Yes</td>
<td>Limited</td>
</tr>
<tr>
<td>ChemChaste</td>
<td>OS</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 3.1: Summary of a selection of agent-based simulation software focusing on the cell descriptions. Description of the cells changes the accuracy of capturing the biological details into the simulation. The specific cell model effects the simulation capabilities of the software and the ability to model cell bound reactions using chemical equations enables greater user model implementation. The cell-cell interactions change the shape and location of the cell community and the cell physiology, such as cell cycle model or death and birth conditions, change the state of each cell in the community.

are summarised in tables 3.1 and 3.2. The packages considered both open and closed source software implementing CA, CPM and OS cell models. Different environment models were also considered, from simple diffusion to reaction-diffusion-advection system. However, no one software implements the coupling of agents to a reaction-diffusion system with enough flexibility to consider reactions occurring both in the environment and within cells.

The software descriptions were also presented alongside a selection of studies that were performed by using the software packages. A selection of these studies may be implemented in a range of software with few changes to the underlying model. For example, the Cellular Potts studies can often be performed in any of the packages that supports CA simulations. The choice of software is often due to software specificity, interface usability, and model exploration tools. The specific tailoring of functionalities; such as cell linking in CC3D (see section D.2) or implementing the hexagonal lattices and 3-dimensional simulations of Morpheus (see section D.3), may only be necessary in a small niche of settings.

The user interface experience can inhibit adoption by the community, the code heavy Chaste (see section 3.3) requires complex scripts to be written. The reduction of mathematical complexities and minimisation of a "learning curve" are thus critical for an effective software. In addition, maximising the ease of installation and portability across operating systems is vital for a wide adoption. Meanwhile, model exploration is a key aspect for large scale, large cell number, simulations and designing code that may be tailored to high performance computing is a necessity (Cytowski and Szymańska, 2014).
<table>
<thead>
<tr>
<th>Model</th>
<th>Environment reactions</th>
<th>Diffusion</th>
<th>Environment-cell feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;well-mixed&quot;</td>
<td>Yes</td>
<td>No</td>
<td>Limited</td>
</tr>
<tr>
<td>PhysiCell</td>
<td>No</td>
<td>Homogeneous</td>
<td>No</td>
</tr>
<tr>
<td>CompuCell3D</td>
<td>No</td>
<td>Homogeneous</td>
<td>No</td>
</tr>
<tr>
<td>Chaste</td>
<td>Limited</td>
<td>Homogeneous</td>
<td>No</td>
</tr>
<tr>
<td>Morpheus</td>
<td>Limited</td>
<td>Homogeneous</td>
<td>No</td>
</tr>
<tr>
<td>iDynamics</td>
<td>Limited</td>
<td>Homogeneous</td>
<td>No</td>
</tr>
<tr>
<td>ChemChaste</td>
<td>Yes</td>
<td>Heterogeneous</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 3.2: Summary of a selection of agent-based simulation software focusing on the environment descriptions. The capabilities for modelling the environment in order to simulate cell-environment systems is considered. The ability to model environment reactions, using user defined chemical equations, is necessary with modelling of molecular transport between cells through the environment which requires an appropriate diffusion solver. Diffusion coefficients may be used in a homogeneous manner, where the same rate coefficient is used throughout the domain, or heterogeneously, where the value changes due to location. To understand the impact that the environmental concentrations have on the cell state, and vice versa, the simulations should be capable of modelling the feedback between both systems.

In summary, current software tend to have small niches where they are unique and present a large overlap in capabilities at the cost of repeated development time. Each implementation vies for widespread adoption but the lifespan of these software tools depends on their adoption by the scientific community that desire the lowest learning curve, most general capabilities, flexibility to adapt the software to their study, and usability. These end-user aspects are coupled with development needs for strong software standards and an open source nature. Therefore we selected one of the largest communities, Chaste, and utilised the software to provide a general chemical reaction implementation.

### 3.5.1 Developing software to model the environment-cell interactions in biofilms

Bacterial biofilms are able to build complex local environments, as discussed in chapter 1, where the chemical concentrations in these environments are capable of generating complex dynamical phenomena as described in chapter 2. The previous software platforms are not suitable to model the emergent environment dynamics as they implement a simplified model for the environment. As the general "rule-of-thumb" for computational modelling follows an Occam’s razor approach, the simpler the model the more effective the simulation, these simpler models require fewer parameters to fit and are more robust to over-fitting. However, as these
complex dynamics requires non-linearities and environment reactions such models may miss the emergent dynamics and the effects of coupling cell populations to the environment may remain elusive.

In order to understand the chemical dynamics in biofilms and other spatially aggregated systems the simulations need to focus on the chemical interactions and environment modelling which is a departure from the cell focus of the packages considered. Readily available packages tend to lack the ability to model arbitrary non-linear reaction systems in both the environment and the cell agents. The common approach adopted by packages is to limit the general reaction systems to occur only within cells which then act as a simple source to the environment. Therefore there is a need to develop a software suite to enable the study of such models and we propose the development of a new simulation package, ChemChaste, that builds upon Chaste.

ChemChaste will aim to span the environment and cell modelling gap present in the reviewed software with a focus on modelling chemical dynamics. This change of focus does not require the full development of an ABM simulation but may leverage the power of previous software offerings and provide an extension for chemical dynamic modelling. Therefore we build upon a general framework rather than develop a new software, saving development time and enhancing, rather than diluting, the software selection. The Chaste software (see section 3.3) provides a suite of finite element routines which may be adapted to solve the necessary environment models and provides a wide variety of cell based simulations.

3.5.2 Why build upon Chaste in particular?

Chaste is a mature agent based modelling software that balances computational efficiency against model usability where a wide array of biological phenomena may be investigated (Cooper et al., 2020). The toolkit has been applied to an array of different studies and has subsequently developed both a large user and development communities with active engagement and community support via the Chaste mailing list. The wide software application scope has evolved from the low level access afforded to the underlying solver routines and easy to merge additions, however this flexibility comes at the cost of lower user accessibility requiring advance C++ features to be written. Therefore by utilising the power of the simulation suite to form an application with an approachable front end interface the platform can be productively harnessed for environment-cell simulations.

Chaste provides a range of cell models, see tables 3.1 and 3.2, each with different optimal application scales. Cellular Potts type models are readily applied
to connected tissue studies, such as epithelial buckling investigations, overlapping spheres for microbial interactions and mechanical interactions, and cellular automata for computationally efficient large scale studies. As the aim of this work is to study the metabolic-environmental chemical feedback in biofilms, we build upon the overlapping spheres model subset of Chaste. The OS model is appropriate for considering the fine grain cell-environment reaction and we wish to perform chemical reaction network within each cell which required cell based ODE solvers, such as the Forward Euler methods deployed in Chaste (Osborne et al., 2017). The overlapping sphere model was selected to provide a balance between accurate cell modelling and computational efficiency while providing a strong couple to reaction-diffusion modelling provided within Chaste.

While the development focus is on the OS model, there is large applicability for the chemical methods to be adapted to cell based simulations forming an adaptable framework. Therefore following the modular design of Chaste and future proofing for the adaptation to the other cell methods will be considered in the design process. The application of the chemical interactions across a range of biological scales and consider potential merging with other development branches of Chaste will provide ample future work.

3.5.3 Development aims for ChemChaste

We set some development aims for ChemChaste;

1. Couple the cell simulation to a general environmental reaction-diffusion system

2. Support general reaction kinetics in order to model complex chemical reactions in both the environment and in cells

3. Implement heterogeneous diffusion and reaction rates within sub-domains of the environment

4. Develop a user interface to support adoption by non-programmers

5. Provide a range of summary metrics to support the analysis of large simulation data sets

6. Provide parameter sweeping functionality capable of sweeping over reactions and kinetics

7. Portable code that may run on any computer operating systems and cloud based high performance settings
8. Provide a tailor-able simulation capable of future development

9. Minimise the needed changes to the Chaste trunk code to prevent incompatibilities

Aims 1-3 will be fulfilled by introducing a hybrid continuum-discrete method to Chaste. The general reaction kinetics will take the form of bespoke reaction rate laws that a user may write as a C++ file and incorporate in to the ChemChaste system through constructing an ODE system from the reactions. Heterogeneous diffusion and reactions will be implemented through customising the Chaste solvers to alter the diffusion constant and reaction ODEs for different locations within the simulation domain.

Model analysis will be considered through aims 4 and 5. The user interface will intake a user’s chemical equations and cell specification through a file based system and a Python command script will be provided to execute the simulations and provide parameter sweeping. This will be a code-lite user experience with additional specific functionalities possible by a user coding new rate laws. These simulations may then be analysed through simulation metric outputs. These outputs will be the environment chemical concentration fields, internal cell concentrations, cell positions, and summary metrics. The simulation summary measures may be used to evaluate large data sets from multiple separate simulation runs. To facilitate this data acquisition aims 6 and 7 ensure large sets of simulations may be run on a variety of hardware using parallel algorithms within Python. As Chaste may be run on HPC or cloud based computing services, such as the Amazon Web Services (AWS), these parallel algorithms may be leveraged. Aims 8 and 9 are future facing and concern the longevity of ChemChaste as a software package. The modular nature of Chaste will be followed with the ChemChaste codes to ensure an effective merging between the underlying trunk code and the extension.

3.6 Outlook

This section has reviewed a range of software packages that may be suitable for modelling the environment-metabolic feedback in biofilms and other spatially aggregated biological systems. After considering the advantages and disadvantages of these packages the case to introduce an add-on to the Chaste software has been made. This extension, ChemChaste, will extend the ABM and PDE solver features of Chaste to focus on chemical interactions. The development aims for ChemChaste will be to produce a reproducible, parallisable software in order to implement hybrid
continuous-discrete simulation of coupled cell-reaction-diffusion systems within an
ABM while being accessible to users with a low programming experience. This
software is developed and tested in chapter 4.
Chapter 4

The development of ChemChaste

4.1 Introduction

Bacterial biofilms are systems that couple cells with a shared environment as discussed in chapter 1. Each cell in the film contains an internal chemical reaction system which changes the chemical concentrations internally to determine the cell’s behaviour, such as death and reproduction, which may alter the location or number of each cell in the film. These cell intake chemicals from the environment and excrete reaction products into the environment. These excretions may react with chemicals in the environment; such as during nutrient harvesting (Pentland et al., 2021) or the excretion of extracellular matrix substrates (Hung et al., 2013; Limoli et al., 2015), to drive internal cell reactions or inhibit motility of cells or chemicals. In large groups these chemical exchanges can strongly affect the environment to produce conditions such as those found in a biofilm, see figure 4.1. These films, and other general cell systems, have been shown to exhibit spatial patterning through interactions with their environment (Karig et al., 2018; Xavier et al., 2009; Thomen et al., 2020). A suitable simulation framework needs developing in order to explore the potential complex spatial dynamics of a reaction-diffusion system when coupled to a cell population (see chapter 2).

The framework would need to combine a reaction-diffusion system capable of solving arbitrary non-linear reactions with point sources. These point sources would need to be able to move through the simulation space to account for the motility of cell agents and couple the environment to internal reaction systems within each cell. This combination of environment and cell systems would account for complex chemical dynamics in the environment while retaining the ability to model the biology of a cell. These cells can move through the domain, either through
passive cell-cell shunting or active motility, and divide to produce new cell offspring. These methods need to handle the emergence of travelling chemical waves or spatial concentration patterning in the environment while revealing possible cell segregation and cell type heterogeneity. As shown in chapter 3, such a general framework platform is not readily accessible. Therefore we build upon the Chaste software (see section 3.3) to develop ChemChaste to fulfil this research software gap.

Figure 4.1: A 2D schematic showing the biofilm problem to be simulated. a) The cells within a biofilm interact between themselves and the film environment. The in-take of nutrients from the environment and excretion of metabolic byproducts change the chemical composition of the local cell surroundings. Changes in a cell’s local environment can affect neighbouring cells through changing the chemical availability for the neighbouring cells. b) As cells in the biofilm grow they divide to produce offspring cells. The repeated divisions push the envelope of the biofilm to cover more of the environment space and provide more nutrients. Different cell types within the film may prefer the environment conditions and have a greater growth rate leading to imbalances in cell count for the different cell types.

ChemChaste adds a suite of C++ classes to the Chaste trunk code (see figure 4.2). The new classes are integrated into the adaptable modular structure of Chaste which was chosen due to the powerful underlying solver and cell simulation methods. We combined these solvers and cell-based methods to introduce a hybrid continuum-discrete simulation which can implement reaction-diffusion models in heterogeneous domains while coupling to cell based models. However, Chaste is presented with a purposeful "toolkit" structure. This may present a steep learning curve to a new user due to a reliance on the user possessing a strong programming ability. Therefore we introduce a file based user system to assist with model development and model reproducibility by facilitating the sharing of simulation files and results. Simulations are initiated through a command script that may implement parameter sweeping routines and be incorporated into a user’s pipeline.
A set of development aims was declared in section 3.5 and the development of ChemChaste was performed to meet these criteria. The ChemChaste codes are available from the GitHub repository https://github.com/OSS-Lab/ChemChaste under the BSD license and may be readily deployed in an architecture free manner using the docker container system. The codes are presented in section 4.2 with a description of model implementation and the user interface is discussed in section 4.3. The stability and convergence of solutions for both Chaste and the ChemChaste add-on are verified in section 4.5. To demonstrate the capabilities of ChemChaste, a set of exemplar simulations are provided in section 4.6.

Figure 4.2: ChemChaste builds upon the Chaste framework. The Chaste classes (red boxes) provide the finite element PDE solvers and agent based models and are built upon by the ChemChaste additions (green boxes).

### 4.2 The ChemChaste simulation

ChemChaste is a simulation framework that couples the Chaste reaction-diffusion solvers to a centre-based cell simulation to model groups of cell agents in a chemical environment. This includes an expansion of Chaste’s solvers to consider any number of chemical reactions in the environment. ChemChaste expands Chaste by:

1. Modelling any number of chemicals diffusing in the environment
2. Implementing reaction spatial heterogeneity; heterogeneous diffusion rates for
each chemical and a system to implement this

3. Adding an internal reaction system to control cell behaviour through modelling a metabolic network

4. Providing a file based user interface to facilitate software adoption

where the expansions are necessary for ChemChaste to be applied to bacterial biofilm modelling.

The formation of chemical gradients in the environment produce varying local concentrations that may impact cell growth. These concentrations are used by cells to drive complex internal reaction systems. These reactions can feedback into the environment through metabolic overflow and membrane transport which perturb the environmental reaction system (Newman, 2016; Höfer et al., 1995; Glock et al., 2019).

To fulfil the simulation needs, a ChemChaste simulation consists of 4 steps that implement "rules" to control the simulation progress:

1. The performance of cell dynamics; the cell mechanics and internal cell dynamics
2. The solving of reaction-diffusion systems in the environment
3. The exchange of cellular flux between the cell and environment models
4. Adding or removing agents within cell division and death steps

where each step occurs within one simulation timestep (see figure 4.3). These steps update the location and the chemical concentrations internal to each cell, the environment reaction-diffusion system, the transport of chemical concentrations across the cell membranes, and implement cell division and death to introduce a new agent or remove an existing one. These processes may be customised to implement a particular system model through editing the simulation configuration files or the underlying ChemChaste executable.

All of the ChemChaste simulation steps occur for each simulation cycle so that the simulation time step is used for both the reaction-diffusion PDE solvers and as the mechanical time step. Therefore in ChemChaste we do not explicitly consider the different time scales inherent to cell mechanics and reaction systems. The decision to use the same time step was made to closely follow the underlying Chaste code structure and ensure a smooth integration between cell and environment. However, using the smallest time step in multi-scale simulations can lead to unnecessary computation for the larger scale sub-systems and using separate time steps may be considered in future releases.
Cells are represented using a centre based model where the properties of the cell are associated with a point in the domain. One of the properties is a radius value to model a spatial extent through the implementation of overlapping spheres of influence to determine which cells may perform pairwise interactions. While a range of different cell models are available in Chaste the overlapping sphere model was used to simplify the cell-environment coupling as only one exchange point is required for the RD solvers. As these cells are point sources distances between cells are arbitrary and can be decided upon in relation to the other simulation units in order to model the density of the biofilm.

The standard measurement units in ChemChaste are inherited from Chaste. In ChemChaste time is measured in hours and space steps are in terms of the order of one bacterial cell diameter, 1\( \mu \)m both of which equate to a unit simulation time and distance. These units are however arbitrary and used as a guide for converting diffusion rates and mechanics time scales. The scale of the values should be chosen such that the simulation results are free from artefacts and pass "rule-of-thumb" tests.
Figure 4.3: Schematic for ChemChaste algorithm. A single timestep in the ChemChaste algorithm follows a cycle of 4 steps. 1. The cell dynamics which solve the internal cell bound reaction system and determine the cell behaviours. 2. The environment concentrations are determined by implementing a reaction-diffusion PDE system. 3. The coupling of the cell model and environment is performed by a concentration exchange across the membrane. 4. The cell behaviours, division or death, are performed resulting in the addition or removal of cell agents from the simulation.

4.2.1 Cell dynamics

The first step of the ChemChaste algorithm is to update the agent based models inherited from Chaste. This includes applying the cell behaviour rules to determine whether a cell division or a death event is triggered and updating the internal chemical concentrations by applying the cell reaction networks. These cell processes are part of the ABM and implemented separately from the PDE domain.
Cell mechanics

ChemChaste utilises the NodeBasedCellPopulation class from Chaste which models cell agents as point-like particles. This class supports an overlapping sphere model for cell-cell interactions (see section 3.3). A connectivity cut-off distance, $d_{\text{Cutoff}}$, is provided to denote the region of influence for implementing. The default value $d_{\text{Cutoff}}$ is 1.5 however a user may change this by altering the value node_cutoff_length in the simulation configuration file.

Cell mechanics are implemented using the Hookean linear spring to describe the interactions between the OS particles. In Chaste this is implemented using the GeneralisedLinearSpringForce class whose form is derived from Meineke et al. (2001). This force class is defined with a cutoff distance, linear_force_cutoff (with a default value of 1.5) to describe a limit to the extent that the force acts which is available to be changed in the configuration file. Parameters for this force acting between two cells $(i, j)$ are the rest spring length $l_{ij}$, the spring growth duration for new cells $g$, and the spring stiffness coefficients $k_{ij}$. In ChemChaste these parameters are given pre-determined values; $(l_{ij}, F_{\text{Cutoff}}, g, k_{ij}) = (0.5, 1.5, 1.0, 15.0)$. While these values are not freely chosen through the configuration files, they may be changed through altering the ChemChaste executable. Likewise, any of the Chaste forces (see section 3.3) may be added by a user through editing the ChemChaste.cpp executable.

Cell internal reactions

Cells can contain complex internal reaction systems. This network of reactions may possess general reaction kinetic laws which is supported by ChemChaste. Two commonly used reaction types are mass action kinetics using the MassActionReaction class for general reaction systems and Michaelis-Menten reactions using the MichaelisMentenReaction class. ChemChaste is, however, deployed with a suite of reaction laws (see table 4.1) and provides a system for a user to incorporate a custom law (see section 5.3).

Each reaction (see equation 2.5 for details) performed within ChemChaste follows a canonical form given by equation 2.6 where $R(u; k)$ describes the reaction rate model used. The change in concentration over the course of the reaction for chemical $c \in |C|$ is given by an ODE system:

$$\frac{du_c}{dt} = (\beta_c - \alpha_c)R(u; k),$$

where $R(u; k)$ is the reaction rate and $\beta_c, \alpha_c \in \mathbb{N}_{\geq 0}$ are the stoichiometry coefficients...
for the reaction (see section 3.2).

The change in the chemical concentration is determined by the difference in the number of molecules used as reaction substrates and reaction products with the rate of the change determined by the reaction model. For \( R(u; k) > 0, \beta_c \geq 0 \) if chemical \( c \) is a product reaction and \( \alpha_c \geq 0 \) if \( c \) is a substrate consumed within the reaction. These coefficients are provided to consider the event that a reaction may use a chemical both as a substrate and a product, such as in an autocatalytic reaction. If \( R(u; k) < 0 \) the reaction direction is reverse and the products become the substrates and vice versa. The rate coefficients \( k \) may be constants describing the reaction dynamics or capture aspects of the reaction environment such as the thermodynamic conditions.

A reaction system may contain chemicals that are not directly involved in the reaction process. These chemicals may be fully removed from the reaction, and therefore have a concentration value in the system but do not feature in the set of reactions (passive spectators), or are involved during rate calculation without being consumed/produced (chemical catalysts or enzymes). In ChemChaste nomenclature we refer to these chemicals as "spectator species", forming a chemical subset \( \tilde{S} \subseteq C \), however a reaction rate can be dependent on the concentration of these species. Spectator species can be used for example to model signalling metabolites where the set of enzymes is denoted \( \tilde{e} \subseteq \tilde{S} \).

The cell metabolism may be dependent on the environmental conditions. For a simplified cell model, the concentration of chemical species in the environment, such as signalling molecules or environment conditions (such as \([H^+]\) for pH sensitive conditions), may affect the reaction rates of cell bound reactions. Therefore in ChemChaste we introduce a set of environmentally dependent reactions where the cell has a "preferred" concentration for an environment species.

For a chemical \( c \) from the set of environment chemicals \( E \), we consider the species concentration in the environment \( u_c^{env} \in \mathbb{R}_{\geq 0} \) and allow a cell to have a preference for a particular concentration \( u_c^{pref} \in \mathbb{R}_{\geq 0} \). We calculate the distance between the cell preference and the environment value, \( |u_c^{env} - u_c^{pref}| \in \mathbb{R}_{\geq 0} \), and use this as a weighting factor to reduce the reaction rates if the environment concentrations differ strongly from the preferred conditions. When there is a large difference between the preferred and actual concentrations the weighting factor can reduce the reaction rate and inhibit the cell’s growth. The reduction in rate will be the same whether the actual concentration is higher or lower than the cell’s preference.

The addition of the preferred reaction class adds a dependency for the
particular environment onto the cell metabolism. These reactions can be used to switch off particular reactions when there is a divergence from preferred conditions and the remaining reaction system may drive the system towards, or away from the preferred concentrations. While in general the reactions in table 4.1 may be used in both the cell and environment reaction systems, the environment dependent reactions may be defined within the cell only.

<table>
<thead>
<tr>
<th>Reaction name</th>
<th>Reaction rate formula $R(u)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZerothOrderReaction</td>
<td>$k_f$</td>
</tr>
<tr>
<td>ZerothOrderReversibleReaction</td>
<td>$k_f - k_r$</td>
</tr>
<tr>
<td>MassActionReaction</td>
<td>$k_f \prod_{c \in P} u_c^{\alpha_c} - k_r \prod_{c \in S} u_c^{\beta_c}$</td>
</tr>
<tr>
<td>SpectatorDependentReaction</td>
<td>$k_f \prod_{c \in S} u_c$</td>
</tr>
<tr>
<td>MichaelisMentenReaction</td>
<td>$k_{cat} u_{\tilde{e}} \sum_{c \in S} u_c^\beta_c / (K_M + \sum_{c \in S} u_c^\beta_c)$</td>
</tr>
<tr>
<td>EnvironmentDependentReaction</td>
<td>$k_f \prod_{c \in E} u_c^{env}$</td>
</tr>
<tr>
<td>PreferredEnvironmentReaction</td>
<td>$k_f \exp \left( - \prod_{c \in E} u_c^{env} - u_c^{pref} \right)$</td>
</tr>
<tr>
<td>PreferredEnvironmentMassActionReaction</td>
<td>$k_f \prod_{c \in P} u_c^{\alpha_c} \exp \left( - \prod_{c \in E} u_c^{env} - u_c^{pref} \right)$</td>
</tr>
</tbody>
</table>

Table 4.1: Reaction kinetic laws deployed in the current version of ChemChaste. Forward reaction rate coefficients are given by $k_f$ while reversible reactions use $k_r$ for the rate coefficient for the reversed direction. $k_{cat}$, $K_M$ are Michaelis-Menten kinetic parameters where $u_{\tilde{e}}$ is the enzyme concentration.

### 4.2.2 The environment system

A fundamental contribution by ChemChaste is the coupling of the Chaste ABM to the Chaste PDE solvers to develop the so-called hybrid continuum-discrete simulations. Chaste includes a suite of finite element (FE) solvers capable of solving both parabolic and elliptic PDE systems in 1, 2, and 3-dimensional spatial domains.
(see section 3.3). Part of the FE method is to cover the domain with a mesh and the PDEs are solved on the mesh nodes at the element intersections with the solution being interpolated onto Gauss points of the elements. The current version of ChemChaste considers 2-dimensional domains with a triangular mesh and couples this to an off-lattice centre-based overlapping sphere agent based model (see section 3.3). The cell agents contain an internal concentration system that determines the cell behaviours and the interpolated Gauss points are used as locations for the cell agents.

4.2.3 The environment reaction-diffusion system

A reaction-diffusion PDE system is implemented to model the environment dynamics where the source term are composed of both the reaction and cell contributions. In general, reaction systems may occur at all points of the domain $\Omega$ while cell sources occur at the discrete location of each cell. Let the cells be denoted by $i \in A$ where $A$ is the set of all cells in the population. As a centre-base representation is applied to the cell they may be associated with a point location in the domain $x_i \in \Omega$. Therefore environment reaction systems occur over all points $x \in \Omega$ and cell contributions are considered at point locations $x_i$ for $i \in A$.

The Chaste FE routines are used to solve the coupled reaction-diffusion system for each chemical in the environment. The state variable $u \in \mathbb{R}^{|C|}$ describes the concentrations of each environment chemical and evolves through the ChemChaste environment PDE system in $N = 2$ spatial dimensions. The PDE system governing the change in environment concentrations is given by;

$$\frac{\partial u}{\partial t} - \nabla \cdot [D(x) \cdot \nabla u] = R(x, u, t) + \sum_{i \in A} T_i(u, t)\delta(x - x_i), \quad (4.1)$$

where $R(x, u, t)$ and $T_i(u, t)\delta(x - x_i)$ provide the source concentration vector due to reactions or coupled cell $i$ at position $x$. The environment change due to reactions and transport laws that describe the transport of chemicals across the cell membrane (the cell-environment chemical exchange) are determined during one simulation cycle.

The chemical reactions in the environment utilise the same set of reactions as the cell bound reaction systems (see table 4.1) and the cell-environment transport laws (see table 4.2) are implemented after the reaction systems are solved. For each location the source terms are determined by the presence of a cell where the source due to transport is added to the environment reactions (see figure 4.4).
Figure 4.4: The PDE source term depends on the presence of a cell. The source term in the reaction-diffusion PDE depends on whether there is cell present for each point \( x \) in the domain. Without a cell the source term is solely composed of the environment reaction term, \( R \) at the point. However, when a cell is present the source due to transport across the cell boundary, \( T \), is also incorporated.

The diffusion coefficient matrix is given by \( D(x) \in \mathbb{R}^2 \times \mathbb{R}^2 \) and while systems with general anisotropic diffusion rates are solvable using Chaste, ChemChaste restricts \( D(x) \) to a diagonal diffusion matrix and isotropic cases where the diagonal elements are equal. The environmental dynamics (equation (4.1)) are considered as an initial boundary value problem (IBVP) subject to a set of initial conditions and boundary conditions for each of the chemical concentrations. ChemChaste supports the use of BCs of either the Dirichlet or Neumann types (see appendix A.2). For chemicals without a user defined BC, ChemChaste defaults to Dirichlet BCs with a zero value. BCs can be defined on each side of the rectangular (or square) domain or for the whole boundary.

4.2.4 Implementing a heterogeneous domain

In addition to coupling the PDE system and agent based model to form the hybrid continuum-discrete simulation, ChemChaste enables the construction of a heterogeneous domain. The domain is label into a set of regions, \( S \), and a lookup table (LUT) is used to provide the spatially labelled diffusion rate and initial concentration value. Both the initial environment conditions and the diffusion rates use separate LUTs so that they can both vary separately.

For the diffusion values, the boundaries between the different regions are modelled as discontinuities in the diffusion value. These discontinuities are considered using a top-hat distribution over the locations \( x \) where the amplitude values are the diffusion rates \( D(x) \) determined from the LUT. The initial conditions use the same regions and top-hat distributions using the initial values in the equivalent LUT as the values \( u(x) \). For the region \( \Omega_s \subseteq \Omega \) where \( s \in S \), the initial concentration state variable vector and the diffusion rate for position \( x \in \Omega_s \) is;

\[
\begin{align*}
  u_0(x) & = \sum_{s \in S} u_s 1_{\Omega_s}(x) \quad (4.2) \\
  D(x) & = \sum_{s \in S} D_s 1_{\Omega_s}(x) \quad (4.3)
\end{align*}
\]
where $\mathbf{u}_L^s$ and $D_L^s$ are the LUT values for the initial condition vector and diffusion rate for region $s$ respectively and;

$$1_{\Omega_s}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_s \\ 0 & \text{if } \mathbf{x} \notin \Omega_s \end{cases}$$

is the region indicator function providing the discontinuities in the top-hat distribution. This method is also used to select the environment reaction systems using a separate LUT.

### 4.2.5 Solving the coupled system through the finite element method

The Chaste software is built upon a suite of underlying finite element solvers. The finite element (FE) method utilises variational procedure to produce the integral of a differential equation system over a domain. The method discritises the spatial domain through applying a mesh and an implicit time integration scheme is used across these mesh elements. The finite element method is favoured compared to finite difference schema in simulations where the diffusion coefficients is either discontinuous, as produced from the ChemChaste LUT, or rapidly varying (Ženíšek, 1990; Bonito et al., 2013; Tzou and Stetchmann, 2019).

