Stochastic Computational Modelling of Complex Drug Delivery Systems

Marija Bezbradica

M.Sc.E.E.

A thesis submitted in partial fulfilment

of the requirements for the degree

of

Doctor of Philosophy (Ph.D)

to the

Dublin City University

Faculty of Engineering and Computing, School of Computing

Supervised by Prof. Heather J. Ruskin and Dr. Martin Crane

April 2013

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of PhD is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____ (Candidate)

ID No.: _____

Date: _____

Abstract

As modern drug formulations become more advanced, pharmaceutical companies face the need for adequate tools to permit them to model complex requirements and to reduce unnecessary adsorption rates while increasing the dosage administered. The aim of the research presented here is the development and application of a general stochastic framework with agent-based elements for building drug dissolution models, with a particular focus on controlled release systems. The utilisation of three dimensional Cellular Automata and Monte Carlo methods, to describe structural compositions and the main physico-chemical mechanisms, is shown to have several key advantages: (i) the bottom up approach simplifies the definition of complex interactions between underlying phenomena such as diffusion, polymer degradation and hydration, and the dissolution media; (ii) permits straightforward extensibility for drug formulation variations in terms of supporting various geometries and exploring effects of polymer composition and layering; (iii) facilitates visualisation, affording insight on system structural evolution over time by capturing successive stages of dissolution. The framework has been used to build models simulating several distinct release scenarios from coated spheres covering single coated erosion and swelling dominated spheres as well as the influence of multiple heterogeneous coatings. High-performance computational optimisation enables precision simulations of the very thin coatings used and allows fast realisation of model state changes. Furthermore, theoretical analysis of the comparative impact of synchronous and asynchronous Cellular Automata and the suitability of their application to pharmaceutical systems is performed. Likely parameter distributions from noisy *in vitro* data are reconstructed using Inverse Monte Carlo methods and outcomes are reported.

Посвећено Тати, Мами, Секи и Андрејy,
за сву њихову предивну љубав и подршку.

(Dedicated to Dad, Mom, Seka and Andrej,
for all their wonderful love and support.)

Acknowledgements

Firstly, I would like to thank my supervisors, Prof. Heather J. Ruskin and Dr. Martin Crane. I am deeply grateful for their support and guidance during my PhD studies which made this journey a very pleasant one. Their expertise and research insights helped me tremendously in directing the course of my work and developing my scientific personality. I am also grateful to my undergraduate Professor, Dr. Dejan Raković, Faculty of Electrical Engineering, Belgrade, whose enthusiasm and dedication to science taught me that we should always aim for gaining new knowledge and broadening our horizons.

I would like to express my gratitude to the Irish Research Council for Science, Engineering and Technology, through the Enterprise Partnership Scheme, with Sigmoid Pharma Ltd. as an Enterprise partner. This work would not be possible without them. I also want to thank Dr. Monica Rosa and Magda Wasik Klimek from Sigmoid Pharma for useful discussions and their assistance with pharmaceutical topics and drug development.

I wish to thank my many current and former colleagues from DCU and Centre for Scientific Computing & Complex Systems Modelling (Sci-Sym) group. It was enjoyable talking with them and sharing our different experiences.

And finally, my very special gratitude goes to my family who kept surrounding me with so much love and support throughout my life. Making them proud and happy is my ultimate goal. I am therefore even more blessed to have a new addition to the family, my fiancé Andrej, to whom I am ultimately grateful for being there with me not only to share my enthusiasm but also to be my support during stressful periods as well.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to drug dissolution modelling

Drug delivery systems (DDS) are systems for transporting drugs into the body. The main components of a typical DDS include one or more active agents (i.e. drugs) and one or more polymers forming the vehicle by which the agents are delivered to the desired area of the body. The behaviour and breakdown of those polymers, as they traverse the route of administration (e.g. parenteral, transdermal, oral, through the nervous system, etc.), influence the time to, and the area of, delivery of the active substances. The intricacies of drug development require collaboration between scientists of different expertise, notably, pharmaceutical scientists, bioengineers, computer scientists and others, because there is a growing need to bring the drug to market faster while maintaining safety of treatment. At the same time, complex factors that influence drug pharmacokinetics and dissolution must be taken into account.

The computer modelling of a DDS is a constantly developing field with potential to become an integral part of pharmaceutical research. *In silico* simulation of DDS can be useful for a number of reasons; specifically, (i) it has the potential to reduce the cost of experiment by reducing the amount of *in vitro* testing needed, (ii) to reduce the time needed for bringing the drug to market, (iii) to increase performance during the design phase by allowing more complex analysis, (iv) to provide better understanding of drug transport processes and (v) to help identify the required DDS composition and manufacturing procedure. Drug

dissolution modelling can provide good predictive capability, and improved agreement with experiment and can help in achieving the desired drug release profile.

A benefit of modelling is that it can be used to explore the influence of many parameters, such as dissolution device[1], geometry, dimension and composition during the drug design phase and well before the *in vivo* phase of drug testing.

This is of particular importance as pharmaceutical companies today face a growing demand for more complex drug designs. Novel formulations are required to reduce side effects, such as unnecessary absorption, while improving the dosage administered, through controlled and targeted release. Knowledge about drug release kinetics from such formulations is limited, due to complex interactions between different underlying phenomena and the environment, often known only through their combined effects. This makes computer models essential tools in the design of experiments. Unlike classical differential-equation models, stochastic modelling is applicable to a wide range of systems without requiring detailed initial knowledge on dissolution mechanisms.

1.2 Controlled DDS Examples

Controlled DDS in general can be classified into reservoir and matrix systems. *Reservoir systems* are defined by utilising one or more coating polymers to surround a core of active substance, which acts as a release-rate controlling material. *Matrix systems*, also known as *monolithic* systems, are characterised by a structural network consisting of a mixture of polymers and drugs, (Siepmann and Siepmann, 2012a).

Each of these groups can have various different geometries and various release scenarios and therefore very different complexities. Figure (1.1) illustrates the amount of possible combinations for diffusion controlled DDS. It can be observed that systems are distinguished depending on their inner structure and amount of initial drug concentration, together with its solubility and dispersion, resulting in a relationship which is not always entirely understood.

[1]The term *device* is a generic term used here for any kind of drug formulation, including capsules, tablets, spheres etc. It refers to the formulation consisting of an active substance (the drug) and various carriers (usually polymers).

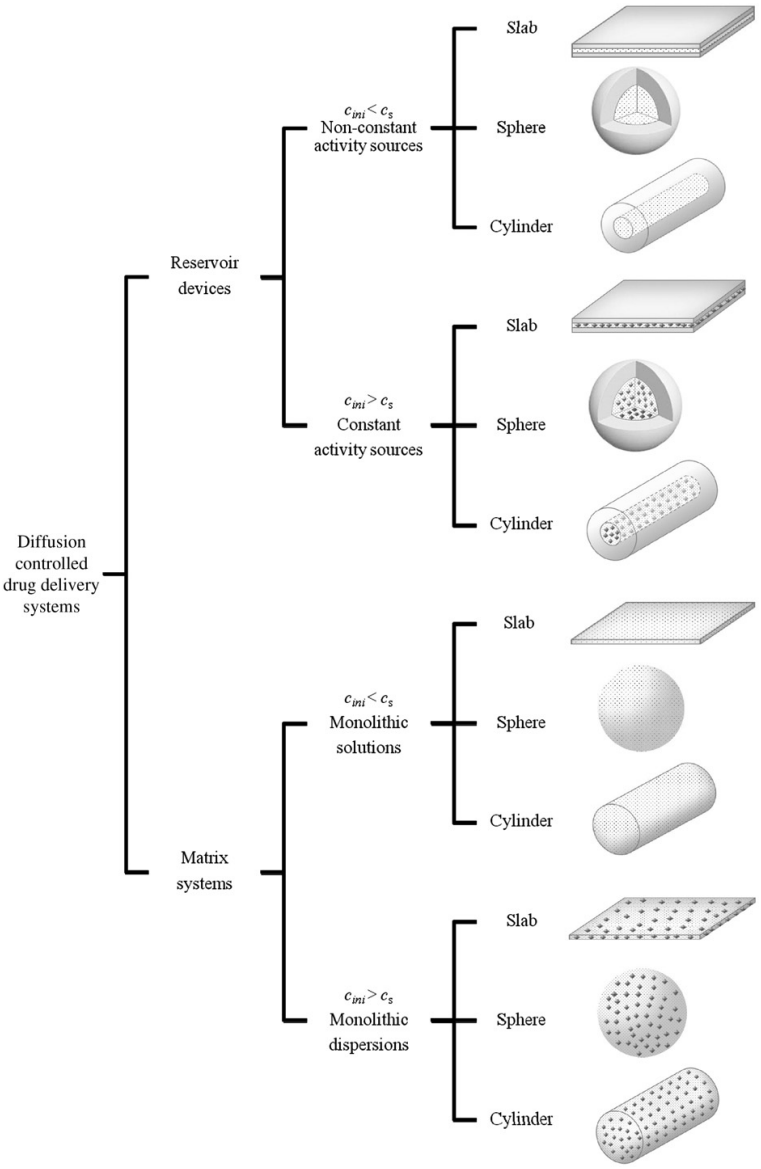


Figure 1.1: An example of DDS classification for diffusion controlled DDS, (Siepmann and Siepmann, 2012a) described by the ratio of initial (c_{ini}) vs saturation concentration (c_s) of the active substance.

1.3 Thesis scope and contribution

A detailed overview of the status of DDS modelling is given in Chapter 2, which discusses the main classification of models and analyses key examples from the literature. The chapter also includes an explanation of the main pharmaceutical phenomena, (as these will be discussed throughout the thesis), together with the role of *in vitro* testing in understanding these. We describe the modelling of polymeric behaviour in general and, specifically, in the the gastro-intestinal tract, as this plays a central role in controlled drug delivery systems.

In Chapter 3 we cover probabilistic modelling methodology in detail, as this will be the main focus of the thesis. We introduce key terms and application of Cellular Automata, Monte Carlo methods, (both in *direct* and *inverse* variants) and Agent-Based modelling. At the end of the chapter, we give an overview of the project performed in collaboration with the industrial partner which served as initial motivation and reference for this work.

Chapter 4 introduces a novel Cellular Automata based integrative framework and a meta-model used to create and analyse a series of stochastic models applied to different controlled release DDS. The meta-model introduces common features which capture the essence of controlled drug delivery and serve as a basis for deriving the intrinsic models. The framework allows for easy parameterisation, separation of release mechanisms or their superposition, (whichever is of interest), graphical visualisation and high performance execution. Several models are constructed and described in detail, covering the main pharmaceutical phenomena, namely erosion, swelling and diffusion in the presence of single and multiple polymeric coatings. The influence of the environment on release is also investigated.

In Chapter 5 we discuss computational and implementation aspects of the framework itself. We introduce, for the first time, a discussion of parallelisation strategies in the context of pharmaceutical models and compare several viable solutions that can be applied depending on the available computational power. As an addendum, we compare the advantages and disadvantages of synchronous and asynchronous Cellular Automata implementation, something that has not been addressed previously for pharmaceutical models.

Chapter 6 presents a detailed high-level analysis of the data, obtained from the various models described in Chapter 4. Each model was validated against experimental data, which

served as a test case for performance and predictive potential.

Chapter 7 introduces a different approach to stochastic modelling, which uses Inverse Monte Carlo methods to "reverse engineer" unknown model parameters from (generally) noisy experimental data. This approach is then integrated with the previous framework and models, connecting all stages of the stochastic modelling process, allowing for its use in concrete industrial applications.

Finally, we present the main conclusions for the work and outline possible future directions, improvements and developments in Chapter 8.

The appendices include the additional material, including UML diagrams and description of the underlying code, glossary of abbreviations, definitions and terms, and a list of publications with selected abstracts, Appendices A - D.

Chapter 2

Literature review: Pharmaceutical and Modelling background

In this chapter we present the fundamental theories used in drug dissolution modelling and review the various approaches available. Representative equations and theory, described here, have been a basis for all main DDS modelling approaches to date. The focus of this thesis is in building *stochastic models* for *controlled* drug delivery, (where polymers play an important role), targeting the gastro-intestinal tract. In consequence, the chapter also reviews models for polymer behaviour in DDS and gives an overview of the main phenomena involved in simulating dissolution in the gastro-intestinal tract.

2.1 Important phenomena in drug dissolution

In order to choose or develop an appropriate release model, it is of fundamental importance to understand the underlying mechanisms of release, (Kaunisto et al., 2011). Therefore, before reviewing DDS modelling itself, we provide short explanations of the key physical phenomena involved:

1. **Diffusion** represents the motion of the molecules from a region of higher concentration to one of lower concentration (i.e. flux). In modelling DDS, diffusion is often described

by Fick's laws, (Higuchi, 1960). In one dimension, Fick's first law can be written as:

$$J = -D\frac{\partial c(x,t)}{\partial x} \tag{2.1}$$

where J is the diffusion flux, D is the constant diffusion coefficient and $\partial c(x,t)/\partial x$ is the concentration gradient.

Fick's second law predicts how diffusion causes the concentration to change with time:

$$\frac{\partial c(x,t)}{\partial t} = D\frac{\partial^2 c(x,t)}{\partial x^2} \tag{2.2}$$

To calculate diffusion mass transport processes in modelling DDS, Fick's second law is used in different forms, depending on the geometry of the device, (e.g. Muschert et al. (2009), Cuppok et al. (2011), Kreye et al. (2011), Seidenberger et al. (2011)). A finite-difference approximation for determining the drug mass transfer rate, from cylindrical components consisting of multi-layers, was applied, (McMahon et al., 2003, 2007), and (McMahon, 2008).

Mathematical solutions of the diffusion equations, including detailed explanations of diffusion processes in solutions, have previous been derived, (Crank, 1975) and have been used as the basis for similar scientific problems. This work is particularly relevant to the area of drug dissolution, given that the author investigated different geometries including plane-sheets, cylinders and sphere and examined both Fickian and Non-Fickian diffusion.

2. **Advection** in its general sense represents transfer of material from one region to another due to the bulk motion of the surrounding fluid. In drug release, it is a second possible mechanism involved in mass transport, (in addition to diffusion), and refers to transport with specified velocity along the surface of the drug device, (Crane et al., 2004b).

3. **Degradation** is one of the two most important processes in dissolution of polymeric drug delivery systems. A widely used definition states that it is the process by

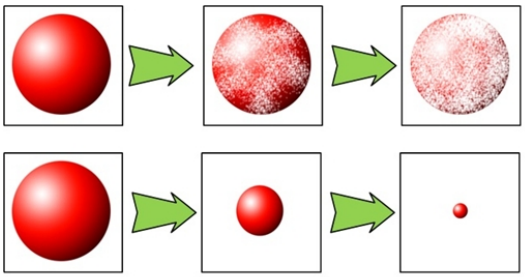


Figure 2.1: Bulk erosion vs. surface erosion.

which chain scission occurs, (during which polymer chains become monomers[1] and oligomers[2]). Monomers and oligomers have the ability to break down and therefore dissolve easily, (Göpferich, 1996).

4. **Erosion** is a consequence of degradation and represents the loss of material from a polymer. It is an important parameter because it determines the release rate of the drug, (Siepmann and Göpferich, 2001), (Lao et al., 2011). When a polymer erodes it leaves space for the drug to be released from the device or for water ingress. Two types of erosion are defined, (Langer and Peppas, 1983), (von Burkersroda et al., 2002):

   Surface erosion is a homogeneous process and represents the stage during which the size of the device decreases while preserving its shape; the device loses material only from its surface;

   Bulk erosion is a heterogeneous process, with material being lost from the whole device, although the device dimensions remain unchanged as the polymer erosion occurs throughout, (Figure (2.1)). For instance, surface eroding polymers like PLA, (polylactide), are often used with a co-polymer such as PLGA, (polylactic-co-glycolic acid), to achieve bulk erosion. PLGA is a bioerodible polymer and is therefore often used in controlled release.

   In general, it is more complex to model erosion than degradation (Siepmann and Göpferich, 2001), since there is a long list of possible factors that can influence the

[1] Monomers are the simplest units of polymer. A number of monomers form complex networks of polymer chains.

[2] Oligomers are simple molecules, formed of a few monomers only.

former, (Göpferich and Langer, 1993, 1995a), (Siepmann and Göpferich, 2001); these include:

- the nature of the chemical bonds - for example, poly(anhydrides) have highly reactive bonds and a short half-life so that this determines velocity of degradation
- pH changes
- polymer chain length
- bond cleavage velocity
- swellability
- crystallinity
- water diffusivity in the polymer matrix
- composition of the co-polymers - affects the velocity of release rate
- water uptake - influences the speeding of hydrolysis.

In the work presented in this thesis, both bulk and surface erosion play a role in controlled drug delivery within the gastro-intestinal tract and are modelled explicitly.

5. **Swelling** represents a process driven by the disentanglement and diffusion of individual polymer chains from non-hydrated polymer material, caused by solvent intake (Braido, 2011),(Kimber et al., 2012). It occurs as a response to changes in environment acidity or temperature conditions, (Gehrke and Cussler, 1989). During this process, polymer chains transition from a state of low chain motion (glassy) to one of a higher motion (rubbery). This occurs until thermodynamic equilibrium (relaxation) of the chains is achieved (Lee and Peppas, 1987). Long polymer chains disentangle and detach outwards from the main mass, causing a natural concentration gradient to form from the inside of the device outwards. At the very end of the gradient, polymer erosion becomes a dominant factor as polymer concentration is very low. This process essentially characterises swelling as a polymer diffusion process, not unlike drug diffusion.

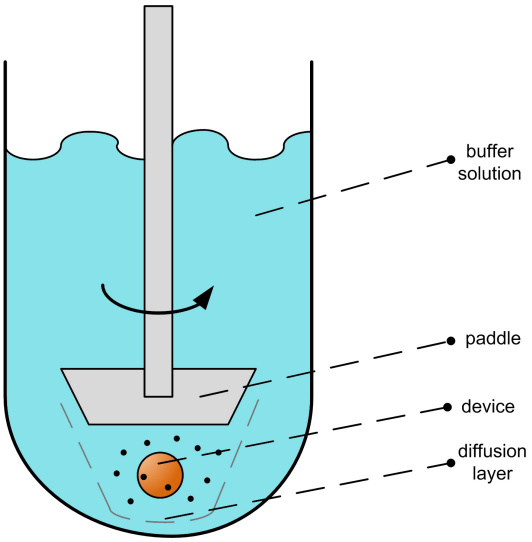


Figure 2.2: Schematic representation of the USP II Paddle Apparatus.

2.2 *In vitro* dissolution testing

The first step in modelling dissolution of a drug is to determine the *in vitro* dissolution rate under various external conditions. To emulate these conditions, a dissolution test apparatus is used, with standardised measuring methods outlined in the pharmacopoeias, United States Pharmacopeia (The United States Pharmacopeial Convention, 2007), and European Pharmacopoeia (Council of Europe, 2007). Conditions in the intestine and stomach, (change of temperature, pH and dynamic flux), are mimicked in the paddle apparatus II, (USP II), by varying either the speed of the paddle, or the physico-chemical properties of the buffer and by monitoring their influence on the drug dissolution rate, (Figure (2.2)). Whilst these instruments produce overall release profiles, they do not give good insight into the underlying physical properties and processes within the device, (coating, filling etc.), that determine the dissolution characteristics, (Kimber et al., 2011b). Diffusion layer thickness, (the region immediately surrounding the drug device), in the USP II apparatus is important as not only diffusion but also advection can have significant impact on velocity in the boundary layer, (McMahon et al., 2003), (Crane et al., 2004a), (Crane et al., 2004b).

2.3 Fundamental modelling methods in DDS

Existing DDS models can be classified into three broad categories: The first two are based on a *top-down* approach where the main underlying phenomena must be known in some detail. Both *mechanistic* and *empirical models* are included in this category. Models in the third category are stochastic and simulate the *probabilistic* behaviour of individual particles in the system with system effects dependant on aggregation.

Modelling drug dissolution is a topic which has attracted attention for more than a century, so that various alternatives exist; it is necessary to define the area of interest more precisely, in order to provide a focused review of relevant developments. In the following sections, therefore, we describe the main equations and models used in complex drug delivery. For additional information on possible models, a number of reviews have appeared in the literature over the last ten years or so, (e.g. Narasimhan (2001), Siepmann and Göpferich (2001), Siepmann and Peppas (2001), Costa and Lobo (2001), Grassi and Grassi (2005), Barat et al. (2006a,b), Siepmann and Siepmann (2008, 2012a), Sackett and Narasimhan (2011), Lao et al. (2011), Kaunisto et al. (2011), and from these different classifications of DDS models can be summarised.

2.4 Mechanistic and empirical models

Mechanistic, sometimes called "phenomenological models" (Sackett and Narasimhan, 2011), use differential equations for explaining dissolution processes, providing some insight into their nature at a lab-based level. The main characteristics and advantages of such models is that they enable investigation of the physical mechanisms influencing release and, by varying the relevant parameters, can be used not only to mimic experimental conditions but also as *predictive tools*. In order to validate such predictions, two options are possible: (i) comparison of modelling results with experimental data and (ii) validation of all parameter values included in the model, (as these have physical significance, (Kaunisto et al., 2011). Another important advantage of using these models is that good matches with experiment can be obtained since equations are developed to describe accurately the underlying phenomena for a specific drug formulation. However, this is also the main

disadvantage as considerable initial knowledge of the important properties is required either from the manufacturing process itself or to be determined by calculation. Therefore, a major consideration is to determine the appropriate level of complexity to be reproduced, in relation to the main release rate-limiting processes, (such as the boundary conditions and the main mass transport mechanism). Defining models, which accommodate these requirements is non-trivial, particularly where multiple-parameterisation and fitting are essential.

Newer pharmaceutical compounds have even more complex behaviour, (Barat et al., 2006b), so that defining all processes involved is extremely difficult, particularly if knowledge of the drug delivery system is incomplete, as is often the case. It is sometimes more appropriate to use *empirical models* that do not reflect all possible processes and therefore are easier to implement, while still giving good prediction for drug release. This parsimonious approach to modelling the complex behaviour has obvious attractions. Such models usually assume *zero-order release kinetics*[3], where zero-order release can involve superposition of various mass transport processes, such as diffusion of liquid/drug, polymer swelling or polymer degradation, (Siepmann and Göpferich, 2001), without requiring exact information on the way in which each of these influence the release. By paying careful attention to the assumptions made in these models, sufficient information is obtained for generation of comparative release curves from different laboratory designs, (Siepmann and Siepmann, 2008). The main disadvantage is limited applicability, i.e. models are based on experimental behaviour of the particular system studied, so cannot be used to predict the effects of hypothetical changes on the system behaviour, e.g. influence of coating thickness on the drug release rate, (Kaunisto et al., 2011).

2.4.1 The principal classical equations

Noyes-Whitney equation

The earliest mathematical models, mechanistic in nature, used to describe system behaviour in drug dissolution research appeared more than a century ago, (Noyes and Whitney,

[3]Zero-order release kinetics implies that the release rate is steady with time and is independent of the concentration of drug. Increasing the concentration will not speed up the rate of the reaction.

1897), with the derivation of an equation to describe the dissolution of multiparticle systems, (powders). The authors discovered the dissolution rate of a solid in solution to be proportional to the difference between the current concentration of the solution and the maximal concentration of the saturated solution:

$$\frac{dc}{dt} = k_{NW}(C_s - c), \tag{2.3}$$

where C_s is the saturation concentration, c is the concentration of the solute at time t and k_{NW} is a constant. Since this equation follows first order kinetics[4] k_{NW} is considered to be a first order proportionality constant.

Later models emphasised the assumptions for which the Noyes-Whitney equation (N-W) is applicable namely (i) a constant area available for dissolution and (ii) a constant and intense rate of stirring, (Hixson and Crowell, 1931). The N-W equation is used widely in many applications, particularly to investigate the influence of the particle size distribution on dissolution profiles, (Hixson and Crowell, 1931), (Higuchi and Hiestand, 1963) where the complexity of the particle size effect is emphasised. In the analysis of polydisperse multi-sized systems, (using complex reverse engineering techniques and taking into account variations of the number of particles), dissolution profiles were found to be strongly influenced by particle size distribution, (Simões et al., 1996), (de Almeida et al., 1997). An estimation of particle size from multi-sized powder dissolution was published recently, (Avdeef et al., 2009) for two scenarios in which particle size either changes or remains constant during dissolution. A good historical review of applications of the Noyes-Whitney is given by Dokoumetzidis and Macheras (2006).

Higuchi equation

A second important equation, empirical in nature, and used to describe controlled release is due to Higuchi (1961). Considered "the father for a mechanistic understanding of controlled DDS", (Siepmann and Peppas, 2011), Higuchi derived an equation, which follows Fick's first law and relates diffusion-controlled release of drugs from non-swellable and

[4]First order kinetics describes a release proportional to the concentration of active substance.

non-biodegradable films under perfect sink conditions, to the diffusivity and solubility[5] of a drug and its initial concentration. With increasing drug solubility, drug release occurs as a linear function of the square root of time:

$$M_t = k_H \sqrt{t} \tag{2.4}$$

where M_t is the cumulative amount of drug released from the ointment film during time interval t and where constant k_H has a specific meaning and should not be ignored, i.e.

$$k_H = A\sqrt{2 c_{ini} D C_s} \tag{2.5}$$

where c_{ini} is the initial drug concentration and A is the surface area available for dissolution.

The Higuchi equation must be used with caution, however, since several assumptions are made: (i) the initial drug concentration in the system is much higher than the drug solubility, $c_{ini} >> C_s$; (ii) the diffusivity of the drug is constant; (iii) no swelling occurs; (iv) perfect sink conditions apply; (v) the geometry is thin film. In consequence, the equation cannot be applied readily to systems with complex release, e.g. in describing biphasic release from coated formulations, (Siepmann and Siepmann, 2008), (Siepmann and Peppas, 2011).

Peppas equation

The last of our general equation was derived by Korsmeyer et al. (1983) and Ritger and Peppas (1987). Adapting the Higuchi equation, the authors provided a generalisation, which permits application to any geometry, (thin films, cylinders and spheres) and enables the dominant nature of the release process, (Fickian or non-Fickian/anomalous), to be determined. This adaptation represents the fraction of drug release M_t/M_∞ as a simple

[5]Solubility is a property of a substance (the solute, e.g. drug) to be dissolved within another substance (the solvent). The amount of drug which can be dissolved in some volume is defined by the saturation concentration (Cs). This means that further increase of drug concentration will not result in faster drug dissolution for given volume.

Value of n for different geometries			Drug release mechanism
Thin film	**Thin Cylinder**	**Sphere**	
$n = 0.5$	$n = 0.45$	$n = 0.43$	Fickian transport: diffusion is the leading process
$0.5 < n < 1.0$	$0.45 < n < 0.89$	$0.43 < n < 0.85$	Non-Fickian transport, anomalous behaviour: interference of more than one mass release mechanism
$n = 1.0$	$n = 0.89$	$n = 0.85$	Zero-order release

Table 2.1: Values for exponent n in the Peppas equation for different geometries and analogous drug release characteristics in polymeric controlled delivery systems.

power-law equation:

$$\frac{M_t}{M_\infty} = k_P t^n \tag{2.6}$$

where M_t and M_∞ represent the amounts of drug released at time t and the total amount of drug contained in the DDS, respectively, k_P is an experimentally determined parameter and n is an exponent that depends on the system geometry and the mass transport mechanisms, (as represented in Figure (2.1)). It can be seen from Table 2.1 that if the exponent n, for the case of spherical geometries for instance, has a value of 0.43, Fickian diffusion is the leading transport process, while a value of 0.85 refers to zero-order release, i.e. concentration independent drug release. Values larger than 0.85 indicate the erosion controlled (also referred to as *Case II*) transport, (Kosmidis et al., 2003b).

The Peppas generalisation is used to describe the drug release mechanism and provides guidance on which model to use, with many authors using the *power-law* form as the basic equation. For example, Kosmidis et al. (2003b) used this generalisation to analyse radial and axial release from cylinder geometries, while Casas et al. (2010) analysed the effect of shape on drug release. It became the standard method of analysis of any pharmaceutical device. In recent work, swellable systems were modelled using a probabilistic CA approach, (Laaksonen et al., 2009a) and Peppas equation was used to estimate the linearity of the release for drug fractions. Even though the equation cannot fully explain all swellable

systems of interest, it has proved very useful in the investigation of complex formulations, where adequate experimental data are not available. In this way, the Peppas exponent n has been used to explain the trends in different release curve phases, i.e. whether these are of zero-order or anomalous, (typical for swellable systems), Laaksonen et al. (2009a).

2.5 Probabilistic models

With the development of high-performance computers, a new approach to modelling DDS was introduced, namely probabilistic models. While these can be both mechanistic and empirical in nature, they have the advantage of using a *bottom-up* approach, simplifying representation of the system and looking at microscopic rather than macroscopic behaviour. Probabilistic models use statistical techniques such as Monte Carlo (MC) and Cellular Automata (CA) to describe drug release properties. Moreover, these approaches are flexible, since differential equations may also be used to define specific elements of release phenomena, but with stochastic features incorporated, (Siepmann et al., 2002).

The basic premise of probabilistic models is the assumption that we cannot always determine the precise parameter values when modelling complex systems, and that the outcomes of individual system reactions can follow stochastic, as well as deterministic rules. Such models are thus applicable to systems where: (i) the complexity of the modelled device prohibits usage of differential equations due to the inherent unknowns of many-element interactions; (ii) where the design formulation of the complex device is undetermined (most importantly, during the wet-lab experimental stage) making derivation of analytical solutions prohibitively costly in terms of time. In addition to this, probabilistic methods were shown not to lack both precision and correctness in prediction, (Zygourakis, 1990), (Barat et al., 2006a), (Laaksonen et al., 2009a) and thus form a promising alternative to traditional DDS modelling.

We cover the current state of probabilistic modelling of DDS based on MC and CA approaches in detail in Chapter 3.

Designing the parameters in DDS using Weibull function

It is useful also to mention a class of probabilistic dissolution models, which are not based on discrete space simulations using MC and CA algorithms. One interesting group in particular are those which utilise the Weibull function, since this function can be applied to almost all kinds of dissolution and related release curves, (Costa and Lobo, 2001), (Martínez et al., 2009). In investigating drug release for which diffusion is the dominant mechanisms, Kosmidis et al. (2003a) used the form:

$$\frac{M_t}{M_\infty} = 1 - e^{(-at^b)} \tag{2.7}$$

where M_t/M_∞ is defined as for Equation 2.6 and a, b are Weibull parameters.

Starting from the Higuchi equation, the authors linked the Weibull parameters to physical properties of the drug device, showing that the *scale parameter*, a, defines the time scale of the process and depends on the diffusion coefficient, while the *shape parameter*, b has a constant value and characterises the curve, (Kosmidis and Macheras, 2007, 2008), (Villalobos et al., 2009). Furthermore, coefficient b acts as an indicator of the transport mechanism, such that its value $b < 1$, $b = 1$ or $b > 1$ implies diffusional release, (parabolic curve), first order release, (exponential curve), or complex release, (sigmoidal curve), respectively, (Papadopoulou et al., 2006).

2.6 Modelling polymer behaviour in DDS

The primary objective of controlled release devices is to achieve drug release at a desired and sustained rate. These are the most common type of systems with *targeted delivery* being the main benefit, (e.g. in targeting the gastrointestinal tract, we usually do not want the drug to be released too soon after the oral intake, but only at the target site). Controlled drug dissolution is maintained by using polymers with different structures. In Peppas and Langer (1994), the basic principles underlying the usage of biomaterials (such as polymers) in pharmaceutical preparation are outlined.

One of the most important problems to solve in order to achieve controlled drug release, therefore, is that of polymer dissolution. In large part, modelling drug dissolution aims to

understand and predict the physics of drug release from a dissolving polymer, where these are commonly-used materials, due to their biodegradability.

A first step to understanding controlled drug delivery is to review the most important phenomena driving release, (previously defined in the Section 2.1), and their influence on polymers of varied complexity. The main modelling approaches for these fundamental phenomena are outlined. The connections between models of different types (i.e. top-down and bottom-up) are also important as the underlying physical phenomena share similar features, and knowledge obtained using models of one type can be used to derive rules for another. For example, for the simple probabilistic model of diffusion we use Fick's law, which also defines the interaction between individual elements in Monte Carlo models.

The rate of drug release from the polymeric carrier is important as it must be kept in the therapeutic range, (Sackett and Narasimhan, 2011).

Glassy and rubbery polymers

A useful division of polymers identifies two types based on their dissolution behaviour: *glassy* and *rubbery* (Crank, 1975), (Lee and Peppas, 1987). The former have complex, non-Fickian, "anomalous" behaviour where dissolution is time-dependent and swelling occurs as a consequence of penetrant ingress into the drug device. The latter, "rubbery" polymers, have Fickian behaviour, with diffusion as the main transport process. These rapidly respond to changes in the environment, such as a change of temperature, (Narasimhan, 2001). Classification of a given system is needed for the choice of an appropriate value for Peppas exponent n.

2.6.1 Polymer dissolution

Purely diffusion-controlled systems are the simplest to understand and therefore can be modelled by direct application of Fick's laws, Equation 2.1, Equation 2.2, (Siepmann and Siepmann, 2008). These can be applied to different DDS types, including both matrix (monolithic) and coated (reservoir) systems.

In general, there are two main differences between dissolution from polymers and non-polymeric materials: (i) Polymers require an *induction time* before dissolution can occur,

while with a non-polymer this can occur instantly; (ii) Polymer dissolution is controlled by its *disentanglement threshold*, a function of polymer chain length. An extremely high threshold leads to a large amount of water intake while low values imply immediate dissolution. Non-polymeric materials are in general controlled by external mass-transfer resistance through a liquid layer adjacent to the solid-liquid interface, (Narasimhan, 2001).

In *controlled release* devices, under assumptions that the coating does not swell or dissolve, drug permeability is constant and perfect sink conditions are maintained, with release dependant on the ratio between concentrations of solute and drug. For the case where drug concentration is higher, zero order kinetics can be used, otherwise first order kinetics apply, (Heller and Baker, 1980). However, with the need to take into account coating effects such as swelling and/or cracking due to polymer expansion, these simplifications are not realistic, and mechanistic models, which can appropriately describe this level of composite behaviour, are still lacking, (Siepmann and Siepmann, 2008).

2.6.2 Polymer erosion

In early empirical models, Hopfenberg (1976) derived an equation for the analysis of surface-eroding polymers. The author assumed that the drug release rate was proportional to the surface area of the device and, as such, can be calculated for any geometry. One conclusion that can be derived from this model was that only "slabs" have zero-order kinetics. Additional analysis of the geometry influence on erodible devices was due to Cooney (1972), who defined two phases of dissolution. One in which molecules detach from the surface and the other where diffusion into the fluid occurs. The release is described as dependent, in the case of cylinders, on an "initial length to initial diameter" ratio, where the value of this ratio defines the rate of release, (zero-order or otherwise). The Hopfenberg model allowed initial predictions for drug release kinetics and improved understanding of erodible systems, such as that described, (Zygourakis and Markenscoff, 1996). This led to looking at the dissolution of polymers as transport processes controlled by two phases, (Narasimhan, 2001), as described earlier, i.e.

- diffusion of the chains through a boundary layer at the solvent-polymer interface
- disentanglement of the polymer chains

Using the Higuchi equation as a starting point, Heller and Baker (1980) developed a mathematical model for combining erosion with diffusion while taking into consideration particular physiochemical phenomena. The authors assumed that degradation in polymeric matrices with bulk erosion follows first order kinetics. Typical examples of those are PLA and PLGA matrices. In addition to the purely diffusion-controlled Higuchi here, permeability in biodegradable polymeric systems is not constant but increases with erosion. As a consequence drug release initially decreases, (similar to Higuchi), but then increases, (due to increased permeability), (Siepmann and Göpferich, 2001).

Additional molecular phenomena and their effects on drug dissolution were taken into account in some studies, e.g. (Narasimhan and Peppas, 1997). These included chain disentanglement from an amorphous polymer, water diffusion into the polymer and hydrophilic drug release from spherical PLGA bulk-eroding microspheres. Furthermore, Batycky et al. (1997) described the phenomena of microsphere hydration and polymer erosion. Polymer degradation was represented as a combination of two processes - random and end-polymer scission and the conclusion was drawn that models, based on only one of these, cannot explain experimentally observed kinetics of particle mass loss and molecular weight change, so that a combined model was necessary. Finally, a model for predicting controlled release from both erosion and bulk matrices was developed, (Rothstein et al., 2009). Drug release was shown to depend on even more mechanisms, such as hydration of the system, formation of pores, and drug and polymer dissolution.

All these models have improved the understanding of the role of erosion, and its variations to some extent. Nevertheless, they are still limited and fail to include other important features such as: interaction between monomer-pore formation, protein-protein interaction, the influence of pH on polymer chain scission, and the role of polymer swelling among others, (Batycky et al., 1997).

2.6.3 Polymer erosion and diffusion

Importantly, for investigation of moving boundary problems for surface polymers, two dissolution fronts, namely the erosion and diffusion front were introduced, (Figure (2.3)), (Lee, 1980). Here, R refers to the time-dependent position of the moving diffusion front and S

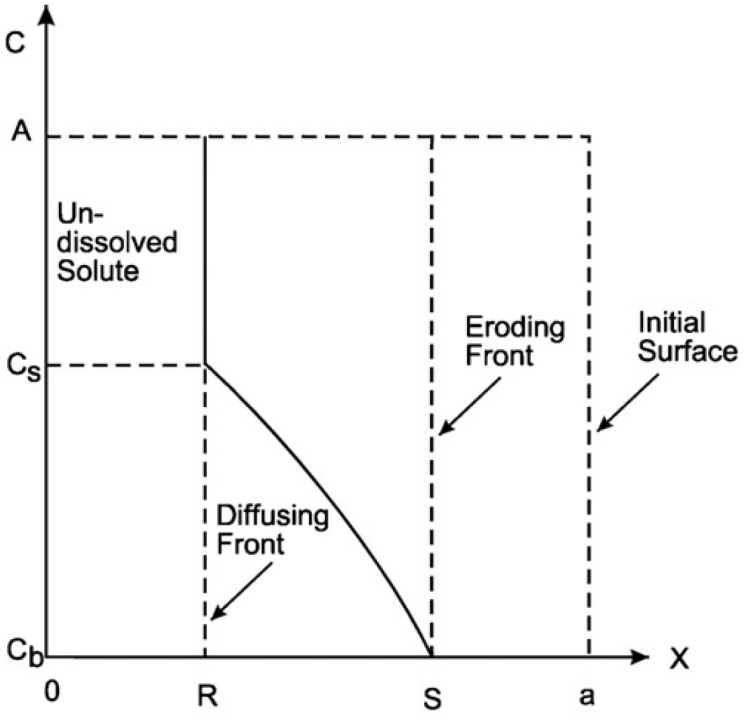


Figure 2.3: General schematic representation of concentration profile for surface-eroding polymers. Two key boundary layers are visible: *R(t)*, the time-dependent position of the moving diffusion front and *S(t)*, the time-dependent position of the eroding front, (Lee, 1980).

to the time-dependent position of the eroding front; A is the initial drug concentration within the polymer, C_s, drug solubility, (as defined previously), and C_b is the drug concentration of the well-stirred solution. Different ratios of A/C_s were investigated and led to the conclusion that increasing this ratio causes decrease in relative drug release rate. The release reaches zero-order for cases when the initial drug loading is much higher than the solubility. This concept of multiple fronts, including also one for swelling, have been in use in later models of dissolution dynamics, for example, Colombo et al. (2000).

2.6.4 Polymer swelling and polymer and drug dissolution

The term *swelling-controlled DDS* usually refers to systems where the swelling step is the only release rate-controlling phenomenon or, in the broader sense, to systems where other phenomena outlined above can play a role, (Siepmann and Siepmann, 2012b). The main idea of using swellable systems is to *control* release, particularly for the case of drugs with low diffusivity, (Arifin et al., 2006). These systems are often based on hydrophilic polymers,

which can adsorb large amounts of water and cause polymer disentanglement, for instance hydroxypropyl methylcellulose (HPMC) (Colombo et al., 2000), (Siepmann and Peppas, 2001), (Chirico et al., 2007), (Kimber et al., 2012).

The key features in polymer swelling are: the *dry, non-swollen* state (with a dense polymeric network, and restricted molecule mobility) and *relaxation of polymeric chains*, or *swollen state* (where molecule mobility and volume increase), (Figure (2.4)). These are fundamentally different states and should be modelled as such, (Siepmann and Siepmann, 2012b).

Following earlier matrix models, (Lee and Peppas, 1987), (Harland et al., 1988) of two phase release consisting of swelling and dissolution, Ju et al. (1995a) developed a mathematical model for drug and polymer release from HPCS matrices. The model gave a quantitative relationship between the polymer disentanglement concentration, $p_{p,dis}$, and equivalent molecular weight of polymer HPCS matrix, M_{eq} where, below the critical polymer concentration, polymer chains are detached from the matrix gel and display simultaneous swelling-dissolution behaviour. Also, a diffusion layer separating the matrix from the bulk solution was incorporated into the transport regime. The model predicts that overall tablet size and characteristic swelling time correlate qualitatively with $p_{p,dis}$. For determining the relation between $p_{p,dis}$ and molecular weight, a power-law was used and its advantages and limitations in this case were discussed. The results showed that drug release is more affected by polymer release than by polymer molecular weight, (Ju et al., 1995a,b).

Subsequently, a model describing the drug release from drug delivery systems consisting of an ensemble of drug-loaded crosslinked polymer particles was presented, (Grassi et al., 2000). As main factors affecting drug release, the following were accounted for: the particle size distribution, the physical state and the concentration profile of the drug inside the polymeric particles, the viscoelastic properties of the polymer-penetrant system and the dissolution-diffusion properties of the loaded drug. Two drugs, completely different in terms of dissolution behaviour, namely, MAP (medroxyprogesterone acetate) and TEM (Temazepam), were used to be crosslinked with polyvinylpyrrolidone (PVP) particles. Interesting use of the Weibull distribution was made to model particle size distribution, in the dry state (corresponding to initial powder conditions). The main novelty of the

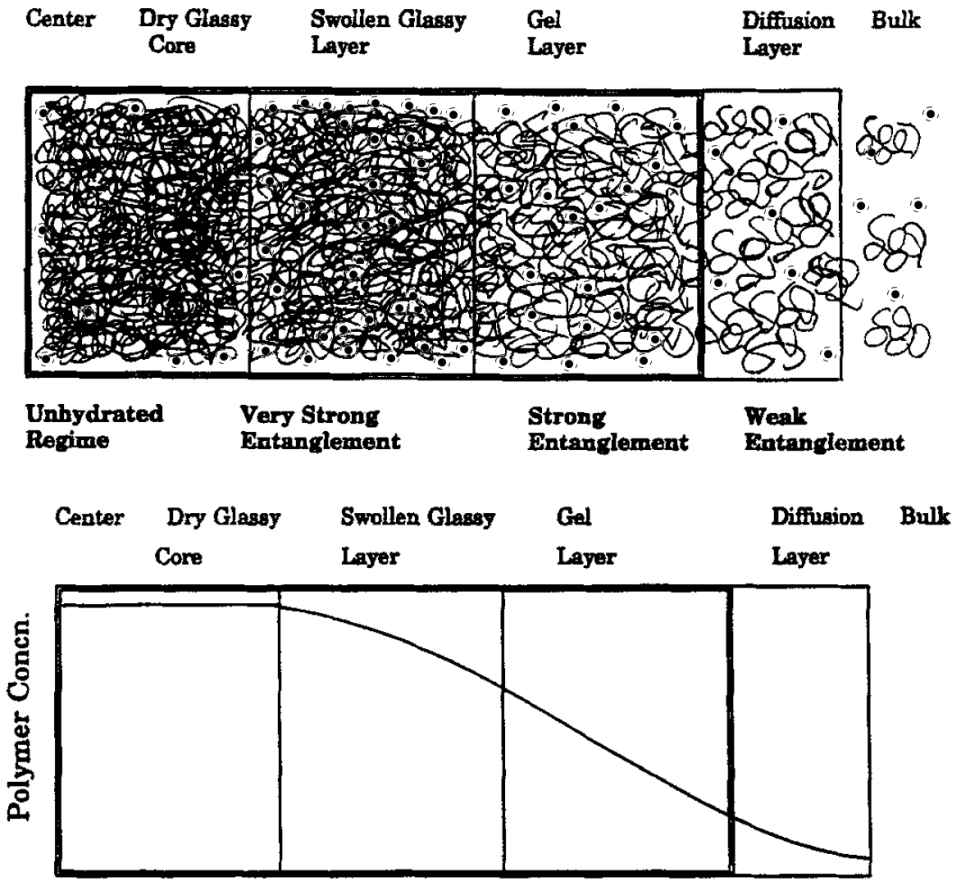


Figure 2.4: (Top) - Detailed schematic of polymer entanglement from *non-swollen* to *swollen* state. Within the *dry glassy* core, polymers exist in an unhydrated regime with dense network structure and low molecule mobility, (left). In the *swollen glassy* layer, solvent diffusion promotes water concentrations leading to a more mobile network, and very strong chain entanglement, (middle). As a result of significant swelling, fewer polymers are present in the gel layer, inducing less though still strong entanglement. Finally, in the water-rich diffusion layer, the chain entanglement becomes weak. At the gel-diffusion layer interface, chain entanglement can no longer hold polymers together causing polymer dissolution to take place, (right). *Black dots* represent drug particles diffusing from non-dissolved to highly dissolved state. (Bottom) - Polymer concentration profile equivalent to the top scheme. The space defined by the double lines represents the undissolved matrix. Adapted from (Ju et al., 1995a).

proposed model was in its versatility, as different situations were modelled, and the particle size distribution analysed. However, its main restriction lies in the fact that it is destined only for (chemically or physically) crosslinked polymer particles that do not undergo any kind of erosion.

Lee and Chakraborty (2002) used *kinetic* Monte Carlo simulation to study the dynamics of polymer chains as a consequence of random disordered media of monomer obstacles. The polymer was modelled as a chain of spherical beads within a 3D lattice divided into cells. The cell can be either occupied by a bead or not. At each MC step the algorithm is performed to check the chain formation and location of the beads. If a chosen bead is found to be at the end of a chain, the bond which connects it to its nearest first neighbour will be rotated to a new position through a randomly chosen angle. Otherwise, the bond will be rotated in a circular motion by rotating the bonds formed with the two neighbouring beads. Each movement can be accepted or rejected according to the Metropolis criterion with a probability ΔU which represents the difference in energy between the old and new chain. Authors looked at several classes of homo- and hetero-polymers and noted that, above a threshold temperature, polymers bearing monomers attracted to sites in a disordered medium and are more mobile than those with neutral or repulsive interactions. This study improved the understanding of polymer dynamics in disordered media.

In delayed release systems, as a consequence of water ingress, a polymeric coating can develop cracks due to the hydrostatic pressure build-up. This phenomenon, together with osmotic pumping, has been investigated in several studies, (Marucci et al., 2010), (Kaunisto et al., 2011) and is investigated in detail in Chapter 4.

Recently, an interesting novel algorithm was presented for simulating the radial swelling and dissolution of cylindrical tablets using the Discrete Element Method. Each particle was allowed to absorb water and swell, pushing against its neighbours and causing an overall expansion up to the disentanglement threshold. When that threshold is reached, the polymer dissolves and the particle decreases in size. Detailed parametric studies were performed to ascertain the main factors influencing polymer dissolution, such as the water diffusion coefficient, the dissolution rate constant and the disentanglement ratio. The model was validated against an exact numerical solution for simpler geometries but the additional

introduction of drug particles is not discussed, (Kimber et al., 2011a, 2012).

2.7 Models for simulation of drug release in gastro-intestinal (GI) tract

Our research focuses on developing stochastic models for simulating drug dissolution processes for controlled release formulations, predominantly those which target the GI tract. Consequently, we now discuss example models used in this context.

2.7.1 Effect of coating on the dissolution

For drugs, specifically targeting transit and dissolution in the gastro-intestinal tract, a planar tablet geometry in the form of coated drug pellets is often used, as it provides reliable results, compared to tablet or capsule forms.