For solving the environment dynamics in ChemChaste, equation (4.1), we consider the time discretised form of the problem,

$$\frac{1}{\Delta t} \mathbf{u}^{n+1} - \frac{1}{\Delta t} \mathbf{u}^n - \nabla \cdot [D(\mathbf{x}) \cdot \nabla \mathbf{u}^{n+1}] = R(\mathbf{x}, \mathbf{u}^n, t^n) + \sum_{i \in A} T_i(\mathbf{u}^n, t^n) \delta(\mathbf{x} - \mathbf{x}_i),$$

where the semi-implicit time derivative has been used for discrete time steps $t^n = n\Delta t$ and concentration state variable vector of $\mathbf{u}^n$ for the $n$-th timestep. Here, $\Delta t$ is the time step for one cycle of the ChemChaste simulation (see figure 4.3). The weak formulation of the problem is then produced from this temporally discretised version by multiplying by a vector of "test" functions, $\mathbf{v}$, where the test function for each scalar field, $v \in \mathbf{v}$, belongs to a Sobolev function subspace, $v \in \dot{V} \subset H^1_0(\Omega)$. The weak formulation is then integrated over the domain, $\Omega$;

$$0 = \frac{1}{\Delta t} \int_{\Omega} \mathbf{u}^{n+1} \cdot \mathbf{v} \, dV - \frac{1}{\Delta t} \int_{\Omega} \mathbf{u}^n \cdot \mathbf{v} \, dV - \int_{\Omega} \nabla \cdot [D(\mathbf{x}) \cdot \nabla \mathbf{u}^{n+1}] \cdot \mathbf{v} \, dV$$

$$- \int_{\Omega} \left[ R(\mathbf{x}, \mathbf{u}^n, t^n) + \sum_{i \in A} T_i(\mathbf{u}^n, t^n) \delta(\mathbf{x} - \mathbf{x}_i) \right] \cdot \mathbf{v} \, dV.$$
where \( \cdot \) and \( \circ \) represent the vector product and the Hadamard (element-wise) product respectively. By re-writing the diffusion term, applying the divergence theorem, and substituting the timestep \( t^n = n\Delta t \) the integrals may be simplified to decompose the boundary effects:

\[
0 = \frac{1}{\Delta t} \int_{\Omega} u^{n+1} \cdot \mathbf{v} \, dV - \frac{1}{\Delta t} \int_{\Omega} u^{n} \cdot \mathbf{v} \, dV + \int_{\Omega} D(\mathbf{x}) \cdot \nabla \mathbf{v} \cdot \nabla u^{n+1} \, dV \\
- \int_{\partial\Omega} [\mathbf{v} \cdot D(\mathbf{x}) \cdot \nabla u^{n+1}] \cdot \hat{n} \, dS - \int_{\Omega} \left[ R(\mathbf{x}, u^{n}, n\Delta t) + \sum_{i \in A} T_i(u^{n}, n\Delta t) \delta(\mathbf{x} - \mathbf{x}_i) \right] \cdot \mathbf{v} \, dV.
\]

Equation (4.5) denotes the variational form of the problem wherein a set of basis functions may be chosen to satisfy the boundary conditions for \( \mathbf{v} \) (Logg et al., 2012). These basis functions may be constructed by considering a finite mesh placed over the domain and the points within the mesh elements being composed of the coordinates of these basis sets.

The FE method benefits from the ability to simulate PDE systems over complex geometries, such as the discontinuous regions considered in ChemChaste, within a domain through triangulation (Shapira, 2012). During triangulation, the domain space is discretised into triangular elements composed of sets of nodes and edges. The nodes sit at the vertices of the triangle where two adjacent triangles share two nodes and an edge and, at the domain boundary, the triangle edge and the edge’s associated nodes line parallel to the boundary. The FE method solves the linear algebraic system on the triangle nodes and interpolates the solution on to points within the triangle element. A triangulation, \( T \), for a 2-dimensional domain and using linear Lagrange elements covers the domain \( \Omega \) (Logg et al., 2012).

The triangle elements covering the domain define the Sobolev space where the linear basis, \( \phi \), is used to describe the triangle elements can be used to construct the test functions in equation (4.5). A labelling of the nodes of the triangle element,
\{\tilde{n}_x, \tilde{n}_y, \tilde{n}_z\} = \{(0, 0), (1, 0), (0, 1)\} alongside a linear basis of the form
\[\phi_x(x) = 1 - x_1 - x_2\]
\[\phi_y(x) = x_1\]
\[\phi_z(x) = x_2\]

provides a coordinate system for interpolated points (the so-called Gauss points) in the element \(x = \sum_{i \in \{x,y,z\}} \tilde{n}_i \phi_i\). Using the same basis, the value of a function defined at the three nodes of the triangle may be interpolated to provide a function value at a point in the triangle:

\[\mathbf{v} = \sum_{i \in \{x,y,z\}} \mathbf{v}(\tilde{n}_i) \phi_i\]

where \(\mathbf{v}\) is the value at the Gauss point, \(x\), and \(\mathbf{v}(\tilde{n}_i)\) describes the function of the variable at the triangle nodes. In a ChemChaste simulation, the test function is determined on a reference triangle element, with standard linear triangle coordinates \(\{\tilde{n}_x, \tilde{n}_y, \tilde{n}_z\}\) which is mapped onto the full domain \(\Omega\).

Using the triangulation and the mapping of the reference triangle to the "global" node location \(\mathbf{n}\), the location of the element nodes can be interpolated into a Gauss point for the domain:

\[x = \sum_{j=1}^{3} \mathbf{n}_j \phi_j\] (4.6)

and a concentration calculated at the Gauss point for time point \(n\):

\[\mathbf{u}^n_c = \sum_{j=1}^{3} U_{c,j} \phi_j\] (4.7)

where \(U_{c,j}\) is the concentration of chemical \(c\) at the \(j\)-th node of the mesh element. The gradient of the concentration state variable may be calculated through the gradient of the basis functions:

\[\nabla \mathbf{u}^n = \sum_{j=1}^{N} U_j^n \nabla \phi_j\] (4.8)

Restricting the "test" function to the basis functions, \(\phi_i\), where the boundary conditions are satisfied on \(\partial \Omega\), the variational formulation used by the ChemChaste
solvers may be written as,

\[
0 = \frac{1}{\Delta t} \int_{\Omega} \sum_{j=1}^{3} U_{nj}^{n+1} \phi_j \cdot \phi_i \, dV - \frac{1}{\Delta t} \int_{\Omega} \sum_{j=1}^{3} U_{nj}^{n} \phi_j \cdot \phi_i \, dV \\
+ \int_{\Omega} D(\sum_{j=1}^{3} n_j \phi_j) \cdot \nabla \phi_i \cdot \sum_{j=1}^{3} U_{nj}^{n+1} \nabla \phi_j \, dV - \int_{\partial \Omega} \phi_i \cdot D(\sum_{j=1}^{3} n_j \phi_j) \circ \mathbf{b} \, dS \\
- \int_{\Omega} \left[ R(\sum_{j=1}^{3} n_j \phi_j, \sum_{j=1}^{3} U_{nj}^{n} \phi_j, n\Delta t) + \sum_{k \in A} T_p(\sum_{j=1}^{3} U_{nj}^{n} \phi_j, n\Delta t) \delta(\sum_{j=1}^{3} n_j \phi_j - x_k) \right] \cdot \phi_i \, dV,
\]

which may be solved through linear algebra operations by converting the variational problem into an algebraic system;

\[
\left( \frac{1}{\Delta t} M + K \right) U^{n+1} = \frac{1}{\Delta t} MU^n + B
\]  

(4.9)

where the stiffness matrix \( K \), the mass matrix \( M \) and the source vector \( B \), are defined as,

\[
K_{ij} = \int_{\Omega} D(\sum_{j=1}^{3} n_j \phi_j) \cdot \nabla \phi_i \cdot \nabla \phi_j \, dV, \quad M_{ij} = \int_{\Omega} \phi_i \phi_j \, dV,
\]

and,

\[
B_i = \int_{\partial \Omega} \phi_i \cdot D(\sum_{j=1}^{3} n_j \phi_j) \circ \mathbf{b} \, dS \\
+ \int_{\Omega} \left[ R(\sum_{j=1}^{3} n_j \phi_j, \sum_{j=1}^{3} U_{nj}^{n} \phi_j, n\Delta t) - \sum_{k \in A} T_p(\sum_{j=1}^{3} U_{nj}^{n} \phi_j, n\Delta t) \delta(\sum_{j=1}^{3} n_j \phi_j - x_k) \right] \cdot \phi_i \, dV,
\]

respectively. ChemChaste applies Dirichlet BCs to the simulation by changing the values of the boundary integral in \( B_i \) to the Dirichlet value. The boundary integral naturally contains the Neumann BC values fulfilling the test function requirements.

### 4.2.6 Cell-environment coupling: transport laws and membrane bound reactions

During a ChemChaste simulation, the cell and environment are coupled through transport processes across the cell boundary. The transport exchanges a portion of the concentrations between the environment and the cell at the Gauss point location of the cell. Additionally, reactions may occur at the membrane boundary that change
the concentrations on both sides of the cell-environment separation at a coupled
reaction rate. Cells in ChemChaste are modelled as point-like particles therefore
they do not have a defined surface area or volume. For the transport and membrane
reaction processes, cells are considered to have a unit area surface for the transport
to occur over. Therefore the surface area of the cell is not used to scale the process
rates and an ODE system is solved to determine the amount of material passing
through the cell boundary or changing due to reactions at the boundary.

Transport laws are solved to determine the flow of chemical concentrations
across the cell boundary. They couple two chemical environments with chemical sets
\( C \) and \( C' \) for the environment and the cell bound system respectively with associated
concentration vectors \( u \) and \( u' \). As the chemical environments are not identical the
sets of chemicals may differ in general, such that \( |C| \neq |C'| \), and a chemical only
defined in one system cannot be transported into the other. For example, a scenario
where a charged molecule may not pass through a membrane but the neutral form
can be implemented through a reduction in the charged molecule and an increase
in the neutral form (Herrera-Valdez, 2018). In ChemChaste these two forms would
constitute two separate species.

The transport laws provided in the current release of ChemChaste are
provided in table 4.2 and are used to produce an ODE system. As shown in equation
(4.1), the form of the law is given by \( T(u, u', t) : \mathbb{R}_+^{|C|} \times \mathbb{R}_+^{|C|} \rightarrow \mathbb{R}^{|C|} \) that determine
the rate of the transport process within the ODE system,

\[
\frac{du_c}{dt} = (\beta_{c'} - \alpha_c)T(u, u'),
\]

\[
\frac{du'_{c'}}{dt} = -(\beta_{c'} - \alpha_c)T(u, u')m
\]

for environment chemical \( c \in C \) and cell bound chemical \( c' \in C' \) that are consumed
in the environment in amount of \( \alpha_c \in \mathbb{N} \) per unit transport rate and sourced from
the cell in amounts of \( \beta_{c'} \in \mathbb{N} \) that are motivated by the stoichiometric coefficients
for chemical reactions. As the cells are provided with a unit area and volume to
account for the point-like description, the units of the rate constants, \( k_f, k_r \in \mathbb{R} \), are
constructed such that the units encode for the correct flux units of "concentration
per unit time per unit area" where the unit area is 1.
<table>
<thead>
<tr>
<th>Transport process name</th>
<th>Transport rate $T(u, u')$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZerothOrderTransportIntoCell</td>
<td>$k_f$</td>
</tr>
<tr>
<td>ZerothOrderTransportOutOfCell</td>
<td>$k_r$</td>
</tr>
<tr>
<td>ZerothOrderReversibleTransport</td>
<td>$k_f - k_r$</td>
</tr>
<tr>
<td>MassActionTransportReaction</td>
<td>$k_f \sum_{c \in C} u_c^{\alpha_c} - k_r \sum_{c' \in C'} u_{c'}^{\beta_{c'}}$</td>
</tr>
</tbody>
</table>

Table 4.2: Transport laws for cell-environment molecule exchange provided in ChemChaste. The transport laws model the exchange of molecules between the environment and the cell across the cell boundary and are provided by the user as a environment to cell "reaction equation".

Reactions at the membrane, such as chemical signal or proton mediated intake, indirectly couple the two chemical systems (Bondar and Keller, 2018; Chaudhry et al., 2014). Two reaction sets, one on the outer environment side of the cell boundary and one on the inside of the boundary, occur which share the same reaction rate. Concentrations in both chemical systems change through an implementation of the reaction ODE system however there is no passage of molecules through the boundary. These reactions are provided to support the implementation of membrane coupled reactions such as hydrogen and electron shuttles and cell signalling at the cell surface.

The reaction system implements an ODE system to determine the change in concentrations both inside and outside the cell,

$$\frac{du_c}{dt} = (\beta_c - \alpha_c)M(u, u'),$$
$$\frac{du'_c}{dt} = (\beta'_c - \alpha'_c)M(u, u'),$$

where $c \in C$ are the chemicals in the environment with concentrations $u_c$ and stoichiometric ratios for the consumption $\beta_c$ and production $\alpha_c$ of the molecules in the forward reaction direction. Likewise $c' \in C'$ denote the cell bound equivalents.

The membrane reaction rates, $M(u, u', t) : \mathbb{R}_{\geq 0}^{\lvert C \rvert} \times \mathbb{R}_{\geq 0}^{\lvert C' \rvert} \rightarrow \mathbb{R}^{\lvert C \rvert}$, provided with the release version of ChemChaste are given in table 4.3.
<table>
<thead>
<tr>
<th>Membrane reaction name</th>
<th>Membrane reaction rate $M(u, u')$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZerothOrderCoupledMembrane</td>
<td>$k_f$</td>
</tr>
<tr>
<td>ZerothOrderReversibleMembrane</td>
<td>$k_f - k_r$</td>
</tr>
<tr>
<td>MassActionCoupledMembraneReaction</td>
<td>$k_f \prod_{c \in C'} u^a_{c'} \prod_{c \in C} u^a_c - k_r \prod_{c \in C'} u^b_{c'} \prod_{c \in C} u^b_c$</td>
</tr>
</tbody>
</table>

Table 4.3: Reaction rate laws for reactions occurring at the membrane of the cell. The table shows the membrane reaction rates usable within ChemChaste where reaction processes occur at the membrane without a flux across the boundary. Therefore reaction substrates and products are confined to the different cell-environment compartments.

4.2.7 Cell division and death

The final step of the ChemChaste simulation cycle is to implement cell birth and death processes. Cell division and death are implemented as rules where the event decisions are based on maximum and minimum threshold conditions for internal chemical concentrations. For each internal chemical species, $c$, with internal concentration $u_c$, a maximum and minimum concentration threshold, $u^{\text{max}}_c$ and $u^{\text{min}}_c$, may be provided. If $u_c \geq u^{\text{max}}_c$ for any cell bound chemical at the beginning of the timestep the cell is marked for division which occurs at the end of the current simulation cycle. During the division process a new cell agent is created and placed at a random orientation around the parent cells (see figure 4.5). If $u_c \leq u^{\text{min}}_c$ the cell is marked for death and does not participate in any internal cell dynamics or concentration exchanges with the environment. These internal concentrations change based on transport exchange with the environment and the implementation of cell internal reactions.

When a new cell is added to the cell based simulation through cell division the internal concentrations of the parent may be divided between the two cells or duplicated. Consider a cell dividing at the end of the $n$-th simulation cycle with internal cell chemical $c' \in C'$ with parent concentration $u_{c'}(n\Delta t)$ at time $n\Delta t$. The offspring is introduced and concentrations are shared with a concentration vector at the start of the next cycle $u'_{c'}((n+1)\Delta t)$, where,

$$u_{c'}((n+1)\Delta t) = pu_{c'}(n\Delta t) \quad \text{and} \quad u'_{c'}((n+1)\Delta t) = (1-p)u_{c'}(n\Delta t),$$

where a split ratio $p$ is introduced to control the degree of sharing. This ratio defaults to $p = 0.5$ however this may be changed in the simulation configuration by a user.
and random perturbations to this value may also be implemented. Alternatively, the
concentrations may be duplicated and copied from the parent to the offspring:

\[ u_{c'}((n+1)\Delta t) = u_{c'}(n\Delta t) \quad \text{and} \quad u_{c'}'((n+1)\Delta t) = u_{c'}'(n\Delta t) \]

where the duplication or sharing is decided for each of the internal chemicals so that
a subset may be duplicated and a subset may be shared. The default behaviour is
to duplicate the chemical.

Figure 4.5: Schematic of the cell division process. During cell division a new cell
is added to the simulation (c) at a random orientation around the parent cell. The
mechanical interactions between the cell increase the pressure of the system. This
pressure is reduced through the natural relaxation of the cell-cell force laws by moving
the cell centre locations which pushes the neighbouring cells (b) in order to make
space to accommodate the new cell which has the effect of expanding the biofilm
envelope.

4.3 Model implementation and the user interface

ChemChaste has been developed to produce an adaptable framework in which to
implement a variety of chemically orientated simulations. To support the adoption of
ChemChaste a file based user interface has been developed to aid the implementation
of a user’s model. The modular nature of ChemChaste also facilitates future code
expansions by user’s experience with writing C++ code. For example, a user may
write new reaction laws which may be readily integrated into the ChemChaste
algorithm in order to supplement the range of laws provided within the current
ChemChaste edition (see tables 4.1, 4.2, and 4.3).

The docker container system simplifies the deployment of ChemChaste onto a
wide range of computer architectures and high performance computing systems. For
example, ChemChaste has been deployed on an 8 core laptop CPU Ubuntu system,
a 12 core desktop machine running the Windows operating system, and on a 32
core elastic cloud based computing server using Amazon Web Services. Methods to explore a model through parameter sweeping have also been implemented. However, using high performance computing facilities and parameter sweeping can produce a wealth of data which may require extensive post-processing.

To assist with the exploration of this data a selection of simulation summary statistics have been provided. These measures present a detailed overview of the simulation output without the need for immediate further post-processing. Similar to Chaste, the recommended method to visualise and analyse ChemChaste simulation is through utilising the Paraview software. However, methods developed to process Chaste outputs may be directly used on ChemChaste outputs.

4.3.1 Docker

Docker is a software development environment that may be utilised in the same way as a virtual machine or virtual environment but without the computational overhead (Merkel, 2014). Within the lightweight container system, a user can install software, package libraries, and file dependencies are stored within self-enclosed containers. These container images may then be distributed and deployed on different computer architectures facilitated by the container repository Docker Hub https://hub.docker.com/ (accessed on 14/04/2022). Docker images contain a file system within which a Dockerfile is used to determine the environmental variables and runtime applications for when the image is run as a container instance (Nüst et al., 2020). A Chaste docker volume has been produced to utilise the Chaste-Docker method which may be accessed after installing the Docker engine (Cooper et al., 2020). Install instruction may be found on the Chaste-Docker page https://github.com/Chaste/chaste-docker (accessed on 14/04/2022). ChemChaste may be added to a new Chaste project for the Chaste file structure after the Chaste Docker is installed (see installation documentation https://github.com/OSS-Lab/ChemChaste (accessed on 14/04/2022)).

4.3.2 Command and configuration files

ChemChaste simulations are performed within a Python command script and constructed within a configuration file. The simulation command file (see figure 4.6), called RunChemChaste.py in the ChemChaste-docker image, contains the configuration information for each simulation to be run and compiles and executes the necessary ChemChaste executable scripts, ChemChaste.cpp. Parameter sweeping is implemented by calling Python functions from within the command script and an
array of simulation runs is constructed. This array of simulations is then distributed amongst a pool of CPU cores to implement a so-called coarse-grained, or naive, parallelism. While the command file controls a set of simulation and higher order features such as parameter sweeping, the individual simulations are determined by the particular simulation configuration file. The configuration file (see figure 4.7) refers to the elements of the user file interface that determine the simulation model.

```python
# This script generates and submits jobs for a parameter sweep of our new force
import multiprocessing
import subprocess
from ChemChasteDefinitions import *

# generate a list of bash commands
command_list = []

# config files for each of the simulations
swEEPConfigFile = "ParameterSweeping(configEnvironmentParameterSweep)

for paramSimConfig in sweepConfigList:
    simulationExecutable = str(AbsNameExecutable(paramSimConfig))
    command = simulationExecutable + " 1"
    # set config
    command += "-config=" + str(paramSimConfig)
    # set simulation type (default "coupled cell")
    command += "-simulation_type:environment_cell"
    # set additional commands to override config
    command += "-simulation_timestep:1"
    command += "-sweeping" parameter="1"
    command += "-sweeping" parameter="1"

    # add simulation to the list
    command_list.append(command)

# use 'count' of processes
nproc = 8
# use multiprocessing.cpu_count() for the number of cores on your machine
nproc = 8
# generate a pool of workers
pool = multiprocessing.Pool(processes=nproc)

# ... and pass the list of bash commands to the pool
pool.map(execute_command, command_list)
```

Figure 4.6: The command file for a ChemChaste simulation. a) Importing the functions used by the Python script, including ChemChaste input/output functions. b) The simulation configuration files used in this command script. The "ParameterSweeping" function is used to convert a single simulation into a series of sweeps depending on the "Parameters.csv" file. c) The parameter sweeping command call d) The Python routines to perform the simulations in parallel using a number of processor cores.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>simulation_type</td>
<td>coupled_cell</td>
<td>Simulation type as a String keyword</td>
</tr>
<tr>
<td>simulation_timestep</td>
<td>1/120</td>
<td>Timestep used for the simulation solvers, $\Delta t_{sim}$.</td>
</tr>
<tr>
<td>sampling_timestep</td>
<td>1/120</td>
<td>Timestep used for simulation result output, $\Delta t_{sampling} \geq \Delta t_{sim}$.</td>
</tr>
<tr>
<td>output_filename</td>
<td>ChemChasteExecutable</td>
<td>Results directory location</td>
</tr>
<tr>
<td>simulation_end_time</td>
<td>10.0</td>
<td>The maximum simulation time</td>
</tr>
<tr>
<td>number_of_reaction_pdes</td>
<td>1</td>
<td>The number of diffusing chemicals</td>
</tr>
<tr>
<td>spatial_dimensions</td>
<td>2</td>
<td>The domain spatial dimension</td>
</tr>
<tr>
<td>FE_element_dimensions</td>
<td>2</td>
<td>FE mesh dimension</td>
</tr>
<tr>
<td>node_cutoff_length</td>
<td>1.5</td>
<td>The node cutoff length $d_{cutoff}$.</td>
</tr>
<tr>
<td>linear_force_cutoff</td>
<td>1.5</td>
<td>The linear Hookean spring force constant</td>
</tr>
</tbody>
</table>

Table 4.4: ChemChaste simulation parameter options available in the current release. The parameter options control the simulation and are defined within the configuration file.

Figure 4.7: The configuration file for a ChemChaste simulation. a) Simulation files are contained within a directory structure, showing separate cells \{CellA, CellB\} and the environment domain DomainField. b) Simulation parameters and file directories are provided in the "Simulation Configuration File". c) Parameter sweeping is implemented by defining a "Parameters.csv" file.
4.3.3 User file interface

The user file interface was developed to support the adoption and sharing of models by users less experience with C++ coding. Each ChemChaste simulation model has two components; a configuration file that is called by the command script which contains simulation parameters and file paths (see figure 4.7, and the model directory. The model directory contains individual cell models, the cell topology, the environment domain information, and an optional parameter sweeping file. These combined files are sufficient to define a ChemChaste model and simulate using the command file and therefore may be archived or distributed between computer systems.

4.3.4 Specifying the cell model

Cells in ChemChaste follow the overlapping sphere model and contain internal chemical reactions. The spatial arrangement of the cells in the ABM are provided within the Cell directory of the simulation model (see figure 4.7). The initial arrangement of the cells is provided in the CellLayerTopology.csv file. This file uses a key label to provide a matrix of initial cell types. These types are determined through connecting the key labels to a cell type name in a corresponding CellLayerKey.csv file. A user may edit the topology and key files to input the initial spatial arrangement of the cells and any number of cell types may be utilised. When the simulation begins the cell mechanics will relax the position of the cells into a more natural arrangement therefore the initial file arrangement provides the approximate locations for the cells. The particular cell properties are stored in sub-directories for each cell type.

For each cell type a set of property files are included (see figure 4.8). The initial cell concentrations are provided in the InitialCellConcentrations.csv file for each of the cell bound chemical species. The reaction for these species are stored in the srn.txt file using the cellular chemical reactions (see table 4.1). Transport laws are provided in a TransportReactions.txt file and membrane reaction that occur at the cell boundary are provided in the MembraneReactions.txt file following their respective reaction forms. Cell behaviours are implemented in the SpeciesThresholds.csv, SpeciesDivisionRules.csv, and Environment.csv files. These files provide the minimum and maximum cell concentrations for determining division and death conditions, the rules that govern concentration sharing after division, and any preference in environment concentration values as used by environment dependent reactions.
Figure 4.8: An example of an individual cell directory. a) "InitialCellConcentrations.csv" provided the initial concentrations within the cell specified by the chemical name, base initial concentration, and whether to add a uniformly random value. b) "SpeciesDivisionRules.csv" describes how concentrations are treated on cell division. The rules are specified by the chemical name, and the whether to "share" between parent and offspring or to "duplicate" and copy the parent value to the offspring. c) "SpeciesThreshold.csv" specifies the chemical name, the minimum concentration below which cell death is implemented, and the maximum concentration above which cell division is performed. d) The cell chemical reaction system. Parameter sweeping acts on this system through the parameters. e) The transport laws which implement the exchange of concentrations between cell and environment. f) The reactions occurring at the membrane which couple the cell and environment systems. Here, this reaction set is switched off due to the zero reaction rates.

4.3.5 Specifying an environment system

Following the model definition method used for defining the cell simulation, the environment can be defined using a sub-directory of text and CSV files (see figure 4.9). A heterogeneous domain is specified using the same key-label system as used for the cell topology. The Domain.csv file contains labelled regions where the labels may be associated with different diffusion or reaction conditions. These regions are named in the DomainKey.csv file. The chemical diffusion rates are provided in the DiffusionDatabaseFile on a region basis with the chemical name, region name, and diffusion rate order. The labels are also tied to a reaction file name where the reactions are provided as in the cell Srn.txt file. These reactions are initialised on a
region basis with the InitialConditionFile.csv providing the initial concentration through a list of condition strings: chemical name, region, value, and whether to provide a uniformly distribution random perturbation to the value. The uniform perturbation is implemented at the nodes of the FE mesh. Boundary conditions are defined for each environment chemical in the BoundaryConditionsFile.csv file and may be of either the Dirichlet or Neumann type and defined for the whole boundary or for each edge (see appendix A.2).

Figure 4.9: An example environment domain configuration where the different regions of a heterogeneous domain may be defined. a) The "Domain.csv" file where each matrix entry is mapped to a cluster of nodes of the FE mesh. The value of these nodes refer to different region types. b) "DomainKey.csv" the values used to refer to domains are translated into region names. c) The "BoundaryConditionFile.csv" contains the boundary conditions around the environment domain. Here, zero-Neumann (no-flux) conditions are implemented for chemicals \( E \) and \( ES \) and there is a flow into the domain for the Substrate and Oxygen chemicals around the whole perimeter of the domain. d) The "DiffusionDatabaseFile.csv" where the diffusion values for the different labelled domain regions are specified. The entries denote the chemical, the region, the diffusion rate coefficient, and whether to perturb the rate by adding a uniform random value. e) "InitialConditionFile.csv" provides the initial conditions for the concentrations in the environment domain; specified by chemical name, region, and initial value. f) Each region may possess a different reaction system labelled in "OdeReactionFileKey.csv" which provides a list of the associated reaction files. g) "ExtracellularReaction.txt" the reaction file referred to in f)
4.3.6 Providing the chemical reactions and transport laws

The environment domain and cell bound reaction systems both utilise the same reaction file structure; for the cell Srn.txt file and the environment ExtracellularReaction.txt file. The files contain a list of different reaction string that include a reaction type name (from table 4.1), the chemical equation string that may be either reversible "<=>" or irreversible "->" and including stoichiometric coefficients, and a list of reaction parameters. Each reaction string may use a different reaction type name, chemical equation, and parameter set.

Cells may be provided with transport laws and (see table 4.2). Transport processes are provided in the cell TransportReactions.txt file following a list similar to the chemical reactions. The chemical equation in the reaction files is replaced by a similar equation where the "substrates" are replaced by the species on the environment side of the boundary and the "products" the species on the cell side of the boundary. The stoichiometric ratios are replaced with ratios that denote the quantity of each chemical used in the transport process. Membrane reactions at the cell-environment boundary can also be implemented that combine two chemical equations, one for each side of the boundary, and a coupling reaction rate law (see table 4.3).

4.3.7 Adding new reaction rate law

Reactions in ChemChaste build upon an inheritance structure to share functionality and gradually build complexity. A new reaction may be added to this structure by branching from an existing reaction class and adapting the functionality to suit the new reaction needs. These branches may be further developed to produce more specialised reaction rate laws (see figure 4.10).

The Reaction Tablet function is used to inter-convert the reaction types utilising the pointer management and class inheritance features of C++. This method allows the seamless integration of a new reaction type, custom written by a user as a reaction C++ class, into the ChemChaste system. The user integrates these new classes by writing their types into the reaction tablet after which these new types may be utilised in the same way as the current reactions in the file system. In addition to writing new cell and environment reactions, the same method may be applied to writing new transport laws and membrane reactions (see the ChemChaste user documentation for details: https://github.com/OSS-Lab/ChemChaste (accessed on 14/04/2022)).
Figure 4.10: The ChemChaste reaction and transport law classes follow an inheritance structure. The reaction class a) and transport classes b) can be built upon through the inheritance structure. New laws are added by inheriting from one of the classes in the tree.

4.4 Model exploration and output summary statistics

Biochemical systems contain many parameters each of which has the potential to greatly change the system dynamics through the introduction of bifurcation dynamics and inherent nonlinearities in chemical reaction systems. Therefore the ability to consider multiple values for each parameter in turn is a necessity for robust modelling. This can be achieved in ChemChaste through performing parameter sweeps. The combined PDE system and ABM model simulated in ChemChaste produces a large quantity of data per simulation run and parameter sweeping may produce many simulations per parameter set. To assist in the analysis of the large amount of data a range of summary statistics to describe heterogeneity amongst the cell types have been implemented.

One advantage of the ChemChaste setup is that a user can use the command file and parameter sweeping technique to link a simulation to optimisation routines, such as genetic algorithms. This linkage would be easy to accomplish due to the command file being written in Python which has access to a wide array of optimisation packages. This optimisation would be facilitated through the summary statistics which may be used to evaluate the simulation run for objective refinement.
4.4.1 Parameter sweeping

Parameter sweeping is implemented through a symbol replacement scheme similar to the method used in CompuCell3D (see section D.2). Within the simulation directory a Parameters.csv file is added that contains a series of lines specifying a identifier to replace; where the first character is the & symbol, and then either a comma separated list of replacement values or a range for the values. For each parameter value all other parameters are varied leading to a combinatorially large number of simulations. The identifiers are used throughout the simulation directory, and both the cell and environment sub-directories, and substituted for the parameter values. The simulation directory is duplicated for each realised set of parameters to minimise data access issues through parallel algorithms accessing the simulation files at the same item.
Figure 4.11: Parameter sweeping is performed through value substitution in the ChemChaste configuration and model directory files. a) The simulation directory contains the "Parameters.csv" file which contains the parameters to sweep over. Each simulation uses one of each parameter $C, P$ and substitutes these for the symbols in the simulation directory files b). c) The simulation directories are copied for parameter set and are referred to by a configuration file for each set d).

4.4.2 Simulation summary statistics

Simulation summary statistics have been provided to assist the analysis of the ChemChaste output. This has lead to the implementation of a selection of indices to capture the cell species spatial heterogeneity. These indices are calculated at the end of each simulation cycle therefore the total number of cells are calculated and may vary for each time step. For summary statistics we consider the population of cells denoted by $A$ where each cell is labelled with a type from the cell type set $\mathcal{T}$. The number of cell types in a simulation is given by the cardinality $|\mathcal{T}|$, and $n_t$ is the number of cells of type $t \in \mathcal{T}$.

The type of a cell and the number of each cell type are categorical measures
of the cell properties. However a subset of the summary statistics rely on continuous values to capture the spatial diversity in the cell populations. Therefore a continuous value that capture the local diversity around a cell has been constructed. The cell diversity measure,

\[ d_i = \frac{\sum_{j \in N} 1(t_i \neq t_j)}{|N|}, \]  

(4.10)

encodes the cell type difference in the neighbourhood of a cell. The diversity for cell \( i \), \( d_i \in [0, 1] \) is maximal if the neighbouring cells have types different to cell \( i \) and minimal if the cell is surrounded by cells of the same type. The mean cell diversity is calculated through the sum of the cell diversity measures;

\[ <d> = \frac{1}{|A|} \sum_{i \in A} d_i \]  

(4.11)

the probability of a cell being of type \( t \) is given by;

\[ p_t = \frac{1}{|A|} \sum_{i \in A} n_t \]  

(4.12)

where some indices use the probability density of finding a neighbour of a particular cell type.

The autocorrelation of the cell diversity values across the cell population is considered using the Moran, Geary, and Lee-Ogburn indices (Lee and Ogburn, 2020; Dhawan et al., 2016). These indices are defined over a continuous domain in two dimensions and relate continuous properties to measures on continuous intervals. The Moran index varies over the interval \( I_{Moran} \in [-1, 1] \) with an expected value that varies with the number of cells in the simulation, \( E_{Moran} = -1/(|A| - 1) \) (Cliff and Ord, 1973). The expected value is based on a hypothesis of no spatial autocorrelation of the cells within a population. Therefore values above the expectation suggest a positive autocorrelation where cells with similar neighbourhood heterogeneity coexist and values below the expectation suggest highly changeable diversity in neighbourhoods (Li et al., 2007). The related Geary index provides a measure on the interval \( I_{Geary} \in [0, 2] \) where \( I_{Geary} < 1 \) suggest positive autocorrelation with a spatial clustering of similar diversity neighbourhood or \( I_{Geary} > 1 \) suggesting negative autocorrelation (Cliff and Ord, 1973). The Lee-Ogburn index was developed to expand the Moran index to consider categorical data, therefore rather than using the cell diversity values the Lee-Ogburn index uses an indicator function whose value is 1 for cells whose neighbour is of the same type and 0 if the neighbour is of a different type (Lee and Ogburn, 2020).
The Moran, Geary, and Lee-Ogburn indices attempt to capture the autocorrelation between heterogeneity across the domain. However, ChemChaste also implements measures of the global diversity of the population based on the occurrence of each cell type. In the results presented in this article (see chapter 5) we consider the interactions between two cell types at most, however neither ChemChaste nor these summary statistics place a limit on the number of cell types. However, simulating a greater number of cell types may be computationally expensive and these statistics may be less informative and exhibit greater noise to signal ratio for low frequencies of a large number of cell types. The diversity index as defined by iDynomics has been provided (Lardon et al., 2011) alongside the Shannon and related Gini-Simpson indices used in ecology studies (Jost, 2006). Higher values of these indices suggest greater diversity in the cell populations. These indices benefit from being simpler to calculate and interpret than the autocorrelation measures.

<table>
<thead>
<tr>
<th>Global summary statistics</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Cells of Type t</td>
<td>( n_t )</td>
</tr>
<tr>
<td>Moran’s index</td>
<td>( \frac{</td>
</tr>
<tr>
<td>Geary’s index</td>
<td>( \frac{</td>
</tr>
<tr>
<td>Lee-Ogburn index</td>
<td>( \frac{1}{\sum_{i \in A}</td>
</tr>
<tr>
<td>Diversity index</td>
<td>( 1 - \frac{\sum_{i \in A} n_i(n_i - 1)}{</td>
</tr>
<tr>
<td>Shannon index</td>
<td>( -\sum_{t \in T} p_t \log(p_t) )</td>
</tr>
<tr>
<td>Gini-Simpson index</td>
<td>( 1 - \sum_{t \in T} p_t^2 )</td>
</tr>
</tbody>
</table>

Table 4.5: Global summary statistics describe the whole simulation output. These statistics provide a single value to describe the simulation domain and are calculated for each time step. The values describe the autocorrelation or the heterogeneity of cell types through the domain. All quantities described are described in the main text.

The local measures (see table 4.6) may be used in visualisation software to colour the cells to display the values over the cell population. The local measures provide a fine grain understanding of the diversity in the neighbourhood of a single
cell. The **Strain Type** value gives the categorical type of the cell where regions of the same colour in visualisations show the regions where particular cell types proliferate. The local Moran index highlights regions where the neighbourhood diversity of a cell differs from the average neighbourhood diversity (Schmal *et al.*, 2017). Index values of $I_{Local\,Moran} > 0$ suggest that the neighbourhood of cell $i$ have similar diversity values and $I_{Local\,Moran} < 0$ the neighbourhood are heterogeneous in their diversity. Another diversity related measure is the Getis-Ord index where index values $I_{Getis-Ord} > 0$ suggest that the neighbourhood of the cell shows a tendency for greater than average diversity and conversely $I_{Getis-Ord} < 0$ for below average diversity (Chen, 2020).

<table>
<thead>
<tr>
<th>Local summary statistics</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strain Type</td>
<td>$t_i$</td>
</tr>
<tr>
<td>Local Moran’s index</td>
<td>$(d_i - &lt;d&gt;)</td>
</tr>
<tr>
<td>Getis-Ord index</td>
<td>$\frac{\sum_{j \in N_i} d_i}{</td>
</tr>
</tbody>
</table>

Table 4.6: Local summary statistics describe each cell in the simulation. The local measures reflect the difference between the cell type and the cell neighbourhood types and are calculated for each cell in the simulation per time step. All quantities are described in the main text.

### 4.4.3 Model input and relation to COMBINE and SBML

The distribution of models amongst the scientific community is a key element to produce reproducible robust simulations. Efforts have been made to develop standards and produce computational frameworks to facilitate model sharing, for example through the COMBINE archives and the SBML standards (see section 4.4). ChemChaste does not directly support these standards due to the input methods for the file based interface being custom made. However, there are shared elements between the COMBINE, SBML and ChemChaste descriptions which provide a foundation to map models between the different descriptions.

The mapping between models would require the addition of reader and writer scripts to be written. These scripts may act upon the file based user interface and therefore would be independent of the ChemChaste C++ codes. The reaction components of SBML are contained within reaction compartments that is mirrored by the ChemChaste cell systems and environment reactions through the region
specifications. The current set of reaction rate and transport laws (see tables 4.1, 4.2, and 4.3) implemented in ChemChaste may not be sufficient to implement all reactions encoded in the SBML specification and for these cases the addition of new ChemChaste reactions would be necessary.

The environment aspects of ChemChaste can be converted into the SBML-spatial specification. The sub-domains in SBML are defined with boundary conditions between the domains which is not implemented within ChemChaste. Therefore a form for these boundaries would need to be assumed during the conversion process. Furthermore, SBML-spatial can support the definition of more boundary conditions types than ChemChaste, such as Robin BCs, therefore some model distributed through SBML may not be possible to simulate in ChemChaste. This incompatibility is also present when SBML models utilise advection in the PDE system which ChemChaste does not support.

The capability to produce an archive directory that contains the necessary simulation metadata and settings under a complete package is a strong advocate for the COMBINE system. A similar method has been developed for ChemChaste where the simulation configuration file contains the simulation settings and file directory paths. The individual simulation files found in an accompanying simulation directory which contains the full model to be simulated. The COMBINE archive however relies on supported model formats of which the ChemChaste file based system is not one of them. Therefore archives are not readily constructed with the current release of ChemChaste. Further integration of SBML and the COMBINE system can be included in future releases.

4.5 Simulation stability and convergence

The PDE solvers in ChemChaste are inherited from Chaste and require verification to ensure the chemical concentrations fields of the coupled cell-environment simulations are modelled to an adequate accuracy. During the development of ChemChaste, the Chaste solvers were used to model a simple heat-diffusion system where the simulation results can be compared to known analytic solutions. However, when these solvers were used for the simple source less heat diffusion equation the total heat of the system produced some discrepancies. In particular, when an isolated box with zero-Neumann boundary conditions was considered with the source less form, the total heat was seen to decay instead of remaining constant as would be expected from such a source less isolated system (see section 5.5). Therefore when conservation of heat was expected the Chaste simulation results failed to produce
the correct dynamics which warranted further investigation (see figure 4.14). The output of the Chaste PDE solvers was corrected through changing the point in the Chaste algorithm where the data was exported. This change was implemented in the ChemChaste additions to incorporate the correction. Additionally, the issue was reported for further investigation by the core Chaste developers.

After the output from Chaste solvers were verified the convergence of the ChemChaste PDE solver was considered. The Fisher-KPP PDE system was implemented to demonstrate the correct functioning of the ChemChaste system and produce travelling wave phenomena as a prerequisite dynamics for complex biofilm dynamics. An analytic solution for the wave can be derived and these solutions were used to verify the simulation output. Additionally, the convergence of the output under reducing time and spatial steps was demonstrated (see figure 4.15).

4.5.1 Simulating the total heat in a source-less heat diffusion system

To investigate the PDE solution issues in Chaste the implementation of a test PDE with a known solution is required. A simple form of a reaction-diffusion PDE used in ChemChaste is the source-less heat diffusion equation. The heat equation may be considered a case of the more general reaction-diffusion equations wherein the source/sink term is removed. To solve the PDE we consider a 2-dimensional rectangular domain \( \Omega \in 10 \times 100 \) with the standard triangular mesh used by ChemChaste. In this domain a point is represented by \( x \in \Omega \) and the heat state variable \( U \) is traced for a time, \( t \). The heat state variable is governed by,

\[
\frac{\partial U}{\partial t} - D \frac{\partial^2 U}{\partial^2 x} = 0, \tag{4.13}
\]

where \( U(x, t) \in \mathbb{R}_{\geq 0} \) is the heat at a point in the system subject to diffusion where \( D \in \mathbb{R}_{\geq 0} \) is the thermal diffusivity of the medium in the domain. For an isolated system, we consider the domain to be bounded by zero-flux Neumann boundary conditions such that there is no heat flux across the boundary \( \partial \Omega \). Therefore, the heat distribution in the system may change with position but the total heat inside the domain remains constant and is determined by the initial conditions.

For a simple initial heat distribution, \( U_0(x, t = 0) \), we define a strip across the domain to defined a sub-domain \( \Omega_i \) with an initial amount of heat \( Q_0 \in \mathbb{R}_{>0} \) distributed across the slice,

\[
U_0(x, t = 0) = Q_0 \mathbb{1}(x - x_i), \tag{4.14}
\]
for $x \in \Omega$ and $x_i \in \Omega_i$ where $1 \in 0,1$ represents the indicator function which evaluates to 1.0 for a point $x$ in the sub-domain $\Omega_i$ and 0.0 if the point is outside of the sub-domain. For the source-less isolated PDE (see equation (4.13)), there is no addition or removal of heat in the system so the expected total mass of the system would be constant;

$$U_{total} = \int_{x \in \Omega} U_0 dx = \int_{x \in \Omega_i} Q_0 dx = Q_0,$$

(4.15)
equal to the unit area of the sub-domain due to the value of $U_0$. While the heat distribution is expected to change through diffusion from regions of high to regions of low, the total heat would be expected to remain constant. However, when the equation was simulated a non-constant average heat was observed (see figure 4.12).
Figure 4.12: Example output for the Paraview software for the heat equation see (equation (4.13)) with a strip initial condition. a) Snapshots showing the passage of a diffusive heat wave solution for equation (4.13). The plane output from Paraview shows the domain space, $\Omega \in 10 \times 100$, and initial ($t = 0$) distribution of the heat across the space. The strip initial condition is defined with a value of unity over the column of nodes with a finite width due to interpolation over the Gauss points. The boundary node values are determined through zero Neumann boundary conditions. The snapshots show the passage of a wave from the left hand side of the domain to the right hand side of the domain. b) The average heat value over the whole domain for each timestep. The plot compares the strip initial conditions (solid) where the initial heat, $Q_0$, is confined to a small section of the domain (see $t=0$ of subplot a)) with the result when the initial heat is distributed throughout the domain. The strip trace demonstrates a non-conserved behaviour while the uniform heat scenario shows conserved behaviour.
4.5.2 The Chaste PDE solver routines

The PDE solving routines of Chaste centre around a set of FE assembler and solver classes. The main assembler of interest for solving equation (4.13) is the AbstractFeVolumeIntegralAssembler class which controls the interpolation and solver methods on a given mesh element. The interpolation Gauss point is determined within this assembler from which the location and state variables are also interpolated. After the domain elements are assembled, the Chaste PDE simulation calls the linear solver AbstractDynamicLinearPdeSolver to set up the temporal solver conditions; initial conditions, start and end times, and solver time steps. As part of Chaste’s implementation of the FE method, the total mass at time $t$, $U_{total,t}$, is approximated by the summation over all elements of the Gauss point value in a given element,

$$ U_{total,t} = \int_{x \in \Omega} U_t(x) dx \approx \sum_{e \in \text{Elements}} \sum_{i=0}^{2} U_{i,t} \phi_i, \quad (4.16) $$

where $U_{i,t}$ are the nodal coefficients values of the heat at the nodes of an element at time $t$.

During each solver time step in Chaste, a bespoke interpolation scheme is called. During this scheme the state value, $U$, is determined at the Gauss point $\mathbf{x}$ through implementing the Chaste GetCurrentSolutionOrGuessValue method. These interpolated values, such as position in the domain and heat value $(\mathbf{x}, U)$, are used by Chaste to construct the solution vector to be used as the initial condition for the next time step, with the concentrations determined at Gauss point locations. The initial condition vector is mapped to the nodes of the FE mesh. These mesh node values for the state variables are updated through implementing a forward Euler scheme to update the concentrations from the source-sink ODE systems. These updated node values are then interpolated onto the Gauss points in order to implement the affects of source/sinks at the Gauss point level. In the heat source example, these Gauss point source/sinks are null as well as the source-sink ODE systems, providing no affect to the heat value. For modelling cell systems, the cells act as local Gauss point sources while the environment chemical reactions are implemented as source-sink ODE systems.

The results presented in this article utilise Paraview v5.10 to produce the 2-dimensional plots for the spatial distribution of chemicals and cells. After a simulation is performed the results are stored in a vtu file. In these files the simulation data is stored on a nodal basis, with the node ID, location in the 2D
plane and a vector for the scalar values for the state variables. Paraview applies operations to the input data through a pipeline of filters which act upon the data. The Interpolate to Quadrature Points filter uses the node data from the *vtu* and applies a weighted interpolation scheme to map the node state variable values onto the Gauss points (see section 4.2.5 and figure 4.13).

Figure 4.13: ChemChaste utilises the finite element method to solve PDE problems. During a ChemChaste simulation a mesh is placed over the domain, a), where a triplet of nodes (grey circles) define an element. Each element, \(\{n_0, n_1, n_2\}\) is mapped to a reference triangle, b), where the node locations, \(\{x_0, x_1, x_2\}\), and state coefficient values, \(\{U_0, U_1, U_2\}\), are mapped onto a Gauss point (red square) using a set of basis functions \(\{\phi_0, \phi_1, \phi_2\}\). The interpolation flowchart is given in c) where the nodal state coefficients from the previous timestep, or the initial conditions, are updated through the ODE solvers before interpolating onto the Gauss point for updating through the diffusion solvers. Then these values are mapped back onto the nodes ready for the next solver cycle.

### 4.5.3 Inspecting simulation output for heat loss

To test the Chaste output solvers simulations were performed where the results are known and simple to interpret. We implemented a simulation to model the distribution of heat within a closed isolated domain where the total heat in the domain is a quantity that is expected to be constant. The results from this simulation were seen to show variable total heat so tests were performed on the solver routines to locate the reason for the non-constant total heat. These tests were performed at different points in the Chaste routine. First, the functions used to calculate the heat
was considered, and shown to be correctly constant, and then tests were performed at three different output points the solver algorithm with the results exported for analysis. The purpose for the three methods was to determine whether there was any deviation from the expected (constant total heat) results where the simulation output was determined and whether a change in the routine would affect the total heat.

The summation of the nodal coefficients were recorded as the simplest output because the nodal values may be calculated using the "Plot Data Over Time" filter in Paraview to calculate the total heat over time which provides two independent methods that, to some degree, process values that were considered correct (i.e the calculated heat values). The other two output methods exported solution values at different points (functions) of the Chaste PDE solver routine and therefore these solutions are exposed to some post-processing by Chaste. For the purpose of testing the Chaste output, these node values are stored and summed for the second output method. Finally, during a Chaste simulation the previous solution vector of the solver is used as the next initial condition. For the third output analysis method these values are then manually interpolated and output from the software, bypassing the Chaste export routine. These three output methods were tested to determine which method would produce the constant total mass as expected for the system.

The simulations of the heat equation, equation (4.13), were performed for consistent parameter sets. A rectangular domain, \( \Omega = [100] \times [10] \), was considered for a simulation time of 10 units with timestep 0.1 units. Unit distance between two mesh points was used for a total of 1000 mesh nodes. The results did not change when the number of mesh nodes was varied, resulting in a finer or coarser mesh. The thermal diffusivity was set to 100 with initial mass given by (4.15) for a line source sub-domain. A one Gauss point thick slice across the major axis of the rectangular domain was considered for one dimensional plots. The main test results are presented in figure 4.14. The different solver output methods are presented with the final method, interpolating at the end of the solver routine, yielding the constant total mass as expected from equation (4.15).
Figure 4.14: The direct outputs of Chaste were tested to bypass Paraview analysis. a - c) The varying output methods were used to test the heat output. a) Both using the sum of the nodal coefficient and b) recording the value at interpolation produce decaying traces of the domain heat. c) Determining the total heat through summation of Gauss points at the end of the solver step produces a constant total heat. d) The Gauss point heat values across a slice of the domain. The sequence of times demonstrating the diffusion of heat from the initial strip distribution, figure 4.12 a).

The simulation results show that the diffusion component of the solver produced the conservation error and that manually interpolating the heat value at the end of the timestep produced the desired conserved total heat (see figure 4.14). Therefore these manual values that bypassed the Chaste output routines were used for future simulation runs. The results of this study were presented to the Chaste core development team for follow up investigations.

4.5.4 Chemical waves: the Fisher-KPP model showing convergence

A further test for the ChemChaste solvers was performed to check the accuracy of the PDE solvers. A single reaction-diffusion PDE with a known analytic solution was implemented to solve the Fisher-Kolmogorov-Petrovsky-Piskunov (Fisher-KPP) equation showing a strong agreement with an analytic series expansion (Fisher, 1937). The Fisher-KPP equation is known to admit wave solutions (Murray, 2002; Chandraker et al., 2015) and has been used in ecology to model species invasion (Venegas-Priz et al., 2017, 2013). The Fisher-KPP equation couples a diffusion equation to a logistic growth term for the propagation of a state variable.
$U(x,t) \in \mathcal{R}_{\geq 0}$; 
\[ \frac{\partial U}{\partial t} - D \nabla^2 U = U(1-U), \]  
\[
(4.17)\]

for normalised coefficients in the logistic equation. Equation (4.17) admits a pulled wave solution that propagates at a known minimum velocity, $c_{\text{min}} \in \mathcal{R}_{\geq 0}$, whose observation is dependent on the initial conditions (El-Hachem et al., 2019). The solution for the tip of the wave may be determined through an expansion;
\[
U(x,y,t) = \frac{1}{1 + \exp(z/c)} + \frac{c^{-2} \exp(z/c)}{(1 + \exp(z/c))^2} \ln \left( \frac{4 \exp(z/c)}{(1 + \exp(z/c))^2} \right) + O\left(\frac{1}{c^4}\right)
\]

where $z = x - ct$ denotes the travelling wave coordinate (Loyinmi and Akinfe, 2020). Where the minimum velocity is given by;
\[ c_{\text{min}} = 2\sqrt{D}, \]
\[
(4.18)\]

for the normalised logistic growth equation (see equation (4.17)).

ChemChaste was used to solve the Fisher-KPP equation for a rectangular bounded domain $\Omega \in [0,10] \times [0,100]$ with zero-Neumann BCs. The Fisher-KPP equation is not provided as a model for simulation in Chaste as the PDE classes in cell-based Chaste do not allow for complex source terms. Therefore a new PDE class was developed and tested to accommodate the Fisher-KPP equation and extend the simulation capabilities of Chaste. This class was incorporated into the ChemChaste PDE solver to allow of spatially variable diffusion rates and initial conditions. The Fisher-KPP equation was simulated for an initial distribution of the state variable $U$ that was set to zero throughout the domain except for a slice chosen to demonstrate the simple propagation of a wave where;
\[ U(x,y,0) = U_0 \text{ for } 0 < x < 1, 0 < y < 100. \]
\[
(4.19)\]

and $U(x,y,0) = 0$ elsewhere, the simulation was performed such that there was a minimum wave speed of $c_{\text{min}} = 2$.

The Fisher-KPP PDE was solved using a Forward Euler scheme to solve the temporal terms. The Forward Euler scheme was implemented using the CVODE package from the Sundials software suite (Hindmarsh et al., 2005). The CVODE is an adaptive solver wherein the time step is varied to avoid instabilities due to the stiffness of the ODE system. In ChemChaste, the simulation time step is used as a maximum step, of 0.01 seconds, for the CVODE solver and smaller steps may be used by the solver when determining the simulation solution. The simulation output
was post-processed in Paraview to plot the state variable across a 1-dimensional slice from the $[10] \times [100]$ domain. We used Paraview’s "Plot Over Line" filter placing the line along the major axis of the rectangular domain (along the x-axis) and Paraview took the mean average across the minor direction (along the y-axis). These averaged values were then inspected as a spreadsheet data type and exported from Paraview as a CSV file type. These values were then compared against the first 2 terms of the analytic expansion (see equation (4.18)). The results show the wave propagation with correct wave speed showing that ChemChaste can demonstrate complex dynamic phenomena (see figure 4.15). The stability of the ChemChaste solvers was then considered by inspecting the convergence of the numerical solution to the analytic solution for decreasing temporal and spatial numerical step sizes. These showed good agreement for a range of step sizes and step sizes for all subsequent simulations were drawn from this agreeable range.

Figure 4.15: ChemChaste simulations of the Fisher-KPP equation. a) The progression of a wave through the system originating from a spike initial condition at $x = 10$. The wave front produced by the simulation aligns with the analytic solution for the wave tip, equation (4.18). b) Convergence of the Fisher-KPP system for different spatial step sizes ($dx$) and time steps ($dt$). The $L_2$-norm for the Fisher system was calculated by comparing the simulation output with the analytic solution. Two computational issue arise, the low tolerance due to the Chaste linear algebra routines used to solve and low convergence in the Forward Euler ODE solvers. For appropriate values of $dx$ and $dt$ the solvers converge.

4.6 Exemplar simulations to demonstrate the ChemChaste system

To demonstrate the capabilities of ChemChaste a selection of simulations were implemented within the ChemChaste system. To test the coupled chemical reaction PDEs capabilities and reaction input file based user interface the
Sel’kov-Schnakenberg system was solved to demonstrate the onset of spatial patterning and oscillations in chemical reaction systems (see figure 4.16). To demonstrate the coupling between the ABM and reaction-diffusion PDEs a simple cooperator-cheater model was implemented involving both environment reactions and cell bound reactions for two different cell species. The cooperator-cheater model shows the beginning of spatial patterning in the cell population through the excretion and uptake of an substrate-scavenging enzyme (see figure 4.19). These models were further explored using the model exploration features of ChemChaste (see figure 4.22).

4.6.1 Spatial patterning and oscillations of the Sel’kov-Schnakenberg reaction system

The Sel’kov-Schnakenberg reaction system can exhibit both oscillations and Turing patterns in the concentration fields for different regions of parameter space (Al Noufaey, 2018). In ChemChaste, we consider a square domain bounded by zero-Neumann BCs and perform simulation for parameter sets in each dynamic region to demonstrate the onset of oscillations and spatial patterning. For each simulation we consider a 2-dimensional domain covered by a $100 \times 100$ node mesh and perturb the initial conditions for each node by adding a uniform random value. That is for the initial concentrations $U_0, V_0 \in \mathbb{R}_{\geq 0}$, we draw random values $\xi, \zeta \sim \text{Uniform}(-1, 1)$ for each mesh nodes and add them to the scalar field value at the node;

\begin{align}
U(x, y, t = 0) &= U_0 + \xi, \\
V(x, y, t = 0) &= V_0 + \zeta,
\end{align}

where the random perturbation is bounded by the interval $[-1, 1]$. Following standard linear stability analysis, the regions of parameter space that produce the spatial patterns are determined in appendix B while oscillations in the total domain concentrations are present when the homogeneous reaction system displays a limit cycle behaviour. A point in the parameter space for both cases was used to produce the figures 4.16 and the point uses the associated parameters provided in Table 4.7. The expected dynamic phenomena was confirmed through bifurcation analysis as shown in figure 4.17. The parameter file input used to perform the Sel’kov-Schnakenberg reaction system are shown in figure 4.6. Here, the `domain_only` simulation type is used which only requires the domain section of the simulation directory to be defined (see section 5.3).
Table 4.7: Parameters used in the Sel’kov-Schnakenberg reaction diffusion simulation (equations (2.14) and (2.15)). The values were selected based on analytical solutions of this system and to demonstrate the possible oscillatory and patterning dynamics.

<table>
<thead>
<tr>
<th>Case</th>
<th>$k_1$</th>
<th>$k_{-1}$</th>
<th>$k_2$</th>
<th>$k_3$</th>
<th>$D_U$</th>
<th>$D_V$</th>
<th>$U_0$</th>
<th>$V_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oscillations</td>
<td>0.5</td>
<td>2.2</td>
<td>1.5</td>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
<td>0.91</td>
<td>1.67</td>
</tr>
<tr>
<td>Patterning</td>
<td>0.1</td>
<td>1.0</td>
<td>0.9</td>
<td>1.0</td>
<td>1</td>
<td>40</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Figure 4.16: The Sel’kov-Schnakenberg reaction diffusion system (see equations (2.14) and (2.15)) showing both oscillatory and patterning dynamics. The two simulations were performed using the same equations with only the initial conditions are parameters sets changed (see table 4.7). a) The simulations were initiated by perturbing about the fixed point by adding uniformly distributed noise at each node of the FE mesh (see equations (4.20)–(4.21)). For both traces, the solid line indicates the mean average for the state variable across the domain and the shaded regions show the range of the values. b) For the oscillation parameter set in table 4.7, oscillations of both concentration across the whole domain may be seen. c) For the patterning parameter set in table 4.7 patterning in both the $U$ and $V$ concentrations may be seen.
Figure 4.17: The Sel’kov-Schnakenberg reaction diffusion system (equations (2.14) and (2.15)) simulated through ChemChaste aligned with a separate bifurcation analysis for the homogeneous system (equations (2.12) and (2.13)). The bifurcation diagrams a) and b) show the steady states for chemical species $U$ and $V$ respectively for a range of values for the parameter $k_1$. The values $k_1 = 0.5$ shows an oscillating state as shown in figure 4.16-b) and the values for $k_1 = 1.5$ shows a stable state as shown in figure 4.16-c). For the oscillating scenario, the stable states oscillate with an approximate amplitude of 1.8 and 2.8 for $U$ and $V$ as seen in figure 4.16. The oscillation periods are shown in c) where a value $k_1 = 0.5$ provides the period for figure 4.16-b) oscillations.

The Sel’kov-Schnakenberg test case demonstrates the ability for ChemChaste to model complex reaction phenomena that are only present when spatial effects are considered. This test also motivates the complexity in chemical gradients that may be produced by simple reaction systems and when cells are coupled to these gradients the spatial effects change the local cell environment. Therefore coupling the cells to the environment may affect the cell population composition in ways that the homogeneous reaction environment would fail to model. This test case also presented an example of a simulation directory for the domain_only simulation type. ChemChaste correctly parsed the chemical reactions and modelled the PDEs producing results in accordance with the expected results from linear stability analyses.
Cooperator-cheater model through enzyme excretion

A clear example of the coupling between an environment and cell population is demonstrated in the scavenging enzyme, or siderophore, mechanism. This mechanism is used by cells to harvest some material from the environment, whether iron containing compounds or energy sources such as glucose, that are not readily intaken by the cell (Schiessl et al., 2017; Page, 2019). An enzyme is excreted that binds to the target molecule and facilitates the uptake through membrane effects.

When the cell excretes the enzyme, diffusion dynamics will transport the enzyme away from the cell and, after binding to the target molecule, the enzyme may be taken in by a different cell. In a heterogeneous population a subset of cells may not spend metabolic energy to produce the scavenging enzyme but benefit from the harvesting mechanism. Therefore the deficient cell types may exhibit a greater growth rate when the enzyme source is readily available in the environment. When small enzyme concentrations are present and spatial diffusion affect the distribution of the enzyme a cooperator-cheater game may be present (Griffin et al., 2004; Tudge et al., 2016; Rubin and Doebeli, 2017).