Ethylcellulose(EC) is a frequently-used material for coating, as it represents a generally non-invasive polymer with good film-forming capabilities. It is both inert and hydrophobic, although able to expand as a consequence of water ingress when maintained under high humidity, (Geraghty, 2004). Its viscosity depends on its molecular weight during processing, while mechanical properties such as flexibility, elongation and tensile strength depend on the degree of polymerisation, (polymer molecular weight). Ethylcellulose is also stable up to 50 °C and, its properties, in general, make it ideal for use in matrix agents, both for prolonged release or as a coating material.

The influence of different drug types on the release profiles of EC-coated pellets was analysed, Sadeghi et al. (2003), while a model which considered constant and non-constant diffusivities using ethylcellulose-based mini-matrices was reported, Verhoeven et al. (2009). Further, to facilitate the prediction of drug release from coated tablets, (Muschert et al., 2009) formulated an analytical solution of the behaviour of ethylcellulose-coated thin-film drugs, using chemical and mathematical analysis. This could be used to speed up the development of DDS for the treatment of diseases of the GI tract by "minimising" the *in vitro* effort needed in laboratory trials. The authors showed that diffusion through the polymeric film coatings represents the sole uniform drug release mechanism, and fitted the

results of experimental trials to Fick's law equations. Furthermore, they managed to show that the mathematical solution is essentially one-dimensional, due to the dispersion of drug molecules within polymeric networks and the fact that the surface of the coating is very large, compared to the thickness of the tablet.

For other types of coatings, Ensslin et al. (2009) demonstrated the direct influence of film coating thickness and polymer blend ratio on drug release rates. They also gave a simple mathematical model that provides a way to calculate release rates from initially known coating parameters, although applicable to a certain type of drug pellets only. Other drug parameters that affect drug release, specifically in the GI tract, have been studied by authors such as Ahmed (2005) who investigated the effect of pH of gastric fluid on polymer swelling and hydration, and Leopold (1999) who gave an overview of the effect of different types of colon-specific polymers on the rates of swelling, dissolution and erosion. In Liu et al. (2007) the authors also provided a review for drug delivery mechanisms of the use of pectin, a polymer well-suited to the gastrointestinal tract due to its reaction with colonic bacteria.

Most recently, Loney and Susarla (2009) presented a mathematical model for diffusional release of spherical drug particles which incorporated the innovation of physiological characteristics in the simulation of the GI tract. These authors highlighted the need for justification of the assumptions of earlier models, in particular that of *perfect sink conditions* in a constantly *changing environment*, such as the GI tract. The developed model also accounts for the important role of adsorption.

2.8 Summary

To date many different theories have been performed to tackle different aspects of dissolution. The choice of adequate models for developing a particular DDS strongly depends on required accuracy and predictability. Most theories take into account only the most important properties of the system and are still not oriented toward routine usage by pharmaceutical scientists, (i.e. empirical models). Therefore, there is a need for optimisation of existing approaches and research on new ones.

Starting from this viewpoint, but with application to novel formulations in mind and in

an effort to improve realism, we describe additional molecular phenomena for inclusion in the models, described in Chapter 4.

Chapter 3

Probabilistic modelling: Methodology

This chapter reviews the main methodologies used in the thesis, namely Cellular Automata and Monte Carlo methods. It discusses the difference between Direct and Inverse Monte Carlo and reviews in more detail their use in the context of DDS modelling. As building blocks for controlled release in GI tract are reviewed, at the end of the chapter we present an overview and motivation for the work presented in the thesis.

3.1 Cellular Automata

In computation theory, cellular automata (CA) are important type of discrete model used in a number of scientific fields that consist of a regular n-dimensional grid of *cells*, each having one of a finite number of *states*. The state of each cell evolves over *discrete time*, starting from an initial state and then progressing through a series of generations, according to a fixed set of *rules* that take into account the state of the cell itself and the states of cells in its *neighbourhood*.

As a type of modelling tool, cellular automata were first discovered and described by Ulam and von Neumann in the 1940s, while working on models to describe the growth of crystals and calculate liquid motion, (von Neumann, 1966). The concept was to describe the liquid as a group of small, discrete units and derive the motion of each unit in the flow,

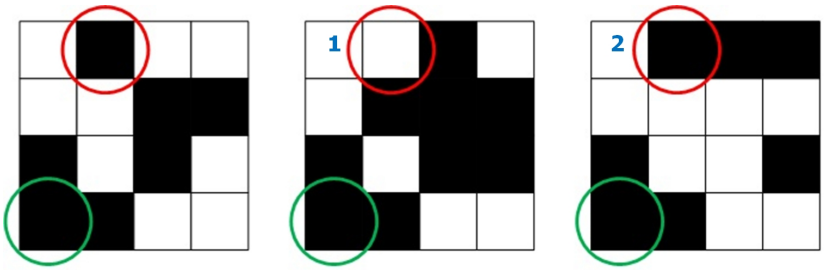


Figure 3.1: An example of Conway's "Game of life". Each cell can have one of two states, alive or dead, i.e. black or white, respectively. As an example of dynamics: If a cell has two live neighbours, its state remains unchanged, (green case). If a live cell has less than two live neighbours it dies, "starves" (red, case 1). If a dead cell has exactly three live neighbours it becomes live, (reproduction occurs), (red, case 2).

according to the behaviour of the neighbouring units.

Perhaps the best known example, frequently used is a two-dimensional, two-state cellular automaton created by Conway in the 1970s and named "Game of life", (Gardner, 1970). It consists of a 2D matrix of cells which can be either alive (black) or dead (white). If a cell has two living neighbours, its state does not change. If it has three living neighbours it is revised. In other situations, such as four or more living neighbours (overcrowding) or a single or no neighbours (starvation) it dies. Thus the set of states includes being alive or dead, and the set of rules includes various cell transition probabilities changing from one state to another, Figure (3.1). The initial state of the "Game of life" can be set arbitrarily, with certain initial configurations receiving wide coverage in scientific literature (Ninagawa et al., 1998), (Bosch, 2000), (Adachi et al., 2008), (Burguillo, 2013).

Due to the "bottom up" nature of the system description, CA make suitable and powerful tools for simulating the influence of the microscopic scale on macroscopic behaviour of complex systems (Hoekstra et al., 2008), (Désérable et al., 2011). Thus, they have a wide application in scientific fields including traffic modelling (Burstedde et al., 2001; Korcek et al., 2011; Vasic and Ruskin, 2012), bacteria growth (Margenstern, 2011), fluid dynamics (Leon et al., 2011; Désérable et al., 2011), modelling of tumours (Patanarapeelert et al., 2011), gas lattice dynamics and many others.

Cellular Automata can be divided into categories based on a number of properties, with fundamental distinctions being made by the type of neighbourhood and nature of the rule

set itself. There are two classical neighbourhood types of interest in multi-dimensional CA (Figure (3.2)): von Neumann, which considers the orthogonally adjacent cells only (4 in the case of two-dimensional automata, and 6 the in case of three dimensions), and Moore, which considers the entire adjacency lattice (so both orthogonal and diagonal neighbours - 8 for two-dimensional automata, and 26 for three dimensions). While the Moore neighbourhood potentially offers more realistic interactions (the CA applied to physical systems for example) it requires more computations to evaluate the target state of the cell. Before the era of high performance computing, this posed a disadvantage in applications to complex systems, so von Neumann was generally preferred. However, with today's HPC clusters this is no longer crucial to the decision of which to use, with choice governed instead by the physical realism of the problem space to be modelled.

In terms of other possible multi-dimensional neighbourhoods that could be used to describe the space of a physical system, it is worth noting the Margolus neighbourhood which considers the cells of a 2D or 3D lattice divided into 2^2 or 2^3 cell blocks, (Toffoli and Margolus, 1987). Rules are applied locally, and the dynamics of the system are achieved by shifting the partitioned grid itself (i.e. "cells" contain different "blocks" in each time step), (Figure 3.2) (bottom).

Another useful division of CA concerns the nature of the state transition rules. *Deterministic* CA employ rules which are fixed and do not change over time (as in the "Game of life" for example). In contrast *probabilistic* Cellular Automata rules may or may not be applied, depending on random number generation, and relevant for example, to situations where knowledge of the real system is incomplete. Stochastic CA properties are particularly useful for drug dissolution modelling and will be extensively covered in later chapters.

A final categorisation of CA type is based on the order and nature of cell state updates. *Synchronous* Cellular Automata (SCA) apply transitions in an atomic way by calculating for the current iteration and applying changes in the next. Thus the state of the system is not dependent on the order of updates. In contrast, *asynchronous* Cellular Automata (ACA) apply updates in the current iteration, so that other cells are aware of the neighbour state change immediately. A detailed analysis of asynchronous automata and discussion of their advantages and disadvantages compared to their synchronous counterparts is presented in

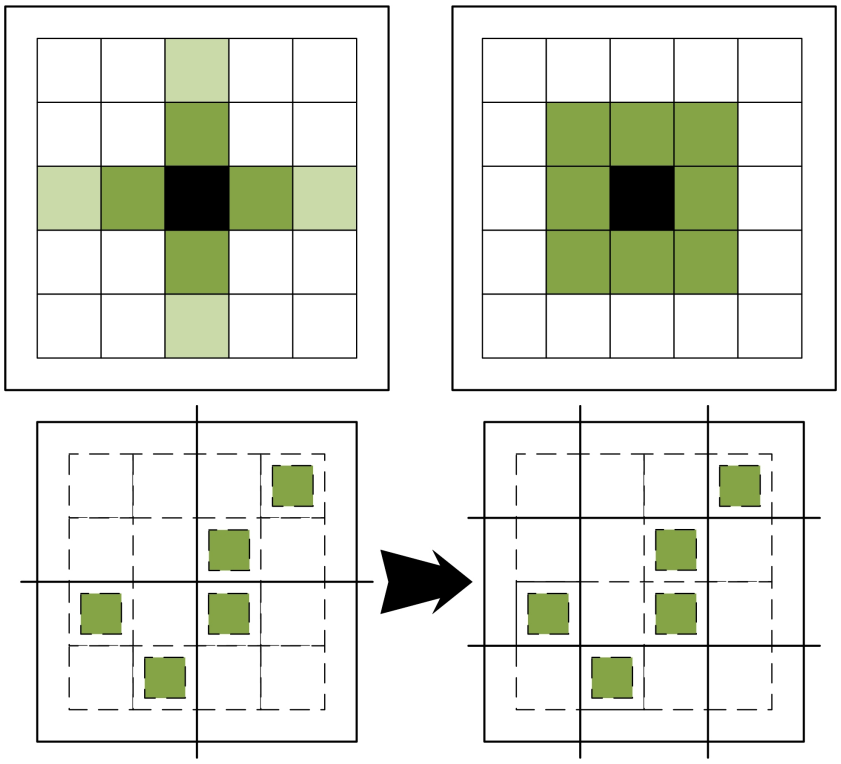


Figure 3.2: Different variations of CA neighbourhoods for 2D matrix. (Top left) Extended von Neumann neighbourhood, (Top right) Moore neighbourhood, (Bottom) Margolus neighbourhood, illustrating cell repartitioning to contain different blocks.

3.2 Agent-Based Modelling

Agent-Based Models (ABM, also referred to as *multi-agent systems*) present a natural extension of the cellular automata approach. CA can be viewed as a way to model physical system dynamics by "passively" propagating a change through the modelled system by influencing a neighbouring cell state. Agents provide "active" dynamism, representing independent physical entities that are able to move from cell to cell in a single iteration and are thus better suited for describing active processes, such as drug diffusion. Including a new feature to an existing CA model, however, introduces additional computational requirements, and requires good code optimisation and, usually, parallel re-implementation of the model (Perrin, 2008). The gains, however, are significant. The resulting models permit large scale simulations and inclusion of various localised effects, such as those found in the gastro-intestinal tract.

ABM is a model in which the key unit of abstraction is an *independent entity* (an agent) that has both spatial and temporal positioning. The generally-accepted properties for an intelligent agent, (there being no unique definition), are given by Wooldridge and Jennings (1995):

- *Autonomy*: acting without intervention with some control over its actions and the internal state.
- *Social behaviour*: interacting with other agents through a specific language.
- *Reactivity*: ability to see part of its environment and change its behaviour according to the environment state.
- *Proactivity*: not only reactive to the environment changes but capable of acting itself and taking the initiative, in order to satisfy identified goals.

ABMs, like CA, provide a generic enough paradigm to be used in a wide variety of fields. Notable examples include economics modelling (Holland and Miller, 1991), traffic congestion (Wen, 2008), biomedical systems such as the human immune system (Kim, 2009),

social and other networks (Ojanen et al., 2010), organisational structures (An, 2012) and social behaviour (Hughes et al., 2012), as well as medical diagnostics (Rodríguez-González et al., 2012).

The agent-based approach requires explicit reference to agent-like entities in the system. Advantages of ABMs include modelling efficiency, robustness, interoperability between existing systems, and reasonably intuitive solving of problems for which data, expertise and control are distributed (Jennings and Sycara, 1998). The approach is thus particularly useful in the context of natural sciences, and permits reciprocity between agents and biological entities and between real-system interactions and exchanges between agents types. Agent-based models implementing several agents are referred to as *multi-agent systems*. These systems provide a generic framework for model development, as noted by Perrin (2008).

Although suitable for modelling molecular diffusion processes and naturally compatible with stochastic models, ABMs have not so far made significant impact on DDS modelling although an example of their use is given in a seminal paper by Barat et al. (2008). In this thesis, we use a subset of agent behaviours (namely the *autonomy* and *reactivity*) for describing certain active processes within the modelled device. It would be of interest, however, to further examine their full potential in DDS system modelling.

3.3 Monte Carlo methods

Monte Carlo methods represent a large class of computational algorithms that utilise random sampling as a means of optimisation, numerical integration, and generation of probability samples (important in our case). Initial MC simulations were also done by Ulam and von Neumann in the 1940s, while working for the Los Alamos National Laboratory.

MC methods have numerous applications, ranging from the physical sciences e.g. fluid dynamics and astrophysics through traffic modelling, statistical physics to financial analysis and business, (Sopasakis, 2004), (Tezuka et al., 2005), (Li et al., 2007). The essence of each MC method lies in generating a set of inputs over a certain domain by sampling from a probability distribution to define a starting state. A set of deterministic computations is then performed and aggregated to obtain the "system" result.

There are two broad method groups for Monte Carlo modelling. The *direct* methods use probabilistic distribution sampling for setting initial spatial properties of the model as well the state of the modelled entities (cells, flows, agents, etc.). Direct, deterministic or stochastic computations, are then utilised for describing the evolution of such entities over time to produce the aggregated end result for the system. Conversely, *inverse* MC methods utilise the sampling process to try to derive the unknown but feasible distributions for model parameters, from which the known aggregated result was obtained. In other words, *direct* models start from a known initial state and attempt to derive the end result, while *inverse* methods start with a known set of one or more end results and try to determine the possible, but unknown, initial states.

3.3.1 Application of Direct Monte Carlo methods in DDS

When applied to drug dissolution modelling, MC methods bring several advantages. They allow simulation of dissolution problems at increased resolutions by generating natural structural variations, and using parameters that can only be estimated using probabilistic distributions. They describe the system behaviour over time as a stochastic process and enable observation of certain properties of the system at individual time steps. As an example, an MC model might describe the internal location of pores within the polymer structure as random with certain distribution characteristics. Those pores can be allowed to evolve over time using random transitions to increase in size say, taking in polymer or drug particles and similarly. A snapshot of the system state can then be obtained at any time point to determine evolved behaviour.

MC methods have been variously used to take into account mechanisms underlying the rate-limiting steps in DDS such as diffusion, swelling and erosion. One of the first models, developed in 1993, (Göpferich and Langer, 1993) simulated microstructural changes in bioerodible polymers and still remains a basis for all subsequent MC simulation models. In this model, the polymer matrix was represented by a 2D computational grid, divided into individual cells. Each cell/pixel corresponded to one of two possible polymer states: amorphous, (with a higher likelihood of erosion) and crystalline, (with a slower erosion rate). In the polymer matrix, erosion from one particular cell occurs only if that cell was in

contact with a previously-eroded neighbour, where MC was used to represent this erosion as a random phenomenon. The probability of erosion of an individual cell was assumed to follow a Poisson distribution with the characteristic value for the erosion rate being greater for amorphous than for crystalline polymers. Each cell was assigned an expectation of life, (distributed as a first order Erlang distribution).

Improvements on the original model investigated additional factors influencing erosion, such as porosity and pH changes, (Göpferich and Langer, 1995a). Both mechanistic and empirical methods were used to describe release of monomers, using Fick's second law, (Equation 2.2) and given by:

$$\frac{\partial}{\partial t}c(x,t)\epsilon(x,t) = \frac{\partial}{\partial x}D_{eff}c(x,t)\epsilon(x,t)\frac{\partial c(x,t)}{\partial x} \tag{3.1}$$

where c is the concentration of the diffusant as a function of the effective diffusivity D_{eff} and ϵ is the porosity along the diffusion pathway.

The porosity distribution was interpreted as a network of pores, modelled using a 2D grid and MC. Degradation was considered as a spontaneous process of transformation from polymer into monomer with monomer dissolution affected by pH changes and following Fick's first law. The innovation lay in the tracking of changes in the molecular weight of the polymer. The model also enabled determination of the constant rate of erosion, but not of release predictions for the incorporated drug.

Further improvements led to investigation of surface and bulk erosion from polymers, (Göpferich and Langer, 1995b), programmable release from several layers of cylindrical eroding polymers, (Göpferich, 1997a), and erosion from slow and fast-eroding polymers, (Göpferich, 1997b). Starting from these models, many additional MC-based models for bio-erodible microparticles were generated and will be discussed further, for example Siepmann et al. (2002).

3.3.2 Monte Carlo in modelling DDS for the GI tract

At a relatively early stage, Kalampokis et al. (1999a) presented an interesting "heterogeneous tube" model for simulating the *villi*[1] of the gastro-intestinal tract by using direct MC for describing the dissolution and adsorption processes. Transit flow in the tract was simulated using two diffusion models referring to two types of random walk: the blind ant and myopic ant, (Kalampokis et al., 1999b). For simulation of the GI tube, an empty cylinder geometry was used, where the drug, inserted at one end of the cylinder, was allowed to the other end following the rules of a biased random walk.

3.3.3 Cellular Automata (CA) in Direct Monte Carlo methods

The first use of probabilistic CA models was described for drug release from bioerodible pellets. Pellets can be as systems, composed of several components of arbitrary geometry and with different dissolution rates: system behaviour was defined according to local relations, (Zygourakis, 1990). Subsequently, CA and parallel iterations were used, (Zygourakis and Markenscoff, 1996), for the design of bioerodible devices, where the transient behaviour of the surface erosion system was described. As with MC models, (Göpferich and Langer, 1993, 1995a), there are two possible states of polymer cells in the computational grid for CA: amorphous (highly erodible) and crystalline (poorly erodible). However, Zygourakis and co-worker's models were not based on differential equations. Instead, simulations were used to determine the effects of intrinsic dissolution rate, drug loading and porosity. Improvements on early models include assumptions on cell-dissolution neighbourhood, where dissolution of each solid cell depends on the number of solvent-filled neighbours. Increased solvent around the cell leads to more rapid dissolution. The authors suggested that simulations could be used for rapid screening of newer formulations and for speeding up the design process. They pointed out the advantage of the CA approach in terms of its ability to handle multicomponent systems with arbitrary geometry.

MC, together with CA, was subsequently used for setting initial conditions to investigate the behaviour of a binary device system, (Barat et al., 2006a,b). The novelty of these models

[1]The *villi* are small projections which cover the surface of intestinal wall, and serve for absorption of fluids and nutrients. The surface area of the wall in this way is increased and it enlarges the area available for absorption.

was that they enabled creation of pores "inside" the device, not only on the surface, and also emphasised the importance of the surrounding medium, not defined in earlier MC/CA models. The theory on USP II apparatus medium influence was due to Ramtoola and Corrigan (1987) and Healy and Corrigan (1996), who examined the influence of particle size and discovered that, in general, higher dissolution rate occurs for increased excipient particle size. Consequently, two boundary layers were defined in the apparatus: the *concentration layer* where advection governs mass transport and the *velocity boundary layer*, (a small region around the compact) where diffusion is the main mass transport, (McMahon et al., 2003), (Crane et al., 2004a), (Crane et al., 2004b). In parallel work, (Barat et al., 2006b), diffusion was modelled as the sum of the set of particles that can move to a new position. Particles that pass beyond the concentration boundary layer were considered to be dissolved. Advection was modelled using the Pohlhausen equation (Crane et al., 2004a) such that if the particle concentration in the cell is higher than a defined maximum, the particles are removed by advection. Possible states of the system were defined, according to Göpferich et al. (1995), Göpferich and Langer (1995b), but with additional update rules.

Pure Cellular Automata application in DDS modelling

In investigating the erosion and swelling behaviour from binary matrix systems, CA without MC randomisation has been used as well, (Laaksonen et al., 2009a), (Laaksonen et al., 2009b). The main differences include dispensing with arbitrary lifetimes for the drug and polymer and assuming that diffusion was based on the pure random walk, thus eliminating the need for Fick's first law. There is no specified expectation of life, rather the diffusion coefficient and rate of disintegration are calculated. These models were developed for several types of release mechanisms: erodible matrices, diffusion through channels/pores, membrane controlled release and the investigation of swelling-controlled drug release. Although the models provided a new perspective, a major limitation was the restriction to 2D cylindrical geometries due to relatively small simulation space.

3.3.4 Direct Monte Carlo vs. Inverse Monte Carlo

The *Inverse* Monte Carlo approach is used for problems, for which we do not have complete knowledge of the relationship between model parameters and observed data, (Mosegaard and Sambridge, 2002). One of the main reasons for using this MC variant is thus to avoid stringent assumptions of linearity between the two, (Socco and Boiero, 2008). Instead, it is assumed that initial parameters follow a probability distribution, as described in Voutilainen et al. (2001).

In this context, MC methods can be divided into two groups, (i) *sampling methods* and (ii) *optimisation methods*, (Mosegaard and Sambridge, 2002). The latter are used to find a globally optimal solution from a number of local optima. Sampling methods are useful when it is necessary to examine reliability of solutions in order to narrow down the feasible solution space. One technique, often used for sampling, is the Bayesian paradigm, where the principle is to take all unknown variables to be random. In general, the Bayesian framework for posterior distribution evaluation can be represented, (Mitchell, 1997), by:

$$P(h|E) = \frac{P(E|h)P(h)}{P(E)} \tag{3.2}$$

where $P(h)$ and $P(E)$ are prior probabilities of the hypothesis h and observed data E, respectively. $P(h|E)$ is the posterior probability[2] of h given E and $P(E|h)$ is the likelihood of obtaining the actual data E, for the specific h.

The Bayesian paradigm for inverse problems is widely used, with examples ranging from noise reduction, (Davies, 1998), and environmental modelling (Dowd and Meyer, 2003) to geophysical inverse problems, (Mosegaard and Sambridge, 2002).

IMC methods have been used across many scientific disciplines, also, to solve a set of related, inverse model problems. As an example, they may be used to find a near-optimal solution in terms of data fit and adherence to constraints for a posed problem. In another, IMC methods may be used to search for solutions fitting the data within a certain tolerance. IMC is a general method of co-called "structural" modelling based on experimental data and was first described, (McGreevy and Pusztai, 1988), and applied to condensed matter science.

[2]The posterior probability is the conditional probability determined after additional information is used to modify the initial, *a priori*, probability assigned.

The authors developed and applied the methodology to understand how physical properties of materials were determined by their atomic structure. Despite origins in condensed matter physics and solid state chemistry, IMC has also proved applicable to many other different data, as follows.

In geophysics, a large amount of work has been done on investigation and improvement of inverse methods, with generally applicable results (Mosegaard and Sambridge, 2002). Keilis-Borok and Yanovskaja (1967) applied inverse MC methods to determine if "Earth" models were consistent with seismological data. Geman and Geman (1984) applied simulated annealing, a technique for finding approximate global optima of a function that cannot be evaluated directly, to Bayesian image restoration. The authors suggested using the Metropolis algorithm to sample the *posterior* distribution and to determine its maximum estimate.

One of the first algorithms for inverse MC problems was simulated annealing (SA), (Geman and Geman, 1984), a heuristic algorithm used for finding the global optimum of a given function in large, discrete search space. The SA algorithm is based on the idea of a global reduction in the probability of accepting worse case solutions as the solution space is explored. Each point of the search space (s), analogous to a state of the physical system, together with the function to be minimised, is considered.