Cooperator-cheater mechanisms have been studied in the cell population context revealing the organisation of cell types throughout the population (Nadell et al., 2006; Mitri et al., 2016; Momeni et al., 2013). Common organisations are for cheater cells to be expelled to the periphery of the system due to rapid expansion and cooperator monopolies and for stratification of colonies under resource competition. These organisations have been linked to changes in the growth rate of the two cell species where the cooperator experiences a growth rate cost of cooperation and where the cheater does not. The cooperating cell experiences a growth cost to produce the enzyme and excretes this into the environment as a shared "public good" (Drescher et al., 2014). The cheater cell however utilises the public good to archive a higher growth rate without adding to the shared enzyme pool. If however the cheater cells inhabit a region without the enzyme source the growth rate will be inhibited without a mechanism for enzyme production. Conversely, regions of cooperator cells will always be able to produce the enzyme and continue to grow (Borenstein et al., 2013).

This scavenging mechanism has also been investigated as a method to introduce antibiotic molecules into a target cell through a "Trojan horse" mechanism (Miethke and Marahiel, 2007). A binding rate or interaction process between the enzyme and the target molecule may be modelled as a chemical reaction in the environment therefore to effectively model these mechanisms a simulation framework needs to be capable of modelling both cell bound reactions, selective excretion and
uptake, and environment reactions. Therefore developing a simulation framework to understand these spatial effects can have an impact on a range of studies. However, as discussed in chapter 3 this functionality is not readily available in developed software. Therefore we demonstrate that ChemChaste can model this mechanism.

We consider a simple system involving two cell types, a cooperator and a cheater cell, where the cooperator excretes an enzyme that binds to an environment substrate. Both the enzyme and it’s bound form are diffusible throughout the environment and may be taken up by all cells, while the cell boundaries are impermeable to the free substrate. When the cells have internalised the substrate a toy metabolic system is performed in both cell types to convert the substrate into a metabolic precursor. This precursor is utilised to produce a "biomass" molecule which is used to determine cell division events. In the cooperator cell some quantity of this precursor is diverted into enzyme production therefore reducing the biomass per internalised substrate molecule, the metabolic cost of cooperation.
To model the cooperator-cheater mechanism we consider a PDE system of four chemicals representing the environment oxygen, the substrate, $S$, which the cells are impermeable to, and the diffusion of the scavenging enzyme, $E$, and substrate-enzyme bound form, $ES$. The oxygen, enzyme, and enzyme-substrate complex may permeate the cell membranes with a symmetric exchange coefficient of 1.0 with no bias in either direction. A simple metabolic sub-model of the respiration
Table 4.8: Parameter values for the cell internal dynamics in the cooperator-cheater model. The reaction rates for the metabolic sub model are the same for both species except the rate for the enzyme pathway, $k_5$, shown in figure 4.18. The reaction system is described through equations 4.22–4.26.

The simulation values for these parameters are given in tables 4.8 and 4.9. These reaction systems were modelled with zero-Neumann BCs on the domain boundary for both enzyme forms and a small influx of substrate and oxygen, Neumann BCs with a value of 1.0, to replenish the system.

\[
\begin{array}{ccccccc}
\text{Cell type} & k_1 & k_2 & k_3 & k_{-3} & k_4 & k_5 \\
\text{Cooperator} & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\
\text{Cheater} & 1.0 & 1.0 & 1.0 & 1.0 & 0.0 & 0.0 \\
\end{array}
\]
Table 4.9: Environment parameters for the cooperator-cheater simulation. The diffusion rate coefficients for the enzyme, $D_E$, and enzyme-substrate complex, $D_{ES}$, are lower than the substrate, $D_S$, and oxygen, $D_{oxygen}$, coefficients to reflect the size of the structure inhibiting the diffusion. The environment reactions (equations (4.27)–(4.28)) show the reversible enzyme-substrate interaction with a bias in the binding direction, $k^e_1 > k^e_{-1}$, and the degradation of the enzyme in the environment, $k^e_2 > 0$.

Figure 4.19: Heatmaps for the enzyme, $E$, and substrate, $S$, concentrations across the domain at $t = 10000$. A) The enzyme concentration is elevated in the presence of cooperator cells (white) where the size of the cells represent the amount of biomass contained within. The substrate concentration is depleted at the centre due to reactions with $E$ and uptake of $ES$ by the cells. Additionally the substrate is added at the boundary of the domain due to the Neumann boundary conditions. The parameters for the simulation are given in tables 4.8 and 4.9. b) Cartoons demonstrating the boundary conditions (BCs) for the concentrations. $E$ has no-flux BCs while the substrate has a flux into the domain, a Neumann BC value of 1.0.

Figure 4.19 shows the organisation of cells within a cooperator-cheater system with the formation of chemical gradients in the environment. In a well-mixed scenario ($D \to \infty$) the higher growth rate for the cheater cells, due to complete processing of substrate to biomass, will enable them to out-compete the slower growing cooperator cells. Meanwhile, in the isolated scenario ($D \to 0$) the cheater cells will be unable
to grow due to the absence of the scavenging enzyme which results in there being no internalised substrate from which to produce biomass from. Therefore the diffusion rate for the scavenging enzyme will change which cell out competes the other. For the diffusion rates considered (see table 4.9), higher enzyme concentrations can be seen in the surroundings of the cooperator cells due to the excretion of the metabolic product. The simulation results show an increase in the biomass production rate, and therein growth rate, for the cooperator species when compared to the cheater species for simulations where the diffusion of nutrients through space is considered. Therefore when we consider a non-extreme rate of diffusion for the scavenging enzyme model spatial segregation can be seen as a consequence of the explicit modelling of space. Overall, figure 4.19 shows a simple environment-cell feedback mechanism and demonstrates the ability for ChemChaste to model such systems from a set of chemical reactions and cell properties.

4.6.3 Model exploration of the cooperator-cheater system

A common method for exploring computational models is to change the parameter values and inspect the simulation output for changes in behaviour. This process can be time intensive if new simulation files are needed for each change of parameter values. To speed up the model exploration process and assist documenting reproducible simulations, ChemChaste provides a procedure for sweeping over sets of parameters (see section 4.4.1 for details). The sweeping capabilities of ChemChaste was used to explore the cooperator-cheater model (see section 4.6.2) for different growth rates of both species and production rates of the scavenging enzyme. The sweeping procedure automatically duplicates and alters the simulation files for different sets of the swept parameters and constructs a resulting parameter file to document the individual simulations (shown in figure 4.11). These simulations were performed with the output showing spatial organisation of the different cell types may be seen in figure 4.22 with enhanced spatial organisation when compared to figures 4.20–4.19. To further develop the model exploration, a selection of the output summary statistics for the simulation have been shown in figure 4.23.

Figure 4.20 presents a series of snapshots for one instance of the cooperator-cheater model with a set of swept parameters. The cells grow through the metabolic sub-model fuelled by substrate scavenging and cooperator-cheater mechanism with the internal cell biomass being shown by the size of the cells in the figures. The figures were made by post-processing the ChemChaste output in the Paraview software (Ahrens et al., 2005). The simulation data was imported into Paraview with the Glyph tool filter being used to represent the cells in the
simulation. To view all cells in the system, the glyph mode was set to "All Points" and Glyph Type set to "Sphere". The size of the cell sphere is set through the Scale Array value where the cell internal "biomass" value is selected with a Scale Factor set to 1.0. The colour of each cell is determined by the "Strain type" value. For a progenitor cell of both types the internal chemical concentrations are shown in figure 4.21 where sharp discontinuities changes in the concentrations indicate a cell division event which occurs when the cell hits a threshold of $\text{biomass} = 1.0$ and the chemical concentrations are divided equally between the parent and offspring cell. The plot was also made in Paraview by selecting the cell data and using the Plot data over time filter. The filter was used to separate two cell from the community and plot the different concentrations over course of the simulation.

Figure 4.20: Snapshots showing the growth of a cell community with cooperator-cheater dynamics (equations (4.22)–(4.28)). The simulation was initialised with a cluster of 9 cells, 4 cheater cells and 5 cooperator cells, as shown in a). The evolution of the simulation is shown in b) for a total of 58 cells at the end of the simulation. The simulation parameters are the same as in tables 4.8 and 4.9 except with $k_4 = 3.0$ for both cell types and $k_5 = 3.0$ for the cooperator cell.
Figure 4.21: Plot of the chemical concentrations within the progenitor cooperator cells and cheater cells. From the initial cell community, concentration traces for one of the progenitor cooperator cells (solid lines) and one of the cheater cells (dashed lines) are plotted over the simulation. The parameters and initial conditions are the same as in figure 4.20.

The final community composition for each of the swept simulations are shown in figure 4.22. The community grows from a small group of 9 cells into a large cluster of 58 cells each running an internal reaction system and integrating both mechanically, through cell-cell physical Hookean springs, and chemically through the diffusion of excreted compounds such as the scavenging enzyme. The initial group of cells, starting near to the cell division threshold, rapidly divide and spread through the domain through natural mechanical relaxation. The cheater cells grow quicker than the cooperator cells due to their intrinsic growth advantage as energy is not used to produce the scavenging enzyme. As the cheater cells divide and spread through the space the cooperator have greater access to the scavenging enzyme and grow over the rest of the simulation run. The growth of the communities is shown in figure 4.23.
Figure 4.22: Parameter sweeping may be used to explore the dynamics of the cooperator-cheater model (see figure 4.18). The cooperator cells are coloured "blue" and the cheater cells "red". The initial cell topology, of 9 cells, is given in figure 4.20-a) with the simulation results shown for varying the growth rate of the cell (Biomass production rate in figure 4.18) and the enzyme production rate (for the cooperator cell only). The environment in the plots shows the substrate concentration with high concentrations at the boundary and depleted near the cell community.
Figure 4.23: The ChemChaste summary statistics can be used to explore parameter sweeping results. These summary reflect the sweeping results shown in figure 4.22. The \textit{Biomass} production rate is given by \textit{Growth} and the enzyme production rate by \textit{Kine}. a) and b) show counts of cells \textit{A} and \textit{B} respectively. c) Displays the Shannon heterogeneity index for the simulations showing a decrease in heterogeneity when like cells are neighbours and d) the Lee-Oyburn autocorrelation index which is greater when structures emerge through the domain. The heterogeneity index is sensitive to the emergence of local order, such as when cells separate into blocks of their own type, and the autocorrelation index is sensitive to the emergence of complex patterning in the system.

### 4.7 Conclusion

The core of ChemChaste is distributed over 140 class files for approximately 40000 lines of C++ code. These classes are supported by Python scripts, to assist with a user’s simulations, and by unit test files used during code development and verification. The code was written following a test driven development method. Here, the individual classes were written and tested against simple test cases before the next class was written. Therefore a modular structure naturally formed where each constituent element could be incorporated into the wider system with confidence. This unit test method followed the approach taken by the Chaste development team to ensure the same quality of code development. Additionally, care was taken with the class and function names to ensure they followed the Chaste naming conventions.

This development has fulfilled the aims presented in section 4.5. The
deployment of ChemChaste is computer architecture independent through the use of
the docker container system. For computationally expensive simulation with large
domains or cell numbers, the docker deployment assists running ChemChaste on
high performance computing systems. Further development of the simulation suite
is possible through the addition of new reaction or transport laws or by editing
the underlying ChemChaste executable. Expansions are possible by adding to the
inheritance of the class based structure used to add onto the original Chaste code.
These additions are readily integratable to ChemChaste without the need to edit
the underlying codes or affecting the wider Chaste system.

The stability of the inherited Chaste solver routines were explored in section
4.5, where simulation output issues were identified through modelling the sourceless
heat equation. These problems were recognised within the Chaste code and
demonstrated to impose a negligible effect on the results of complex systems. The
convergence of solutions for the Fisher-KPP equation was demonstrated using the
ChemChaste suite. After verification, the ChemChaste simulation was applied to a
range of spatial problems to investigate the possible dynamics present in bacterial
biofilms, where known results may be used as a simulation target. In chapter 5,
ChemChaste is used to study the spatial dynamics of a cell population where the
results is unknown.
Chapter 5

Analysis of cell-environment feedback in cellular systems

5.1 Introduction

The chemical complexity of a cell containing environment can produce myriad emergent phenomena. The formation of chemical gradients due to niche formation and subsequent heterogeneity in cell populations have been discussed in previous chapters. In chapter 1, we considered how changes in environment concentrations can affect the cell state and how cells change their environment through the uptake and excretion of metabolites. Furthermore, these cells can indirectly interact through the sharing of metabolic products during cross-feeding processes where molecules are excreted and diffuse throughout the community before being taken in by neighbouring cells. The flow of these chemicals through a system such as biofilms can be regulated via changes in the diffusion rates. In chapter 2, we discussed how molecular diffusion can lead to spatial organisation in reaction systems and how cells may be coupled to these reaction systems. The presence of cells were also shown to be capable of strongly perturbing the admissible environment dynamics. Therefore we have shown that the environmental chemical reactions and the diffusion of molecules across the domain motivates the onset of the spatial heterogeneity that affects cell proliferation and community composition.

The chemical reaction composition of the cellular microenvironment can influence their own metabolism and can alter their chemical consumption and excretion profiles. Thus, there can be a feedback cycle operating between the cell metabolism and the chemical microenvironment. This feedback can influence both the spatial organisation of cell populations and their local chemical environment
chemical composition in turn. One method to incite feedback is through the creation of chemical gradients across the cell population where cells inhabit spatial niches. Different cell species have different preferences for their environment. They prefer an environment that supplies the necessary nutrients in good supply and have conditions that are favourable for the cell growth such as optimal pH levels, in these environment their growth rate is maximal. However, communities of cells may have a range of preferences while inhabiting the same spatial domain. Therefore the cells will wish to either move towards a preferred niche or tailor their microenvironment to construct the niche around themselves.

Niches are constructed by the cell’s own excretions, reactions in the microenvironment, and diffusion-driven distribution of environmental chemicals which drive these spatial phenomena. In these multispecies communities, the formation of localised niches form heterogeneous communities where the neighbourhood of each cell can vary across the population. Therefore each cell can experience different growth conditions and the interplay of the different regions can cause complex cell communities that cannot be understood through cell interactions alone. This hypothesis is considered from three related angles; how the spatial distribution of cells running fixed metabolic systems interacts in an environment described by the pH, how the coupling of cells to the environment can produce chemical gradients, and how the competition for resources over spatially distributed systems can drive population stratification and niche formation through the diffusion of nutrients.

In this chapter we use our hybrid continuum-discrete modelling of cell-environment coupling to present illustrative results to demonstrate the importance of considering these coupling processes. Using the new simulation capabilities introduced in ChemChaste, we build upon an ODE model to spatially model and increase the model complexity until only the hybrid continuum-discrete formulation can investigate the system. We use the results to drive further investigations into the consequences of coupling in determining community growth and fate.

We discuss the phenomena of bistability in the population of two different cell species as a possible emergent property (see section 5.2). In two species communities each cell has an opportunity to be dominant but we consider how spatial coupling can perturb the bistable dynamics (see section 5.3). Using the same coupled mechanism, we also show how spatial heterogeneity in the cell population can be supported through the formation of local niches and demonstrate how diffusion changes this heterogeneity. We then introduce bias to the cell growth rate to investigate how
a change in the dominant species can alter the chemical niche. Finally, the onset of cellular patterning due to the chemical gradients is then expanded through the introduction of a nutrient in the environment where competition for resources can lead to an enhanced exploration of the domain space (see section 5.4).

5.2 Well-mixed models have limited ability to capture the effects of spatial coupling

Well-mixed models based on systems of ODEs have been developed for cell-environment coupling. These ODE models represent the cell coupling strength as contributions to a general environment variable. The models however do not explicitly consider the domain space and subsequently do not fully capture the heterogeneous effects of spatially distributed communities. Here, we consider an instance of these ODE models and examine the ability for a well-mixed model to investigate the spatial effects of coupled cell-environment systems.

One model developed to study the effects of coupling between cell species and a shared environment has been produced by (Ratzke and Gore, 2018). The authors presented an ODE system to model the proliferation of cells which change the pH of their environment (see equations (5.1)–(5.3) with parameters 5.1). The authors consider the cell species to have a high or low preference for the environmental condition (pH) and through the coupling alter the environment. The alteration can benefit themselves at the expense of the other species or to benefit the other species by converting the environment to the other species preference at their own expense. The authors show that, for a select choice of coupling parameters, the model posits a bistable state for the whole population where one species reaches dominance and the second state extinction and vice-versa.

The model proposed by Ratzke and Gore (2018) involves two cell species, \((a, b)\), whose growth rates are coupled through changing the environment \(pH\):

\[
\frac{\partial n_a}{\partial t} = n_a \left(1 - \frac{n_a}{K}\right) \left(\exp\left(-\frac{(pH - pH_{\text{pref}a})^2}{\sigma^2}\right) - \delta\right) \tag{5.1}
\]

\[
\frac{\partial n_b}{\partial t} = n_b \left(1 - \frac{n_b}{K}\right) \left(\exp\left(-\frac{(pH - pH_{\text{pref}b})^2}{\sigma^2}\right) - \delta\right) \tag{5.2}
\]

\[
\frac{\partial pH}{\partial t} = (n_ac_a + n_bc_b) \left(\frac{pH}{b} (2b - pH)\right) \tag{5.3}
\]

where \(n_a, n_b \in \mathbb{R}_{\geq 0}\) are the cell densities for cells with a preferred environment pH \(pH_{\text{pref}a}, pH_{\text{pref}b} \in \mathbb{R}_{\geq 0}\). The growth rate for the cells is motivated by a logistic
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preferred environment pH for cell &quot;a&quot;</td>
<td>pH_{prena}</td>
<td>[0, 10]</td>
</tr>
<tr>
<td>Preferred environment pH for cell &quot;b&quot;</td>
<td>pH_{prenb}</td>
<td>[0, 10]</td>
</tr>
<tr>
<td>Environment-cell &quot;a&quot; coupling factor</td>
<td>c_{a}</td>
<td>0.1</td>
</tr>
<tr>
<td>Environment-cell &quot;b&quot; coupling factor</td>
<td>c_{b}</td>
<td>-0.1</td>
</tr>
<tr>
<td>Coupling term</td>
<td>\sigma</td>
<td>[0, \infty]</td>
</tr>
<tr>
<td>Environment carrying capacity</td>
<td>K</td>
<td>100</td>
</tr>
<tr>
<td>Environment dilution rate</td>
<td>\delta</td>
<td>0.5</td>
</tr>
<tr>
<td>Population asymmetry factor</td>
<td>d</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 5.1: Summary of the parameters in the ODE model (see equations (5.1)–(5.3)).

growth form for an environment with a carrying capacity $K = 100$ modelling a maximal population due to resource constraints. The environment pH is constrained within an interval $pH \in [0, 10]$ and changes due to cell excretions $c_{a}, c_{b} \in [-1, 1]$. These coupling values show that cell a excretes substances when $c_{a} > 0$ which increases the environment pH while cell b acts to decreases the pH when $c_{b} < 0$, with a asymmetry in coupling applied through the application of a factor $d = 0.8$. This factor prevents the environment from symmetrically responding to the cell types when the coupling coefficient magnitudes, $|c_{a}|, |c_{b}|$, are equal. As the cell couplings may be interchanged through swapping the coefficients the asymmetry factor can be applied without a loss of generality. Cells are removed from the simulations through dilution at a rate $\delta = 0.5$ however this dilution does not affect the environment. This dilution was added to the model by Ratzke and Gore (2018) to consider the death of the cells or from the cells leaving the domain in a chemostat. Mathematically, the dilution rate acts to punish the cell with a lower growth rate but high cell numbers and enable a switch in the long term state of the system. The model parameters are summarised in table 5.1.

The growth of each species is maximal when the environment is at the preferred pH value therefore the cells can obtain a growth advantage by changing the environment pH in their favour. Cell species a prefers a high pH value so acts to increase the environment pH while cell species b prefers a low pH and so decreases the environment pH. As the cells have opposite preferences, the coupling is antagonistic between species therefore we refer to this coupling regime as "selfish coupling". In the well mixed selfish regime, the higher density species will alter the environment to a greater effect. Therefore a higher density species will have a growth advantage
over the lower density species and bistability in a 2 species population may emerge through the extinction of one species and fixation of the other species (see figure 5.1).

The value of $\sigma \in \mathbb{R}_{\geq 0}$ represents the degree to which the cells are coupled to the environmental changes caused by the other cells in the population. In the low coupling limit, $\sigma \to 0$, the growth of each species has a stronger coupling to the pH. In this limit the Gaussian function controlling the growth rate reduces in spread and the changes to the growth rate due to difference in the environment and preferred pH is reduced. In terms of the phenomena model, the Gaussian function captures the ability of a cell to "communicate" and be effected by the other cells in the population through changes to the environment pH. As $\sigma \to 0$ the cell becomes more focused on the local surroundings and act as an isolated cell. Therefore the effects of pH variation become more pronounced. Conversely, in the high coupling limit $\sigma \to \infty$ equation (5.1) tends to:

$$\frac{\partial n_a}{\partial t} = An_a \left(1 - \frac{n_a}{K}\right)$$

(5.4)

where $A = (1 - \delta) \in \mathbb{R}$ is a constant. For high dilution rates, $A < 0$, the population decays whereas for low dilution, $A > 0$, the population growth is fully captured by logistic growth controlled by the resource carrying capacity $K$. In this limit the cells act as if they cells were isolated from the environment and experience their preferred environment and only the carry capacity of the environment effects the growth rate of the cell population. As changes to the environment do not change the growth rate there is no spatial ordering or patterning in the cell types. Therefore the local variation of pH has a lower/no impact. In the region of $\sigma$-space considered by Ratzke and Gore (2018), the growth of the two species are coupled through the environment pH and bistability in the population is evident. This region is considered in figure 5.1 to demonstrate the bistability of the populations for different $\sigma$ values and initial cell ratios. The bistability phenomena was further investigated by determining the bifurcation diagram for equations (5.1)–(5.3) using $\sigma$ as the bifurcation parameter.
Figure 5.1: The Ratzke and Gore (2018) ODE model may be used to determine the population of each species in a community connected through a shared environment. The cell number density of each species, \(a\) and \(b\) is presented alongside the change in environment pH where the cells have environment preferences of \(pH_{\text{pref}} = 9.0\) and \(pH_{\text{pref}} = 1.0\) and with model parameters given in table 5.1. Different initial conditions, shown by different lines types (solid, and dashed, and dash-dot) in the plots, are considered with different long term distributions of the cell types. The three plots a)–c) were produced for three different values of \(\sigma = (1.0, 4.0, 11.0)\) representing perturbations of the parameter in equations (5.1)–(5.3). The plots change with the coupling parameter \(\sigma\) with extinction of both cell types occurring for low values of \(\sigma\) a) and mutual growth for both types at high \(\sigma\) c) for the model parameters considered. For intermediate coupling parameters, b), these plots show bistability in the populations where one cell type achieves fixation and the other extinction.
The authors used their model to investigate the cell densities for different coupling regimes and compared the simulations to experimental results. They supported the view that coupled cell systems can exhibit complex population dynamics (see chapter 1). In the authors model (equations (5.1)-(5.3)) the composition of the population is dependent on the coupling behaviours $c_a$ and $c_b$ and the initial cell ratio $n_a/n_b$ (see figure 5.1). For an initial pH value of 5, coupling of $c_a = 0.1$ and $c_b = -0.1$, and an appropriate value for $\sigma \in [1.2, 10.8]$, the community reaches dominance of cell species $a$ and extinction of species $b$ if $n_a/n_b \geq 1$ and dominance of $b$ and extinction of $a$ if $n_a/n_b < 1$. Changing the coupling behaviour the population composition changes and bistability is lost while selecting $\sigma$ values outside of the interval can also lose stability, as shown in figure 5.2.

To evaluate the ability of ODE models to capture the effects that considering a spatially distributed cell community has on the population we compare the model equations (5.1) and (5.2) to simulations using the ChemChaste suite. We propose that community interactions don’t fully capture cell-environment interactions in the
non-spatial model as mixing of cells affect the ratios of cell a to cell b as shown in figure 5.1. ODE models are limited to the summary statistics that explain the population as a whole, such as the cell densities of a and b or the ratio between them. Additionally, the coupling between cells and environment is not localised except for the introduction of an abstract coupling effect so that excretions affect all cells in the system. Therefore the ability for ODE models to study spatial cell-environment feedback is limited. We question, what additional effects can a truly spatial model reveal? Additionally, spatial models allow for the investigation into the onset of local pH and cell patterns that aren’t provided in ODE models. Spatial phenomena were discussed by Ratzke and Gore (2018) through in environment changing interactions between the cell types however patterning was not explicitly considered. The onset of patterning in the environment may drive spatial heterogeneity of cell types (Barbier et al., 2022) and the summary values cannot demonstrate patterning or wave solutions and therefore restrict the ability to analyse these emergent properties.

5.3 Cell-environment coupling affects the proliferation of cells supporting bistable regimes

The ODE model developed by Ratzke and Gore (2018) described the cell-environment dynamics in a non-spatial population (see equations (5.1)-(5.3)). We suppose that the consideration of the spatial component of the system is a necessity to effectively model cell-environment feedback dynamics in cell populations. To investigate the degree to which the spatial component is a necessity we aim to demonstrate that the bistability seen in the non-spatial ODE cell model (see figures 5.1 and 5.2) displays the same phenomena in an explicit spatial model. To view the explicit spatial system we use ChemChaste to simulate a similar model where the growth rate of the cell is dependent on an environmental pH preference and the cells change their local environment through excreting pH.

In ChemChaste the cells are modelled as individual agents within a shared environment (see section 4). Therefore in place of the population growth rates and environment ODE, seen in equations (5.1)-(5.3), ChemChaste formulates the model as biomass growth rates for each cell with the environment modelled as the diffusion of excreted pH. Therefore the descriptions for the coupling between cells and environment different between the ODE model and the ChemChaste simulation. In place of the coupling factor σ and carrying capacity K, which describe the "communication" between the cells and the total population size that the environment can support, ChemChaste uses the diffusion of the pH through
the environment $D_{pH}$ and the mechanical cell model which inhibits cell division when space is unavailable. The cell dilution rate, $\delta$, is analogous to using a large domain for the spatial model as there is more space for the cells to grow as the community envelope expands. Furthermore, we use a different function to describe the cell growth. Rather than the Gaussian function for the growth terms in equations (5.1) and (5.2) that depend on the squared $pH$ differences we use an exponential function where the growth rate decays when difference between the preferred and environment $pH$ is large. This change was done to simplify the model description. The ChemChaste simulation files are give in Appendix E.

We implement the selfish coupling scenario by making the cells change the pH in their local vicinity. The cells have a preference for the environment pH with the low pH favouring cell B having a preferred pH $[pH]_{B}^{pref} = 3$ and decreasing the environment pH in the surroundings of the cell. Similarly, the high pH condition species, cell A has a pH preference $[pH]_{A}^{pref} = 7$ and increases the pH. The growth rate of the cells is dependent on the environment pH. We utilise the PreferredEnvironmentMassActionReaction class provided in the current release of ChemChaste to model the growth of the cell with a growth rate;

$$\frac{d[Biomass]}{dt} = k_f \exp \left( -\left| [pH]^{env} - [pH]^{pref}_{A,B} \right| \right)$$

is given for cell species $A$ with an equivalent form for cell B and where the maximal growth rate coefficient is given by $k_f \in \mathbb{R}_{\geq 0}$.

The cells divide when the internal biomass concentration hits a threshold $[Biomass] \geq 1.5$ and a new cell is added to the simulation (see chapter 4). After a division event the biomass is shared between parent and offspring, setting the value $[Biomass] = 0.75$ for both cells. We use the Euclidean distance between the environment and the cell’s preferred environment to reduce the growth rate in unpreferred conditions. The cells change the environment through implementing the MassActionTransportReaction transport law with rate coefficients $k_{f_B} = -10$ and $k_{f_A} = 10$ for the two cell species. These rates are equivalent to the negative coupling for cell $B$, decreasing the pH, and positive coupling for cell $A$, increasing the pH. No-flux Neumann boundary conditions (BCs) are implemented for the environment diffusion of $pH$.

The total biomass for each cell type in the system grows exponentially through the biomass production within each cell;

$$[Biomass]_{A}(t) = k_f \exp \left( -\left| [pH]^{env} - [pH]^{pref}_{A,B} \right| \right) t + [Biomass](t = 0)$$

117
for both cell types $A$ and $B$, changing only through the preferred $pH$, and where
the initial biomass is given by the initial number of cell type, $N_A(t = 0)$ with a
starting biomass value of $1.0$ (i.e $[Biomass]_A(t = 0) = N_0^A$). The cells divide when
the cell biomass reaches a threshold $[Biomass]_{Threshold} = 1.5$ and the biomass is
reset to $[Biomass]_{reset} = 0.75$. Therefore when the biomass reached this threshold
the number of cells doubles, $N_A(t) = N_0^A 2^{t/\tau_A}$, where

$$
\tau_{A,B} = \frac{([Biomass]_{Threshold} - [Biomass]_{reset})}{k_f \exp\left(-\left[pH\right]_{env} - [pH]_{pref}_{A,B}\right)} \tag{5.7}
$$

meanwhile for the ODE model in the high coupling limit where molecular crowding
effects aren’t modelled (see equation (5.4))

$$
N_A(t) \sim \frac{e^{At}}{e^{At} + 1} \tag{5.8}
$$

which follows a sigmoid form with $A = 1 - \delta$. Therefore both models follow an
exponential growth for small cell numbers before the environmental limitations inhibit successive growth. The differences in both approaches stem from the
treatment of the environment. The ChemChaste simulations diverge from the
previous model in the implementation of the cell-environment coupling and
qualitatively compare the output. We do not consider an analogue for the asymmetry parameter, $d$, using the passive movement of the cells through the space due to
cell-cell mechanical relaxation to perturb the system. We also do not include an
environment dilution rate, however we terminate the simulation before complete
covering of the domain space (i.e consider a larger domain than the cell population).
We explicitly model the transport of pH through the environment and vary the
diffusion rate of the pH pseudo-species to demonstrate the effect explicit spatial
coupling has on the population composition.

The explicit spatial model was simulated in ChemChaste with the simulation
output processed using Paraview. The effects of the cell-environment coupling on the
ratio of cell $A$ and cell $B$ in the simulation is demonstrated in figure 5.3-a) for different
initial cell ratios shown in 5.3-b). The cell population ratio $n_a/n_b$ is presented for the
uncoupled case, $(kf_A, kf_B) = (0, 0)$, and the coupled case, $(kf_A, kf_B) = (10, -10)$. The
results demonstrate bistable states for the population ratio that depend on
the initial cell ratio with slight variation due to the diffusion rate coefficient for
$pH$. The number of cells of both types is presented in figure 5.4 which shows the
difference that the cell-environment coupling has on the number of cells in the system.
With coupling considered, the cells produce more biomass quicker (see equation 5.5)
leading to earlier cell divisions and increased cell counts. As the population ratio is varied, the effects of diffusion influence the formation of spatial structure (Van Gestel et al., 2014). The variation in diffusion rate shows a slight splitting of cell traces and a divergence in the population dynamics for the length of simulation considered. This is related to exposure of the cells to favourable pH conditions which accelerates the bistable dynamics through the $pH$ diffusion in the environment and cell excretion dynamics.

The spatial distribution of the cell types is shown in figure 5.5 showing the initial cell distribution and the final cell distribution after 2000 time steps. The evolution of a single simulation instance is shown through snapshots in figure 5.3. The figures show that under the simple model of preferential environment growth the initial population ratio determines the final community composition with the prevalent cell type having a greater impact on the cell environment. That is, for a larger initial population of cells preferring a high $pH$ environment and increasing the local environment $pH$, as shown by the black cells in figures 5.5 and 5.3, the final composition will contain a high ratio of the high $pH$ cell.
Figure 5.3: ChemChaste simulations demonstrating the effect of cell-environment coupling on the ratio of two species in cell population starting at different population ratios. a) Comparing the coupled and uncoupled population ratio for the same environment, $D_{pH} = 1.0$, and initial conditions. For the coupled case diverges from the uncoupled case as the cells act to tailor the environment in their favour (selfish coupling) b) The effect of the environment diffusion rate for the coupled cell-environment case. Greater diffusion rates tend to amplify the population ratio change due to selfish coupling. c) The initial cell topology for the three simulations. Each simulation varies the initial proportion of the two cells; cell B the white cell, and Cell A is represented by the black cell where the cell topologies represent a separated inoculation. The environment pH is initialised to a uniform distribution, $pH(x, t = 0) = 5$. 

- Coupled
- Uncoupled

- $D_{pH} = 0.1$
- $D_{pH} = 5.0$
- $D_{pH} = 1.0$
- $D_{pH} = 10.0$
Figure 5.4: Plots showing the number of cells of each type in the environment-cell model comparing the coupled to the uncoupled case. In the coupled case (orange lines) the cells change the $pH$ of the environment towards the cell’s preference while in the uncoupled case (green lines) the environment does not change. The traces show the cell numbers for different initial ratios of cell A to cell B (lines types).

Figure 5.5: Plot showing the initial and final cell populations for a cell-environment coupled simulation with different initial cell ratios. The same initial cell communities from figure 5.3-c) are shown in a) with the final community at time $t = 2000$ shown in C). The cell populations overlay the $pH$ distributions in the environment with high $pH$ around the location of cells of type A (black cells) and low $pH$ around cells of type B (white cells).
Figure 5.6: Snapshots showing the evolution of a cell population with an environment dependent growth rate. The population grows with environment dependent growth rates (equation (5.5)) for two cells with equal initial cell numbers and preferences for high and low pH; \([pH]_B^{\text{pref}} = 3\) for the white cells (Cell B) and \([pH]_B^{\text{pref}} = 7\) for the black cells (Cell A). As the cells grow they change the environment with transport coupling terms \((k_{fA}, k_{fB}) = (10, -10)\).

The results presented in figures 5.3–5.3 build upon the cell-environment model through introducing the hybrid-continuum discrete simulation framework. Both ODE and hybrid models demonstrate bistability in the cell population where the ratio of cells depends on the environment, \(\sigma\) in the ODE model and the diffusion rate coefficient \(D_{pH}\) for the hybrid model. The differences in the model stem from the physical description of the cells with the hybrid model providing a more physical description with few modelling assumptions and parameters than the ODE model.

The simulation results corroborate the findings of the ODE model while introducing the explicit spatial representation of cell agents within a domain, therefore verifying the ChemChaste framework as a test bed for cell-environment modelling. While the figures are suggestive of the spatial impact on the bistable dynamics, more analyses are needed to understand exact effects of diffusion. With the introduction of diffusion we can further explore the cell-environment coupling without the abstractions implemented in the ODE model. For example we can consider how different initial cell topologies affect the spatial heterogeneity of the growing cell population.

The simulation for figures 5.3–5.3 used separated cell clusters for the initial
cell community where like cells were together in two segments. Subsequently, when
the cells grew and divided they had a high likelihood of being surrounded by
neighbouring cells of the same type. To investigate heterogeneity in cell populations
under the cell-environment model simulations were performed for a mixed inoculate
scenario where the cells types are mixed together, alternating type A and type B
cells. Simulations were performed for an initial 50:50 ratio of cells types (5 cells
of type A, 5 cells of type B) alternating in type (see the first column in figures
5.9 and 5.9). Two different environment diffusion rates were considered $D_{pH} = 0.1$
and $D_{pH} = 10.0$ producing snapshots shown in figures 5.9 and 5.9 respectively. We
use the Getis-Ord index from the ChemChaste summary statistics (see table 4.6) to
represent each cell by the heterogeneity of the cell’s neighbourhood.

The figures were produced through post-processing using the Paraview
software (Ahrens et al., 2005), where cells were represented by "sphere" Glyphs
and coloured using the strain type or Getis-Ord index. The growth rate of the cells
is the same as used to produce figure 5.3 (selfish coupling) and the environment
is affected to the same degree. Therefore the simulations only differ through the
initial cell placements. When compared to the figure 5.3, we show the formation
of different chemical gradients and strong heterogeneity on the cell population for
figures 5.3–5.3.
Figure 5.7: Snapshots of the evolution of the cell community and environment $pH$ with low environment diffusivity. Where the white cells (Cell B) favour low $pH$ and the black cells (Cell A) prefer high $pH$ conditions and the cells are coupled to the environment with coupling terms $(k_f^A, k_f^B) = (10, -10)$. The first (top) row shows the evolution of the cell community overlaying the environment $pH$ which is shown in the second (middle) row, where the diffusion rate coefficient for the $pH$ is $D_{pH} = 0.1$. The Getis-Ord heterogeneity index (see table 4.6) is used in the third (bottom) row to colour the cells to show the diversity of the cell’s neighbours.
The spatial information presented in figures – provides a more in-depth understanding of the origin for the spatial arrangements and patterning in cell types and chemical gradients than the well-mixed ODE system. The chemical gradients are formed when clusters of one species collectively change their neighbourhood culminating in changing the wider environment (Ratzke et al., 2020). As the coupling is selfish, regions that alter the environment increases the growth rate of the cells in the region and cell proliferation. When the diffusion rate is low, the pH variations form strict defined regions where the cells types differentiate. At high diffusion, the pH variations are smoothed and strong pH variations are only formed in regions where the different cell types became dominant. Niche formation is related to the diffusion of chemicals through the environment and cells have been shown to excrete compounds to affect chemical diffusion rates. In complex communities such as bacterial biofilms, different diffusion regimes may be formed through the process of EPS excretion and collection necrotic material (Stewart et al., 2016; Süß and Visscher, 2019).

In addition to the formation of chemical niches, asymmetry in the growth rates of the two species can affect the ratio of the species within the population (Mitri et al., 2011). To demonstrate the effect of varying the cell growth rates, a simulation where the growth rates constant were varied was implemented in ChemChaste for the
50-50 mixed inoculation. These simulations produced the results shown in figure 5.9 producing population ratios shown in figure 5.10. These figures show that changing the growth rates affect which cell dominates the system, independent of changes in diffusion rate coefficients.

Figure 5.9: The growth rate of cells displaying selfish coupling affects the fixation of the cell types. The initial 50-50 mixed inoculation of cells, as seen in figures and , diverge due to heterogeneous growth rates. The cell growth rates are given by the coefficients \( a \) and \( b \) which change the coefficient, \( k_f \in (a, b) \), for the biomass growth equation (see equation (5.5)) for cells \( A \) and \( B \) respectively. The different rows show two instances of different diffusion rate coefficients for the environment \( D_{pH} = 0.1 \) for the first row and \( D_{pH} = 10.0 \) for the second row. The growth rates for both species \( A \) and \( B \) is the separate simulations. Changes in the plots are due to the growth rate and diffusion changes only. Both cells are coupled to the environment with coupling terms \( (k_fA, k_fB) = (10, -10) \) in all snapshots.
Figure 5.10: The population ratios vary depending on the growth rate of the individual species. Solid line denotes the population ratios for low diffusion $D_{pH} = 0.1$ and dotted lines denote the ratios for high diffusion cases $D_{pH} = 10.0$. The uncoupled case can also be seen by the dashed lines where the environment is not changed by the cells.

The simulations that combine the effects of both diffusion rate coefficient changes and growth rates changes can be seen in figure 5.11. The Lee-Ogburn index captures the autocorrelation between neighbourhood heterogeneity throughout the population (see table 4.5) and changes in the growth rates may be seen to dominate the diffusion rate changes in figure 5.11b.
Figure 5.11: The diffusion of pH through a cell community affects the heterogeneity patterning of cell systems. a) The spatial plots for the two species community where each cell is coloured by their Getis-Ord index value. White colouring indicates cells within a neighbourhood of low heterogeneity and red-black colouring indicates high heterogeneity. b) The effect of pH diffusion and growth rates changes on the heterogeneity autocorrelation through the community, where the changes in growth rates prefactors \((a, b)\) are discussed in figure 5.9.

As discussed previously in section 5.3, the formation of localised conditions provides a means for cells to survive in systems that would otherwise lead to their extinction. By forming small clusters and tailoring their surroundings, a low growth rate species can survive deleterious effects of an out-competing antagonistic species in a spatial setting. This is however dependent on the diffusion rate. As diffusion is increased niche formation is inhibited and the protective effects are muted. Therefore these simulations suggest that a low growth rate species can survive through coupling to the environment in scenarios where extinction is the expected result.

This demonstrates how the invasion of a higher growth species can affect the niche to be antagonistic to previous inhabitants if the diffusion rate is high. This emphasises that the chemical niche is independent of the species and a minority cell in an unfavourable niche may struggle to proliferate. However when diffusion rates are low, an out-competing invasive species would struggle to completely supplant the inhabitants whose survival is helped by the formation of more localised niches. This is demonstrated by the greater chemical heterogeneity for lower diffusion rates in the spatial plots, figure 5.9. Overall, these illustrative cases show patterns that are more ‘fragmented’ with low diffusion and more ‘bulky’ with high diffusion showing that spatial segregation of cell species is favoured in high diffusion limits with finer patterning possible in low diffusion limits.

The niche environment constructed in this section allows cell systems to exist when the expected results from the ODE system suggests otherwise. These results
cannot be determined from simple non-spatial systems such as the model presented in equations (5.1)–(5.3). The inclusion of an abstract implicit spatial term is insufficient for the purpose of modelling coupled cell-environment systems similar to the one considered by Ratzke and Gore (2018).

5.4 Diffusion drives niche formation in competitive cell systems

The modular nature of ChemChaste leads itself to readily adapting the simulation to add new mechanisms and additional dynamics without the need to reformulate the whole model. For example, competition for a shared resource may be added to the previous coupled cell-environment model (see equation 5.5) through the inclusion of a second environment variable for the resource. This increased complexity is not readily added to ODE systems without reformulating the model problem therefore ChemChaste is a natural framework within which to build these models. The competition for resources is a driver for the expansion cell communities and the onset of segregation of cell species (Kremer and Klausmeier, 2013; Estrela and Brown, 2013). In addition to the environment conditions, resource competition can affect cell population structure in a spatially distributed cell system. Therefore we consider the inclusion of cells competing for a resource into our cell-environment ChemChaste simulations.

Competition between different species within cellular systems is vital for the formation of bacterial biofilm as well as chemical niches (Hibbing et al., 2010; Oliveria et al., 2015). As shown in section 5.3, the different antagonistic interactions of pH alteration can drive domination and recession of the different species with the formation of different regions producing cell segregation and stratification in populations (Rendueles et al., 2015). Therefore we would expect the strengthening of species segregation when resource competition is considered. In the context of this model the cells "compete" passively where their growth rate is dependent on the environment resource which is depleted by the cells. Therefore the first cells to inhabit an area of the domain will have a higher growth rate than subsequent cells. Therefore the competition is won by the cells that proliferate and explore the domain quickly.

We modelled resource competition in a coupled cell-environment simulation in ChemChaste by adding a nutrient resource into the environment. The nutrient, $S$, diffuses through the environment and is taken up and utilised by the cells. The \texttt{MassActionTransportReaction} class is used to irreversibly exchange the substrate
across the cell membrane, depleting the environment and increasing the cell’s internal concentration. We use no-flux Neumann BCs for both the pH and nutrient S chemicals to represent an isolated cell system. The growth of the cells is dependent on the conversion of the nutrients into a biomass molecule; \( S \rightarrow \text{Biomass} \), where the reaction rate is dependent on the environment pH. The rate of Biomass production is given by:

\[
\frac{d[\text{Biomass}]}{dt} = k_f[S] \exp \left( - \left| \left[ pH \right]_{\text{env}} - \left[ pH \right]_{\text{pref}} \right| \right) \quad (5.9)
\]

\[
\frac{d[S]}{dt} = -k_f[S] \exp \left( - \left| \left[ pH \right]_{\text{env}} - \left[ pH \right]_{\text{pref}} \right| \right) + T \quad (5.10)
\]

where \( T \in \mathbb{R}_{\geq 0} \) is the nutrient transport into the cell (see the MassActionTransportReaction class in table 4.2) from the environment which is dependent on the environment nutrient concentration and \( k_f = 1.0 \) is the rate coefficient with inverse time units. The same system of equations is used for cell species B with interchange of pH preference \( (\left[ pH \right]_{\text{pref}}^A \rightarrow \left[ pH \right]_{\text{pref}}^B) \). The cell change the environment pH through the same selfishly coupled cell-environment model, as discussed in section 5.3, with the additional cell-environment interaction through depletion of the resource \( S \). Both cell species divide when the Biomass reaches a threshold of 1.5 while the resource \( S \) is depleted to produce the Biomass and is therefore consumed at a rate equal to the Biomass production rate. When the cells divide, the current parent Biomass and internal nutrients is shared between parent and offspring. The difference between the proliferation of both cells is the pH preference and the changing of the environment pH. The simulation results for resource competition, shown in figure 5.12, were produced from simulations whose files are provided in appendix F.
Figure 5.12: The introduction of resource competition on the spatial distribution of a selfishly coupled 50-50 mixed inoculation. The spatial snapshots at the end of the simulation (t=6000) demonstrate the environment pH distribution in the first row, the cell types (high pH = 7 preference for the black cells, low pH = 3 preference for the white cells) in the second row, and the resource S distribution in the third row. These snapshots are presented for varying diffusion rate coefficients, D, which are the same for both the pH and the resource S.

When competition is included into the simulations, the limited chemical diffusion through the environment drives the formation of chemical niches while high diffusion rates tend to more uniform chemical distributions. In regions depleted of nutrients, the formation of a preferred environment to bolster the growth rate is a necessity. This can be see in figure 5.12 through larger and more uniform regions when the diffusion rate is increased as the local perturbations due to cell growth are smoothed out. The results suggest a minimal model for stratification in biofilms where cells can affect their environment and are dependent on an environment resource. The introduction of a consumed resource which all the cells compete over leads to the cells to spread to the periphery as the cells experience a growth rate that is proportional to the nutrient concentration. Successive cell divisions of one type lead to trails expanding from the core to the periphery (see figure 5.13).

The movement of cells to exploit nutrient rich space has been studied by Gude et al. (2020) who varied cell motility to allow cells to traverse the space. However instead of considering explicit motility of the cells or chemotactic responses, we show that the variation in growth rate due to niche conditions can also lead to this spatial
competition. This resource induced patterning produced stratified populations as cell growth is promoted in the direction of higher nutrients, when comparing the resource dependent case (figure 5.12) with the nutrient independent case (figure 5.9a). This stratification is seen experimentally (see figure 5.13), suggesting that the resource competition is a driver for this expansion mechanism (Teal et al., 2006; Müller et al., 2014; Mitri et al., 2016). This stratification is a purely spatial phenomena that cannot be seen in ODE models. Furthermore, the dynamics are driven by the cell-environment coupling of cells with the chemical domain and adding further complexity that can only be performed using ChemChaste.

![Figure 5.13: Schematic showing the impact that resource competition may have on radial stratification in cell communities. A domain with unexploited resources provides a boost to the growth of a biofilm. As cells grow they deplete local resources which impact the cell’s maximal growth rate. a) As a line of cells extend into the fresh media they deplete the resources leading to preferential growth into the domain. The full resource availability (solid line) is depleted (dashed line) as new cells are added. b) In a spatial setting the resource driven growth from a cell community acts in the radial direction leading to stratification of the cell types.](image)

5.5 Conclusion

The central purpose of this work has been to demonstrate the ChemChaste software and the capabilities that the software has when considering diffusive effects of chemical concentrations within cellular systems. To this end we have provided some illustrations to the potential complexity of these coupled systems that produce
noticeable spatial changes in experimental systems. These illustrative results emphasise the importance of considering the coupled cell-environment feedbacks in cell systems and develop methods to explore this phenomena in future work.

We considered a system of ODEs that were developed by Ratzke and Gore (2018) to model the non-spatial coupling of cells to an environment. For selfish coupling the model displayed bistability in the cell population with one cell type reaching dominance and the other extinction depending on the initial cell type proportion. However, when we developed a similar spatial version of the model we showed that cells can see off extinction by changing their local environments to produce favourable conditions. Therefore the formation of chemical gradients can protect the minority cells when experiencing the antagonistic affects of a selfishly coupled dominant species. We extended this hybrid continuum-discrete model to investigate how changes in the cellular growth rates would affect the spatial organisation of the different species.

We illustrated that changes in the diffusion rates can change the chemical concentrations that affect the construction of niche conditions. Finally, the emergence of stratification and patterning of cell types was shown in a minimal model that included resource competition. These illustrative cases may be used to produce more complex simulations looking at different interaction and coupling regimes with ChemChaste simulation files provided in the associated appendices.

The simulation results presented here were produced to qualitatively describe the phenomena that were modelled and do not provide quantitative insight. This is because the parameters used in the simulations are not derived from empirical data but are introduced as toy model values. The scaling of the parameters also requires further development for quantitative results. For example, the impact the cells have on their environment is small suggesting that the coupling terms are too small to have a strong impact on the environment. These weak coupling terms are enough to demonstrate that the ChemChaste simulations are capable of modelling the phenomena but more work is require to provide an accurate representation. This is demonstrated in figure 5.11 where the cell growth rates provide a dominant effect in determining the growth of the communities rather than pH diffusion rates. Consequently, the auto-correlation is effected more by the growth rates than the diffusion rates (see figure 5.11-b)) with no consistent ordering of the traces due to diffusion changes however a noticeable change in the pH distribution may still be seen (see figure 5.11-a)). Therefore a robust connection between diffusion and cellular patterning (auto-correlation) cannot be concluded. Nonetheless, the results do indicate the additional capabilities that ChemChaste provides to the research tool.
ecosystem, as intended.

The illustrative results provided in this chapter have demonstrated the role that cell-environment feedback cycles could play on the formation and composition of cellular systems;

• We proposed that the non-spatial models for community interactions don’t fully capture cell-environment interactions as the mixing of cells affects population ratios (see section 5.2)

• We included the spatial effects of chemical diffusion and cell locations into a simulation to investigate the changes that the inclusion of spatial effects have on the population dynamics of cellular systems (figures 5.3–5.3).

• We motivated that the environment effects are mediated through the diffusion of chemicals through the population. Therefore we considered the formation of niche environments and we tested the variation and affects of niche construction through changes to the chemical diffusion rates (figures 5.9 and 5.9). We also show that variations in growth rates determine the population makeup however cells of a lower growth rate can still proliferate in local niches (figures 5.9 and 5.10). We suggest that the onset of cell type heterogeneity throughout a population is related to these diffusion and growth rates (figure 5.11)

• When simulations were preformed that included resource competition as an addition to the effects of environment variability, the simulations can produce patterning of cell species within communities. By encouraging expansion through competition (as shown in figure 5.13), the formation of local gradients was also supported (see figure 5.12).

Overall, the results presented in this chapter demonstrate the importance of explicitly modelling the spatial aspects of cellular systems. We show the capability of ChemChaste to model cell-environment feedbacks in cellular systems and compared the results to a previous ODE model. The emergence of patterning the cell species population when considering heterogeneous growth rates, mixed inoculation, and changes in diffusion rates.
Chapter 6

Conclusion and Future work

In this work we have considered the role that cellular systems, such as a cell colony in a biofilm, play on changes in their surrounding environment. We have motivated the need to consider these dynamics in the chemical environment of cellular systems and their impact on the cell behaviours through considering how cells affect their environment. We identified a trend in current software packages that overlooks the spatial aspects of chemical reactions of cellular systems; previous simulation software that couple cells to an environment do not explicitly model the chemical reactions simultaneously in every cell system and the environment which is a necessity to fully understand cellular systems. Previous work has also focused on homogeneous domains where the diffusion rates of environment resources are uniform throughout the domain which inhibits studies where heterogeneous diffusion are significant such as investigation in the extra cellular matrix or cellular cross-feeding phenomena in biofilms (see chapter 1). These gaps may be due to simulation complexity leading to slow software, experimental intricacies limiting data availability, or diverging interest in the cell biology community (see chapter 3). We developed ChemChaste (see chapter 4) which builds upon the popular Chaste platform to extend the simulation capabilities of Chaste while implementing novel cell-environment feedback methods.

ChemChaste provides a platform to investigate a large range of cellular systems. The ChemChaste simulations are general and adaptable enabling a user to implement models that are not possible to simulate in current software packages (see chapter 3). To demonstrate the simulation capabilities, ChemChaste was used to simulate systems of cells in a diffusing environment where the cells change their environment and the environment affects the cell growth rate. These simulations included chemical reactions in the environment, sub-metabolic models for cell reaction systems, and the growth and division of cells leading to heterogeneous cell
populations. Alongside the introduction of the ChemChaste software, this work has provided contributions to the theory of chemical reaction systems. More specifically, we consider reaction systems that couple cell systems and emphasised the importance of considering the reactions in the environment in tandem to considering the reactions that form the cell metabolism.

To demonstrate the capabilities of the general ChemChaste software simulation we considered bacterial biofilms as an exemplar cellular system that is coupled to a chemical environment. Biofilms can develop complex environment niches that protect and influence the proliferation of the constituent cell species (see chapter 1 for details). The cell inventory of a biofilm is related to the chemical reactions occurring within the environment and within the cells (i.e the cellular metabolism). Cells contribute to the niche construction by changing the chemical composition of the environment and these changes in turn can affect the environment reaction-diffusion system. The environment reaction systems can possess bifurcations in the chemical dynamics, such as bi-stable states or oscillations and the presence of cells can change the bifurcation dynamics (see section 2.3) (Martinez-Corrall et al., 2018; Garde et al., 2020).

When diffusion is considered, patterning and bifurcations can become coupled. This case was discussed for spatial Savannah grass systems by Rietkerk et al. (2021) (see chapter 2 for details). The growth dynamics for the grass ecosystem produces a bistable bifurcation state where the grass is healthy and growing in one case and unhealthy and dying in another. Based on environment parameters the grass can tip from healthy to unhealthy which is catastrophic for the ecosystem. The grass is however a spatial system and may produce diffusion-driven Turing patterns based on the growth conditions. When patterning occurs in the Savannah grass the changes in the environment which would normally cause the catastrophic tipping can be avoided through undergoing a sequence of transition between patterning states, therefore allowing the ecosystem to survive an environmental perturbation. We view the Savannah grass ecosystem as an analogy for a cellular system with the cells taking the place of the grass with the diffusion and cell-environment coupling producing the diffusion-driven patterning. For a sufficiently complex cellular system, it would be possible for the passage of a chemical wave to provide the necessary perturbation to trigger patterning or tipping in cell growth dynamics (by changing the concentrations in the associated bifurcation space) (Ward et al., 2003). This tipping can drastically change the environment, which may be catastrophic for the cell populations (Tytgat et al., 2019). The chemical dynamics in biofilms is not fully understood therefore an analysis of how cells affect the local concentrations is of paramount importance to
anticipate these strong ecological tipping response. ChemChaste was developed to investigate these systems.

After we discussed cell-environment models, we explored the capabilities of current software suites (see chapter 3). As we did not find a readily available general framework within which to implement our models we initiated the development of ChemChaste. ChemChaste (see chapter 4), presents an extensive framework that is adaptable to a wide range of cell based simulations. We developed ChemChaste in close collaboration with the development team behind Chaste to expand the already extensive capabilities. The powerful finite element PDE and agent based models contained within Chaste could not model the cell-environment coupling and a significant amount of work was required to achieve this. The expansive ChemChaste framework is written in a total of $\approx 40000$ lines of C++ code supplied with 140 class files with additional Python functions for parameter sweeping and input/output operations. As a hybrid continuum-discrete modelling suite, ChemChaste models the cells in a system by individuals in an agent based model with the capability of complex behaviours and cell-cell interactions. In the ChemChaste system, the environment is modelled by a general PDE system that implements nonlinear chemical reaction systems. The cell and environment models are coupled to perform the hybrid model where the cell behaviours are dependent on the local microenvironment, and the exchange of molecules between cells and the environment.

We then used ChemChaste (chapter 4) to illustrate the possible complexity of the chemical gradients in cell systems by considering cells in cooperator-cheater and resource competition scenarios. These simulations, described in chapter 5, were shown to produce the chemical gradients and environment dynamics seen in bacterial biofilms where we consider the effects of varying the molecular diffusion rates. The formation of heterogeneous cell populations was modelled where the different cell species are distributed across the domain with the formation of sub-regions of different dominant cell types. We demonstrated the abilities for cells to avoid extinction through inhabiting niche environments when in competition with high growth species. These species separations were furthered when modelling radial stratification under resource competition. These simulation results were corroborated by experiments widely presented in literature (see chapter 5 for details).

In summary this work motivates the study of, and provides the means to further investigations into the chemical dynamics of cellular systems.

- We discussed the introduction of cells within a reaction-diffusion environment and highlighted the link between cell presence and changes in the bifurcation dynamics and admissible reaction conditions. Therefore the non-trivial effect
that a presence of cells has on the environment dynamics and steady states changes the local environment experienced by the cells. Therefore we conclude that to disregard the effect on the reaction system can results in incorrect predictions for the system dynamics.

- We analysed a range of software frameworks and discussed their usability with regards to modelling chemical dynamics in cellular systems. We subsequently identified a gap in the simulation capability space and identified the need to modify existing software to plug this gap.

- We introduced ChemChaste, a hybrid continuum-discrete simulation framework for modelling the chemical dynamics in disperse cellular systems. The capabilities of ChemChaste were shown through modelling a range of spatial phenomena in chemical systems and we applied ChemChaste to demonstrate spatial segregation in cooperator-cheater systems.

- We utilised ChemChaste to investigate cell-environment feedback in biofilms and the onset of population bistability in a range of model scenarios. As bacterial cells are known to excrete extracellular materials (ECM and EPS) in biofilm structures, they have a method to influence the diffusion of substances throughout the film. Therefore we considered the role diffusion plays in the onset of complex gradient formation and demonstrated this by varying the diffusion rate of environment markers throughout the cellular system.

There is a wide range of applications to the work presented here. When considering cells as chemical factories engaging in chemical synthesis, increases in metabolic efficiency can improve the process yield of select metabolite products. As the chemical dynamics in these films are complex a thorough understanding is required to guide the necessary metabolic and population engineering. By modelling the cellular systems using ChemChaste, we can determine the chemical profiles of a simulation and better inform experiments that probe the yield of metabolic products. These experimental developments can yield new insights into the dynamics of biofilms and the robustness of multicellular systems. Overall, a tandem development of experimental methods and agent-based simulations promises to advance the mechanistic understanding of microbial communities and bacterial biofilms.
6.1 Future work

The work presented here can be further developed through exploring the cell-environment feedback dynamics and implementing new features in the ChemChaste software. The results we have presented are illustrative of the complexity of the cell-environment system and further studies are required to quantify the dynamical effects at play. After quantifying the simple system used here as illustrative cases, the development of more biologically realistic models can be performed to investigate the dynamics of real world systems.

Future work in ChemChaste:

- To develop routines to conform to SBML and COMBINE standards to further develop the model distribution capabilities.

- To develop a graphical user interface to assist the implementation of models by a wider range of users

- To expand the underlying ChemChaste capabilities in order to consider more complex (than linear spring) force interactions. Both cell-cell interactions and cell-environment processes such as biased diffusion or chemotaxis (see section 3.3)

- More insightful model evaluation and parameter testing routines

One of the development aims behind ChemChaste was to produce a framework that was readily accessible and expandable to a broad range of users and therefore applicable to a range of cell-environment systems. Therefore the introduction of a graphical user interface for ChemChaste (and/or the wider Chaste package) would greatly increase accessibility to users of a lower coding experience. The design of this interface would however require careful consideration of the "Chaste as a simulation toolbox" doctrine to ensure the adaptable philosophy of Chaste is not compromised. To ensure this fundamental principle is met, the expansion would need to include the ability for users to implement their own cell models, cell-cell interaction forces, and reaction kinetics with the functionality readily integrated into a graphical interface. For such a system the simulation output writer may be used to implement more complex parameter sweep methods, such as so-called "expensive optimisation" algorithms using simulation outputs to compose fitness measures.