Another algorithm, used to draw samples from a desired probability distribution of possible parameter values, as long as the function proportional to density is calculable, is the Metropolis-Hastings algorithm (MH) (Metropolis et al., 1953). Here, the aim is to generate a series of Markov Chain samples for which the distribution matches that of the target distribution over an extended period of time. In Bayesian sampling models, the density proportionality constant, also known as the *normalisation factor* needs to be known, so that the MH category of algorithms offers a significant advantage.

An improvement on Metropolis-Hastings is Gibbs sampler, first introduced by Geman and Geman (1984), which represents a special case of the MH algorithm. It can be extended to a general framework for sampling from a large set of variables in turn, and can incorporate the steps of Metropolis-Hastings as part of the algorithm. It is applicable when the conditional distribution of each variable is known and can be sampled.

It is worth mentioning that not all inverse methods apply Bayesian techniques. Early work was based on simple searches of the parameter space for "best fit" models, (Keilis-Borok and Yanovskaja, 1967), while a related set of methodologies, namely *genetic algorithms* and the *neighbourhood algorithm*, have more sophisticated evolution characteristics and a reduction-based approach.

3.3.5 Inverse Monte Carlo in investigating the effect of particle size distribution

The main difficulty when modelling novel formulations for drug devices lies in unreliable and inconsistent *in vitro* data. Data obtained from the pharmaceutical partner have been extremely noisy and relatively sparse; thus extensive model assumptions have proved necessary and their validation has been far from straightforward. Determining correct dependencies is a highly complex process, exacerbated by the inability to measure certain critical parameters of the system *in vitro* (notably the diffusion coefficients for drugs and all polymers involved) and it is often difficult to utilise direct MC techniques because of this.

In Barat (2006), the authors created a model, based on Inverse Monte Carlo (IMC) simulations in a Bayesian context, which was capable of extracting knowledge from experimental data, (time series of dissolved quantities). The work leverages de Almeida et al. (1997), by applying a Bayesian framework and inverse techniques to a powder dissolution system to reconstruct the set of initial particle size distributions from a large number of possible states, in order to provide the best fit to the experimental data. Fitting is done by empirically updating key parameters through repeated comparisons with the experimental release curves. This approach provides substantial flexibility in determining the important parameters of the DDS, based solely on available "drug release vs. time from administration" curves.

The model of Barat (2006) thus took the observable dissolution data as the starting point and tried to discover a set of key parameters that describe the total number of particles. The observable dissolution process was decomposed into two components: a deterministic part, defined by relationships between the state parameters at a given time, and a random noise

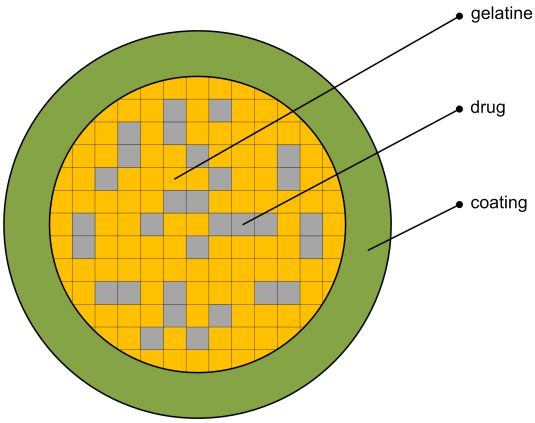


Figure 3.3: Simplified design of the modelled device: yellow cells represent the gelatine carrier with dispersed drug packets (grey). The device is coated with EC/Pectin layer (green).

component, assumed to have a Normal (Gaussian) distribution. The size distributions were updated, according to the de Almeida et al. (1997) algorithm, simulating the deterministic part of the dissolution process, with the likelihood then calculated for each distribution, (to determine the most probable values of key parameters together with their confidence intervals). In the next iteration time step, these probabilities were reused as the starting point for the next prediction. This cyclic process enables algorithm values to be updated, while values of parameters are simultaneously measured for each time step.

The main novelty of this earlier work by DCU group was applying IMC techniques to the field of drug dissolution, showing (importantly) that crucial parameter values could be determined through analysing a number of release curves under different conditions. However, since it was done as a proof of concept study only and was not applied to real data many questions remain to be fully investigated, not least in terms of the practical application value and potential for adaptation to different types of formulation, release or similarly.

3.4 Project motivation and overview

In summary, the main goal of this project is to develop novel stochastic Cellular Automata models for simulation of drug delivery systems and apply these in an industrial context to predict drug release behaviour from coated beads, targeting the gastro-intestinal tract. The

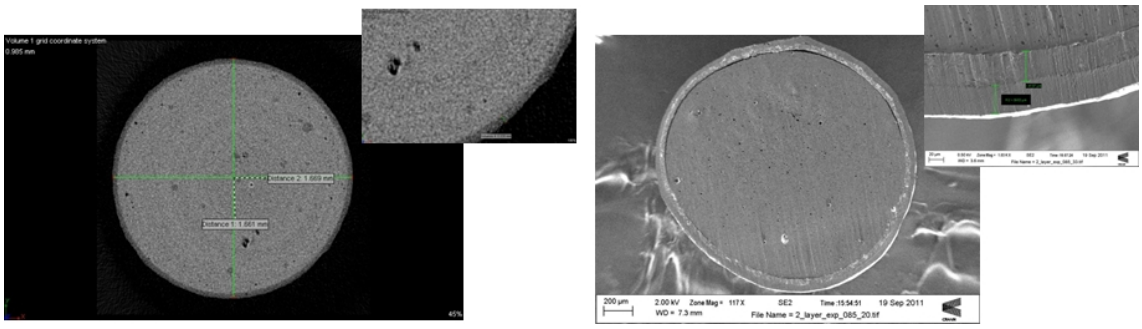


Figure 3.4: X-ray Tomography of modelled device. (left) A single bead with one layer of coating; (right) variant of the bead with two coatings.

project was carried out in collaboration with the pharmaceutical partner, Sigmoid Pharma Ltd., which has acknowledged expertise in developing controlled release formulations, specifically for treatment of GI tract diseases such as Crohn's disease and ulcerative colitis. One of the objectives here is the incorporation of experimental features, characterising the drug device and collected by the pharmaceutical partner, into a probabilistic model by mapping these to a set of independent, simple CA rules. The "device" consists of a capsule, containing hundreds of coated spherical beads, which carry the active substance, Cyclosporine A (CyA). Due to the complex nature of the device, the underlying release mechanisms cannot be precisely determined, prompting the need for probabilistic modelling methods, in order to understand the device behaviour.

A simple cross-section view of the drug design of interest is shown in Figure (3.3) and X-ray tomography images of actual device (provided by Sigmoid Pharma) in Figure (3.4). A mixture of ethylcellulose and pectin is used as coating materials. As discussed in Chapter 2, considerable work has been done by chemists to investigate the pharmaceutical potential of this type of coating, (specifically its influence on release profiles of different drug types encapsulated within EC-coated pellets, and on gastric fluid enzymes). All such features must be taken into account when developing a suitable model. Encapsulated by the coating, the active substance, Cyclosporine (poorly soluble), is dispersed within a swellable/hydrophilic gelatine carrier, (Coulter, 2010). While the coated formulation is pH independent and hydrophobic, the capsule content is highly hydrophilic and dependent on temperature. To our knowledge, no previous model exists for this type of complex

formulation and this project comprises original work in the area.

The work to date has included the definition, building and application of a modelling framework, which provides a foundation for the simulation of physical properties for specific designs and their effect on drug release kinetics. So far, the framework has been used to build several models describing specific types of drug behaviour. Details on this implementation are presented in Chapter 4 of the thesis. Further, the framework has also been applied to the investigation of other device design parameters and drug release patterns. IMC methods were used in the context of extracting the unknown parameters, (such as solubilities and diffusion coefficients), from noisy *in vitro* data using reverse engineering methods, (Chapter 7).

Chapter 4

Models for Controlled DDS

In this chapter a novel, integrative CA framework for drug dissolution is presented with several subsequent models derived from it. In the context of achieving realistic dynamics, comparisons are drawn between synchronous and asynchronous Cellular Automata and their relative merits are discussed. Such analysis is lacking in the literature but can lead to real improvements in simulation.

4.1 The Meta-Model

In this section, we describe the fundamental basis and initialisation of the *computational framework* used for simulations, together with the *meta-model*[1] which includes the CA modelling features, (such as representation of diffusion), common to all subsequent derived models. Several *models*, describing drug release driven by different dissolution phenomena, (such as erosion and swelling moderated by one or more coating layers for the active component), are extensions of the meta-model and are presented in detail in Sections 4.2, 4.3, 4.4, respectively, while the influence of these specific features on release behaviour is discussed. Additionally, a model which looks into the influence of *in vitro* media on the drug device is also developed, and is described in Section 4.5.

[1] In this context, we consider "metamodelling" as a practice of construction of a collection of basic "concepts". These concepts are used as a mechanism for derivation of different sub-models of interest.

4.1.1 Fundamentals of developed models

The basis of all probabilistic models developed to date can be described in terms of a discrete CA framework, satisfying the following conditions:

- Initial cell states are assigned randomly, according to a specified probability distribution, within the structural constraints of the device. (This is because the local distributions of device features, e.g. drug density or polymer composition at a given site, are unknown);
- The different dissolution phenomena are modelled independently. (We can separately investigate the influence of polymer swelling while ignoring coating erosion and vice-versa);
- The three-dimensional discrete space allows for models of various geometries and device compositions;
- An efficient large scale simulation. (This, in order to explore molecular level effects).

Recalling Chapter 3, the general formulation is given by Figure (3.3) where a number of drug beads are contained within a quickly erodible capsule. The models themselves simulate the dissolution of a single, idealised, drug bead represented in a three-dimensional space, assuming its release profile to be representative of the cumulative release from all beads present in the capsule. The simulation space itself is represented by a lattice, divided into discrete cells, with each being described by a state $\Psi_{(i,j,k,t)}$ defining the aggregate condition of the bead structure at discrete space coordinates *(i, j, k)* and at a discrete point in time *(t)*. The meta-model defines the Cellular Automata rules:

$$\Phi(\sigma) : \Psi_{(i,j,k,t)} \rightarrow \Psi_{(i,j,k,t+\Delta t)} \tag{4.1}$$

for each cell which are applied to cell state (Ψ) in order to re-evaluate it after each time increment (Δt) of the simulation. When a particular rule is applied to the cell state, this is in the context of the states of all neighbouring cells (σ) in the 3D space, (26 in all, corresponding to a Moore neighbourhood), which also determines the effect of the rule for

the given cell. The rules, as applied, can thus model any distinct process that causes a change in the bead state, including drug diffusion, chain-scission of ethylcellulose polymers, erosion or swelling of the device core.

The bead itself is represented as an "idealised sphere", consisting of a number of layers containing the drug molecules. The possible states of each cell represent different structural forms of the bead that affect the behaviour of these molecules as they diffuse out of the sphere. The Meta-model defines several common cell types unique to the modelled device, and re-used by all the derived models:

1. **Gelatine cells** (designated $|P$, as of polymer form) which represent the internal structure of the bead that serves as initial carrier of the drug (Cyclosporine).

2. **EC/pectin cells** (designated $|C$), comprise the coating and are used jointly to model the coating layer of the sphere.

3. **Wall cells** (designated $|W$) mark the boundary of the simulation space and block any cellular transition occurring outside this.

4. **Buffer cells** (designated $|B$) which model the dissolution space around the cell.

5. **Solvent cells** (designated $|S$) which represent the internal solution inside the bead coating that is created as a consequence of dissolution of gelatine material.

Each cell, under certain circumstances, (with the exception of those in the wall layer), can hold a certain concentration of drug (C_d) represented as discrete "packets" that can model an arbitrary precision of drug volume loading *(v/v)*[2] within the cell dimensions. Between subsequent time points in the simulation (Δt), a packet can move (diffuse) to a neighbouring cell. Thus, the size of the packets and the time step of the simulations are primary factors in controlling the simulation speed, and we can choose either to simulate large, bulk diffusion processes or very finely-grained ones, as required.

The models themselves have been developed in several stages, designed to facilitate our understanding of key behaviour at increasing levels of internal complexity and inter-dependency, and starting with simple erosion-based transitions. Additional rules were

[2]In chemistry, volume-in-volume ratio (denoted as v/v) represents the volume of the substance relative to the total volume of the solution.

developed to consider specific variants and to represent more realistic, composite interactions between processes such as diffusion, erosion and swelling.

Meta-model initialisation

At the beginning each cell is initialised to a start state, depending on its location.

1. The coordinates of cells that satisfy the spherical equation:

$$(i-o)^2 + (j-o)^2 + (k-o)^2 - (r-d)^2 \leq 0 \tag{4.2}$$

   r is the sphere radius with o the centre of the sphere, and d the thickness of the coating layer are initialised with starting gelatine state ($\Psi_{(i,j,k,0)} = |P$), (representing the polymer inside the bead). For each of these cells, a random number (R) between 0 and 1 is generated and compared to the drug volume (v/v) loading ratio of the sphere:

$$V_d = \frac{m_{drug} \cdot \rho_{sphere}}{m_{sphere} \cdot \rho_{drug}} \tag{4.3}$$

   where m and ρ are, respectively, the mass and the average density of the sphere and the drug itself.

   If $R \leq V_d$, the cell is initialised with a number of drug packets, where this number depends on the desired resolution of the simulation. This ensures that the desired percentage of cells corresponding to (v/v) ratio will contain drug packets. Drug diffusion is not permitted through $|P$ cells.

2. Cells which are contained outside the gelatine sphere, but within boundaries defined by the larger, coating sphere, satisfy the equation:

$$(r-d)^2 \leq (i-o)^2 + (j-o)^2 + (k-o)^2 \leq r^2 \tag{4.4}$$

   and belong to the coating layer ($\Psi_{(i,j,k,0)} = |C$). Coating layer cells cannot contain drug packets.

3. Cells which satisfy the boundary property ($i = 0 \vee n, j = 0 \vee n, k = 0 \vee n$) are initialised as walls: $\Psi_{(i,j,k,0)} = |W$.

4. All other cells are initialised to represent the buffer solution: $\Psi_{(i,j,k,0)} = |B$.

5. Solvent ($\Psi_{(i,j,k,t)} = |S$) cell types do not occur during model initialisation, but only during cell transitions in the course of model execution.

Transition rules between cell states are represented using dimensionless ratios, (one of the main advantages of CA models, as this allows comparison of processes occurring at different velocities).

Modelling diffusion

As described in Section 2.1, the diffusion of a drug in a governed environment is given by Fick's laws, which gives the change in concentration gradient over time. Although, at macroscopic level a given drug formulation may follow either Fickian or non-Fickian (anomalous) behaviour, we assume that Fick's laws always hold on smaller scale. Movement can occur in any one of the directions in three dimensional space, so we assume that there exists probability of movement in any direction where there is a concentration gradient.

We model the influence of Fick's laws by allowing each drug packet to have a probability (P_l) of movement to a given neighbouring cell (l) with lower drug concentration. This probability is directly proportional to the concentration gradient between two neighbouring cells, i.e.

$$\Delta C_l = C(i, j, k, t) - C(i^*, j^*, k^*, t), \tag{4.5}$$

where *C(i, j, k, t), C(i*, j*, k*, t)* represent dimensionless drug concentrations in the current and adjacent cell, respectively. The sum of all movement probabilities to permitted neighbouring cells, (i.e. to those for which CA rules do not prohibit movement, non $|C$ cells), has to satisfy the law governing the sum of probabilities: $\sum_{i=0}^{n} P_l = 1$. Therefore, the

influence of concentration differences is normalised, giving:

$$P_l = \begin{cases} \dfrac{\Delta C_i}{\sum\limits_{j=0,\Delta C_j>0}^{k} \Delta C_j}, & \text{if } \Delta C_i \geq 0 \\ 0, & \text{if } \Delta C_i < 0 \end{cases} \tag{4.6}$$

where $i \in \{0, k\}$ and k is the number of neighbours to the given cell. For certain types of cells ($|W$, $|C$ and cells where $\Delta C_i \leq 0$), P_i is assumed to be 0 by default. To summarise: according to this rule, if there exists a non-zero concentration gradient between the cell and any of its valid neighbours, there is a non-zero probability that a drug packet will move to that cell.

The speed of diffusion itself (i.e. the frequency of drug packets moving from one cell to another) is calculated using the formula defined by Laaksonen et al. (2009a), which relates the simulation time step (Δt) with diffusion coefficient (D), and the resolution of the cellular grid (a):

$$\Delta t = \frac{a^2}{4D} \tag{4.7}$$

Key release indicators

In order to categorise the release curves obtained from the models (as anomalous, Fickian or case-2 relaxation) we use Peppas Equation (2.6) (Chapter 2) to classify the drug dissolution, based on the parameter n and the resulting curve behaviour in achieving zero-order release rate. The parameters M and M_∞ represent the amount of drug released at time t and total amount of drug respectively, while k_p is the kinetic constant. Values of the parameter n are obtained from resulting release curves, using linear regression methods, where n is the slope of the log-log line, of cumulative release against elapsed time, for the period during which the majority of the drug dissolution occurs.

A dimensionless Deborah number (Vrentas et al., 1975) can be used as an additional descriptor of the release, indicating Fickian or anomalous transport. The Deborah number

is defined as:

$$De = \frac{\lambda_e}{\tau_e} = \frac{\lambda_e D_w}{\delta^2} \tag{4.8}$$

where λ_e represents the *characteristic relaxation time* of the polymer, τ_e the *characteristic water diffusion time* and δ is gel layer thickness. Changes in Deborah number can be estimated directly from the water diffusion and polymer swelling probabilities using the derived version of the equation, where D_w is estimated from the water diffusion probabilities, while λ_e and δ are read from the resulting simulation measurements.

4.2 Modelling erodible coated drug devices (ECDD)

The first model specifically developed to address project aims (as stated in Section 3.4) is designated the ECDD and considers the case for polymeric chain erosion in the coating and subsequent drug diffusion, where these were assumed to be the primary factors influencing release[3]. The main rationale for taking this initial, simplified approach was the availability of experimental data that showed significant sensitivity of drug release to changes in the coating thickness, implying erosion due to water intake rate as a primary contributor to release. There is a long list of possible effects that can influence rates of degradation and erosion (e.g. pH changes, water diffusivity, composition of polymers, polymer chain length, (Göpferich and Langer, 1993, 1995a) and so on) and it is of particular importance to determine the most influential and to establish the dominant release kinetics.

In the theoretical introduction, (Section 2.1), we noted that erosion represents the scission of polymeric chains, allowing solvent penetration into the coating layer and, further, into the sphere. On the microscopic scale, we can essentially look at this as a stochastic process, where the probability of chain breakage is a value following a known distribution. Such events occur independently of each other and at a specified rate within an interval of time. Following (Göpferich, 1997b), bulk erosion effects are represented by the Erlang

[3]This work has been published in (Bezbradica et al., 2011).

distribution:

$$e(t) = \lambda e^{-\lambda t} \tag{4.9}$$

where parameter lambda (λ) in the CA model, describes the rate of polymer breakage, i.e. porosity formation. This parameter is characteristic of the *material* and can be changed if a different coating material is used. In our case, a coating cell is considered to consist of the mixture of ethylcellulose and pectin. The individual characteristics of each of these components influence erosion, but we focus on their combined role. The lifetime (τ_c) of a coating cell can be determined as inverse by proportion to λ given by:

$$\tau_c = \frac{1}{\lambda} ln(U), U \in Unif[0,1] \tag{4.10}$$

where $Unif[0,1]$ is an Uniform distribution taking values between 0 and 1.

Further, we introduce a new cell type (type 6), namely a *coating channel cell* ($|CH$), to model cracks and pores in the polymer structure caused by erosion of the coating material or partial chain disentanglement. Unlike $|C$ cells which act as a barrier to drug diffusion, $|CH$ cells permit drug movement, thus allowing for diffusion through the coating layer.

At the beginning of the simulation, (t=*0*), each $|C$ cell is assigned an individual value of τ_c, based on Equation (4.10). In every iteration, this value is decreased by the time step of the simulation (Δt). When τ_c falls to zero, the coating cell is considered to be fully eroded and is thus transformed into a channel cell:

$$\Phi(\tau_c = 0) : |C_{(i,j,k,t)} \rightarrow |CH_{(i,j,k,t+\Delta t)} \tag{4.11}$$

The drug matrix is held together by gelatine, another polymer inside the coating, which is considered to undergo surface erosion. At t=*0*, each $|P$ cell is assigned a fixed lifetime (τ_p). A process, of progressive solvent penetration through the coating layer, eventually allows $|P$ cells to come into contact with the solvent cells ($|S$), hence providing a trigger for their erosion. As with $|C$ cells, the speed of erosion of a $|P$ cell is determined by the change in the gelatine lifetime, where this lifetime starts to decrease as soon as the gelatine cell comes into contact with a $|CH$ or $|S$ cell. Additionally, the decrease in lifetime is

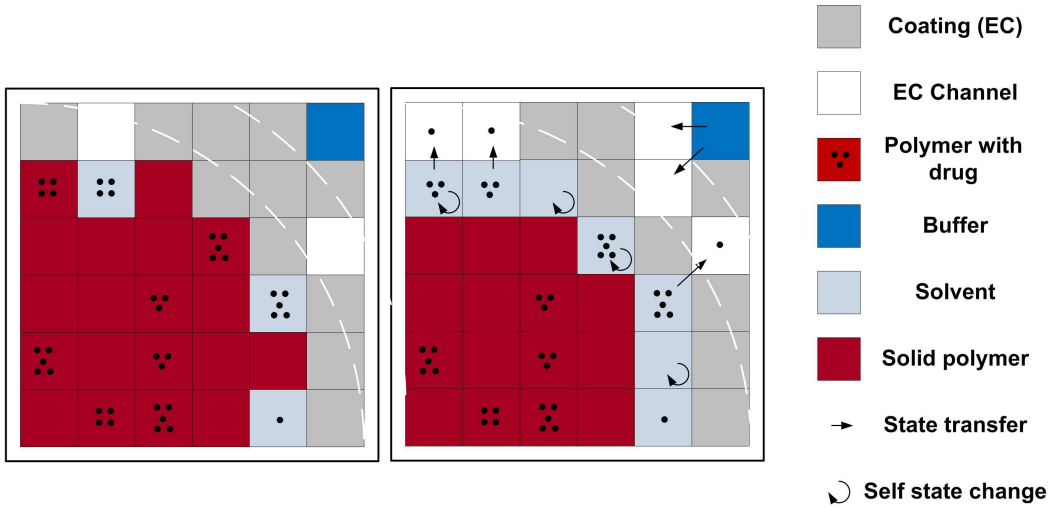


Figure 4.1: Schematic representation of CA rules for ECDD model. The main transition rules given in Table 4.1 are visible. Arrows are showing drug and water diffusion (straight) and polymer erosion (circular).

proportional to the number of those cells in the direct neighbourhood. Thus,

$$\tau_p(t + \Delta t) = \tau_p(t) - k_s \Delta t \tag{4.12}$$

where k_s is the number of all surrounding $|CH$ or $|S$ cells. When a $|P$ cell is eroded it becomes a solvent cell, $|S$. Table 4.1 summarises the CA rules of model ECDD and Figure (4.1) shows a schematic example of transitions between two model iterations.

4.3 Modelling swellable coated drug devices (SCDD)

Building on the basis of the previous ECDD model, which laid the foundation for investigation of some release phenomena of the target device, as well as accessing the efficiency of the model framework, proposed in Chapter 5, a more complex set of release mechanisms could then be formulated. It was anticipated that these would improve the simulation model description of the device behaviour observed by experiment.

The principal addition made was incorporation of polymer swelling phenomena into the CA system, as gelatine (and related substances) are known to swell under hydration, (Klepko and Mel'nichenko, 1995). Hydration causes a general increase in the polymer volume, as water penetrates deeper into the device core, creating a hydrogel layer. The

Cell type	Behaviour description
Buffer ($\|B$)	Drug is considered released when it reaches buffer zone. Cell type acts as a perfect sink.
Ethylcellulose ($\|C$)	Drug-free coating layer. Assigned lifetime, (τ_c), based on λ (degradation rate) parameter. Forms a $\|CH$ cell upon complete erosion.
EC channel ($\|CH$)	Drug can diffuse through $\|CH$ cells.
Gelatine ($\|P$)	Assigned fixed initial lifetime (τ_p). As it erodes into $\|S$ cells, it facilitates movement of drug "packets".
Solvent ($\|S$)	Occurs as result of gelatine erosion. Drug can diffuse through the solvent.
Drug packet	Drug, initially dispersed in gelatine cells. Each cell can hold a maximum (saturation) amount of drug "packets".

Table 4.1: Cell types and rules of behaviour for ECDD model.

thickness and permeability of this layer control the rate of drug release. It should be noted at this point, that such gelatine swelling phenomena categorise the system as true, non-porous and swelling-controlled DDS, in contrast to the swelling matrix systems, where drug diffusion occurs though the porous matrix structure even if there is no solvent penetration. Characterising the difference is the fact that the drug particles are immobile while a polymer is in its glassy state, and can diffuse out only after a phase transition to the rubbery state takes place, (Colombo et al., 2000).

The modelling of such swelling systems is a non-trivial task. Unlike the other major release mechanisms of polymer erosion and drug diffusion, which can be modelled in relative isolation, swelling acts in conjunction with these mechanisms to create a complex set of state transition fronts inside the device. Two of the most important fronts are: (i) erosion, taking place at the outside water/gel interface, where the disentangled polymer chains erode, and (ii) diffusion, occurring at the internal solid/dissolved drug interface where polymer transition from glassy to rubbery state occurs. The difference in radii of those two fronts, i.e. the gel layer thickness, acts as a major controller of the drug release rate, so describing its behaviour is crucial (see Figure (4.2)).

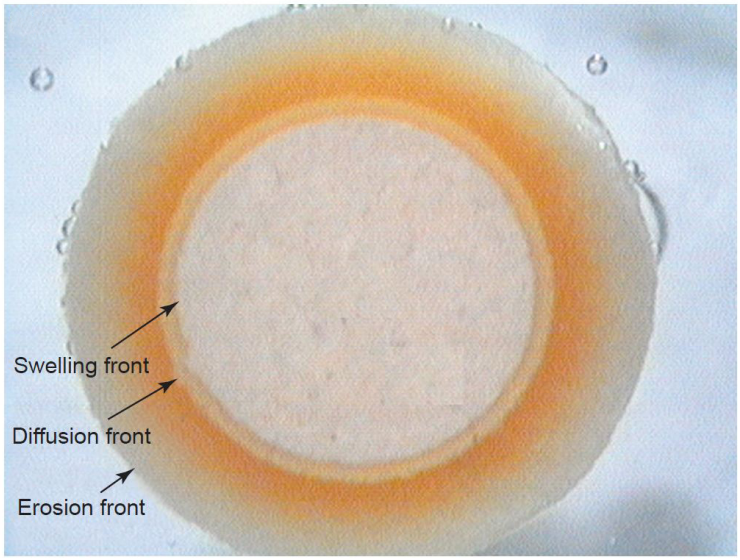


Figure 4.2: Photograph of an HPMC matrix tablet loaded with buflomedil pyridoxalphosphate, after one hour of swelling release. Three dissolution fronts are presented allowing for visualisation of gel-layer thickness, (Colombo et al., 2000).

Swelling has attracted significant attention relatively recently, mostly in mechanistic and empirical models, ranging from a simple power-law equation, proposed by (Korsmeyer et al., 1986), to more complex cases, which have used a mechanistic approach linked to the particular hydration properties of a given device. Examples describing swelling as the essential parameter in probabilistic models of CA type, are few, with only one major model described recently, (Laaksonen et al., 2009a). Of course, existing models have limitations, both in terms of the amount of initial knowledge needed to "feed" the model and also in terms of limited applicability to complex formulations for targeted delivery, (such as the one we are investigating).

We thus investigate two related models, one assuming a constant rate of swelling and the other assuming a varying rate related to the physical behaviour described in the literature (Singh and Weber, 1996). Both the uncoated and coated drug formulations are modelled, with the former used mostly for validation of the modelling assumptions. A schematic representation of CA rules and behavioural transition is presented in Figure (4.3) and Figure (4.4). The colours represent different cell types, the arrows the direction of transfer and dots within a cell the concentration. An example of both self state change and lifetime transfer is apparent from the right hand panel.

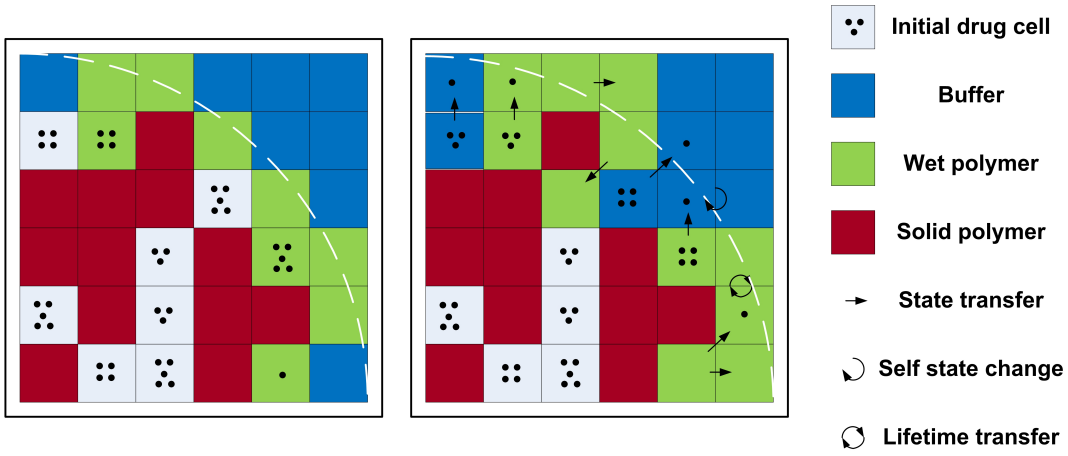


Figure 4.3: Schematic representation of CA rules for uncoated devices. Saturation loading is 4 drug packets per cell. Initial cell can be under or over saturated.

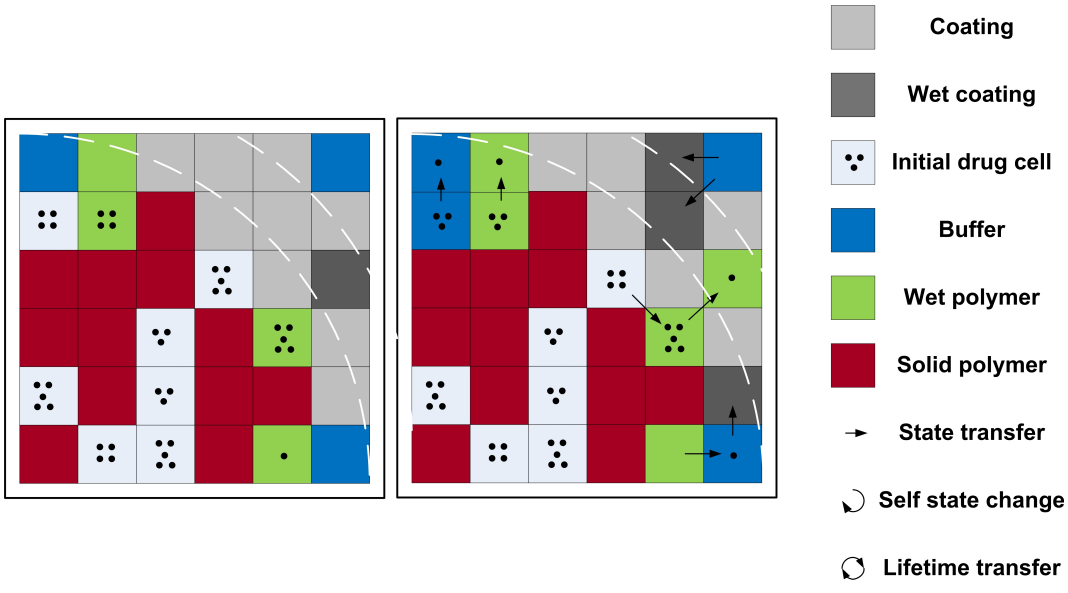


Figure 4.4: Schematic representation of CA rules for coated devices. The initial cell can be under- or over-saturated.

4.3.1 Effect of swelling kinetics - "constant-rate" (SCDDa)

To model swelling, we introduce additional states and cell transition rules to the existing CA model:

1. Similar to (Laaksonen et al., 2009a), we describe the swelling process by two probabilities - namely the probability, P_W, of water diffusing into the polymer cell (core or coating), and P_S, the probability of core polymer chains diffusing out of the initial core volume, proportional to the swelling rate of the polymer.

2. Each $|P$ cell is assigned a swelling potential (S_p) representing the effect of the stress field[4] formed between the glassy/rubbery interface on one side and rubbery/solvent interface on the other. This potential allows the cell to swell a certain number of times, and is linked to the amount of polymer left within the $|P$ cell, which is represented by the lifetime (τ_p) (i.e. $|P$ cells with higher lifetime have a higher swelling potential). This rule combines the swelling and erosion processes into one common parameter, allowing for a simplified description in the event of difficulty in comparing with unreliable experimental observations for individual processes.

3. A new cell state representing the rubbery polymer stage, ($|WP$, as "wet polymer"), was introduced. This state enables the diffusion of drug cells through the wetted environment, as well as erosion and swelling of the gel polymer chains themselves. A separate glassy to rubbery polymer cell transition rule is defined:

$$\Phi(p \leq k_w P_W) : |P_{(i,j,k,t)} \rightarrow |WP_{(i,j,k,t+\Delta t)} \tag{4.13}$$

   where k_w represents the number of neighbouring water cells, (either $|B$ or $|S$), P_W the water intake probability, and p, a uniformly distributed random variable, selected for each cell in every time step. We consider the $|P$ sites (glassy polymer) to be mechanically and chemically inert, with no swelling, erosion or drug diffusion taking place.

[4]The stress field is formed as a consequence of the concentration gradient between two fronts.

4. Diffusion through $|WP$ cells occurs with probability based on the remaining lifetime of the cell (τ_p):

$$P_{WG} = 1 - \frac{\tau_p(t)}{\tau_p(0)} \tag{4.14}$$

where P_{WG} represents the probability that a drug packet present in the $|WP$ cell will move. The drug diffusion rate within the gelatine thus ranges from 0 (immobile) to 1 (rate of drug diffusion in water).

5. As in the ECDD model, $|WP$ cells erode when their lifetime reaches zero, and are converted into $|S$ cells.

6. The behaviour of gelatine cells is further enhanced as an initial value for τ_p is set using the Erlang distribution, Equation (4.10), which describes the nature of the gelatine erosion more realistically, as this includes both surface and bulk (Ulubayram et al., 2002).

7. The process of swelling causes the $|WP$ cell to transfer part of its material to a neighbouring $|B$ or another $|WP$ cell, simulating chain disentanglement in a random direction:

$$\tau_p(i^*, j^*, k^*, t + \Delta t) = \tau_p(i^*, j^*, k^*, t) + L_c, \tag{4.15}$$

and

$$\tau_p(i, j, k, t + \Delta t) = \tau_p(i, j, k, t + \Delta t) - L_c \tag{4.16}$$

where L_c is a simulation constant representing the lifetime of polymer chains transferred during individual cell swelling. This value is determined by calculating the average lifetime of gelatine cells at the beginning of the simulation and dividing this quantity by S_p.

The lifetime transfer only occurs between cells where there is a "lifetime gradient". Swelling is thus represented as the diffusion of polymer from higher "concentration" to

lower. Further, we choose the target neighbour in the same way as for drug diffusion - the sum of all probabilities of swelling affecting neighbouring cells must satisfy the probability sum rule and the movement is proportional to the difference in lifetime. The formula is the same as in Equation 4.6, with ΔC replaced by $\Delta\tau_p$. Moreover, there are types of cells ($|W$, $|C$, $|P$ or cells where the lifetime difference is negative) for which the probability is set to zero.

8. The mechanism of drug diffusion through the coating layer is changed. Instead of the bulk porosity used in the ECDD model, we now assume a more realistic sequence of events, where the coating layer first takes in water through the ethylcellulose chains with a probability P_W, allowing it to reach the core layer after a certain period of time. The core layer then begins swelling, according to the rules 1-7 described above, creating a hydrostatic pressure difference, which has the potential to cause polymer chain breakage ("cracking") in the coating, resulting in expansion of gelatine out of the coated sphere.

9. Finally, we consider the phenomena of *polymer blocking* and *core fragmentation* which are specific to the Cyclosporine formulation modelled. Polymer blocking occurs when the undissolved Cyclosporine chains attempt to diffuse through the ethylcellulose membrane. This can lead to significant chain entanglement. This, in turn, causes micro-pores within the coating to become congested and drug diffusion through the outer membrane to be reduced significantly. Core fragmentation is the process whereby large chunks of core gelatin tend to clump together, which results in larger clumps with reduced solubility that trap the drug inside the bead, causing release rates to plateau after a certain percentage. We introduce an additional modelling parameter, P_C, representing the probability of a $|P$ cell going into the non-dissolvable state in each iteration. Such a cell does not loose polymeric material over time and remains in the same state throughout the simulation.

4.3.2 Effect of swelling kinetics - "varying-rate" (SCDDb)

For various pharmaceutical applications, it is of interest to know the actual swelling kinetics of the hydrogel which will be used as the drug carrier, as this may have a significant impact

on drug release itself, (Martínez-Ruvalcaba et al., 2009). There are several theories to date that deal with hydrogel swelling kinetics. The physics of this process was studied initially in isolation by Tanaka and Fillmore (1979), Li and Tanaka (1990) and Singh and Weber (1996). These papers documented two characteristic swelling curves, differentiated by the ratio (F) of gelatine volume in equilibrium (maximum swelling point) and its initial volume. If this is less than a characteristic value of 2.5, (based on the inflection point of the swelling curve for spherical geometries, Singh and Weber (1996)) then swelling can be considered as a Fickian process, with the change in sphere radius given by:

$$\frac{a_e - a(t)}{a_e - a_0} = \frac{6}{\pi^2} \sum_{k=1}^{\infty} exp(-\frac{k^2 t}{\tau_T}) \tag{4.17}$$

where a_e, a_0 and $a(t)$ represent the distance from equilibrium radius at the initial, last and interim swelling times, and τ_T is the characteristic swelling time of the polymer. However, as pointed out, (Singh and Weber, 1996), this law holds only for small swelling rates, and does not give a good prediction for polymer swelling where F is larger than 2.5. In this case, the dissolution process becomes non-Fickian, with swelling predominantly controlling release, resulting in a slightly different, sigmoidal curve.

We observe the influence of swelling probability changing with time, (as in Tanaka and Fillmore (1979) and Singh and Weber (1996)). We assume that the probability of polymer swelling (P_S) is linked to the change in radius between two time points, with the exact mechanism depending on the total ratio of volume change:

$$P_S = e^{-C_1 \frac{t}{\tau_T}}, F < 2.5, \tag{4.18}$$

$$P_S = C_2 \left[s(t) \left(1 - s(t) \right) \right], F > 2.5 \tag{4.19}$$

where F represents the ratio of the device volumes at equilibrium and initial states respectively; C_1 and C_2 are empirical constants that link the swelling kinetics to model probabilities, (limited by factors such as lifetime and swelling potential), and must be determined, (a non-trivial task). Finally *s(t)* represents a logistic curve, used to describe the sigmoidal nature of the process.

4.4 Modelling multiple coatings (MCES)

While single coated beads permit the fundamental controlled release scenario (based on the properties of a single polymer or of a polymer mixed with additives (such as a Surelease®/Pectin combination), there are often undesirable side effects that occur as a consequence of interaction between the core polymer and the coating polymer or between the coating polymer and the drug. Adding extra layers of different polymers can help alleviate this issue. Also, these additional coatings can provide other benefits by augmenting behaviour, such as boosting the dissolution of the drug or by modifying action of the coating. As an example, a common multi-layer device design includes a core surrounded by two layers - the inner one is a swellable polymeric layer and the outer one made up of a non water-soluble polymer. Upon water penetration, the inner coating will swell, causing the outer one to rupture. Another possible use of multiple coatings is to trap several stages of a drug between each of these, causing a multi-stage release at different moments of GI tract transition.

In our specific case, the use of a second coating on the target device was to remove the negative effects of *polymer blocking* and *core fragmentation*, discussed in previous Section, which were shown to be the major causes of incomplete Cyclosporine release from the beads. To negate these effects, a second coat of Opadry® film coating system was added to the modelled device. The Opadry layer dissolves quickly after the water penetrates the bead and enhances the gelatin and cyclosporine dissolution within the internal solvent, allowing the chains to diffuse more quickly and preventing clustering of gelatin within the bead (Colorcon Inc, 2009).

The multiple coating model (MCES) thus builds on the previous erosion and swelling models (EC- and SCDD), and we present the additional model characteristics as incremental delta terms only. It is assumed that those features not explicitly revisited here remain the same as defined in the general swelling model presented above, (SCDD). The dominant release control process in MCES is assumed to be either erosion or swelling, with the same control mechanisms (including the λ factor and water diffusion probability (P_{WG}) for the gelatin erosion, and swelling potential and swelling probability for gelatin swelling).

4.4.1 Geometry parameterisation

While the MCES model retains the geometrical characteristics of the previous models (3D space, variable geometry), the method of specifying the thickness of the coating layers is simplified somewhat by defining each layer thickness directly. Previous calculations, based on weight gains (wg), required a knowledge of average polymer density (d), which is generally unknown as beads undergo a thermal process after layering, ($d \sim f(\rho, wg)$, where ρ is unknown). The thickness of the beads under various coating configurations, (none, single or multiple), can be easily measured under lab conditions and the thickness of the coatings derived from the difference.

4.4.2 Influx of water into EC/P layer

The coating cracking mechanism, caused by gelatin expansion and noted in reference to the single coated swelling model (SCDDa), is augmented by allowing the diffusion of active substance through micro-pores caused by EC polymer separation. The transition is defined as cell state change from EC cell to a wet EC cell. The change occurs with water diffusion probability P_W, if there is at least one neighbouring solvent, buffer or another wetted EC cells. The diffusion probability P_W can be estimated by the following equation:

$$P_W = \sqrt{\frac{D_{water/EC}}{D_{CyA/water}}} \tag{4.20}$$

i.e. the square root of the ratio of speed of diffusion of water molecules through ethylcellulose relative to the overall speed of drug diffusion in water, for the given simulation setup.

4.4.3 Opadry® layer erosion

As noted earlier, incorporating a further coating requires an additional cell state and associated rule to model the coating behaviour. The behaviour of Opadry® polymer is controlled through erosion only. The role of Opadry was discussed in Section 4.4. Each Opadry cell is assigned a lifetime, similar to that of gelatin cells. However as the primary effect of Opadry is to act as a buffer between the outer coating and the inner core, it dissolves

quickly and a change to the properties of the selected solute parametric distribution is not needed. Instead, a simplified uniform probability range $(life_{lower}, life_{upper})$ is sampled to assign a lifetime to each Opadry cell. A transition rule "Opadry cell" $\rightarrow$ "wet Opadry cell" is defined and controlled via an associated probability P_{OP}. This parameter is used to delay the start of layer erosion only. A mean lifetime is assigned, based on *in vitro* observations from experiments carried out with the industrial partner.

4.4.4 Diffusion through EC coating permeated with solvent

The two approaches to modelling diffusion of drug through the wetted ethylcellulose chains used by the ECDD and SCDD models showed qualitatively good results but parameters are not straightforward to derive from experimental data. Thus, in the MCES model we chose to describe the process by a similar paradigm to that of Equation (4.20). In order to move to a neighbouring cell in the EC layer, drug packets have to overcome the P_P probability. They can only move through wetted EC cells. The same rule in selecting direction of movement, based on concentration difference, still applies (i.e. first we determine if the drug packet can move while inside a wet EC/P area and then select in which direction). The P_P parameter is critical in order to get the correct range of drug release rates between different coating thicknesses.

The additional model rules are summarised as a schematic in Figure (4.5).

4.5 Influence of dissolution environment in MCES models

Of utmost importance for drug devices which have to transition between different dissolution media is the distinction between how each environment influences drug behaviour. In augmentation of the MCES model, (and indeed any of the presented models so far, as the changes are orthogonal to the specific model rules), we can also mimic the influence of the different environment (e.g. in the case of the Sigmoid Pharma project, this includes acidic, $(pH = 1.2)$ and enteric, $(pH = 6.8)$ environments). Two parameters are proposed. The first is the relative ratio of drug diffusion coefficients between the two environments, (effectively allowing for slow-down or speed-up of the dissolution). The second is related to the presence of chemical additives in the environment, which increase the relative capability of the

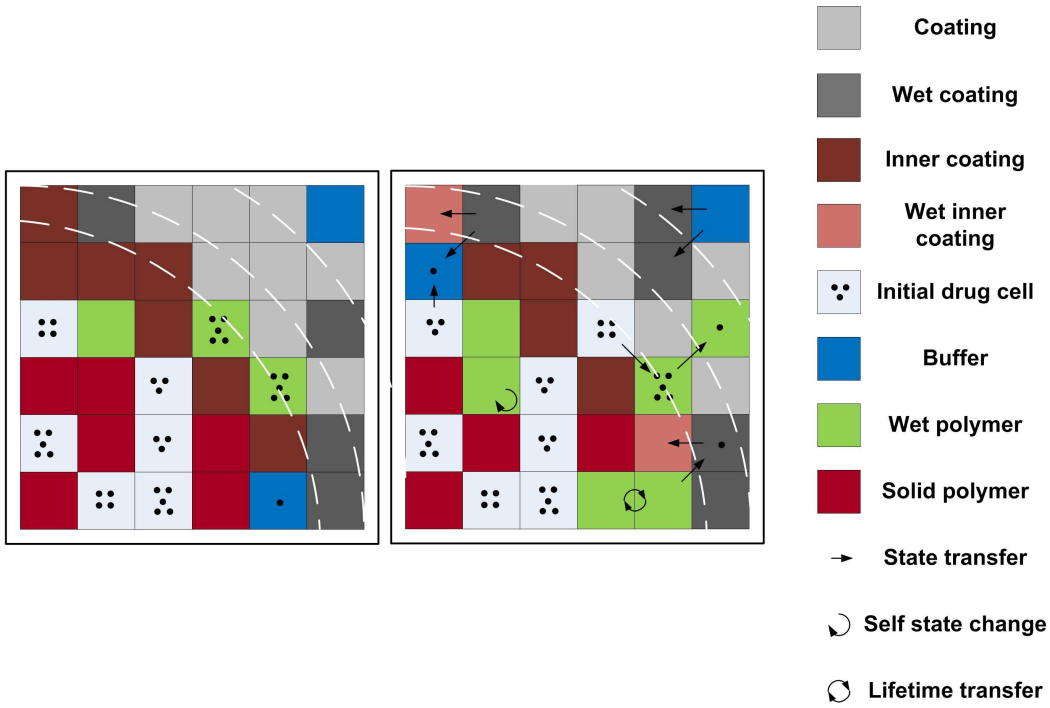


Figure 4.5: A schematic representation of the CA rules for MCES model. The additional rules of drug diffusion and Opadry erosion are displayed. Arrows show drug and water diffusion (straight), polymer erosion (semi-circular) and polymer transfer (swelling, circular).

current environment to dissolve the target drug. As an example, Cyclosporine solubility is low in acidic environments, but dissolution is fast with the addition of surfactants, such as sodium dodecyl sulfate (SDS).

We have modelled the following properties of the dissolution environment:

- **Dissolution rate** - i.e. the speed of dissolution. The parameter defines the probability of a CyA packet dissolving in given time step and it is directly proportional to the dissolution rate which can be determined empirically using a variation of the Noyes-Whitney Equation (2.4.1), given by:

$$D_R = \frac{AD}{h}\left(C_s - \frac{X_d}{V}\right) \tag{4.21}$$

where D_R is the dissolution rate, A is the surface area of drug, h the thickness of boundary layer, C_s the saturation concentration of drug in different media, X_d the amount of dissolved drug and V the volume of the active dissolution media.

- **Solubility** - the maximum amount of drug that can dissolve in a defined volume of solvent. It is defined and parametrised as the dissolved drug percentage after which no more dissolution occurs due to saturation of the media. The solubility is estimated using the formula:

$$S_{tot} = k_m(C_{SURF} - CMC) \tag{4.22}$$

where C_{SURF} is the concentration of surfactant, CMC the critical micelle[5] concentration of the solubiliser used, (for example SDS surfactant used in the Sigmoid experiments has CMC of 0.008M/l) and k_m is the molar solubilisation capacity of drug.