A long term aim for ChemChaste would be to apply ChemChaste simulations to experiment data. This may required further developments of experimental methodologies but would a vast expansion of investigative scope. The capability of
relating the simulation output to experimental data may suggest future experiment protocols, suggest new growth media conditions to support hard to grow cell communities, or prove a test-bed for developing new analytic technologies such as chemical probes and cell imaging techniques. These advances will strongly support both industrial applications in the biotechnology and healthcare spheres as well as advance our fundamental understanding of cell communities.
Appendix A

Mathematical modelling of a continuous chemical environment

Partial differential equations (PDEs) can be used to model the evolution of a distribution of chemicals both temporally and over a spatial domain. We associate a scalar field, \( u \in \mathbb{R} \), for the concentration of each chemical in an \( N \)-dimensional domain, \( \Omega \subset \mathbb{R}^N \) bounded by \( \partial \Omega \subset \Omega \) where a location in the domain is represented by the position \( x \in \Omega \) at a time \( t \in \mathbb{R} \).

A.1 Classification of Partial differential equations

The general partial differential operator for a 1-dimensional domain may be presented by:

\[
A(x, t) \frac{\partial^2}{\partial t^2} + B(x, t) \frac{\partial^2}{\partial t \partial x} + C(x, t) \frac{\partial^2}{\partial x^2} + D(x, t) \frac{\partial}{\partial t} + E(x, t) \frac{\partial}{\partial x} + F(x, t))u + G(x, t) = 0
\]  
(A.1)

where \( \{A(x, t), \ldots G(x, t)\} \) are coefficients for the differential terms for the PDE. An analogy between equation (A.1) and the general equation for a conic section

\[
Ax^2 + Bxy + Cy^2 + D(x + Ey + F = 0
\]  
(A.2)

may be drawn by considering the differential operator as a polynomial in terms of \( \partial x \) and \( \partial t \) and mapping the coefficients, \( \{A(x, t), B(x, t), C(x, t), \ldots\} \rightarrow \{A, B, C, \ldots\} \).

The general conic equation may be broken into classes; elliptic, parabolic, hyperbolic, depending on the geometric behaviours. This is performed by considering a matrix form for the polynomial and inspecting the determinant of the 2nd-order coefficient.
matrix $Z$, i.e equation (A.2) may be written as;

$$x^T \cdot Z \cdot x + W \cdot x + F = 0 \quad (A.3)$$

where vector $W = [D, E]$ contains the coefficients of the 1st-order polynomials and matrix

$$Z = \begin{bmatrix} A & B \\ B & C \end{bmatrix} \quad (A.4)$$

contains the 2nd-order coefficients. The determinant of the matrix $|Z| = AC - B^2$ is used to classify the geometries; elliptic for $AC > B^2$, parabolic for $AC = B^2$, and hyperbolic for $AC < B^2$, for all points that the class condition holds. Therefore for 2nd-order PDEs, by analogy, the general family of PDEs in N dimensions may be described as elliptic, parabolic, or hyperbolic depending on a subset of the coefficients, $\{A(x, t), B(x, t), C(x, t)\}$ (see table A.1). Each of the different PDE types has differing solving strategies to provide computationally efficient solutions. Furthermore, the possible dynamic phenomena and modelling applicability also varies across the families. Subsequently, a simulation suite implementing the chemical models may only consider a subset of the families and therefore will only be applied to a subset of potential models.

<table>
<thead>
<tr>
<th>Classification</th>
<th>Condition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elliptic</td>
<td>$A(x, t)C(x, t) &gt; B(x, t)^2$</td>
<td>Steady-state i.e $D\nabla^2 u - f(x) = 0$</td>
</tr>
<tr>
<td>Parabolic</td>
<td>$A(x, t)C(x, t) = B(x, t)^2$</td>
<td>Reaction i.e $\partial u/\partial t - D\nabla^2 u + f(x) = 0$</td>
</tr>
<tr>
<td>Hyperbolic</td>
<td>$A(x, t)C(x, t) &lt; B(x, t)^2$</td>
<td>Advection i.e $\partial^2 u/\partial t^2 - D\nabla^2 u + f(x) = 0$</td>
</tr>
</tbody>
</table>

Table A.1: PDE coefficients for 2nd-order PDEs are related to the type of PDE which require different solving procedures.

For 2nd-order PDEs the classification and solution behaviours are dominated by the higher order terms. However, not all PDE systems used to track the transport of chemical concentrations are 2nd-order or the lower order terms can be readily ignored. For low diffusion systems with an advective flow, the changes in the chemical distribution due to advection are significant.

For chemical systems we consider a set of $m$ chemicals, $\mathcal{C}$ of cardinality $|\mathcal{C}| = m$, distributed over the domain $\Omega$. To track the dynamic evolution of the chemical concentrations we construct a vector of coupled scalar fields. Consider the N-dimensional reaction-diffusion-advection equation for a set of scalar concentration
fields, where the field for chemical \( c \in \mathcal{C} \) is given by \( u_c \);

\[
\frac{\partial u_c}{\partial t} - \nabla \cdot D(x, t) \cdot \nabla u_c + \nabla \cdot (v(t)u_c) + R(x, t; k) = 0
\]  
\[ (A.5) \]

where the surrounding fluid advection velocity is given by \( v \in \mathbb{R}^N \) and \( R(x, t; k) \) denotes a source/sink reaction term. The generalised spatial differential operator;

\[
\nabla = \sum_{i=1}^{N} \frac{\partial}{\partial x_i}
\]

\[ (A.6) \]

where the general coordinates \( \{x_1, x_2, x_3\} \rightarrow \{x, y, z\} \) may be mapped to the standard Cartesian coordinate system.

Considering equation \((A.5)\) under a non-negligible advection flow, the 2nd-order diffusive term, \( C(x, t)\partial^2 u/\partial x^2 \), is comparable to the 1st-order advection term, \( E(x, t)\partial u/\partial x \), and the 1st-order effects need consideration.

For 1st-order PDE systems of the form:

\[
\frac{\partial u_c}{\partial t} + \nabla \cdot F_c(u_c, t) = 0
\]  
\[ (A.7) \]

where \( F_c(u_c, t) \) represents a spatial flux vector for each chemical concentration. A 1st-order PDE of the form of equation \((A.7)\) may be classified as hyperbolic if the linear superposition of Jacobian matrices for each spatial dimension;

\[
J = \sum_{i=1}^{N} \alpha_i \begin{bmatrix}
\frac{\partial F_{1i}}{\partial u_1} & \cdots & \frac{\partial F_{1i}}{\partial u_{|C|}} \\
\vdots & \ddots & \vdots \\
\frac{\partial F_{|C|i}}{\partial u_1} & \cdots & \frac{\partial F_{|C|i}}{\partial u_{|C|}}
\end{bmatrix}
\]

\[ (A.8) \]

for \( \alpha \in \mathbb{R}^N \) has only real eigenvalue solutions, \( \lambda \in \mathbb{R} \) defined such that \(|J - \lambda I| = 0\) for identity matrix \( I \), that produce eigenvectors \( P \) enabling \( J \) to be cast into a diagonal form \( D \) such that \( J = PDP^{-1} \).

For a single scalar concentration field in 2-dimensions, \( \{x, y\} \), we may write the flux vector as;

\[
F(u, t) = v(t)u - D(x, t) \cdot \nabla u + R(x, t; k)x
\]  
\[ (A.9) \]

producing the \( 1 \times 1 \) matrix \( J = \alpha \cdot v(t) \) dependent only on the fluid velocity \( v(t) \). Therefore, the general reaction-diffusion-advection system, equation \((A.5)\), is hyperbolic for negligible diffusion rates and parabolic for small fluid velocities.
For general solver routines, the equation is considered as a hyperbolic system.

**A.2 Boundary and initial conditions**

Chemical reactions are described by systems of ordinary differential equations (ODEs) which may be used to describe well-mixed chemical systems. Alongside the ODE system, solvers require the provision of the initial concentrations of the chemicals at a time \( t = 0 \). As PDEs describe the spatial evolution of the concentrations they require the additional information of the chemical behaviour as the boundary of the domain, \( \partial \Omega \). We consider these PDEs as initial value problems and provide both an initial starting concentration, at a time \( t = 0 \), over the whole spatial domain and the boundary conditions on the domain edge.

The boundary conditions (BCs) may be classified as the Dirichlet, Neumann, Periodic or the combined Robin type. A combination of these conditions may be implemented on segmented parts of the domain, \( \Gamma_{\partial \Omega} \subset \partial \Omega \), and separately for each chemical species.

Dirichlet BCs fix the value of the scalar field at the boundary and represent the physical condition of the concentration being "clamped" at the boundary;

\[
u(x) = f(x), \quad \forall x \in \partial \Omega \tag{A.10}\]

where \( f(x) \) describes the variation of the constant concentration along the boundary segment. Dirichlet BCs may be used to describe the conditions at a bulk source, such as the input into a system of oxygen from the lab atmosphere or "no-slip" condition in fluid systems.

Neumann BCs control the flux into, or out of, the system at the boundary describing a permeable non-isolated system.

\[
\frac{\partial u(x)}{\partial n} = f(x), \quad \forall x \in \partial \Omega \tag{A.11}
\]

where the unit normal out of the domain \( \Omega \) is given by \( n \) with flux function \( f(x) \). Neumann BCs are used to describe the flux across a boundary for the system, such as the edge of a cell membrane. Additionally, zero Neumann conditions, where \( f(x) = 0 \), may be used to describe an isolated system with no flux of concentration into or out-of the domain.

Robin BCs are a mixed condition type, combining both Dirichlet and
Neumann components.

\[ \alpha u(x) + \beta \frac{\partial u(x)}{\partial n} = f(x), \quad \forall x \in \partial \Omega \quad (A.12) \]

where \( \alpha, \beta \in \mathbb{R} \) are weighting coefficients used to balance the Dirichlet and Neumann contributions, described in equations (A.10) and (A.11) respectively.

Periodic BCs map the concentration at one side of the boundary to another. For two segments of the domain boundary, \( \Gamma^a_{\partial \Omega} \subset \partial \Omega \) and \( \Gamma^b_{\partial \Omega} \subset \partial \Omega \), for \( x \in \Gamma^a_{\partial \Omega} \) and \( x' \in \Gamma^b_{\partial \Omega} \);

\[ u(x) = u(x') \quad (A.13) \]

such that the concentrations at the two locations are equal. Periodic boundary conditions are used to model continuous domain where no physical boundary exists.

### A.3 Courant–Friedrichs–Lewy condition

Numerical PDE methods, such as the commonly used finite-difference and finite-element methods, discretise the spatial domain by applying a mesh grid. The Courant-Friedrichs-Lewy (CFL) condition introduces a description for the relation between diffusion rate and the simulation numerical step-sizes for both the spatial and temporal components (Courant et al., 1967).

For a reaction-diffusion equation defined in an N-dimensional space with a step-size in the \( i \)-th spatial dimension being given by \( \Delta x_i \) with a dimension dependent diffusion rate \( D_i \), the CFL condition may be given by;

\[ C_{max} \geq \sum_{i=1}^{N} \frac{D_i \Delta t}{\Delta x_i} \quad (A.14) \]

where the Courant number, \( C_{max} \geq 1 \), is the maximum value for the ratio.

When the grid is regular the discretisation process may be described by a constant spatial step-size, \( \Delta x \). As the numerical solvers utilised to solve the spatial components for a PDE simulation are often more complex than the temporal solvers, the CFL condition is used to determine the maximum temporal step-size, \( \Delta t_{max} \), for the simulation;

\[ \Delta t_{max} = \frac{\Delta x}{D} \quad (A.15) \]

assuming the diffusion rate, \( D \), is independent of the dimension. As the CFL condition is a necessary condition for PDE stability, for simulations of the reaction-diffusion system a time-step below \( \Delta t_{max} \) is required as a necessary condition for stability.
Appendix B

Derivation of spatial patterning conditions for the Sel’kov-Schnakenberg reaction system

B.1 General spatial patterning

To investigate the onset of spatial patterning we consider a fixed point in the reaction term and perform a linear perturbation to analyse stabilities (Krause et al., 2021). Consider a fixed point $U_0$ with a small perturbation $\epsilon u$ where $|\epsilon| << 1$. As $dU_0/dt = 0$ we obtain from equation (2.9) the evolution of the perturbation;

$$\frac{\partial u}{\partial t} = D \cdot \nabla^2 u + R(u)$$  \hspace{1cm} (B.1)

The stability of the perturbations may be considered by looking at the eigenvalues of the Jacobian matrix. For $i \in |\mathcal{C}|$,

$$J_{R(u)} = \begin{bmatrix} \frac{\partial R_1}{\partial u_1} & \cdots & \frac{\partial R_1}{\partial u_{|\mathcal{C}|}} \\ \vdots & \ddots & \vdots \\ \frac{\partial R_{|\mathcal{C}|}}{\partial u_1} & \cdots & \frac{\partial R_{|\mathcal{C}|}}{\partial u_{|\mathcal{C}|}} \end{bmatrix}$$  \hspace{1cm} (B.2)

The dynamics of the fixed point, and therein the temporal stability of the system, may be considered by calculating the find eigenvalues, $\lambda_k$ where $k \leq |\mathcal{C}|$, of the
reaction Jacobian (equation (B.2)) such that

$$|\lambda_k 1 - J_{R(u)}| = 0.$$  \hfill (B.3)

For concentration perturbations about the equilibrium position to be stable and decay back to the equilibrium they each eigenvalue must satisfy;

$$\mathcal{R}e\lambda_k < 0.$$  \hfill (B.4)

Bacterial biofilms are immersed in a surrounding aqueous environment. Therefore when considering spatial perturbations in the chemical concentrations boundary effects of the flow of chemicals into the system should have a negligible affect. Spatial patterning is affected by boundary conditions due to the admissible wavelengths of waves being domain dependent. Here, we consider an infinite domain which posits a continuous spectrum of wave modes independent on the boundary conditions. The spatial perturbations may be constructed through wave modes by assuming a wave-like ansatz where the waves decay with the stability of the temporal perturbations (Murray, 2003).

We use the ansat;

$$u = A \exp(\lambda_k t + ik \cdot x)$$  \hfill (B.5)

for temporal eigenvalue, $\lambda_k$, wavenumber $k$ and wave vector $k$.

The stability of the wave modes may be determined through,

$$|\lambda_k 1 + \rho_k D - J_{R(u)}| = 0,$$  \hfill (B.6)

with the perturbation being unstable to spatial affect, and therein pattern forming, if with this introduction of spatial affects and for each spatial eigenvalue $\rho_k$ presenting a dispersion relation for $\lambda_k$ in terms of $\rho_k$. From analysing system specific dispersal relations, conditions for stability may be found when;

$$\mathcal{R}e\lambda_k > 0$$  \hfill (B.7)

that is the real component of the eigenvalues are positive. This analysis may be considered for any number of chemical scalar field in a PDE system (Kuznetsov and Polezhaev, 2020).
B.2 The Sel’kov-Schnakenberg system

In the absence of diffusion \((D_U = D_V = 0)\), equations (2.14)–(2.15) have the unique steady-state solution

\[
(U_0, V_0) = \left( \frac{k_1 + k_2}{k_1 - k_2}, \frac{k_2 k_1^2}{k_3 (k_1 + k_2)^2} \right),
\]  

(B.8)

By requiring both eigenvalues of the Jacobian to have negative real part, we find that (B.8) is linearly stable in the absence of diffusion if

\[
\frac{\partial R_U}{\partial U} + \frac{\partial R_V}{\partial V} < 0
\]  

(B.9)

and

\[
\frac{\partial R_U}{\partial U} \frac{\partial R_V}{\partial V} - \frac{\partial R_U}{\partial V} \frac{\partial R_V}{\partial U} > 0,
\]  

(B.10)

where the partial derivatives of \(R_U\) and \(R_V\) are evaluated at (B.8). By requiring at least one eigenvalue to have negative real part, we find that (B.8) becomes linearly unstable in the presence of diffusion if

\[
D_V \frac{\partial R_U}{\partial U} + D_U \frac{\partial R_V}{\partial V} > 0
\]  

(B.11)

and

\[
\left( D_u \frac{\partial R_V}{\partial V} + D_V \frac{\partial R_U}{\partial U} \right)^2 - 4 D_U D_V \left( \frac{\partial R_U}{\partial U} \frac{\partial R_V}{\partial V} - \frac{\partial R_U}{\partial V} \frac{\partial R_V}{\partial U} \right) > 0.
\]  

(B.12)

We consider the Sel’kov-Schnakenberg system on a domain \(\Omega\) with boundary \(\partial \Omega\) with the unit normal pointing out of the domain denoted \(\mathbf{n}\), and consider zero-flux BCs for both state variables.

We linearise for small perturbations about the homogeneous steady state, \((U_0, V_0)\), and introduce the small perturbation vector

\[
\mathbf{W} = (U - U_0, V - V_0)
\]  

(B.13)

and map the homogeneous form for the concentrations to a stability matrix, \(S\), for the perturbations

\[
S = \begin{bmatrix}
\frac{d}{d\tau} R_U & \frac{d}{d\tau} R_V \\
\frac{d}{d\tau} R_V & \frac{d}{d\tau} R_U
\end{bmatrix}
\]  

(B.14)
providing the time-dependent evolution for perturbations about the steady state

\[
\frac{d\mathbf{w}}{dt} = S\mathbf{w}
\]  

(B.15)

where the stability matrix \( S \) is evaluated at the steady state, \((U_0, V_0)\). We look for solutions of the form

\[
\mathbf{w} = c e^{\lambda t}
\]

(B.16)

where \( \lambda \) is the eigenvalue of the stability matrix, \( S \), and \( c \) is a vector of reaction rate constants. To determine the solution concentration vector we need to calculate the eigenvalues, \( \lambda \), through the appropriate characteristic equation;

\[
|S - \lambda I| = 0
\]

(B.17)

where \( I \) is the two-dimensional identity matrix and \( |.| \) denotes the determinant.

The stability of the solutions can be classified based on the eigenvalues. If \( Re(\lambda) < 0 \) then the system is stable to small concentration perturbations. This occurs for parameter sets that satisfy the inequalities

\[
tr(S) = \frac{d}{dU} R_U + \frac{d}{dV} R_V < 0, \quad \text{(B.18)}
\]

\[
|S| = \frac{d}{dU} R_U \frac{d}{dV} R_V - \frac{d}{dU} R_V \frac{d}{dV} R_U > 0, \quad \text{(B.19)}
\]

where \( tr(S) \) denotes the trace of the matrix \( S \).

When stability for the homogeneous case is confirmed we consider the heterogenous case, equations (2.14)–(2.15). Considering the small perturbations about the same steady state, \((U_0, V_0)\), but using a different solution form to include the spatial effects

\[
\mathbf{w} = c e^{\lambda t + i \vec{k} \vec{x}}
\]

(B.20)

where \( \vec{x} \) is the position vector in the domain \( \Omega \) and \( \vec{k} \) is a wave-velocity parameter.

We solve

\[
\frac{d\mathbf{w}}{dt} = S\mathbf{w} + D \nabla^2 \mathbf{w}
\]

(B.21)

where \( D \) is the matrix of diffusion coefficients, we assume here that \( D \) is the diagonal
as is the case for isotropic diffusion, given by
\[ D = \begin{bmatrix} D_U & 0 \\ 0 & D_V \end{bmatrix} \]  
(B.22)

with \( D_U \) and \( D_V \) being the bulk domain diffusion coefficients for \( U \) and \( V \) respectively as shown in equations (2.14)–(2.15). As in the homogeneous case, the modified characteristic equation
\[ |S - \lambda I - D\tilde{k}^2| = 0 \]  
(B.23)

from which the eigenvalues, \( \lambda(k^2) \), can be determined as a function of the squared wavenumber, \( k^2 = |\tilde{k}|^2 \). In particular, the \( \lambda \) eigenvalues can be determined through solving the polynomial produced by the particular characteristic matrix,
\[ \lambda^2 + \lambda(tr(D)k^2 - tr(S)) + h(k^2) = 0 \]  
(B.24)

where the term independent of \( \lambda \) is given by \( h \)
\[ h(k^2) = |D|k^4 - Ck^2 + |S|, \]  
(B.25)

which includes a term which is formed by the coupling the matrices \( S \) and \( D \),
\[ C = D_U \frac{d}{dV} R_V + D_V \frac{d}{dU} R_U \]  
(B.26)

For DDI spatial patterning, we require the solutions, \( \lambda(k) \), of equation (B.24) to satisfy \( Re(\lambda(k^2)) > 0 \) for some \( k \neq 0 \) and \( Re(\lambda(k)) < 0 \) for \( k = 0 \). This occurs when,
\[ \frac{D_V}{D_U}(2k_2 - 1)k_3^3 > k_3(k_1 + k_2)^2 > 0. \]  
(B.27)

for the reaction rate constants \( k_1, k_1, k_2, k_3 \) and diffusion coefficients \( D_U, D_V \).
Appendix C

Agent based modelling: On-lattice and Off-lattice models

During the development of a model for cell systems, one approach is to model the cells as discrete individual agents within a continuous domain environment. When using agent based models the cell models are singular and dynamic; where each unique cell can grow and divide according to personal behaviours that may depend on a set of dynamic rules. A wide array of agent based models (ABMs) have been used over a range of fields from describing social interactions and healthcare policy (Jager, 2021; Tracy et al., 2018) to physical systems (Klein et al., 2019) and have previously been used to describe heterogeneous biofilms (Koshy-Chenthittayil et al., 2021). The wide range of fields has lead to the development of a mosaic of software packages that implement these models and subsequently overlaps in the software capabilities may be seen. A selection of these software frameworks is discussed in chapter 3 with a focus on modelling cells in a chemical environment. In this section we describe two modalities of agent based descriptions commonly used in biological modelling; on-lattice methods and off-lattice centre-based modelling, summarised in figure C.1.
Figure C.1: Agent based models come in two forms, on-lattice and off-lattice, which defined how the cell agent is described. a) The on-lattice cellular automata model represents each lattice point as a single cell with neighbouring lattices forming the cell’s neighbourhood. b) The on-lattice Cellular Potts model considers a connected set of neighbouring cells to be a single cell agent with cell interactions represented by neighbouring cells (black and red, not grey). c) The off-lattice overlapping sphere model represents cells as particles (black) with cell-cell interactions occurring when two cells have overlapping regions of influence (grey).

C.1 On-lattice methods: Cellular Automata and the Cellular Potts Model

On-lattice agent models consider a lattice formed by a $N$-dimensional regular grid whose lattice points, $\mathbf{x} \in \mathbb{Z}^N$, each represent a location in the space. A cell agent is represented by at least one lattice point within this grid and the presence of a cell is described by the state of that point. We consider a finite set of states $\mathcal{S}$ to describe the agent dynamics of the on-lattice simulation; for example a 2 state model, $|\mathcal{S}| = 2$, may describe an empty lattice point $S(\mathbf{x}, t) = 0$ and $S(\mathbf{x}, t) = 1$ for a lattice point containing a cell. Therefore the mode is described by the state set $S = \{0, 1\}$. Each lattice point also has a well-defined regular neighbourhood, $\mathcal{N}$, which denotes the connected lattice points. Therefore on-lattice simulations may be reduced to tracking the evolution of the state for each lattice point where rules are provided during model definition to govern the interactions of the lattice point agents.
C.1.1 Cellular Automata

A simple form of an on-lattice ABM developed to study biological systems was introduced by von Neumann to model self-replicating entities (Sotomayor et al., 2020). The cellular automata (CA) model was developed as a model for artificial life but has found further applications in the social sciences (Epstein and Axtell, 1996) with the introduction of the NetLogo (Wilensky, 1999) software platform and the SugarScape model (Epstein and Axtell, 1996). Cellular Automata were developed as a primitive form of agent-based models and found widespread adoption due to their simple implementation and capability to generate complex spatial patterns of different agents with a wide array of possible agent dynamic phenomena (Wolfram, 2002).

As an on-lattice ABM, each agent in the CA is associated with a point in the lattice and is represented by a lattice point state. In the CA model each lattice point may only have a single state representing a distinct cell, therefore points may indicate the absence and presence of a cell \( S = \{0, 1\} \). The cell-cell interactions are restricted to interactions between a cell with their connected neighbours within a set neighbourhood. Two commonly implemented neighbourhoods for a regular grid lattice are the Moore neighbourhood, composed of the 9 cells within a square centred on the principal cell, and the von Neumann neighbourhood, composed of the 5 in a diamond configuration.

In CA models, the agent behaviours are determined by a set of conditions determined for each of the agent states. These so-called "rule-based models" may also be built upon an arbitrary rule set whose rules limit or promote the complexity of agent decisions (Liu et al., 2019). While decisions such as "consider the state of a neighbour lattice point" may be applicable for interaction models in the social sciences they struggle to capture more than simple physical interactions in biological models. CA models cannot effectively capture mechanistic aspects of cell-cell interactions with no adhesion processes or interactions outside of the set of simple rules (Wolfram, 1985). This limits the usefulness of CA models as a tool for understanding the chemical processes of biological systems and mapping to experimental results. If restrictive rule-sets are used to formulate the CA model, rather than rules derived from biological principles, then applying the simulation results to assist with the analysis of experiment data will not be fully synergistic.

An exemplar case of the cellular automata model is the 2-dimensional Conway’s "Game of Life" (Gardner, 1970). Conway’s game consists of two agent states representing the live, \( S(x, t) = 1 \), and dead, \( S(x, t) = 0 \), cells. During the simulation, each lattice point is visited by the algorithm and the value of the
neighbourhood is calculated. For lattice point \( \mathbf{x}_{i,j} \) the state is evaluated by a set of rules:

1. If \( S(\mathbf{x}, t) = 1 \) and \( T(\mathbf{x}, t) < 2 \) then \( S(\mathbf{x}, t + \delta t) = 0 \)
2. If \( S(\mathbf{x}, t) = 1 \) and \( T(\mathbf{x}, t) \in [2, 3] \) then \( S(\mathbf{x}, t + \delta t) = 1 \)
3. If \( S(\mathbf{x}, t) = 1 \) and \( T(\mathbf{x}, t) > 3 \) then \( S(\mathbf{x}, t + \delta t) = 0 \)
4. If \( S(\mathbf{x}, t) = 0 \) and \( T(\mathbf{x}, t) = 3 \) then \( S(\mathbf{x}, t + \delta t) = 1 \)

where rules 1 and 3 lead to a decrease in the total population through environment attrition or starvation through resource competition and rule 4 increases the population through reproduction. The rules capture the biological reasoning:

1. Any live cell with fewer than two live neighbours dies, as if caused by underpopulation
2. Any live cell with two or three live neighbours lives on to the next generation
3. Any live cell with more than three live neighbours dies, as if by overpopulation
4. Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction

The simple set of rules is capable of producing order from random initial cell distribution with cell motility and oscillations in cluster size being emergent phenomena.

To simulate the coupling of CA to an environment the lattice may be placed over a reaction-diffusion PDE. By solving the PDE using a grid based method (such as the finite difference method) the two grids can be overlapped and this overlap can utilise the same spatial grid for both algorithms (Varahan et al., 2019). This provides computational efficiency in sharing a single grid. The numerical method can be used to solve the reaction-diffusion PDE and the presence of a CA state can influence the terms in the PDE.

### C.1.2 Cellular Potts

The Cellular Potts model (CPM) is an on-lattice agent based model used to simulate extended cell agents and complex cell-cell adhesion processes. The CPM was developed by Glazier and Graner (Glazier and Graner, 1993) to model cell sorting, from initial multi-strain clusters of cells (i.e random initial state configuration) to mono-strain aggregates (i.e formation of mono-domains) through
differential-adhesion. The agent dynamics are built upon a discrete-time Markov chain tracking the stochastic evolution of a configuration state space. Unlike the cellular automata, CPM is less mathematically understood making a formal analysis treatment difficult due to all of the simulation dynamics being modelled by different stochastic jump-type transition processes.

The association between lattice points and states is similar to the CA model. For each lattice point \( x \) in the grid covering the simulation domain we associate an agent state, \( S(x, t) \) that characterises the point’s behaviour. However unlike the CA model, in the CPM this lattice point is only an element of a single cell (rather than a whole cell in the CA model) and the agent state is shared with a set of connected lattice points. As the connected set of lattice points constitute a single cell agent we can provide the cell with a well defined volume and shape. The cell shape, size, and surface area drive the mechanical interactions in the CPM which capture more biologically realistic cell-cell interactions than the simple state switches in a CA simulation.

In a CPM simulation each distinct cell (connected set of lattice points) is provided with a representative energy term, the Hamiltonian, which is calculated from the properties of the constituent lattice point states. The Hamiltonian captures the "energy" of a physical process and provides a lower level description of the cell mechanical dynamics than the rules based system used in the CA model. The Hamiltonian is used to test possible lattice point state switch where a state associate the lattice point with a neighbouring cell is attempted. The bespoke simulation models are then implemented by appending to the Hamiltonian different dynamical energies formulated by considering the physical processes. The energies are additive and each capture a different aspect of cell behaviour. This ensures CPM is readily adaptable to qualitatively simulate experimental systems and cell population dynamics.

For each "timestep" of the simulation, a lattice point is selected and the state of the point is switched. The change in the effective energy of the system, \( \Delta H \), is then calculated with the aim of reducing the overall system energy or probabilistically accepting an increase in energy to model membrane fluctuations. The CPM algorithm follows a Metropolis-Hastings like algorithm to accept or reject the state switch depending on the change in energy (Hastings, 1970). The form of this energy term changes depending on the particular model implementation and in practice phenomenologically derived to fit a desired set of agent behaviours or experimental outcome.