- **Changes in diffusivity with temperature**: Temperature changes directly influence the diffusion coefficient, so this can be partially modelled if we understand the connection between the temperature (T) and the drug diffusivity of the environment (D). For that purpose, we can use Stokes-Einstein formula to estimate D for different T:

$$D = \frac{k_b T}{6\pi\eta r} \tag{4.23}$$

where η is the viscosity of the medium, k_b is the Boltzmann constant, T is the temperature of the environment and r is the radius of the spherical particle.

4.6 A comparison of SCA and ACA in probabilistic pharmaceutical modelling

In the literature, both synchronous (CA) and asynchronous (ACA) types of automata have been used so far, (Section 3.1), without analysing their comparative impact on resulting model outputs. Choosing the appropriate update mechanism, besides having an impact on the perceived realism of the simulation, also has practical benefits for the applicability

[5] Aggregation of molecules in a colloidal solution.

of different model parallelisation algorithms and their performance when used in large-scale simulation context. High-fidelity models described in the previous sections have further stressed the need for analysing differences between the two. In this section, we compare several variants of probabilistic CA and ACA update algorithms for building models of complex systems used in controlled drug delivery. Advantages and disadvantages, related to different synchronous and asynchronous modelling scenarios, are outlined and the performance obtained reported.

In research papers on application of CA to the DDS field, synchronous and asynchronous update methods have been used, (Göpferich, 1997a; Siepmann and Siepmann, 2008; Barat et al., 2008; Laaksonen et al., 2009a), but their effectiveness has not been analysed. Choosing, the algorithm of cellular matrix iteration and the order of application of the local rules, affects how temporal realism of the physical process is represented. As a DDS is physicochemical in nature, chaotic or random updates might represent the system dynamics better than synchronous, "all-at-once", changes. On the other hand, as the size and complexity of the models grows, scalability and the need for efficient parallelisation of model space restricts use of the asynchronous methods due to performance, (Bandman, 2006b). Understanding the effects of synchronicity and asynchronicity on model outputs, can enable optimal choices to be made during model development. It is also worthwhile to examine the equivalence of substituting one iteration rule for another while maintaining model outputs such as drug release rates and device geometry changes within acceptable ranges. The transition of CA to ACA has been investigated in the literature in other modelling contexts (Kalgin, 2008; Alba et al., 2002; Bandman, 2006b).

In this part of the chapter we compare behavioural characteristics, model outputs and performance for different synchronous and asynchronous CA update mechanisms in the context of probabilistic models used in controlled drug delivery and their parallel implementation, (where differential equations are not applicable due to the inherent unknowns in the parameter space). We introduce the mechanisms as components of a wider meta-model presented and describe how their logic fits into the CA system.

Section 4.6.1 presents the design methodology used for developing the CA rule sets, together with comparison of different CA and ACA update mechanisms when used in the

context of the meta-model and a theoretical analysis of their properties and variations in parallel and sequential implementations. We summarise findings in the final section and consider directions for future improvement.

4.6.1 CA and ACA modelling

Design methodology

As for any model build, the first stage involves transfer of the domain knowledge of the structural and behavioural characteristics of the DDS to the CA model. Figure (4.6) outlines the main modelling classes of interest, where the DDS consists typically of different polymeric coatings, carriers or matrices, (Bezbradica et al., 2011). There are several distinct DDS characteristic categories to be considered:

- the shape and geometry of the system (slab, cylindrical or spherical) - this is captured in the shape and size of the model cellular matrix;
- the polymer composition of the device defined by states of the matrix cells;
- the polymer physico-chemical interaction mechanisms (laws) - described by characteristic behaviours that occur inside the DDS;
- the drug loading and initial dispersion within the device;
- the influence of the dissolution environment on polymer behaviour.

The models, with above characteristics, are classified as *kinetic* CA or ACA models (Bandini et al., 2010). Based on the way we choose to represent the physical phenomena modelled, we can adopt rules, which are either deterministic or probabilistic, (or a combination of both) affecting the individual cell behaviour and the surrounding neighbourhood. Both classical CA neighbourhood mechanisms (Von Neumann and Moore) can be used for either model type.

Another important aspect of the meta-model in context of the CA or ACA used is the establishment of the initial configuration and dispersion of elements within the modelled device as well as the existence of boundary conditions. Although it is common for various

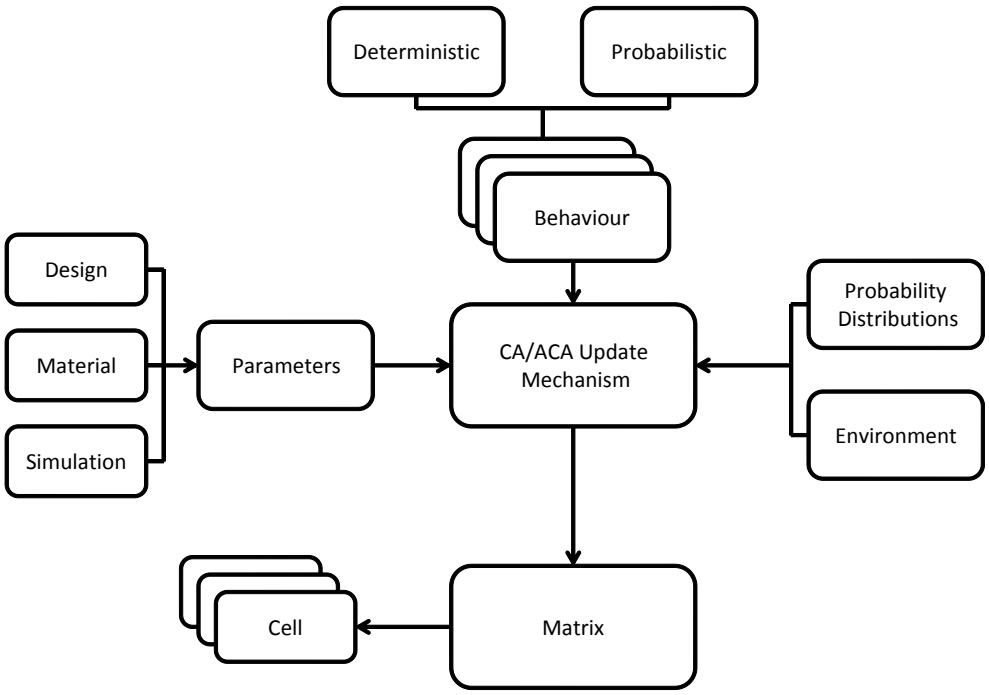


Figure 4.6: Meta-model diagram with detailed components.

families of CA update mechanisms to operate under periodic boundary conditions (Xiao et al., 2005; Voorhees, 1990; Burstedde et al., 2001), pharmaceutical models benefit from using the fixed equivalent, as drug movement across the matrix boundaries is used as a direct method for calculating release rates. To satisfy the condition that the models need to mimic the inhomogeneous nature of the physical device, with exact distributions not available from experimental data, the meta-model establishes the initial cell states using stochastic distributions within the known device geometry. The stochastic aspect usually has a lower and upper limit which is deduced from *in vitro* experiments, and the distribution of values within the limits follows a probabilistic distribution, chosen to mimic known chemical properties of the material, (for example its diffusion coefficients or degradation rate). Therefore, various direct Monte Carlo algorithms can provide a natural solution to the initial condition problem, by establishing a random starting state for each simulation run. The stability of the obtained models can thus be tested, by validating that variation in model outputs is not statistically significant for multiple runs using the same parameter set.

4.6.2 Update methods

Crucially, a good description of the model dynamics, i.e. the closest qualitative match to the behaviour of the pharmaceutical system modelled, relies on choosing the appropriate update method of the cellular automaton itself. While both synchronous and asynchronous modelling have been used for this type of systems, an evaluation of the influence on model performance and correctness is lacking. This is addressed here, with particular emphasis on the correctness of the update rules as we consider several standard synchronous and asynchronous CA update methods (Schönfisch and de Roos, 1999; Bandini et al., 2010; Baetens et al., 2012).

Mathematically, the principal features of the 3-dimensional DDS models can be represented as a cellular automaton using a tuple representation, given by:

$$ACA = \{G, A, U, \Theta, F\} \tag{4.24}$$

where G denotes a set of cell coordinates (the model matrix in our meta model). In the case of a three-dimensional system:

$$G = Z^3 = \{(i, j, k) | 1 \leq i \leq N_x, 1 \leq j \leq N_y, 1 \leq k \leq N_z\} \tag{4.25}$$

A is the model alphabet - a finite set of possible cell states (the aggregate polymer states in the conceptual model) and U denotes the cell neighbourhood (including the cell itself). Then $A\left(U(x), t\right)$ denotes the state of a neighbourhood of cells U around a given cell x at a moment in time t. The behaviour of the system is described by a set of elementary transition rules for the conceptual model, F, where these are applied to the states of a neighbourhood of cells U. For the **sequential case of synchronous updates**, we can write the following:

$$F_s = \{f(x) : A(U(x), t) \rightarrow A(U(x), t + 1) | x \in G\} \tag{4.26}$$

Finally, Θ denotes the CA/ACA update order function, applied to G and A in order to advance the global model state. As a basis for Θ we investigated the application of

several random and ordered asynchronous update methods, (see e.g. (Cornforth et al., 2005; Valsecchi et al., 2010)) and compared these to the well-used synchronous method.

For a clearer description of the implementation of the different forms of Θ, we move from the mathematical to the algorithmic description:

- We implement the different update forms as modifications of the basic **synchronous CA two-phase update algorithm** of the main matrix G (Table 4.3a);
- The first ACA update method investigated is the **random order algorithm** which involves updating cells of G in a random order which is changed every time a full cell sweep is finished. All cells are updated in each time step of the simulation (Table 4.3b);
- The **random cyclic algorithm** is a variation of the random order algorithm, with the difference that a single random permutation of G is always used (a *cycle*). Random permutation of G is chosen at the beginning of the simulation (Table 4.3c);
- In the **random independent** method (Table 4.3d), one cell is chosen at random for updating at each time step. In the overall simulation, each cell should thus be updated approximately the same number of times. However, over shorter time periods a given cell may be updated significantly more often. To achieve uniform selection, the algorithm thus depends heavily on the size of the cycle of the random number generator implementation. The *Mersenne Twister* algorithm has been used in this case, to reduce bias (Matsumoto and Nishimura, 1998);
- Finally, the **fixed cyclic sequential algorithm** (Table 4.3e and 4.3f) was used in two forms: in the first one, cells of G are visited in sequential order of their coordinates, (first width, then depth, then height in 3D). In the second form, cells are sorted based on their state (A), so that polymers of certain types are given simulation priority over polymers of other types (outer coating, then the inner coating, then the core). The order of cell simulation within the same polymer type is sorted by its coordinates.

(a) Synchronous algorithm:

for each time step t
  $S \leftarrow$ random permutation of cells in G
  $S_d \leftarrow$ temporary copy of S

  for each cell x in matrix S
    for each rule f in F
      $A(U(x), t+1)|_{S_d} = f(U(x), t)|_S$

  for each cell x in matrix S_d
    $A(U(x), t+1)|_S = A(U(x), t+1)|_{S_d}$

(b) Random order algorithm:

$S \leftarrow$ random permutation of cells in G

for each time step t
  $x \leftarrow$ next cell from S
  for each rule f in F
    $A(U(x), t+1)|_S = f(U(x), t)|_S$
  if $x =$ last cell from S
    $S \leftarrow$ random permutation of cells in M

(c) Random cyclic algorithm:

$S \leftarrow$ random permutation of cells in G

for each time step t
  $x \leftarrow$ next cyclic cell from S
  for each rule f in F
    $A(U(x), t+1)|_S = f(A(U(x), t))|_S$

(d) Random independent algorithm:

for each time step t
  $x \leftarrow$ random cell from G
  for each rule f in F
    $A(U(x), t+1)|_S = f(A(U(x), t))|_S$

(e) Fixed cyclic sequential algorithm (1):

for each time step t
  $x \leftarrow$ next cyclic cell from G
  for each rule f in F
    $A(U(x), t+1)|_S = f(A(U(x), t))|_S$

(f) Fixed cyclic sequential algorithm (2):

$S \leftarrow$ type-sorted set of cells from G

for each time step t
  $x \leftarrow$ next cyclic cell from S
  for each rule f in F
    $A(U(x), t+1)|_S = f(A(U(x), t))|_S$

Table 4.2: Synchronous/asynchronous Cellular Automata update algorithms.

Equivalence of Sequential and Parallel Implementations

In a concrete, model execution context, the mechanisms presented above only apply as long as the simulation is *sequential*. Once the algorithm has to scale up to be applicable to large data sets, the inclusion of parallelisation will have fundamental impact on the update logic. It can be shown that in our case synchronous matrix updates are more suitable for parallelisation, as the effect of parallel updates on the resulting state should be equivalent. Consider a parallel version of the fundamental rule set given in (Equation 4.26):

$$F_p = \{f(x_1, ...x_n) : \bigcup_{i=1}^{k} A(U(x), t) \rightarrow \bigcup_{i=1}^{k} A(U(x), t+1) | x_1, ...x_n \in G\} \tag{4.27}$$

Essentially, parallelising the update mechanism by splitting the CA space into disjoint domains, each having a set of boundary cells, introduces a simultaneous update of k cells at a time, where n represents the degree of parallelisation. The exact selection of cells x_1, ... x_n depends on the particular parallel algorithm being used. In the synchronous case, the state of a neighbourhood of cells $U(x)$ at moment t depends only on the same state for the previous instant $t-1$, and not on any currently updated state of any of the other neighbourhoods. Therefore, for synchronous updates, it holds that $F_S \Leftrightarrow F_P$, which is in line with Bandman (2006a).

For asynchronous updates, the equivalence of sequential and parallel implementation breaks down. As the parallel version of the rule set presents a set of functions applied simultaneously, the order of their application can result in a different overall state of the matrix. This is always true if any of the chosen neighbourhoods $U(x_i)$ overlap.

Therefore, in the case of random order and random cyclic updates, we expect the parallelisation to always result in a slightly different model output. The same holds for different variations of *fixed cyclic rules*, presented in Table 4.3, where the idea of the underlying algorithm essentially breaks down. The only exception to this rule is the *random independent* order of updates, which might produce equivalent results, but only if, in each individual iteration, cells x_i are chosen from each parallel domain so that their neighbourhoods $U(x_i)$ are non-overlapping.

From an implementation point of view, for parallelisation of pharmaceutical models using some of the industry standard parallel APIs, such as Message Passing Interface (MPI), synchronous updates are preferable from the execution speed point of view, as simulations have a practical wall-clock time limit of 24hrs, (the amount of time it would take to run a single *in vitro* experiment). Synchronous updates are extremely efficient in terms of execution speed, especially as they can utilise two-sided communication using MPI to send and receive primitives. Asynchronous parallelisation schemes require one-sided communication primitives such as MPI "put" and "get", utilising the remote memory access mechanism (RMA), which, although slower, allows for the cell state to be asked for or provided on demand, without the need to wait on some eventual update (Thakur et al., 2004).

Finally, it is important to note, according to Toffoli and Margolus (1987) that, for relatively slow changing stochastic CA models, the expected variance in outputs between synchronous and asynchronous update methods would be small. This results from the fact that large-scale, low-probability models do not have too many cell state updates in each iteration, which in turn limits the number of cases where overlapping neighbourhoods are updated.

4.6.3 Model properties

Table 4.3 revisits the main CA rules used, in the meta-model described, with respect to their impact on neighbouring cells and the type of update performed (probabilistic or deterministic). Following the notation from Section 4.6.2, each of the transition functions is applied to an alphabet of CA states:

$$A = \{P_{COAT}, P_{CORE}, S, D\} \tag{4.28}$$

where P_{COAT} and P_{CORE} denote the coating and core polymers respectively, S represents solvent cells and D drug molecules. The rules can also be described using a more formal notation:

- **erosion:** polymer lifetime of a given cell x ($l(x)$) decreases linearly with time according

Phenomenon	Rule behaviour	Type
Erosion	Applies to all polymer cells. Amount of polymer present represented through randomly distributed cell lifetime. Polymer degradation directly mapped to lifetime decrease.	Initial state - probabilistic; Subsequent updates - deterministic; Neighbour-independent.
Diffusion	Applies to drug carrying cells. Drug moves from cells with higher concentrations to cells with lower concentration. Destination cell chosen with proportional probability. Initial drug dispersion set using probabilistic distribution.	Initial state - probabilistic; Subsequent updates - probabilistic; Neighbour-dependent.
Swelling	Applies to core polymeric cells. Amount of polymer present in cell is transferred to a neighbouring cell according to concentration gradient. Destination cell chosen with proportional probability.	Initial state - derived from erosion; Subsequent updates - probabilistic; Neighbour-dependent.
Dissolution	Applies to cells outside of drug device boundaries. Drug is removed after it crosses the boundary. Amount of drug removed depends on set probability.	Initial state - derived from diffusion; Subsequent updates - probabilistic; Neighbour-independent.

Table 4.3: Elementary CA rule matrix for the meta-model together with descriptions and type quantifiers.

to the following function: $f_e(x) : \{l(x) \rightarrow l(x) - t | \forall x \in \{P_{COAT}, P_{CORE}\}\}$

- **diffusion:** the amount of drug present in cell x_a ($d(x_a)$) partially transitions to a neighbouring cell x_b with probability p_{diff}: $f_{diff}(x_a, x_b) : \{d(x_a, x_b) \xrightarrow{p_{diff}} d(x_a) - \Delta d, d(x_b) + \Delta d | \forall x_a \in \{D\}, x_b \in U(x_a), \Delta d > 0\}$

- **swelling:** the amount of polymer present in a cell is distributed in a similar fashion, using probability p_s: $f_s(x_a, x_b) : \{l(x_a, x_b) \xrightarrow{p_s} l(x_a) - \Delta l, l(x_b) + \Delta l | \forall x_a \in \{P_{COAT}, P_{CORE}\}, x_b \in U(x_a), x_b \in \{P_{COAT}, P_{CORE}, S\}, \Delta l > 0\}$

- **dissolution:** finally, the process of partial or total drug dissolution is described as the reduction in drug molecule count of a given cell once it transitions to solvent state: $f_{diss}(x) : \{d(x) \xrightarrow{p_d} d(x) - n | \forall x \in \{S\}, n \leq d(x)\}$

Multiple rule combinations can be superimposed (e.g.$f(x) = fe(f_{diff}(f_s(x)))$) to fully define a cell behaviour during a single iteration if the given cell state satisfies all the alphabet preconditions of the rules given in Equation (4.28).

4.6.4 ACA to CA transition

The initial model developed, ECDD (Section 4.2), used a random order ACA update scheme, (Bezbradica et al., 2011). This was used for two reasons, (i) to more faithfully represent the nature of chemical processes occurring inside the device (ii) for performance reasons, with the initial model using thread-level parallelism, deemed more suitable for ACA update types as cell state exchange is not as expensive as in process-level implementations.

However, since further model extensions and increasing complexity required a transition to synchronous updates in order to achieve faster running times, certain compositions of elementary rules were found to be no longer applicable (Bezbradica et al., 2012). As rule applications in ACA lead to an immediate change in the matrix state, readily visible to all other rules, the possibility of having "undefined" states, where two consecutive state changes lead to an invalid cell neighbourhood in the next iteration, cease to apply. With synchronous models, on the other hand the possibility exists that an elementary rule changes the state of the cell in the next iteration, such that subsequent executions of other elementary rules on the same cell no longer apply. For the case of the erosion rule, the complete loss of

polymer now has to block further swelling updates originating from the cell, even though the swelling rule would be applicable to the cell state in the current iteration. Accordingly, diffusion has to adapt to its valid neighbours changing to the polymer state due to swelling in the next iteration, which would then make any drug movements invalid.

4.7 Summary

In the first part of this chapter we incorporated all main dissolution phenomena, (defined in Chapter 2, Section 2.1), into our meta-model. As such, we created a novel comprehensive and complex modelling framework which allows easy extrapolation of a base model to different desired geometries, dissolution scenarios or device structures used in complex drug design, (and with capability well beyond that required by the Sigmoid Pharma project). Within these models, we have been able to combine stochastic prediction of outcomes, with empirical information, from observation of *in vitro* experiments, in order to access the impact of parameter changes.

The rationale for taking this approach was to address certain limitations of earlier model applicability to complex structures, (such as coated swellable formulations) and to supplement sparse *a priori* experimental data needed for calibration.

In the second part of this chapter we presented an overview of methodological considerations, important to modelling drug delivery systems using CA and ACA, and have analysed the advantages and disadvantages of each update method. While some flexibility is possible in choosing between asynchronous and synchronous methods for approximate solutions in this context, this is governed by structural requirements.

Chapter 5

Model Implementation and Analysis

Currently, a "hot-topic" in computing, *Data Analytics* typically refers to the use of *large scale computing* to solve problems involving "big data". The use of Data Analytics in drug design and discovery was of limited use historically, as the process of *in vitro* drug release was considered akin to a black box, where different inputs were varied in order to tailor desired outputs[1]. In this sense, the internal dynamics of the pharmaceutical device could not be deduced solely by observation as this presents a superimposed picture of the phenomena involved.

As described in Chapter 1 pharmacokinetic modelling of drug release from a DDS, from the drug discovery phase through to development and manufacturing, provides both support for the decision-making process as well as an improved and effective usage of drug development time together with reduction of design parameter space. The basis for analyses both within and between input data sets is also covered. This has the potential to reduce the amount of *in vitro*/*in vivo* testing required, important to cost-effective and flexible solutions. Hence, *in silico* modelling and analysis could enable extensive offline testing and evaluation of parameter sensitivity, (Bader, 2012).

In our case, *in vitro* and *in silico* data analysis and development of the current models were performed based on the workflow presented in Figure (5.1). This continuous feedback loop is composed of several distinct stages, separated into two main parts (enterprise partner and modeller).

[1]This work has been published in (Perrin et al., 2012).

The initial drug design and *in vitro* testing in the USP II apparata (The United States Pharmacopeial Convention, 2007), has been performed by the enterprise partner. The data are filtered to determine the variables of interest, intrinsic to the drug device design process, (input parameters such as geometry, size, drug loadings, and coating thicknesses), and relevant to the modeller (e.g. the types of physical processes/chemical reactions present, release curves for different parameter sets, etc.). The data obtained were used in the construction of the CA models presented here and shed light on different assumptions about the dissolution processes.

The models are then optimised for parallel execution and simulated on high performance computing clusters for various input data sets, in order to maximise the gain from running many simulations at once. As the laboratory testing is performed for periods of up to 24 hours, in order for *in silico* simulations to be cost-effective, the workflow and the underlying framework must be able to support large numbers of simulations over a relatively short time period.

Two main sets of simulation results are generated: (i) a control set, which validates results against the known experimental data points and is used to assess model usability for predictive purposes. For controlled release, a sigmoidal ("S" Shaped) release curve is expected, demonstrating properties of zero-order release kinetics during a period when the device passes through a region of maximum adsorption; (ii) a generic data set showing variations of release profiles, dependent on parameter changes, which can then be used to feed back into the drug design process.

If the model validation shows promising results, extensive optimisation can also be performed in order to ensure model robustness and reusability for similar drug device analysis. An insight on system evolution is also provided by graphical representation of the model, which captures successive stages of the dissolution.

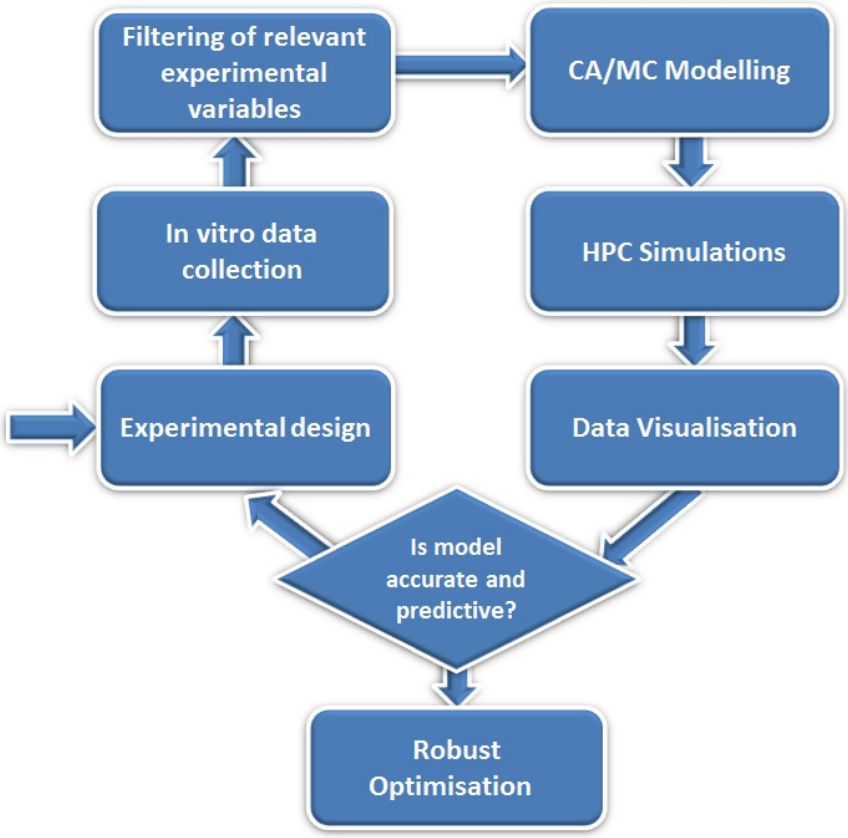


Figure 5.1: Data analysis and modelling workflow. The workflow starts at initial design of the pharmaceutical device and preliminary *in vitro* testing. The resulting data are filtered for the experiments that include relevant experimental variables, and are used to construct the CA models. High Performance *in silico* simulations and data visualisation are then used to investigate the influence of parameter change to the release profiles. If the control results accurately describe the experimentally observed data, the simulated scenarios can be used to reduce the space of further experimental designs.

5.1 Parallelisation strategies for large scale Cellular Automata frameworks in pharmaceutical modelling

Here, we focus further on the implementation aspects of the simulation stage, with resulting data analysis and visualisation covered in subsequent chapters[2]. One of the main challenges in pharmaceutical data analytics is that uncovering structures and patterns in complex, heterogeneous datasets is a computer-intensive task, requiring significant computational resources. Recent advances in high-performance computing provide part of the solution. Multicore systems are now more affordable and more accessible. Advances

[2]This work has been published in (Bezbradica et al., 2012).

in high-performance computing (HPC) have resulted not only in improved performance but also in greater availability of such resources. The cost per GFLOPS[3] dropped below $1000 for the first time in 2000, below $42 in 2009, (Nakasato, 2009), and below $2 in 2011, (Stevenson et al., 2011). Ongoing development of cloud-based solutions is likely to further increase availability and affordability of HPC resources. HPC solutions are already having significant impact on *Data Analytics* in a range of disciplines. This served us the motivated for providing an overview of the challenges and present a detailed case study for pharmaceutical R&D, since practical applications are lacking in the context of optimisation and parallelisation of large-scale, high-precision, high-fidelity DDS CA simulations. The key obstacle to parallelisation of pharmaceutical models is not only the amount of data involved, but also the fact that many of these models incorporate agent-like behaviour within the CA framework in order to describe pharmaceutical components. This makes communication across process boundaries expensive.

In this chapter, we investigate how to develop more advanced methods for data analytics and focus on a specific case of model-driven analysis. We apply two parallel-computing application programming interfaces (APIs), namely OpenMP and MPI, to partition the simulation space and analyse the applicability of each API to the problem individually, as well as in the hybrid solution. We examine the speedup potential and overhead for local and global communication for simulation speed and solution scalability.

5.1.1 Advantages of High Performance Computing

If we are speaking in Data Analytics terms, there are two sources of "big data" in general in pharmaceutical analysis. The first is due to novel drug formulations generating considerable experimental data from the design stage (Coulter, 2010), which require processing in order to deduce relations between design parameters. Synthetic sources depend on the device, where fine resolution is needed for micro- and nano- scale simulations, requiring high performance, large scale infrastructure to execute. Even though such resources today are readily available, either from local scientific computing clusters or commodity public cloud infrastructure, the algorithmic solutions are less well-reported in the scientific literature,

[3] 10^9 floating point (i.e. real number) operations per second.

(Bader, 2012).

A second source of data comes from the problem of implementing parallelisation of CA systems themselves, where the key challenge is that of the efficient transfer of large amount of cell state information, which contain high-precision data, across process boundaries where communication is expensive, (Bezbradica et al., 2012). This problem becomes more complex for multiple scales as we switch from classical CA models, to ones incorporating agent-like behaviour, (such as those of particle diffusion within the drug device).

Depending on the available infrastructure and on the need for granularity, the general model framework, described in Chapter 4, can be adapted for different parallelisation strategies, (Bezbradica et al., 2012). On one level, *process level parallelism*, which requires more extensive computing resources, can be used at the development and analysis stages, for processing of large data sets of experimental data, in order to reduce input requirements from wet-lab experiments. Lighter, *thread*[4] *level parallelism* can be utilised even under laboratory conditions and is adequate for further analysis of single parameter ranges. Finally, there is a possibility of adopting a hybrid approach, allowing selection between the two strategies, which maximises the utilisation of available high performance infrastructure by applying both thread and process level parallelism, (Silberschatz et al., 2012).

Inter-thread and inter-process communication overhead occurs as a consequence of the cell state being exchanged during an execution rule and is solved by introducing boundary cell layers, where communication is either sequentialised, (i.e. uses locking[5]), for the case of threads, or delegated to one of the two neighbouring processes for process level parallelism.

5.1.2 Parallelisation Schemes

In order to develop the most efficient parallelisation algorithm for the CA model presented, we need to understand the fundamental types of cell communications and cell transfers occurring. Traditional synchronous CA models usually follow a standard pattern where a cell state is passively dependent on its neighbours and, based on their state values, is

[4] In a programming context, *threads* represent a lower level parallelisation abstraction than *processes*. A process would represent a general sequence of execution (i.e. an application) while a thread would represent a sequence of execution within the process itself. Each process can have multiple threads.

[5] *Locking* is a general thread programming concept, where given thread blocks waiting for another thread to perform a dependent action.

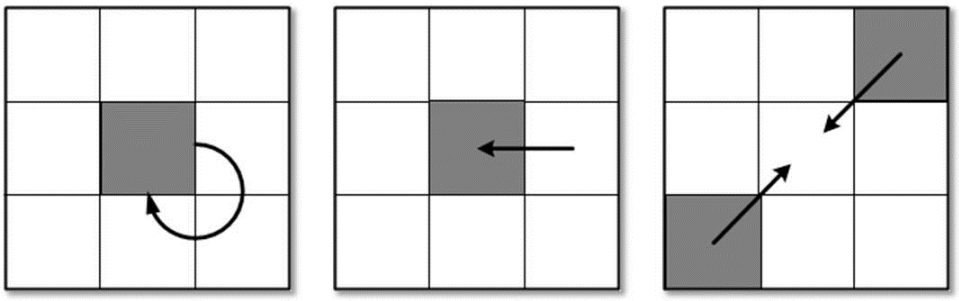


Figure 5.2: Elementary Types of CA Rules present in the Model.

updated itself in the following iteration. These kinds of models should fit into the category of so called "embarrassingly parallel" problems, where parallelisation of the problem space is straightforward, (Massaioli et al., 2005). However, differences in the rule types in our case mean that certain types of shared state and synchronisation between different parallel processes is necessary.

Rule Types

In the models presented in the earlier chapters, each cell has a state describing the current status of several different physico-chemical phenomena:

- The amount of degradation of polymer chains;
- The extent of expansion potential of polymer chains;
- The number of drug particles present in the cell.

These sub-states are mutually independent and can be updated separately from each other. Thus, we cannot treat whole cell state updates as "atomic"[6], only sub-state ones. Therefore, every single cell has the potential to be updated several times during a single simulation iteration.

The model rules that affect those sub-states can generally be classified into three fundamental types (Figure (5.2)):

- Temporal rules which change the state of a single cell without neighbour influence. This type of rule is always deterministic (e.g. polymer degradation over time)

[6]The *atomic* instruction is "indivisible". It cannot be interrupted before its completion. For example, if the CPU attempts to read from the memory it can only see operation as completed or not and will not be able to read its inner state.

- Neighbourhood-dependent rules which change the state of a single cell, based on the combination of states from neighbouring cells. This type of rule can either be deterministic (polymer erosion) or probabilistic (polymer wetting)

- Neighbourhood-dependent rules which change both the state of the cell and the state of one of its neighbours. This type of rule is always probabilistic (e.g. drug diffusion, polymer swelling).

It is also important to note, that conflicts can occur in shared cell updates of different types (i.e. swelling and diffusion or erosion and swelling). In this case, we define a permitted order of cell updates, where a state change will block further updates of certain kinds. For example, erosion of a cell which degrades all the remaining polymer will further block any swelling rules which can be executed from that cell.

The particular obstacle to overcome for parallelisation of the model to be efficient is *effective communication* of the third type of rule across parallelisation boundaries. Recall that these rules were introduced to realistically simulate the physico-chemical processes which have an active spatial span, as opposed to being passively affected by the neighbour state. However, the ability of the cell to affect the neighbour state in a probabilistic manner leads to the possible occurrence of multiple updates of a single cell from two or more of its neighbours (Figure (5.2)). To resolve possible conflicts several approaches may be used:

- The neighbouring cell can be marked as updated for all sub-states and no further state changes permitted;

- The neighbouring cell can be marked as updated for a single type of sub-state change (e.g. polymer swelling), but other types of state changes (e.g. drug diffusion) may be permitted;

- The neighbouring cell state can be updated multiple times and all types of state changes may be permitted. In this case, changes are considered to be additive.

In order to preserve the realism of the physical process, all models generally assume the third approach; however there can be cases where the neighbouring cell is precluded from a certain type of update (e.g. due to change of state). To illustrate the point let us consider

Algorithm 1 *Update*(Matrix M)

```
1: Secondary = M
2: shuffled_cell_vector = Shuffle(M)
3: for each c in shuffled_cell_vector do
4:     for each rule in rule_list do
5:         Apply(c, rule) ⇒ Secondary
6: M = Secondary
7: ComputeRelease(M)
```

Table 5.1: The pseudo code for the basic model algorithm.

two examples. Example 1: the drug diffusion algorithm selects the same target cell for drug packet movement from two different neighbours. In order to correctly describe the resulting movement, the destination cell will contain packets from both sources. Example 2: polymer expansion into the coating layer has caused a neighbour state change which prevents any further updates to the destination cell. In general, for the latter updates, we assume a "priority approach" based on which cell caused the first change to the destination cell state. Any subsequent attempts to change the state are ignored. However, from the parallel implementation perspective, this implies both the need for cells to communicate their state across to that of the destination cell, as well as some kind of "locking access" to the destination cell during the update process, in order not to cause partial or incorrect updates and thus loss of data.

5.1.3 Model Parallelisation Algorithms

The main model algorithm uses two 3D matrices, primary and secondary, to hold the state of the dissolution space in the current and next iterations respectively. The algorithm performs multiple pass visits of each cell in the primary matrix. The cells are visited in random order, using a shuffle approach, to avoid any potential bias or rule cycle. In the first pass, when a cell is visited, all possible behaviours defined for the particular cell type are evaluated, and applied based on calculated probabilities. The resulting state is then updated in the secondary matrix. When the first pass is finished, the matrices are swapped,

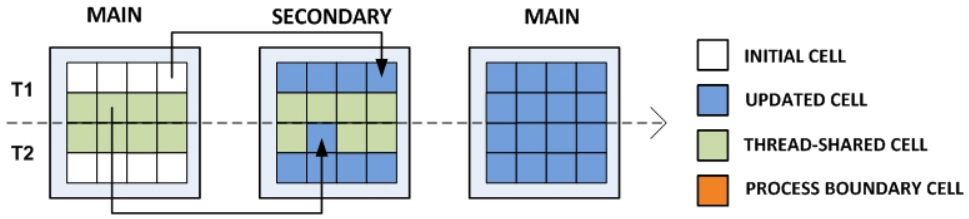


Figure 5.3: Parallelisation Strategies - OpenMP. (Left) Primary matrix is divided into equal regions with boundary (thread-shared) layers, each simulated by a dedicated thread (Centre) Non-boundary updates are applied directly to the secondary matrix, while the boundary updates are exchanged between the threads. (Right) The updated state is gathered back to the primary matrix.

and a second pass is performed to calculate the resulting global state of interest to the modellers, such as new dissolution boundary fronts, and overall amount of drug dissolved. Since only one pass occurs at any one time, there are no conflicts in cell updates. The algorithm pseudo-code is described in Table 5.1.

Thread-level Parallelism (TLP)

The first approach we consider, for solution of the efficiency problem, involves thread-level parallelisation on a symmetric multi-processing (SMP) architecture. The main matrix is kept as is, and the secondary matrix is divided into logical regions of equal size, where updates in each region are controlled by a separate execution thread (Figure (5.3)). Since the matrix state is shared in memory and visible to all threads, we now have the potential for conflicts in cell state updates. In order to minimise the negative effects of explicit thread synchronisation, cell state locking is considered only at region boundaries. Namely, two boundary layers on each side of the region, except in the case of the first and last region, are considered to be "shared" and locking is enforced for any state changes in that region during the single update period.

However, other threads are permitted to make any updates that are considered legal by the rule set, after the lock is lifted. The pseudo-code for the algorithm modification is given in Table 5.2. The standard OpenMP API primitives have been used as a means of implementing the parallelisation of the *for* loops in the main algorithm.

Although the approach provides a straightforward algorithmic solution to the problem, SMP architectures are generally limited by the number of available execution cores, and

Algorithm 2 *UpdateByThreads*(Matrix M)

```
1: Secondary = M
2: parallel for each region in M.Regions
3:     shuffled_cell_vector = Shuffle(region)
4:     for each c in shuffled_cell_vector do
5:         if(IsShared(c) == true) then lock(Secondary(c))
6:         for each rule in rule_list do
7:             Apply(c, rule) ⇒ Secondary(region)
8:             if (IsShared(c) == true) then unlock(Secondary(c))
9: #end parallel
10: M = Secondary
11: ComputeRelease(M)
```

Table 5.2: The pseudo code for the thread-level algorithm.

performance gain, i.e. speedup, from this type of parallelisation has a "hard ceiling" set by the underlying architecture. Hence, in order to improve the execution times further, we have to cross the process and machine boundaries.

Process-level Parallelism (PLP)

In order to split the algorithm execution across the different execution nodes, the focus shifts to solving the communication problem instead of the synchronisation one. In distributed computing platforms, such as MPI, a state has to be explicitly shared by sending and receiving over the communication network. This network now introduces an additional variable in the model performance, and necessitates the definition of a communication pattern between different model regions.

Generally, the CA model space can be efficiently sent to separate execution nodes using the explicit MPI commands $MPI\_Scatter$ and MPI_Gather, i.e. "scatter-gather" algorithm, (Kandalla et al., 2010). Regions of equal size are sent (scattered) from a "master" process, responsible for overall state management, to two or more "slave" processes, which are solely responsible for the simulation of a single region. After the regions have been simulated, their state is returned (gathered) back by the master process, which is responsible

Algorithm 3 *CrossProcessUpdate*(Matrix M, int rank)

```
1: Secondary = M
2: n = Secondary.Regions.Count
3: Matrix region
4: if (rank == MASTER) then
5:     Scatter(M.Regions) ⇒ region
6: for each c in Shuffle(region) do
7:     if(IsBoundary(c) == true) then skip
8:     ApplyRules(c, rule_list) ⇒ Secondary(region)
9: ExchangeBoundary(rank, rank + 1)
10: for each c in Shuffle(region.boundary) do
11:     ApplyRules(c, rule_list) ⇒
12:         Secondary(region.boundary)
13: if (rank == MASTER) then begin
14:     Gather(M.Regions) ⇐ Secondary(region)
15:     ComputeRelease(M)
16: end
```

Table 5.3: The pseudo code for the process-level algorithm.

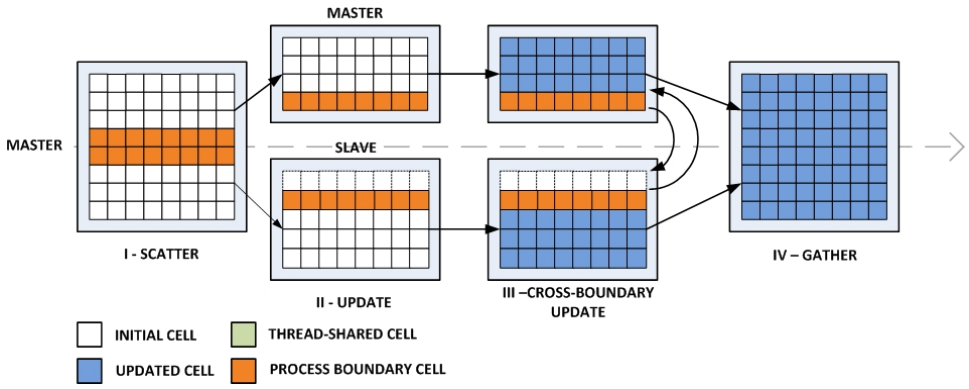


Figure 5.4: Parallelisation Strategies - MPI. (Left) The simulation space is divided into separate processes (master and a number of slaves), with boundary layers shared between them (Centre) After internal cell state updates, neighbouring processes exchange boundary layer information (Right) After the boundary calculations are completed, the state of the simulation space is gathered back by the master process.

for assembling them back together and calculating the resulting output. It is interesting to note that, due to the fact that the master process is idle during the period between the scatter and gather phases, it also assumes the role of the slave, by assigning itself as owner of one of the regions to update.

In our particular case, the scattering of the regions is not enough, because regions share logical space of the CA framework, so that a mechanism to send boundary cell states across this *logical space* needs to be present. Although MPI allows for asynchronous data communication that can be used on a per cell basis (i.e. a cell sends out updated state information as soon as it is changed), this places a prohibitive performance strain, both in terms of the amount of micro-communications that need to occur and due to the fact that any boundary cell update would need to wait on neighbour boundary state updates in order to continue with the execution. To avoid this, we choose to serialise the boundary layer updates to the end of the region update cycle. As with the concept of thread-level shared layers, we introduce *process-level shared layers*. These are constructed of boundary layers in each region. As illustrated in Figure (5.4), the update of the boundary layers is now the sole ownership of just one of the neighbouring process regions that share those layers. Upon completion of region updates, which can be done without the need for data from neighbouring processes, boundary layer information is sent to the "owner" process which completes the region update cycle, and returns the updated state back to the initial

process. The main cycle then completes with the slave processes sending back the resulting state to the master. The resulting pseudo-code is presented in Table 5.3: for brevity, the function *ApplyRules* replaces the behavioural "for loop" from previous code examples, which was responsible for rule applications.

Although this approach allows a much larger parallelisation ceiling, the fact that it involves cross-process communications, usually made across a physical network, means that the bandwidth speed is the main bottleneck that impacts performance. Even when executing on the same node, using inter-process communication, the overhead can be particularly large, especially considering the amount of data present in the model.

Hybrid Parallelism (HP)

The advantages and disadvantages of the proposed parallelisation strategies naturally lead to the solution where both are combined in order to minimise the negative impacts of each (Jin et al., 2011). On the one hand, we want to avoid the "hard ceiling" present in the SMP solution by leveraging cross-process communication, while on the other, we want to minimise the communication overhead present in the message-passing paradigm. In order to achieve this, we execute one process per node via MPI, but implement the threading within each region by using OpenMP. As a result, the process-level communication is reduced to an absolute minimum, while the limit to the amount of cores that can be used is drastically increased. Of course, communication overhead and non-parallelisable elements of the model will result in diminishing returns as the number of cores increase. Also the model geometry will be the ultimate limit, as the number of cores approaches the number of layers present. The resulting hybrid pseudo-code algorithm is shown in Table 5.4 and its schematics are given in Figure (5.5).

The central part of Figure (5.6) shows the CA rule type executions described in the context of the hybrid model. The top panel of the central diagram represents the diffusion process through three distinct update scenarios. In scenario 1, we have the normal case, isolated (i.e. independent of shared space) rule execution, where a set of drug molecules can choose to move to a suitable neighbour. Scenarios 2 and 3 display rule updates which occur inside the *shared layer*. In this case state locking is performed, and cell access is

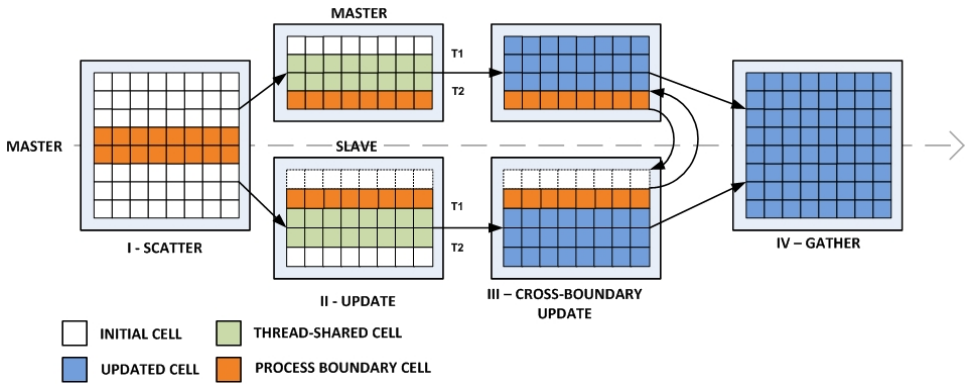


Figure 5.5: Parallelisation Strategies - Hybrid. (Left) The simulation space is divided across master/slave process with each process further dividing the space over multiple threads (Centre) Internal state calculations are performed by each thread in each process with the boundary threads exchanging the state information. (Left) The resulting state is gathered by the master process.

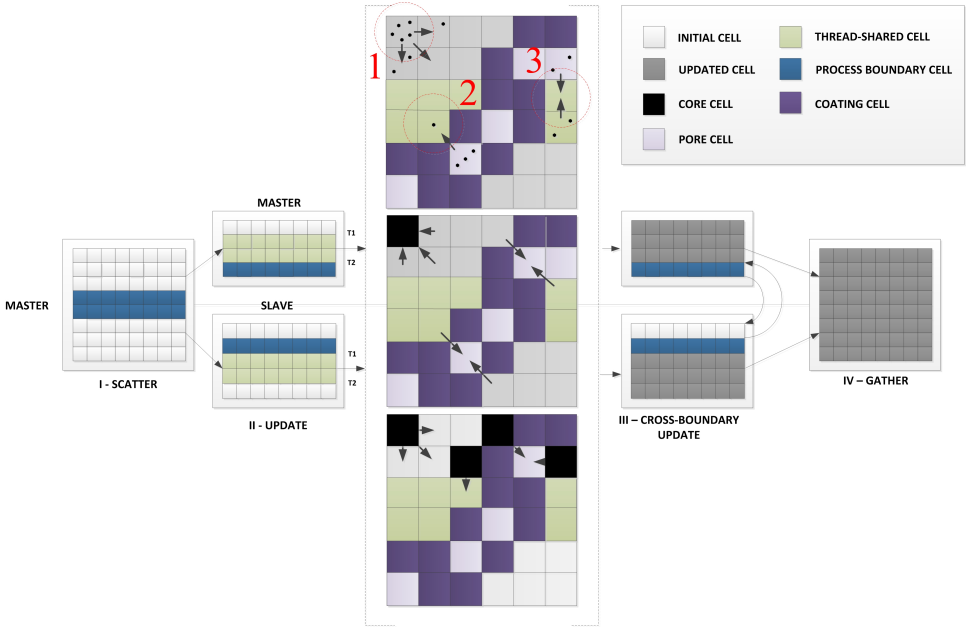


Figure 5.6: Hybrid parallel model incorporating main rule types. Each of the processes computes the internal cell transitions, divided into three core types.

sequentialised on a first-come first-served basis. In scenario 2, there is no additional contest from the other thread, so regular cell state update occurs. However, in scenario 3, as the synchronous CA update mechanism is used, a second update will not be aware of the changes made by the first, resulting in a need to "resolve the update conflict". In this case, we assume the resulting state will be cumulative and both diffusions are permitted, unless

a super-saturation[7] of the cell occurs.