The CPM algorithm randomly selects a set of cells per simulation time step
and checks for state switch events;

1. Select random lattice point $x_i$
2. Select random lattice point $x_j$ in neighbourhood of $x_i$ to copy
3. Calculate $\Delta H = H_j - H_i$
4. If $\Delta H < 0$ set $S(x_j, t) = S(x_i, t)$ set the lattice point $x_i$ to the copied state
5. If $\Delta H \geq 0$ set $S(x_j, t) = S(x_i, t)$ with probability $\exp\left(-\frac{\Delta H}{\Delta t}\right)$

for lattice point $x_i$ belonging to cell state $S$. Each cell in the CPM is described by a cell volume $v_i$ and target volume $V_i^T$, and cell surface area $a_i$ and target surface area $A_i^T$. The energy of changes in cell volume and surface area are calculated from the divergence from target values that motivates the cells to conform to a particular form. This conformation is derived from thermodynamics arguments for the surface energy assuming a particular form. The Hamiltonian, $H$, is calculated for the simulation configuration before and after the cell flip. The difference between the two configurations is used to calculate the change in energy, $\Delta H$, due to the cell flip. $H$ is composed of the summation of additive energy contributions formed from function of the cell property values;

$$H = \sum_S \lambda_{v_i}(v_i - V_i^T)^2 + \sum_S \lambda_{a_i}(a_i - A_i^T)^2 + H_{Contact} + H_{Chemotaxis} + H_{Motility} + \ldots \tag{C.1}$$

where $\lambda_{v_i}$ is the effective energy contribution for a deviation in volume and $\lambda_{a_i}$ is the effective energy contribution for a change in cell surface area. Cell-cell contact energies are calculated for the lattice points in the neighbourhood of the state switched lattice point,

$$H_{Contact} = \sum_{(i,j) \in \text{Neighbours}_i} J(S(x_i, t)(1 - \delta_{ij}), S(x_j, t)), \tag{C.2}$$

where $J(a, b)$ is the surface contact energy of states $a$ and $b$ which may be written in the form of a symmetric matrix,

$$J[a, b] = \begin{bmatrix} J_{a,a} & J_{a,b} \\ J_{b,a} & J_{b,b} \end{bmatrix} = \begin{bmatrix} 0 & c \\ c & 0 \end{bmatrix}, \tag{C.3}$$

where contact energies are symmetric (i.e $J_{a,b} = J_{b,a} = c$ for an arbitrary contact energy $c \in \mathcal{R}$) and the lattice-lattice contact energies for two neighbouring lattice
points belonging to the same cell are zero (i.e. $J_{a,a} = J_{b,b} = 0$). Chemotaxis may be modelled by considering the chemical concentrations at the location of the lattice point to be switched $x_i$ and from the lattice point to be copied $x_j$,

$$H_{\text{Chemo}} = \lambda_{\text{Chemo}}(c(x_i) - c(x_j)),$$

where $\lambda_{\text{Chemo}}$ is the effective energy change due to a chemotactic response and $x_j$ and $x_i$ are the proposed initial and final location of the lattice point switch where $c(x)$ is the chemical concentration at location $x$ that acts as the chemo-attractant/repellent. Directed motility may be implemented within a CPM model by adding an additional work term to the Hamiltonian,

$$H_{\text{Motility}} = \sum_{i \in \mathcal{C}} F_i \cdot x_i,$$

where $F_i$ is a motile force acting on the cell’s centre of mass $x_i$ where the direction of this force will be the migration direction for the cell. These example contributions may be supplement by further contributions in a bespoke CPM simulation.

A key benefit to the CPM over CA models is that cells have a well defined volume and simulations are able to calculate local mechanical pressures forming within a system. The surface contact energies that contribute to the effective energy are motivated by physical principles but are a high level feature and fictitious in nature. The energies represent the contact processes between two dissimilar states types and are not based off of particular membrane or other physical properties. This produces a level of uncertainty in the CPM simulation as there is no rigorous method for finding appropriate values for the energies with users resorting to qualitative justifications. Therefore a range of energies must be trialled motivated by the energy ratios between cell types to ensure the observed simulation output is not due to the free choice of energies. One consequence of this uncertainty may be seen through tissue fragmentation when cell contact energies are low when compared to simulation fluctuations. Attempts have been made to extend the applicability of CPM by using the model to simulate dynamical systems such as the Advection-Diffusion equations (Dan et al., 2005) and CPMs have also be applied to the physical flow problems of bubble foams (Sanyal and Glazier, 2006).
C.2 Off-lattice methods: Overlapping Sphere Model

Off-lattice ABMs describe agents as discrete entities representing particles that may interact through mechanical force laws (Van Liedekerke et al., 2021). Compared to on-lattice models, off-lattice simulation are free from lattice artefacts as agents are placed and move within a continuous space rather than along the lattice lines of the CA and CPM models. While the lattice grid of the CA and CPM simulation provide a natural basis for using finite-difference based PDE solvers, the lack of a lattice definition provides the opportunity to utilise a range of PDE solvers and more sophisticated agent dynamics are possible.

Cells may be represented by shapeless point-like particles, as shaped extended particles, or enclosed regions of the domain space. The array of cell shapes produce a more complex definition of connectivity between cells and cell neighbourhoods without having the grid neighbours to work from. Therefore off-lattice can suffer from more computationally expensive cell definitions but with the freedom to develop more biochemically relevant cell definitions.

A family of off-lattice models that are proficient at modelling biological systems are the centre-based models. Here, cells are represented by the point location of the cell centre of mass and cell movement tracks the location of this point rather than by general rules governing state switches. Consider an $N$-dimensional domain $\Omega \subset \mathbb{R}^N$ and let the centre of cell $i$ be denoted $x_i \in \Omega$. Cells are commonly considered to be contained within a fluid environment, therefore we formulate equations of motion for each cell and include a viscosity term to represent the environment;

\[ m_i \frac{d^2 x_i}{dt^2} = \sum_{j \in N_i} F_{ij} - \eta \frac{dx_i}{dt}, \]  

(C.6)

which are solved numerically according to the discretisation,

\[ x_i(t + \Delta t) = x_i(t) + \Delta t v_i(t), \]  

(C.7)

\[ v_i(t + \Delta t) = v_i(t) + \frac{\Delta t}{m_i} \sum_{j \in N_i(t)} F_{ij}(r_i(t), t), \]  

(C.8)

where the force on cell $i$ of mass $m_i$ is due to resistance to cell motion by the viscous fluid of viscosity, $\eta \in \mathbb{R}$, and the interaction with neighbouring cells $F_{ij} \in \mathbb{R}^N$. A range of pairwise interactions forces may be implemented, from linear Hookean springs to Polynomial forms (Mathias et al., 2020), where the forces act on the centre of the cell. The strength of cell-cell interactions are chosen to ensure artefacts are not generated at high cell density as the pair-wise interactions may lead to numerical
instabilities in the equations of motion.

The forces acting upon the cell agent are composed of the forces due to the environment and the forces due to the agent’s neighbours. The environment forces capture the drag effect acting to resist the motion of a moving cell due to the fluid viscosity. For low Reynolds number fluids, the viscosity opposes the agent’s acceleration by damping the inertial acceleration. Therefore under this fluid damping assumption the inertial term is negligible in comparison to the viscous term, \( m_i \frac{d^2 \mathbf{x}_i}{dt^2} \ll \eta \frac{d\mathbf{x}_i}{dt} \), and the 2nd-order equations of motion (see equations C.8) may be simplified to a first order ODE.

\[
\frac{dx_i}{dt} = \frac{1}{\eta} F_{\text{Drag}}^i(x_i(t), t), \tag{C.9}
\]

which is solved numerically according to the discretisation,

\[
x_i(t + \Delta t) = x_i(t) + \frac{\Delta t}{\eta} F_{\text{Drag}}^i(x_i(t), t), \tag{C.10}
\]

where \( \eta \in \mathbb{R}_{\geq 0} \) is a damping constant associated with the fluid viscosity of the environment (Osborne et al., 2017). The Forward Euler method was used to numerically solve the derivatives and the equations of motion for the centre of the agents may be written as;

\[
\eta \frac{dx_i}{dt} = \sum_{j \in \mathcal{N}(t)} F_{ij}(x_i(t), t) + F_{\text{Drag}}^i(x_i(t), t), \tag{C.11}
\]

with the discretisation,

\[
x_i(t + \Delta t) = x_i(t) + \frac{\Delta t}{\eta} \sum_{j \in \mathcal{N}_i(t)} F_{ij}(x_i(t), t) + \frac{\Delta t}{\eta} F_{\text{Drag}}^i(x_i(t), t) \tag{C.12}
\]

assuming that the timestep \( \Delta t \) is sufficiently small to ensure stable numerics of the Forward Euler method.

For motile cells, the neighbourhood of an agent varies with time, we let the neighbourhood of cell \( i \) be denoted \( \mathcal{N}_i(t) \). The neighbourhood around a cell is dependent on the specific off-lattice model implementation and is used when calculating the cell-cell interactions. By solving the equations of motion for a particular time step the neighbourhood and position of the cell can be determined, and the agent interactions for the next time step can be calculated. The neighbourhood can also be changed through the introduction of a new cell agent due to cell division.
On-lattice methods provide a natural implementation of cell division. In the CA model the neighbouring lattice point switches state while in the CPM a connected area may split into two. For off-lattice models there is not a convenient method of implementing a division event except through the addition of a new agent to the simulation. One complication with adding a new agent is with placing the new agent at a natural location such that the forces between the agents are well defined. This problem is captured in the relaxation time post cell-division which places a limit on the time steps used in a centre based model (Mathias et al., 2020). The aim is to preserve smooth trajectories through the space after division.

Off-lattice models allow for cell position and size to vary dynamically and continuously (Cooper et al., 2020). As each cell is considered as a separate entity rather than a lattice position, complex cell properties may be implemented which control the cell behaviours such as pairwise interactions, growth, and division. Furthermore, these properties may be develop from biochemical principles and parameter ranges may be inferred from experiment due to the physical nature of the equations of motion. The physical nature also assists in model parameter inference as the search space may constrained to physically relevant values and experimentally viable. Different descriptions for centre-based models are possible and one common choice for centre-based simulations is the overlapping sphere model.

C.2.1 Overlapping Sphere model

The overlapping sphere (OS) model is an off-lattice ABM that represents the agents as point-like particles with the cell neighbourhood defined by a radius of influence. The overlap of the influence regions is used to determine connectivity between the cells and the cell-cell interactions are described by force laws. For two cells, \( i \) and \( j \), the cells may be considered as interacting if their influence regions overlap, that is the centres are within a range:

\[
|| \mathbf{r}_i - \mathbf{r}_j || < I_i + I_j
\]  

(C.13)

where \( || \cdot || \) is the \( L^2 \)-norm and \( I_i \) is the radius of influence for agent \( i \). This interaction condition is used to determine whether the agents impart a pairwise force upon one another and represents a cut-off distance within which the agents are considered to be neighbours;

\[
\mathcal{N}_i(t) = \{ j \in \mathcal{A} : || \mathbf{r}_i - \mathbf{r}_j || < I_i + I_j \}
\]  

(C.14)

for the set of cell agents \( \mathcal{A} \).

Cell birth and death are implemented in OS models through the addition and
removal of agents. When the cell birth/division process is triggered by a parent cell, a new cell is placed at a location around the parent. The introduction is accommodated for with the force laws acting to push the cells to equilibrium distances. However when new cells are added more agents are introduced, the resulting force interactions between the cells can induce instabilities into the mechanical interactions under high compression (Pathmanathan et al., 2009). Cell death is implemented by the removal of an agent from the system with a change in mechanical interactions for the remaining cells leading to new relaxation distances.
Appendix D

Agent based software platforms

In the following sections each software package is considered in turn to investigate the cell model, ability to model environment-cell coupling, the possible cell mechanics available, and how a user’s model may be implemented and explored. This exploration is performed with an aim for understanding the applicability, usability, and portability of the software selection.

D.1 PhysiCell

PhysiCell is an open source 3-dimensional centre-based overlapping sphere modelling framework coupled to reaction-diffusion PDE systems (Ghaffarizadeh et al., 2018). The cell based simulation builds upon a previous software, the BioFVM reaction-diffusion solver (Ghaffarizadeh et al., 2015), to incorporate cell agents with mechanical properties. In PhysiCell, cells may excrete and uptake chemicals and perform an internal chemical reaction network with chemical excretions used to simulate the signalling between a population of cells. Additionally, these chemicals may decay in the surrounding environment and are modelled by a continuous concentration field.

Cell motility may be implemented as a cell property which may vary through chemotaxis and random cell diffusive motion. The PhysiCell package is designed to provide an adaptable simulation toolkit with a focus on oncology simulations and developed with tumour models and tissue dynamics in mind. Through the addition of new community made modules the simulation complexity can be increased and used to consider a range of biological systems. In particular, this software has been applied to understand a wide range of biological phenomena; from the fate of hypoxic tumour cells (Rocha et al., 2021), to the progression of liver metastases (Wang et al.,
PhysiCell has been written in C++ and the source code is available via the software’s GitHub page https://github.com/MathCancer/PhysiCell (accessed 14/04/2022) under the BSD licence. The package presents a coding heavy interface to the ABM model requiring some coding familiarity by a user. A range of default demo models are provided and a user’s models are implemented as extension to the base code motivated by a selection of template models. The user writes their particular agent rules as C++ functions within a custom header file and these functions are integrated with the workflow through the use of C++ function pointers. Further functionality may be provided by editing the trunk code functions within the main.cpp and the peripheral files that communicate with the BioFVM solvers. This approach sees a user often editing the underlying trunk code and requires understanding a low level computer language (C++) in order to implement their model. Such a code heavy system may inhibit the software’s accessibility for users and risk the accidental introduction of errors into the trunk code. Furthermore as the software is distributed as source code files, the need to compile the C++ source code relies on the user having an appropriate C++ compiler installed on their computer. These compilers are however not standard and may differ on different computer architectures, however, attempts have been made to enable cross compatibility over a range of operating systems. Parallelisation is supported during the reaction-diffusion solvers of BioFVM as they utilise the open source OpenMP, a multi-platform parallelisation package.

### D.1.1 Cell model

PhysiCell implements the off-lattice overlapping sphere model with each cell having a radius and volume determined by the cell contents (see section 3.3). Each cell is modelled as a compartmentalised sphere of fluid with a solid nucleus and bathed within a low viscosity fluid environment. The cell volume, $V$, may be decomposed into cellular constituent phases,

$$
V = V_F + V_S + V_{NS} + V_N + V_{CS} + V_C,
$$

where $V_F$ is the volume of fluid contained within the cell, $V_S$ is the volume of the solid material within the cell excluding the cytoplasm volume $V_{CS}$ or the nuclear solid volume $V_{NS}$; finally $V_N$ is the total volume of the nucleus and $V_C$ is the total volume of the cytoplasm. These individual volumes evolve during the cell cycle model phase of the simulation according to a set of ODEs that depend on the difference
between the volume and target volume for each phase. The cell volumes are not static and PhysiCell allows for gradual changes in these volumes through ODEs at small enough time steps so as to not interfere with the cell mechanics.

Changes in the cell volume, division and death are linked to the cell cycle model. In PhysiCell the cell cycle is used to set the states, $S$, for each cell and these states may be used to change active parameters in the cell properties. These properties include the volume growth ODEs, velocity terms in motility laws, or cell-cell interaction potentials. These states include a necrotic state and an apoptotic states which trigger target volume terms to change with cell removal after a duration or volume threshold are met. A user may implement further cell states and couple these to a custom cell cycle model to provide bespoke simulations.

A range of PhysiCell extensions provide different methods for intracellular modelling of chemical reactions. Boolean networks may be implemented using PhysiBoSS, ODE modelling is included through the libroadrunner package, and flux balance analysis is provided in the PhysiFBA extension. Flux balance analysis (FBA) is a constraint based approach to approximate the dynamics of reaction networks and predict the production rate of a specific molecule through considering bottlenecks in network dynamics (Orth et al., 2010). While FBA is widely used in whole cell models, the deeper mechanistic understanding behind the reaction dynamics provided by kinetic ODE models is lacking. Kinetic models may be added into a PhysiCell ABM simulation by defining the reaction systems in COPASI (Hoops et al., 2006) (see section 4.4) and importing the models into PhysiCell using the SBML model standard (Hucka et al., 2003) (see section 4.4). However, in the absence of coupled cells, the environment dynamics are limited to chemical decay and complex environment dynamics are only possible due to the excretion and uptake at the cell locations. Therefore simulating a chemical reaction systems, such as the Sel’kov-Schnakenberg system, is not possible.

D.1.2 Environment-Cell coupling

The continuous environment concentration fields are solved through the BioFVM reaction diffusion solver. BioFVM is a PDE solving routine which utilises the tridiagonal matrix, or Thomas, algorithm (Raviart and Thomas, 1977) for the finite volume method to solve reaction-diffusion PDEs of the form,

$$\frac{\partial c}{\partial t} = \nabla D(x) \cdot \nabla c - \lambda \cdot c + E \cdot (c - c') - U \cdot c + \sum_{i \in C} l_i(x)[E_i \cdot (c_i' - c) - U_i \cdot c]$$ (D.2)
in a domain $\Omega$ with zero flux condition $(D \circ \nabla c) \cdot n = 0$, where $n$ is a unit outward normal to the boundary $\partial \Omega$. The possible boundary conditions that may be implemented are limited to be of the zero-flux type for each chemical species in the PDE. The solver utilises a constant global initial condition $c(x, t = 0) = c_0$ (Saxena et al., 2021). Here,

$$1_i(x) = \begin{cases} 
1 & \text{if } x \in \Omega_i \\
0 & \text{if } x \notin \Omega_i 
\end{cases}$$

denotes the indicator function for the finite-element Gauss points $x$ contained within the area associated within the $i$-th cell and the Thomas algorithm is implemented to improve computational efficiency. While BioFVM presents a general reaction-diffusion solver, PhysiCell limits the admissible diffusion constant to be isotropic and homogeneous across the domain and limits the sources to weighted point sources (Ghaffarizadeh et al., 2015). PhysiCell modifies equation (D.2) to simplify the cell based model. The cells act as point sources/sinks to the field at the location of the cell centre with the interaction modelled by,

$$\frac{\partial c}{\partial t} = D \circ \nabla^2 c - \lambda \circ c + E \circ (c' - c) - U \circ c + \sum_{i \in C} \delta(x - x_i)(x) V_i [E_i \circ (c'_i - c) - U_i \circ c],$$

where $V_i$ is the volume of the cell given by equation (D.1) which acts as a source weight factor wherein the strength of the point-wise source is directly proportional to the cell volume/size. Here, $D$ is a matrix of diffusion coefficients which are constants throughout the simulation, $\lambda$ is a vector of constant environment decay rates for the chemical species, and $\circ$ is the element-wise (Hadamard) product. The chemical interactions are formed by the chemical uptake and excretion rates, $U$ and $E$ respectively, $s$ for the cells which in general vary on a cell to cell basis where $c_i$ denoted the internal chemical concentration vector for cell $i$.

The limitations of PhysiCell are shown with the replacement of the indicator function in equation (D.2) with the delta function in equation (D.3). This change is motivated by the cell interacting with the environment at a single point and the volume of the cell is accounted for not by a physical extent to the cell but by a weighting for the cell volume, $V_i$. Therefore the cell chemical uptake/secretion is independent of cell shape (only spherical shapes are permitted) or orientation with the cells only having a "pseudo" size. Furthermore, the homogeneous and isotropic diffusion rates limit the possible environment dynamics. There are no reactions in the environment and this is limited by the underlying solvers and equation (D.2), allowing for decay terms only. Furthermore, the possible boundary conditions are limited to the zero-flux type which reduces the possible environment dynamics when
considering domains that are coupled to a further environment such as media sources or modelling oxygen input from a laboratory environment.

**D.1.3 Cell mechanics**

In PhysiCell, the bulk domain is modelled as a low viscosity fluid free from damping where each cell agent may move through the domain compelled by a set of cell specific velocity rules. This cell motility is implemented as a set of stochastic update rules for the cell velocity. For each cell a migration bias and persistence time is provided with an instantaneous velocity determined through the cell mechanics and by balancing of the cell equations of motion. The persistence time alters a probability for updating the velocity and the migration bias controls the degree of stochasticity about the bias direction.

Mechanical interactions between cells are determined through defining a cut-off distance within which the cells are considered to be interacting. The user can define how this interaction occurs (which defaults to a linear Hookean spring force) by editing the C++ code with the resulting interaction potential being used by the trunk code to determine the updated velocity. These mechanical interactions update the velocity thorough the inertia-less force balance as described in equation (C.12) with the velocity being determined through the cell potentials,

\[
v_i = \sum_{j \in N_j(t)} (1 - \delta_{ij})(A_{ij} \nabla \phi_{ij}(r_i - r_j) + R_{ij} \nabla \psi_{ij}(r_i - r_j)) + A_{iM} \nabla \phi_{iM}(d(r_i)\mathbf{n}(r_i)) + R_{iM} \nabla \psi_{iM}(d(r_i)\mathbf{n}(r_i)) + v_{i,mig}, \tag{D.4}
\]

where \(v_{i,mig}\) is the migration velocity determined as a function of the migration bias and persistence time. \(A_{ij}\) represents the strength of the cell-cell adhesion for cells \(i\) and \(j\) whereas \(\phi_{ij}\) is the associated interaction potential which depends on the cell’s influence and connectivity. Likewise, \(R_{ij}\) is the cell-cell repulsion strength and \(\psi_{ij}\) is the repulsion potential. The gradient of the potentials take the place of the direct force terms, through \(F_{ij} = -\nabla \phi_{ij}\) or \(F_{ij} = -\nabla \psi_{ij}\), in equations (C.8)–(C.12).

As the cells are localised within an environment the environment structure and composition also affects the cell’s motility. A PhysiCell simulation assumes that the cells are placed upon an extracellular matrix and the adhesion/repulsion form the matrix are considered \((A_{iM}, R_{iM}, \phi_{iM}, \psi_{iM})\) with \(d(r)\) representing the distance between the cell and the matrix and \(\mathbf{n}(r_i)\) the unit normal out of the matrix surface. The interaction potentials from neighbouring cells are also used to define a pressure over a cell’s surface.
The cell’s instantaneous velocity is incorporated into the position update term using a linear multistep second-order discretisation (Hairer et al., 1993) instead of equation (C.12) which replaces the force terms with the velocity;

\[ \mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \frac{1}{2}(3\mathbf{v}_i(t) - \mathbf{v}_i(t - \Delta t)), \]  

which retains a higher accuracy than the Forward Euler method employed in equation (C.12) at a higher computation cost.

### D.1.4 Model implementation and exploration

A user interfaces with PhysiCell by creating extensible markup language (XML) files using a simple GUI application, by providing CSV files, or through editing the trunk code. A basic model builder GUI is provided as an addition to the PhysiCell trunk code to define environment and cell types. The GUI populates an XML file with the necessary base simulation parameters and if a user implements custom functions they add their own variables and values within this XML. As these XML files are sufficient to produce a simulation, future simulations may bypass GUI and the XML input facilitates model sharing through distributing the XML file. If more complex interactions are required the user may edit the custom C++ code files.

The issue of parameter space exploration in PhysiCell was considered by incorporating an additional module to the simulation pipeline. The PhysiCell-EMEWS extension was developed to facilitate high throughput hypothesis testing for cancer modelling (Ozik et al., 2018, 2019). The Extreme-scale Model Exploration with Swift (EMEWS) extension was developed to implement a model exploration framework around the PhysiCell ABM environment. EMEWS implements instances of both Active Learning (AL) models and Genetic Algorithms (GA) to ascertain the region of simulation parameter space satisfying model constraints. For AL, the parameter set is determined that would satisfy a user given simulation objective and for GA the optimal parameter set is determined to satisfying an objective fitness measure. The size of the parameter space region was used by the EMEWS authors as a measure for the robustness of the parameter set, where the larger the surrounding viable region the more robust the particular parameter set. EMEWS applies parameter sweeping by modifying the parameter values in the underlying simulation XML file.

The EMEWS extension has been applied to explore models built for oncology studies. In the study of immune-tumour interactions (Ozik et al., 2018), the authors trained a random forest classifier with the objective of keeping the final tumour
cell count below a given threshold to evaluate a viable parameter set. Different parameter sets were then used to determine the structure of the viable parameter sub-space, verifying the space by calculating the optimal set via the EMEWS GA. In addition to GA, Gaussian Processes and Neural Networks have also been applied to PhysiCell simulations for the study of parameter optimisation in Cancer treatment studies (Preen et al., 2019). Here it was shown that computational efficiencies may be found by further developing the evolutionary algorithms acting upon the PhysiCell to reduce the number of instances of the computationally expensive ABM simulation. This method may be implemented in different software platforms to improve the computation of large scale simulations.

While evolving the overall simulation acts to optimise the simulation parameters, evolutionary methods may be applied to the cells to model a biologically evolving model. PhysiBoSS (Letort et al., 2019) is an extension to PhysiCell which incorporates a Boolean network, originally developed within the MaBoSS (Stoll et al., 2017) package, for modelling intracellular signalling. This network is used to model genetic variations in the cells which may be coupled to wider PhysiCell properties such as environmental stimuli or the cell cycle state. This expands PhysiCell to the study of evolutionary population dynamics by providing a means to model mutations as a Boolean switch of implemented pathways and behaviours.

D.2 CompuCell3D

CompuCell3D (CC3D) is an open-sourced CPM simulation suite developed to model large multi-cellular systems with a low computational cost (Swat et al., 2012). The software may be obtained by following the download links on the software’s website https://compucell3d.org/ (accessed on 14/04/2022). Originally developed to understand cell patterning due to differential adhesion (Glazier and Graner, 1993) and the emergence of clusters and stratification in tumour cells (Swat et al., 2015), CC3D has found wide applications across tissue patterning studies; vertebrate segmentation (Hester et al., 2011), somite formation (Dias et al., 2014), and convergent-extension (Belmonte et al., 2016). In particular, CC3D has been used to understand the role that chemical gradients and cell-state switching play on the breakdown of endothelial structure in the tumour microenvironment (Chowkwale et al., 2019). Simulations in CC3D may be in either 2- or 3-dimensions but in practice 2-dimensional simulations are often preferred and are advocated by the developers. This preference is due to the increased computational cost of the increased neighbourhood size for each pixel when projecting into the third dimension.
The CC3D software is written in Python with the speed of C++ leveraged to write underlying algorithms. The code has been designed to have a modular structure. This allows for new cell and simulation properties to be added which integrate with the CPM and are cast into terms which are added cumulatively to calculate the CPM cell energy. These different energy terms are added in the CPM Hamiltonian through including additional "plugins" to the simulation. These provide custom behaviours for the Hamiltonian which is used to determine the pixel flip capabilities for changes to the cell lattice.

Plugins are evaluated on each pixel flip attempt and each attempt constitute the smallest time step in the simulation from which other elements of the algorithms occur at appropriate number of flip attempts. A simulation monte carlo step occurs after a set number of flip attempts and for each complete step a second scale of dynamics is possible. The "steppable" class acts on the connected regions defining cells in the cell lattice affecting the cell dynamics or a "steppable" class may be used to implement a reaction-diffusion system in the field lattices. These steps are also used to save the simulation output. The steppables are formed through Python scripts as part of the CC3D simulation input. A user may alter template Python functions to implement custom models.

D.2.1 Cell model

CC3D utilises the CPM model to represent cells in a simulation. A simulation is composed of a set of regular lattices, each representing a different element of the CPM. The cell lattice contains the agents and evolves through the CPM algorithm. Cells may be modelled as single isolated individuals moving through a bulk or contiguous tissues of cells and CC3D uses the Von Neumann neighbourhood to determine connectivity. Cell to cell adhesion is modelled through cell-cell contact energies while CC3D allows for cells to become attached by linking the cell agents of connected pixels into larger compartmentalised structures. The cell shapes are determined through the cell-cell and cell-bulk contact energies affecting the pixel flip probabilities through the Hamiltonian. An additional term is added to the Hamiltonian to reflect the variation between the number of connected pixels representing the volume of the cell and a target cell volume.

Growth is implemented by increasing the cell "target volume", this provides an energetic advantage to increasing the total number of pixels occupied by the cell. However this is an indirect method to model cell growth and is weighed against other energetic considerations in the system Hamiltonian, such as contact energies. The user may write custom cell growth laws to parameterise the target cell volume, such
as applying pressure terms or using the accumulation of an environment chemical as a factor to increase the target. These growth laws are however not implemented by default in CC3D and require the user to write Python code. Cell division may be implemented by changing a pixels allocation from a parent cell to an offspring cell and setting new target volumes for parent and offspring. For cell death, a target volume may be incrementally reduced to zero to model the evaporation of the cell.

Cell division is added to the simulation by using the MitosisSteppable. Here, the user writes a Python class for the cell conditions under which cell division is implemented. Likewise, CC3D also provides a GrowthSteppable for the user to detail the conditions and processes which lead to cell growth (i.e changing of the target volume). An additional cell property provided in CC3D is the concept of cell links. Links constrain the distance between the cell centres of mass through a Hookean spring potential. For cells $i$ and $j$ with centres of mass at locations $x_i^{\text{COM}}$ and $x_j^{\text{COM}}$ respectively, the spring potential between the two linked cells;

$$H_{\text{spring}}(i, j) = \lambda_{\text{spring}}( |x_i^{\text{COM}} - x_j^{\text{COM}}| - l_{ij})^2$$  \hspace{1cm} (D.6)

where $l_{ij}$ is the target link length and $\lambda_{\text{spring}}$ is the spring stiffness constant. This link property is distinct from the previous properties added to CC3D. By summarising the CPM cells pointwise by their centres of mass and introducing a spring force between them, the cells are treated comparably to off-lattice centre based models. Therefore this property may expand CC3D’s applications from CPM to centre based modalities and increase the scope to model phenomena.

CC3D allows chemical reactions to be defined within a cell agent and implemented with kinetic rates. Reactions are performed within the Python scripts as part of a steppable between simulation update times. The concentrations are associated with a cell agent and may be weighted by the cell’s volume to provide a uniformly distributed pixel by pixel shared concentration. Chemical models may be written explicitly in the Python script or defined by using the SBML standard (see section 4.4) and the developers suggest using Antimony (see section 4.4) to produce the necessary input.

### D.2.2 Environment-Cell coupling

A surrounding environment is implemented in CC3D by labelling non-cell pixels as bulk environment pixels. These pixels share a contact energy with cell pixels but are skipped during the CPM algorithm when considering pixel flip attempts for computational efficiency. This pixel lattice may be accompanied by a set of field
lattices that track the dynamical evolution of chemical species, one field lattice per chemical. These field lattices evolve through a reaction-diffusion equation solved by using the finite difference scheme. Complex environment and diffusion characteristics may be provided thorough implementing blocking regions, such as obstacles or walls around the system. These obstacles exist in the cell lattice with pixels being "frozen" in place. These pixels are then removed from the copy attempt but may impart a surface energy change to agent pixels.

CC3D provides a range of reaction-diffusion solvers where solver parameters are stored within a custom markup language, CC3DML. Each solver uses a finite element method to consider the spatial elements and each routine routine uses the explicit forward Euler method to calculate the change in concentrations over time. The functional form of the PDE is set by the user by calling a respective solver class which implement different underlying routines for computational efficient calculations; the SteadyStateDiffusionSolver2D for time-independent diffusion, DiffusionSolverFE for general diffusion systems, ReactionDiffusionSolverFE for reaction-diffusion systems, and AdvectionDiffusionSolver for advection-diffusion systems. Each solver will be considered in turn.

For simulations requiring a constant chemical gradient across the domain the SteadyStateDiffusionSolver2D solver class is provided. CC3D has the option to calculate steady state concentrations from a source-sink distribution by solving the time independent Helmholtz equation,

$$\nabla^2 c = F(x, c) + \lambda_{\text{decay}} c,$$

(D.7)

where \(F(x, c)\) is a function of source-sinks due to cell excretion and uptake, at location \(x\), and \(\lambda_{\text{decay}}\) is the chemical environmental decay/removal rate. This steady state may be used to provide chemical gradients to the system in a computationally efficient manner by solving the Helmholtz equation and modelling systems with negligible concentration rate change. These are found biologically when the rate of environment chemical diffusion is much greater than the cell dynamics.

When the rate of change of concentrations is non-zero, a standard diffusion equation with decay is solved using the DiffusionSolverFE class. This solver can be used for simulations where the diffusion rate depends on the local cell types. This solver has been made GPU compatible to enable rapid computation;

$$\frac{\partial c}{\partial t} = D \circ \nabla^2 c + \gamma \circ C + E - U,$$

(D.8)

where \(D \in \mathbb{R}^N_{\geq 0}\) is a vector of isotropic and homogeneous diffusion coefficients,
\( \gamma \in \mathcal{R}^N \) is a vector of environment chemical decay/production rates, and \( E \in \mathcal{R}^N_{\geq 0} \) and \( U \in \mathcal{R}^N_{\geq 0} \) cell/source excretions rates respectively. Heterogeneous diffusion rates may be implemented by changing the rate depending on the cell type inhabiting the simulation pixel.

For more complex reaction systems that vary temporally CC3D comes with a selection of different reaction-diffusion solvers. For reaction-diffusion systems the ReactionDiffusionSolverFE class may be used,

\[
\frac{\partial c}{\partial t} = D \circ \nabla^2 c + \gamma \circ C + E - U + R(C, cellTypes), \tag{D.9}
\]

where \( R(C, cellTypes) \) is a vector of reaction rates that may depend on the environment concentrations and the cell types at the pixel location \( x \). There is also a basic advection-diffusion solver implemented in CC3D, AdvectionDiffusionSolver, which solves the advection-diffusion equation on a cell basis rather than lattice grid. The advection term is modelled through the movement of the cells rather than solved explicitly,

\[
\frac{\partial c}{\partial t} = D \circ \nabla^2 c + \mathbf{v} \cdot \nabla c + \gamma \circ C + E - U, \tag{D.10}
\]

where \( \mathbf{v} \) is the velocity of the bulk fluid environment.

Each solver have a range of properties that the user may edit in the CC3DML file. For example, global diffusion rates, initial concentration distributions and boundary conditions are editable under the solver steppable class. Boundary conditions (BCs) in CC3D are specified as the minimum and maximum position of the plane (x, y, or z planes) and are either of the ConstantDerivative or ConstantValue types corresponding to the Neumann or Dirichlet conditions respectively. Local diffusion and decay (for the SteadyStateDiffusionSolver2D and DiffusionSolverFE classes) and reaction terms (for the ReactionDiffusionSolverFE class) may be specified by considering the lattice pixels as cell agents and customising the equation properties to vary on cell type. This ability to change properties on a pixel basis is provided by solving the FE equations on the same lattice as the agents and may be used to simulate chemical PDE systems, such as the Sel’kov-Schnakenberg reaction system. The diffusion properties may also be changed dynamically during the simulation by making the environment diffusion coefficient a property of the cells. One possible exemplar case for this would be to alter the diffusion rate as a model for cellular ECM secretion, and this may vary for different cell types in the simulation.

Reaction-diffusion dynamics have been considered in CC3D and have previously been tailored to experimental systems (Pally et al., 2019). The authors
considered how the extra cellular matrix (ECM) remodelling affects the invasion of cancerous cells, comparing the 2D simulation output with experimental microscopy images. The CPM model considers two agent types, the cancerous cells and inert solid ECM pieces, with reaction-diffusion system considered the diffusion and degradation of two extracellular proteins. The proteins do not react amongst themselves but are secreted at the edge of cells whom are in contact with ECM pixels. The rate of secretion is dependent on the concentrations of the proteins at the local lattice points to the secreting cells. The ratio of these proteins is used to selectively "degrade" the ECM by switching the agent types from ECM to be that of the default surrounding medium. The authors found that the ECM density inhibited the migration of the cancerous cells and that the output simulation structures were qualitatively similar to the experimental time-lapsed images. While this study demonstrates that local chemical gradients can play a strong, non-linear, role in tissue structures the proteins used do not chemically interact within the environment but rather trigger cell type switches.

D.2.3 Cell mechanics
Mechanics in the CPM are controlled by altering the pixel flipping probabilities and cell target volume or surface area. Forces acting upon the agent population are implemented by using the ExternalForcesCollectiveMotion plugin. These forces are used to calculate a change in the system Hamiltonian. CC3D is also capable of implementing cell polarity for the cooperative movement of a collection of cells (Khataee et al., 2020). The authors used the perimeter of the cell agents as a measure to model the contraction of the cell cortex and demonstrated the propagation of a tissue wave front in the absence of cell proliferation. This motility was attributed to the cortex contraction with only a weak contribution by the cell surface energies. This study demonstrated that CC3D is capable of modelling dynamical phenomena beyond the patterning due to cell-cell surface energy changes.

D.2.4 Model implementation and exploration
In addition to downloading a local version, the complete CompuCell3D package is available on the web platform NanoHub (Klimeck et al., 2008). This platform enables the rapid deployment of the CC3D with the inclusion of a cloud based standardised graphical user interface independent of the user's operating system. This user interface includes the "Twedit++" integrated development environment, the "CC3D Player" which provides a built-in simulation visualiser, and a "wizard"
to produce a base simulation XML-like CC3DML file.

CC3D has been adapted for parallel computation through using the Message Passing interface (MPI) (Chen et al., 2007). Inefficiency in MPI computations on compute node clusters is derived from the communication speed. As the simulation is spread over the nodes the state of each separate node needs to be passed between each computer node. This passing time for information reduces the efficiency gained by the paralleisation process. One solution is to reduce the frequency that the nodes communicate allowing separate nodes to continue the simulation without updating one another. This results in the nodal simulations drifting in their accuracy but reduces the inefficiency of communication. Therefore the overall simulation speed is increased at a cost of accuracy which is dependent on the complexity of the simulation.

To benefit model fitting parameter scans were implemented within CC3D (Swat et al., 2015). Parameters scans are implemented by replacing parameters in the CC3DML file with variables and providing those variables with scan ranges and step sizes in an accompanying JSON format file. The full simulation is run for each parameter set and may be evaluated via a metric defined as a steppable in the Python code by the user. This metric may then be output into a log file for parameter fit exploration. The format for implementing parameter scans may also be used to perform a parameter sensitivity analysis.

Chemical reactions may be implemented using the Antimony suite where the chemical equations are written in JSON format directly into the Python simulation file (see section 4.4). Alternatively, a user may write their reactions in an external software that outputs a SBML file (such as Copasi (Hoops et al., 2006) or Tellurium (Choi et al., 2018)) and then import this into CC3D. Boolean networks may be implemented through using the MaBOSS extension which uses its own syntax separate from CC3D (Stoll et al., 2017). The networks from MaBOSS are attached to the cells in CC3D using the same methods as Antimony or SBML model attachment.

While not as coding intensive as the C++ based software, CC3D requires user to be familiar with the Python programming language and also be comfortable editing XML and JSON files. This wide ranging syntax may hinder the deployment by users with less programming experience. However, the use of the online NanoHUB application reduces the installation to be more "plug-and-play" with no need to install dependencies.
D.3 Morpheus

Morpheus is a C++ based open-source multiscale modelling software provided under the BSD licence and available at https://morpheus.gitlab.io/ (accessed on 14/04/2022). The package implements CPM models in 2- or 3-dimensional simulations using a GUI application to construct and control simulations (Starruß et al., 2014). Cells may be provided an intracellular ODE system to model chemical reactions or implement stochastic and delay differential equations to model cell signalling and gene regulation systems. Cell lattices in Morpheus may be either regular rectilinear or hexagonal lattices and may then be coupled to fields for environment reaction-diffusion systems.

Morpheus has been developed with the user experience in mind so that the software would be usable by an audience with a wide range of model development experience. This includes an explicit pipeline to incorporate a model from public SBML repositories (such as BioModels) and scripts to include information from experimental microscopy cell images. Morpheus has been used to study the spread of pathogens through tissue (Imle et al., 2019), track cell recruitment during Drosophila wing growth (Muñoz-Nava et al., 2020), model T cells using the implemented stochastic methods (Beck et al., 2020), model cell signalling and gene expression (Mulberry and Edelstein-Keshet, 2020), and model collective migration of rod bacteria (Starruß et al., 2007).

D.3.1 Cell model

Morpheus simulations represent cells as connected pixel areas according to the CPM. Therefore cell properties are used to construct constraints on the Hamiltonian dictating pixel flip probabilities. In Morpheus, these constraints are limited to volume, VolumeConstraint, and surface, SurfaceConstraint, Hamiltonian contributions. The cells run internal ODE system to determine the cell’s state. Neighbourhoods in the lattice are specified by defining either a maximum distance between two lattice points or by defining an integer order. The lattice distance is defined as the maximum Euclidean distance between two points within which the two lattice points are considered neighbours. The order neighbours label each lattice point with an integer connectivity and the order neighbour denotes the maximum ranking a lattice point can have while still being considered a neighbour.

Cell division is implemented by adding the CellDivision plugin and providing division event triggers. The plugin introduces a new offspring cell to the simulation after a division rule event is triggered. This trigger may be when an
internal chemical concentration reaches a threshold value. The target volume of the parent and offspring are then reset to represent cell shrinkage due to division.

Cell based chemical reactions may be imported through the SBML standard or implemented as a chemical System within the cell. The System class may perform Ordinary, Stochastic, or Delay differential equation models. The differential equations may be solved using a range of numerical solver schemes, based on the Euler, Midpoint, and Runge-Kutta methods. However, the forward Euler numerical method is recommended for potentially stiff ODE systems. Stochastic differential equations are solved through adding a Gaussian distributed random variable term to the solver step, scaled by the timestep. Meanwhile, Delay differential equations may be performed with a constant delay time. These reaction systems are subject to the initial cell conditions which may be homogeneous, where every cell in the simulation starts with the same initial condition, heterogeneous, providing initial conditions by using the cell id or location, or cell-specific, providing condition through the cell type.

Simulations may be further tailored by providing conditions in which update terms in the simulation are ignored. One possible result is to remove the cell from the simulation update and consider the cell as inert using the freezer option. Alternatively, a user may specify a condition, for example preventing update that would break connectivity between two cells using the ConnectivityConstraint option.

D.3.2 Environment-Cell coupling

The environment in a Morpheus simulation is described by the System plugin. This plugin provides a scalar field for each chemical component in the environment and details the reaction-diffusion PDE. These equations are specified by providing an isotropic-homogeneous diffusion rate and an homogeneous ODE term;

\[
\frac{\partial c}{\partial t} = D_c \nabla^2 c + f(c) \tag{D.11}
\]

where \( \frac{dc}{dt} = f(c) \) is the homogeneous ODE for the rate of change of the chemical due to environmental reactions or cell source/sink dynamics. Initial conditions for environment concentrations are provided within the plugin and they can be spatially specified by using a set of inequalities for the different regions.

The solver routines provided within Morpheus use the explicit forward Euler method for both ODE and temporal PDE solutions. The finite difference method with a fixed time step is used to solve the spatial aspects of the reaction-diffusion
PDE. However, Morpheus does not currently support advective terms in the environment PDE. Using the same method as cell based chemical reactions, a user may also implement stochastic differential equations for the ODEs, or the spatially homogeneous term in the environment PDE, by adding a Gaussian distributed random number to the equation which adds a $\sqrt{\Delta t} \Delta W_n$ term into the forward Euler scheme. The cell and environment dynamics may be scaled by the user to control the rate of advancement for the internal cell ODE solvers in comparison to the external reaction-diffusion solvers, thereby bridging the different timescales of cellular and extracellular reactions.

PDE boundary conditions include default periodic, constant Dirichlet, noflux Zero-Neumann which are specified for the entire length of a particular boundary, i.e the x, or -x boundaries etc. Irregular domains obtained from images may also be solved over by providing a pixel mask to Morpheus as a TIFF file. However, the irregular domain may only support Dirichlet or Zero-Neumann BCs. The environment initial conditions may be specified by defining a mathematical expression under the InitPDEExpression plugin option. This may be used to provide constant or randomly varying initial distributions for each chemical and/or set up chemical gradient fields.

The environment modelling capabilities of Morpheus were used to study the role of the extracellular matrix (ECM) on the formation of the vascular network (Köhn-Luque et al., 2011). The angioblast produced ECM was modelled as a non-diffusing continuum with diffusing vascular endothelial growth factor (VEGF) affecting the elongation of precursor angioblast cells through chemotaxis. The simulation output was compared to laser confocal images to show comparable patterning. Overall, this study showed that pattern formation CPM cells using the CPM and reaction-diffusion equations.

D.3.3 Cell mechanics

Cell mechanics in Morpheus are controlled by the Hamiltonian in the CPM. Additional terms to the Hamiltonian depend on whether the System plugin is present and are added using the cellType plugin. For a chemical field, the a chemotaxis or haptotaxis term may be added. The cells may also be provided with a DirectedMotion term to provide a bias to the pixel flipping or a persistence term to provide a resistance to changes in motion. The stochastic cell motility has been considered for investigating the cell phase dynamics in zebrafish (Lupperger et al., 2020).
D.3.4 Model implementation and exploration

Morpheus has been designed with the intention for a user to build up their model within the GUI application. A so-called "middle-out modelling" strategy has been applied wherein a cell-based simulation, cell ODE systems, and environment PDE systems may be constructed and tested separately in-app before explicitly coupling the different aspect into one integrated simulation (Walker and Southgate, 2009). Therefore a user may test the separate components for computational errors to limit the emergence of simulation artefacts in the multi-scale output. This also provides a modular model approach where different sections of the simulation may be readily interchanged to focus on a specific aspect of the model.

Morpheus uses the so-called MorpheusML format, a bespoke XML format, to contain model parameters and can be constructed using the detailed graphical user interface (GUI) which is provided within the software environment. Models may be imported through SBML, which are translated into the native MorpheusML format, or written directly in the software by using the GUI. This interface provided a visualisation plot using a pipeline to the Gnuplot software application. Simulation results may be output using either Tiff or Vtk formats. The VkPlotter plugin may be used to export the cell simulation as a .vtk type for further visualisation with Paraview and is a suggested post-processing software. Tiff files may also be imported into Morpheus in which cells are labelled by integers for reconstructing the cell population morphology. The cell population patterning was compared to experimental images in Köhn-Luque et al. (2011), who also performed a sensitivity analysis by varying a series of parameters and quantifying the change in the output simulation.

Chemical reactions may be written directly into the application by defining the differential equation for the change of the chemical concentration. Therefore this requires a user to manually construct the equation from the chemical reaction and assume a kinetic rate. The parameters in the chemical reaction can be swept over by defining parameter sweeps directly in the GUI, selecting the parameters to be swept over and providing ranges for the sweep values. This does not require any additional editing of the underlying code by the user. Meanwhile, parameter estimates may be made from input experimental data by using the provided FitMultiCell toolbox software. For bespoke simulation a user may develop a plugin for Morpheus, implemented as C++ code with an associated XML file following a plugin template provided in the developer documentation.
D.4 iDynoMiCs

iDynoMiCs is an open source Java based software developed for modelling microbial biofilms and is available for use under the CeCILL license (Lardon et al., 2011). The software, downloadable from https://www.birmingham.ac.uk/generic/idynomics (accessed on 14/04/2022) has been designed to be modular and adaptable while aiming to be accessible to users with low prior programming experience. Simulations are based on a model for a chemostat and the elements are tailored by editing a group of XML files.

The software has been applied to study environmentally motivated metabolic switching by modelling denitrification in a bacterial biofilm grown within a bioreactor (Lardon et al., 2011). The bioreactor chemostat considers an environment composed of three chemical concentrations; Oxygen, Nitrogen, and a pseudo-chemical species reacting to model the consumption of the Oxygen to produce a "Chemical Oxygen Demand" (COD) term. Different environment stimuli conditions were tested. They applied stress to the biofilm under aerobic, anaerobic or pulsed Oxygen intake conditions which were simulated through controlling the Oxygen concentrations in the bulk phase of the chemostat.

iDynoMiCs has been used to study the Warburg effect in tumour cells (Shan et al., 2018). Here, the authors used flux balance analysis (FBA) to determine a set of parameters with which to implement Monod kinetics in a two species ABM. A two species system was considered where one of the species represent healthy Stromal cells while the other species represents the tumour cells with a perturbed cancerous metabolism. Both cells were in competition for finite oxygen and glucose resources which diffused through the biofilm and were taken up by the cells. A reduced metabolism was implemented as a base in both cell types and focused on the glycolysis pathway with a range of precursor metabolites, including pyruvate and lactate. The study investigated how the number of layers of healthy stromal cells between the oxygen source and tumour cells inhibited tumour growth. The increase in healthy cells between oxygen source and tumour resulted in increasing pyruvate/lactate flux proliferation ratios in the tumour cells.

D.4.1 Cell model

iDynoMiCS uses a centre-based cell approach to model the agents as overlapping spheres for cell-cell interactions. Cell agents are multi-compartmental with a collection of particle classes defining the different compartments. There are a range of agent classes to control the possible dynamics for each agent and interactions
between them. Agents may be defined as bacteria, using the Bacterium class, where a new agent is added in discrete division events and the agents may contain multiple compartments, extrapolymeric substance (EPS) that is continuously excreted from the cells to form a growing capsule of polymeric substance BactEPS for agents with a predefined reaction behaviour or BactAdaptable for agents with switchable reactions, finally ParticulateEPS that treat EPS capsules as a particle type for inclusion into a cell agent with predefined reaction dynamics. Microbial cells are provided with a list of different state variables to model a parameterised cell structure. The internal compartment of the cell is considered to be encapsulated by a layer of EPS and the chemical components of the cells are distributed between the two compartments. The total mass and volume of the cell is calculated by considering both cell and compartments.

Reactions occur within the cells by defining a metabolic model with cells dividing when the internal concentrations hit user defined thresholds. The cell division process produces an offspring agent with a random orientation about the parent with parent cell contents between parent and offspring being shared with a Gaussian random noise about equal sharing. A separate set of thresholds are used to determine cell death in which a cell is removed from the simulation. The contents of the cell are not shared with the environment but removed from the system.

D.4.2 Environment-Cell coupling

The simulation environment was developed to model the environment conditions in a chemostat where there is a flow of nutrients through the system. The simulation domain is composed of a regular rectilinear grid and explicitly broken into 4 sub-domains; the inert Support modelling the base substrate from which the initial biofilm grows, the Biofilm itself which contains the cell agents and EPS particulates, the Boundary layer representing an intermediate where chemical concentrations may diffuse into the bulk, and the Bulk which is a well-mixed phase immersing the biofilm and whose chemical concentrations are held at the bulk values. The boundary between these layer pairs is explicitly defined. The boundary layer contains a mixture of chemical concentrations and cell and EPS agents eroded from the biofilm.

While EPS for fluid ECM is modelled through particulates, chemicals in the environment are implemented using a solute class containing the species diffusivity in water. These solutes are represented by scalar fields and the dynamics are controlled differently depending on the simulation layer.

iDynMiCS was developed to model the chemical and cellular growth of biofilms in a bio-reactor. The initial cell cluster are placed onto a inert base and are
submersed in a well-mixed nutrient bath. Diffusion is considered through the forming of biofilm and in an intermediate biofilm-bulk area to reduce the computation. Chemical reactions are mediated by the cells in the biofilm area,

$$\nabla \cdot (D \cdot \nabla c(x)) + E(x) - U(x) = 0,$$

forming a quasi-steady state where $D$ is the vector of diffusion rates for the solutes in water, $c$ a vector of solute concentrations, $E(x)$ cell excretion rates at a cell centre at location $x$ and $U(x)$ cell uptake rates at the cell centre location. The diffusion equation is subject to no-flux (zero Neumann), constant/variable (Dirichlet) concentration, solute-permeable membrane (selective flow), and periodic boundary conditions.

As the bulk phase models the fluid phase of a chemostat, there is a flow into and dilution of the system and iDynoMiCS assumes that no reactions occur within the bulk. As the bulk is well-mixed the change in concentrations in the bulk are given by,

$$\frac{dc}{dt} = d \circ (c_{\text{inflow}} - c_{\text{bulk}}) + R_S \cdot \sigma_R,$$

where $d$ are species dilution rates, $c_{\text{inflow}}$ is the concentration flow into the chemostat. To model the coupling between layers a separation of time scales from fast solute dynamics and slow cell dynamics is performed.

Reactions systems in iDynoMiCS are considered global, acting over the full domain, and are written in the XML file. The user provides a name for a reaction and the cell type where the reaction occurs. Each chemical species within the reaction are provided with a set of multiplicative kinetic terms from a list of pre-programmed rate laws and the yields are provided with the stoichiometry as a set of parameters.

**D.4.3 Cell mechanics**

The overlapping sphere model may be used to provide a radius from the cell centre in which cells may interact. Cell mechanics are formed through a shoving motion proportional to cell overlap with a constant `shoveFactor` determining the base cell motion. To the base motion an advective motion based on the relaxation of biofilm pressure is added. For the set of agents $A = C \cup E$ where $E$ is the set of EPS agents and $C$ is the set of cell agents. To model compaction in biofilms a pressure field is calculated,

$$\nabla \cdot \nabla P = \sum_{i \in A} \frac{r_i}{\rho_i},$$

181
where $r_i$ is the (bio-)mass production rate of the $i$-th agent with density $\rho_i$ with
This pressure term is used to calculate an advection velocity given by $\mathbf{v} = -\nabla P$ for pressure $P$ where the pressure is calculated by considering the radii of the interacting agents.

Cells excrete EPS particulates through ejection of part of the outer EPS coating and the associated mass with the resulting particulates being labelled as non-cell agents which contribute to mechanical pressure and cell shoving. EPS agents are formed with no internal compositions and are solely composed of the outer EPS layer using the non-bacterial agent classes.

Cell agents are also provided with an attachment radius within which they can connect to the biofilm layer. The biofilm undergoes erosion at the intermediate biofilm-bulk surface. The erosion process was expanded with a range of cell attachment mechanisms were implemented to model biofilm structure (Li et al., 2015). These cells may detach to flow along the biofilm layer or be lost to the bulk flow.

D.4.4 Model implementation and exploration

For editing the code the developers suggest using the Eclipse IDE and a series of simulations may be run by calling a Python script from the command line. Parameter sweeps may be performed by collecting simulation XML files, the so-called "protocols", into a single directory and running the iDynoMiCS scripts on that directory. No inbuilt parameter sweeping tools are provided so a user must manually write the different simulation files to be swept over. The developers recommend using MATLAB for post-processing the simulation data and either use MATLAB or POV-Ray for data visualisation.

For complex and data intensive simulation, summary statistics are a necessity to capture the model dynamics. In the Oxygen pulse investigation by Lardon et al. (2011), a diversity index, $D_{Index}$, was implemented to analyse the effects of the variable environment. Throughout the simulations the occurrence of each species were used to calculate the diversity,

$$D_{Index} = 1 - \frac{\sum_{i\in C} n_i(n_i - 1)}{|C|(|C| - 1)},$$

(D.15)

where $C$ is the set of cell types of cardinality $|C|$, and $n_i$ is the number of cell of type $i$. In the study, the authors found that when the frequencies of the oxygen pulses were varied the diversity was lowest for the faster frequencies. This was linked to the quicker response species being favoured with a competitive advantage provided by
the rapid transitions between aerobic and anaerobic pathways. Diversity amongst the species increased when pulse frequency decreased as species slow to respond had time to proliferate between pulses.
Appendix E

ChemChaste Simulation Files: Coupled Cell-Environment Model

The ChemChaste simulation files for the coupled cell-environment model (see section 5.3). The cell directory structure and parameter sweeping file are given in figure E.1. Cell files are provided in figure E.2 for Cell A and domain information provided in figure E.3.
Figure E.1: The coupled cell-environment simulation configuration file system a), with parameter sweeping values b), the cell key for two cell types CellA and CellB c), and cell layer topology showing an "isolated inoculation" starting condition for the cells (blocks of the two cell types) d).
Figure E.2: Cell properties and reactions for Cell A in the coupled cell-environment model. a) The preferred environment conditions of the cell to be used by the "PreferredEnvironmentMassActionReaction" class. b) The initial concentrations for the cell bound chemicals. c) The reactions occurring at the membrane. d) The concentration rules during cell division whether to share or duplicate concentrations between parent and offspring. e) Concentration thresholds for cell death and division. f) The transport laws describing the exchange of concentration between cell and environment. g) The cell bound reaction system.
Figure E.3: The domain files for the coupled cell-environment model. a) The boundary conditions are defined for the whole perimeter of the domain. Here, a no-flux condition for the $pH$ concentration is used. b) The initial $pH$ distribution is uniform without perturbations. c) The environment domain is defined by a mono-region that is named "Bulk" in d). e) The diffusion rate for the $pH$ "chemical" is swept over (see figure E.1-b). f) The "ExtracellularReaction.txt" reaction file is used over the mono-domain and defined in g).
Appendix F

ChemChaste Simulation Files: Resource Competition

The ChemChaste simulation files for the resource competition model (see section 5.4). The cell directory structure and parameter sweeping file are given in figure E.1. Cell files are provided in figure E.2 for Cell A and domain information provided in figure E.3.
Figure F.1: The resource competition model configuration file system a), with parameter sweeping values b), for two cells CellA and CellB c) arranged in a "mixed inoculation" (alternating) structure d).
Figure F.2: Cell properties and reactions for Cell A in the resource competition model. a) The preferred environment conditions of the cell to be used by the "PreferredEnvironmentMassActionReaction" class. b) The initial concentrations for the cell bound chemicals; including the resource, $S$. c) The reactions occurring at the membrane. d) The concentration rules during cell division whether to share or duplicate concentrations between parent and offspring. Here, the concentrations are shared between the parent and offspring. e) Concentration thresholds for cell death and division. f) The transport laws describing the exchange of concentration between cell and environment. The environment $pH$ is decreased when a cell is present and the resource $S$ is taken in by the cell. g) The cell bound reaction system, whose parameters are swept over.

```plaintext
<table>
<thead>
<tr>
<th>a) Environment.csv</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 #species, value, perturb?</td>
</tr>
<tr>
<td>2 pH, 3.0, false</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>b) InitialCellConcentrations.csv</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 #species, concentration, perturb?</td>
</tr>
<tr>
<td>2 pH, 5.0, false</td>
</tr>
<tr>
<td>3 Biomass, 1.0, false</td>
</tr>
<tr>
<td>4 $S$, 1.0, false</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>c) MembraneReactions.txt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 # Reaction at membrane, Bulk &lt;-&gt; Bulk</td>
</tr>
<tr>
<td>2 MassActionCoupledMembraneReaction : Biomass &lt;-&gt; Biomass</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>d) SpeciesDivisionRules.csv</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 # species, rule (share/duplicate)</td>
</tr>
<tr>
<td>2 pH, share</td>
</tr>
<tr>
<td>3 Biomass, share</td>
</tr>
<tr>
<td>4 $S$, share</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>e) SpeciesThreshold.csv</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 # species, maximum, minimum</td>
</tr>
<tr>
<td>2 pH, 0.0, 0.0</td>
</tr>
<tr>
<td>3 $S$, 0.0, 0.0</td>
</tr>
<tr>
<td>4 Biomass, 1.5, 0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>f) TransportReactions.txt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 # Transport across membrane, Bulk &lt;-&gt; Cell</td>
</tr>
<tr>
<td>2 MassActionTransportReaction : $pH$ &lt;-&gt; $S$ ; $k_f = 0.0 \ kr = -10$</td>
</tr>
<tr>
<td>3 MassActionTransportReaction : $S$ &lt;-&gt; Biomass ; $k_f = 0.1$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>g) Sm.txt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Domain : membrane</td>
</tr>
<tr>
<td>2 PreferredEnvironmentMassActionReaction : $S$ &lt;-&gt; Biomass ; $k_f = 6.0$</td>
</tr>
</tbody>
</table>
```
Figure F.3: The domain files for the resource competition model. a) The boundary conditions are defined for the whole perimeter of the domain. Here, a no-flux condition is used for both $\text{pH}$ and $S$. b) The initial concentration distribution is uniform without perturbations for both $\text{pH}$ and $S$. c) The environment domain is defined by a mono-region that is named "Bulk" in d). e) The diffusion rate for both the $\text{pH} "\text{chemical}"$ and resource $S$ are swept over (see figure F.1-b) with the same rate for both. f) The "ExtracellularReaction.txt" reaction file is used over the mon-domain and defined in g) showing no change in $\text{pH}$. 
Bibliography


Bergmann, F. T. et al. (2014). COMBINE archive and OMEX format: One file to share all information to reproduce a modeling project. *BMC Bioinformatics*, 15.


194


Ermentrout, B. (2002). *Simulating, analyzing, and animating dynamical systems: a guide to XPPAUT for researchers and students*. SIAM.


Garde, R. et al. (2020). Extending the minimal model of metabolic oscillations in Bacillus subtilis biofilms. *Scientific Reports, 10.*


197


Herschend, J. et al. (2018). In vitro community synergy between bacterial soil isolates can be facilitated by pH stabilisation of the environment. Applied and Environmental Microbiology, 84.


Momeni, B. et al. (2013). Strong inter-population cooperation leads to partner intermixing in microbial communities. *eLife*, **2**.


Pili, A. N. et al. (2020). Niche shifts and environmental non-equilibrium undermine the usefulness of ecological niche models for invasion risk assessments. *Scientific Reports, 10*.