The middle panel in the central diagram in Figure (5.6) shows similar update scenarios in the context of erosion rules. As rules in this case are passive and affect the self-state only, there is a "smaller" potential for conflicts. In case of shared layer access, however, it is still necessary to lock the cell, in order to prevent other rule types from affecting it.

Finally, the last central panel of Figure (5.6) represents the scenarios for simulation, which incorporates swelling effects. These are fairly similar to the diffusion updates, with added complexity that polymer transfer to neighbouring cell occurs, influencing erosion cycle in the subsequent iteration.

5.2 Performance Metrics

5.2.1 Development and execution environment

The code for the models was written in the C++ programming language. It consists of approximately 11000 lines of code divided across 20 classes. The build environment is a combination of g++ compiler with Eclipse used as an IDE, on Ubuntu Linux, running on VMWare Player within a Windows 7 environment. The compilers used on the compute clusters included Intel and GNU C++. Standard outputs of the simulation program are drug release and structural data (e.g. diffusion and swelling fronts, porosity) at desired time points. In addition to this, there is a need to visually observe the evolution of models for two main reasons: firstly, visualisation enables the interfacing with different subject matter experts, e.g. pharmaceutical scientists and biotechnologists. Secondly, it allows a basic verification during different stages of model development. Visualisation was performed using OpenGL and displays a model cross-section as it evolves over time, to facilitate an in-depth observation of processes occurring within the different layers, Figure (6.1). It is implemented as an optional, separate component of the program, allowing two types of data processing, numerical and graphical.

The parallelised algorithms are based on GNU OpenMP (GOMP) and Open MPI

[7] *Super-saturation* of the cells allows a larger concentration of the drug within the cell volume than that which would result as a consequence of an ordinary diffusion flow. It usually occurs during the drug formulation phase.

libraries. Real simulation runs were performed on SMP architecture cluster provided by ICHEC (The Irish Centre for High End Computing) (Stokes) and DCU (Ampato), with 12 and 8 cores per node, respectively. For each set of parameters, ten runs were performed to assess simulation robustness, showing negligible differences in outputs, (much as expected) as all models are both large-scale and high-resolution.

Model runs were performed using ICHEC Stokes supercomputer. Stokes is an SGI Altix ICE 8200EX cluster with 320 compute nodes. Each compute node has two Intel (Westmere) Xeon E5650 processors and 24GB of RAM. The nodes are interconnected via two planes of ConnectX *Infiniband*. Our model was implemented in C++ (using the gcc 4.3.0 64bit compiler), OpenMPI version 1.4.3 and GNU OpenMP (GOMP) version 3.1. In all cases, 1440 iterations of the simulation were run, with one iteration configured to represent one minute of real time. This choice was based on the need to correlate the data obtained to the *in vitro* experiments which usually run for 24 hours. The bounding volume of the simulation space was 300 x 300 x 300 cells, resulting in around 2.7 GB of in-memory model data.

5.2.2 Performance for TLP

The focus of the first set of runs was on performance gain achievable using OpenMP parallelisation only. Simulations were run for 1-8 cores, in addition to the execution times, we observed the speedup of different code segments and their relative participation in overall program execution. Figure (5.7) (top) shows the improvement in execution times, which is significant and near-linear. At the lower end of the tested spectrum, the initial execution time for non-parallelised code is almost 11 hours, while at the opposite end (8 cores) we gain a 6-fold improvement in wall-time in execution speeds. From the other two measurements, we can see that the potential for speedup in parallel sections of the code can bring further time reductions, as parallel code still represents a larger fraction of the total executed code[8]. However, we are bound by the limitation of the SMP architecture, which prevents further expansion of the thread-parallel approach.

[8] *Parallel* speedup indicates the reduction in speed achieved in part of the code which is parallelised. In contrast, *total* speedup refers to reduction in speed of the overall code, including both parallel and sequential part.

Algorithm 4 *HybridUpdate*(Matrix M, int rank)

```
1: Secondary = M
2: n = Secondary.Regions.Count
3: Matrix region
4: if (rank == MASTER) then
5:     Scatter(M.Regions) ⇒ region
6: parallel for each slice in region.slices
7:     for each c in Shuffle(slice) do
8:         if(IsBoundary(c) == true) then skip
9:         if(IsShared(c) == true) then lock(c)
10:          ApplyRules(c, rule_list) ⇒ Secondary(region)
11:          if(IsShared(c == true) then unlock(c)
12: #end parallel
13: ExchangeBoundary(rank, rank + 1)
14: for each c in Shuffle(region.boundary) do
15:          ApplyRules(c, rule_list) ⇒
16:          Secondary(region.boundary)
17: if (rank == MASTER) then begin
18:     Gather(M.Regions) ⇐ Secondary(region)
19:     ComputeRelease(M)
20: end
```

Table 5.4: The pseudo code for the hybrid algorithm.

5.2.3 Performance for PLP

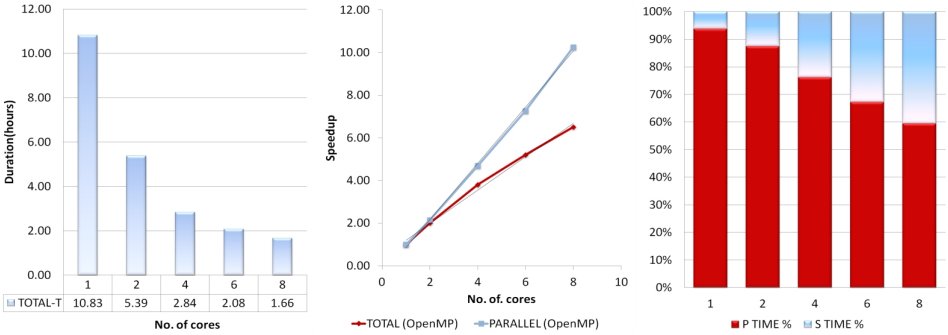
Dividing the algorithm across compute nodes allows us to lift the restrictions imposed by the thread-level paradigm, using the MPI framework. We investigated the performance of a pure-MPI (i.e. no threading) solution over 2-32 cores, with the goals both of comparing the performance to OpenMP as well as examining the potential for speedup and the overhead for local and global communication on simulation speed and solution scalability. Results obtained (see Figure (5.7) (middle)) show that, for this model size, communication overhead has a significant impact on simulation times, although the potential of the MPI solution is ultimately greater as it does not suffer from "ceiling" limitations.

Interestingly, due to expensive communication, the MPI solution is not as efficient as a comparable OpenMP one. Since the "scatter and gather" operations incur a certain overhead (from 1-2 seconds wall-clock time per iteration in our experiments), the sequential portion of the simulation (S-time) is increased, relative to the parallel component (P-time). For the case of 8 cores, the relative performance is almost 40% improved for thread-level parallelism. However, the real benefit occurs for a larger number of cores, where the MPI solution benefits from practically unlimited infrastructure. This holds for all models described in Chapter 4 regardless of the overhead caused by each model complexity.

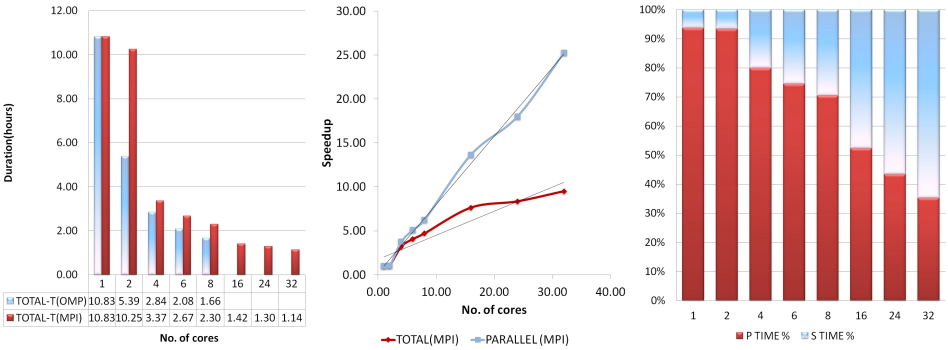
5.2.4 Performance for HP

Combining the two approaches presented above was the ultimate objective of this experiment. By using the fine-grained parallelism within a compute node, and thus avoiding the communication price, we expected to gain some advantage over the coarser-grained approach that MPI provides (Pope et al., 2011). Figure (5.7) (bottom) presents the comparison of the two. From Figure 5.7 (middle), even with MPI for increasing number of cores, there is an increase in the relative sequential execution time. This is in agreement with findings presented in Figure 5.7 (bottom), where the total improvement in execution speeds reaches a plateau after a certain number of cores. Above a certain point, (at around 64 cores), the performance of both MPI and hybrid approach reaches the limit for the model.

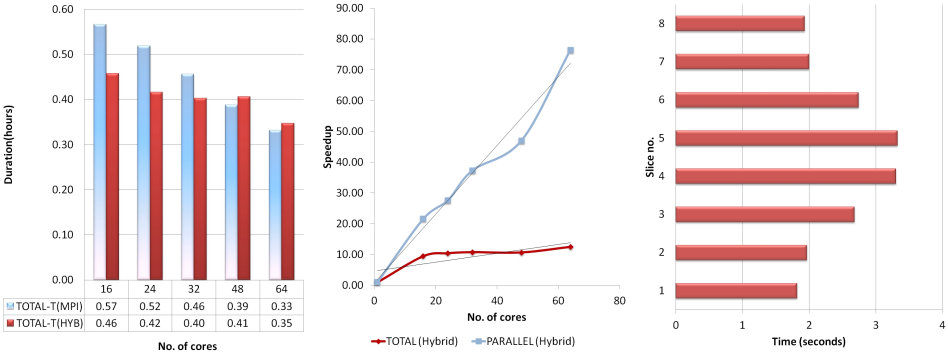
Finally, we investigated the average time taken to simulate each of the parallel slices and conclude that further optimisation is possible by applying an appropriate dynamic load



(Left) Execution Times (OpenMP); **(Centre)** Parallel vs. Total Speedup (OpenMP); **(Right)** Sequential vs. Parallel Execution (OpenMP)



(Left) Execution Times (MPI vs. OpenMP); **(Centre)** Parallel vs. Total Speedup (MPI); **(Right)** Sequential vs. Parallel Execution (MPI)



(Left) Execution Times (Hybrid vs. MPI); **(Centre)** Parallel vs. Total Speedup (Hybrid); **(Right)** Per-Slice Simulation Times (1x8)

Figure 5.7: Performance Results: OpenMP (top), MPI (middle) and Hybrid (bottom). As complexity of parallelisation and number of cores increase, the hybrid approach shows better performance gains over the other two.

balancing solution. For future work, some refinement of the approach is indicated enabling current benchmarks to be re-evaluated for different model-space division strategies.

5.3 Conclusions

We have presented several parallelisation strategies, aimed at improving performance of the CA frameworks, used as a modelling basis for pharmaceutical investigations performed in previous chapters. Each strategy is viable, depending on the speedup requirements and available infrastructure. In general, cluster computing permits the in-depth simulation of complex drug formulations, as well as the evaluation of a wide parameter range through reduction of *in silico* experiment time.

The results show that the hybrid approach offers the best performance, followed closely by the pure MPI based solution (i.e. no threading) over 8-32 cores. With the promised advances in MPI implementation, this gap may reduce further. Further improvements may be possible and can be attempted by experimenting with different load balancing methods, in order to close the gap observed in individual thread/process execution times. Finally, it would be interesting to see the applicability of the proposed solutions when run as part of commodity cloud computing infrastructures, which already promise a flexible and cost-effective solution platform, (Schatz et al., 2010) for the types of research questions of core importance to the pharmaceutical industry.

We also showed the implementation and benefits of a complete cycle of *in silico* data analysis in the stages of drug development, starting from a workflow towards comparison and parameter investigation stage. We have demonstrated both the feasibility and the advantages of combining this process with the *in vitro* development. The amount of possible analysis that can be done by the modelling framework in terms of the investigation of parameter influence on drug release is practically unlimited, allowing the analysis of model data from a variety of different perspectives. The HPC extension to the framework makes it more applicable for a wider set of problems, allowing for a fast and cost-effective modelling solution. Depending on the size of the device simulated (on milli-, micro- or nano-scale), an appropriate parallelisation strategy can be chosen to maximise the processing power available.

Chapter 6

Simulation and Results

In this chapter we discuss results for all of models developed in Chapter 4. For each of these, a detailed sensitivity analysis was performed, demonstrating a key advantage of using the stochastic modelling approach, i.e. the potential to predict the influence of various parameter values and ratios. This allowed for the possibility of "stress testing" the extreme scenarios and combinations, as theoretical model verification is clearly important. All models developed in this thesis were validated against experimental data and showed good matching. Furthermore, as models progress in complexity, (starting from the relatively simple, 'ECDD', to the more complex 'SCDD' and 'MCES'), the difference between simulation and experiment are seen to decrease, demonstrating that this incremental approach, to incorporation of increasing complexity, yielded satisfactory results.

6.1 Simulations for model for erodible coated drug devices (ECDD), Section (4.2)

The initial values for the ECDD model, (Section 4.2), were chosen according to the available *in vitro* data and, unless otherwise stated, the erosion rates of gelatine and ethylcellulose are taken to be 90 minutes and 2 days, respectively. The sphere was set at 1.43 mm in diameter, and to contain 5% of EC coating. The drug loading is kept constant for all simulations at 10.8% of the mass. Visualisation of the erosion model shows the bulk erosion of the coating layer and the delayed erosion process of the gelatine core as the main drivers

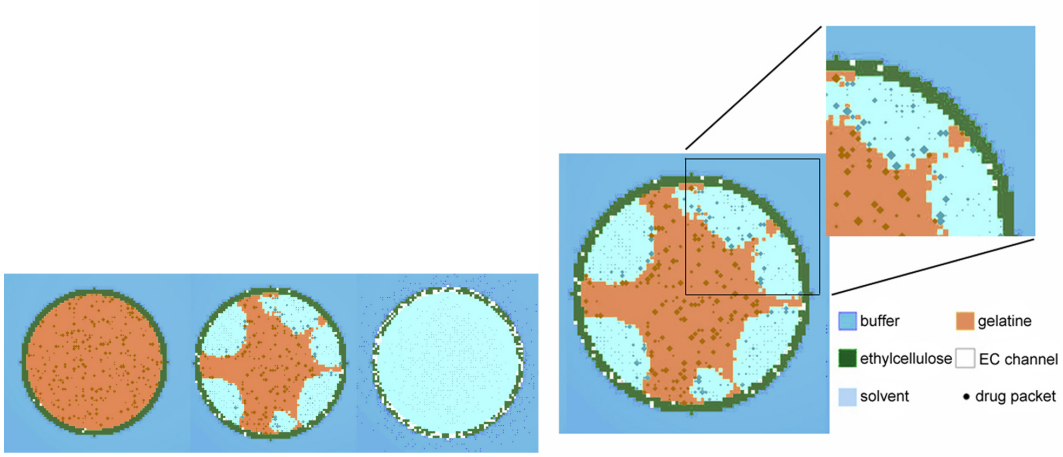


Figure 6.1: Simplified internal morphology of one 3D sphere simulating drug dissolution through coating layer (ethylcellulose). Enlarged: part of the sphere with definition of model cell types.

of drug release, (Figure (6.1)).

The physical size distribution of spheres determines the actual mean and variance of the release rate, (Figure (6.2)). This feature suggests a refinement of initial parameters is necessary, (e.g. coating thickness or drug loading), as these are directly dependent on the size of the sphere. Increasing the sphere size mostly slows down the drug release due both to the capability of the device to hold more drug and the fact that larger spheres have heavier and thus less permeable coatings.

The first step in the modelling was to determine the most appropriate time interval in which cell states should be updated, as this correlates directly with the diffusion coefficient used. This essential parameter directly determines the diffusion process speed (from Equation 4.7) and has order of magnitude of seconds, (from Fick's first law and dimensional analysis). Simulations were then performed for intervals between 5 seconds and 1 minute, (Figure (6.3)). By comparing results against experimental data, (Figure (6.7)), a time interval of 30 seconds was chosen for all subsequent simulations.

The effect of porosity in the coating layer was investigated by varying the lifetime of EC cells, (i.e. varying τ_c, Equation (4.10)), to obtain different release behaviour for different lifetimes of EC chains, (Figure (6.4)). The slower release rate of the drug is due to decreased porosity, i.e. slower channel formation occurring in the coating.

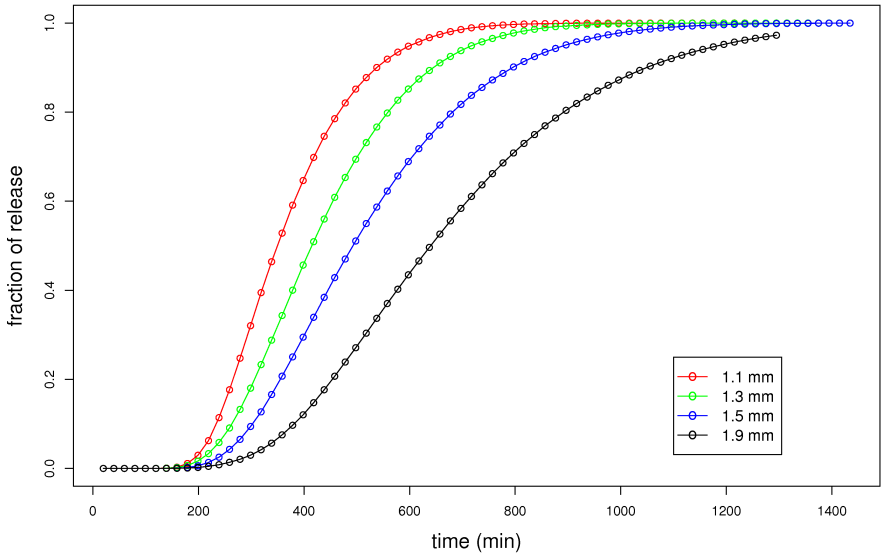


Figure 6.2: Release profiles as a function of the bead size [in mm].

Weight gain of the coating thickness is also one of the primary factors influencing drug formulation and performance. Here, we varied the coating levels from 4% - 8%, (Figure (6.5)). The release curves qualitatively reproduce a reduction in diffusion rate, but the impact of adjusting the weight gain is somewhat smaller than observed *in vitro*. This is expected, to some extent, as perfect sink conditions are modelled in our case. In reality, the concentration of EC in the boundary layer inhibits movement of drug packets and some local saturation occurs. This phenomenon must be taken into account in future model extensions.

Since the gelatine represents a mixture of different substances (peptides and proteins) and the total lifetime was not known precisely from experiment, different values were investigated as parameters of our simulation. As can be seen, (Figure (6.6)), the lifetime of the gelatine carrier can be considered to be negligible in terms of influence on final release rate. However, gelatine is an important controller of the drug release in the initial stages, as it influences the "burst effect" by accelerating drug transfer through EC channels.

This, simplified, model comparison showed good prediction potential. In analysing the results, it should be remembered that measurements of the *in vitro* release rates using different dissolution media in USP II may have certain limitations in terms of release

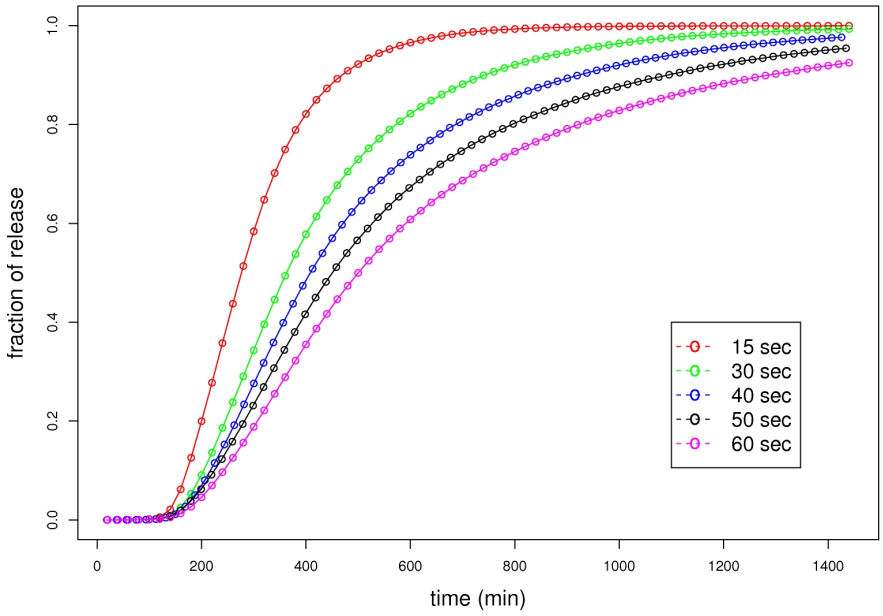


Figure 6.3: Release profiles as a function of the simulation time interval.

detection depending on drug solubility. The insolubility of the Cyclosporine causes a fragmented release, which requires adding surfactants to the dissolution media in order to improve drug solubility. If suitable amount of surfactant is not present, the equipment used is not able to detect exact release rates, (Maggi et al., 1996). As a result, the release curve may reach a platform at 80% of total release, (although 100% is accounted for in the experiment).

Although simulated results do not reproduce quantitative *in vitro* data precisely, (Figure (6.7)), results obtained are promising. These show that the ECDD model predicts qualitatively similar behaviour compared to that seen for experimental release curves. Results also show that the behaviour of the device cannot be modelled using erosion only, but that other phenomena, such as swelling, have to be taken into account (Atyabi et al., 2004). The next focus was thus incorporation of the swelling effect, caused by increased hydration of the broken polymer chains. Additionally, the model should be augmented to include the effect of different polymer coatings and different media surrounding the drug (biphasic release) in order to simulate the different surroundings in the proximal and distal GI tract environments. Nevertheless, the initial approach enables comparison of a set of simulated

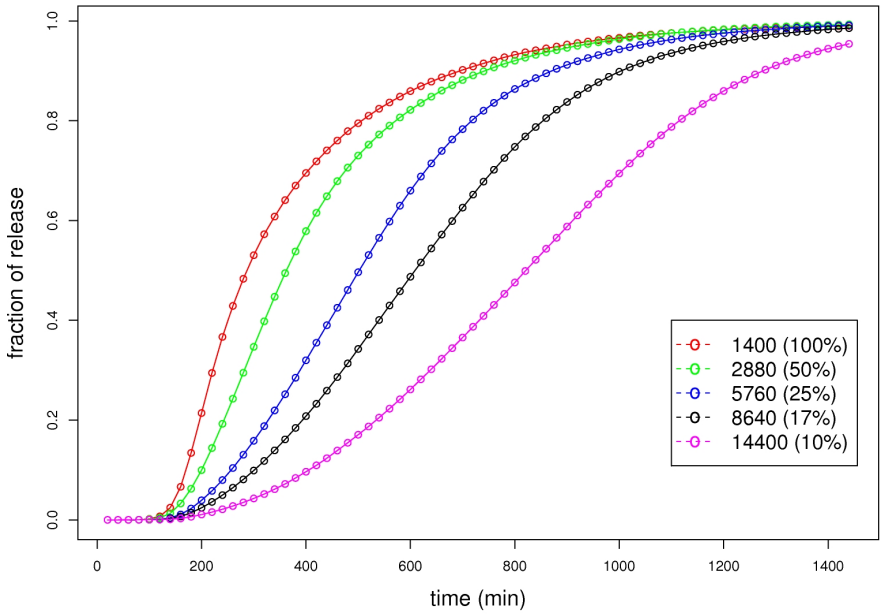


Figure 6.4: Release profiles as a function of the coating degradation rate lambda.

and experimental release curves, allowing us to determine key parameters and their values for this novel formulation, and to reproduce qualitative behaviour. This allowed for the comparisons with a large set of simulated and real release curves in order to estimate best fit parameter values.

6.2 Simulations for model for swellable coated drug devices (SCDD), Section (4.3)

6.2.1 Validation against experimental data

The first set of runs for the SCDDa model was performed against available experimental batch data in order to validate the model prediction capabilities. The data used was provided by Sigmoid Pharma and consisted of two batches of uncoated spheres with different Cyclosporine A loadings (10% and 15% respectively) and three batches of Surelease/Pectin coated spheres of different coating thickness (4.6%, 7.4% and 15% weight gain respectively). Example visualisation of simulations is presented in Figure (6.8).

Due to a lack of experimental data for all bead sizes, the absolute coating thickness

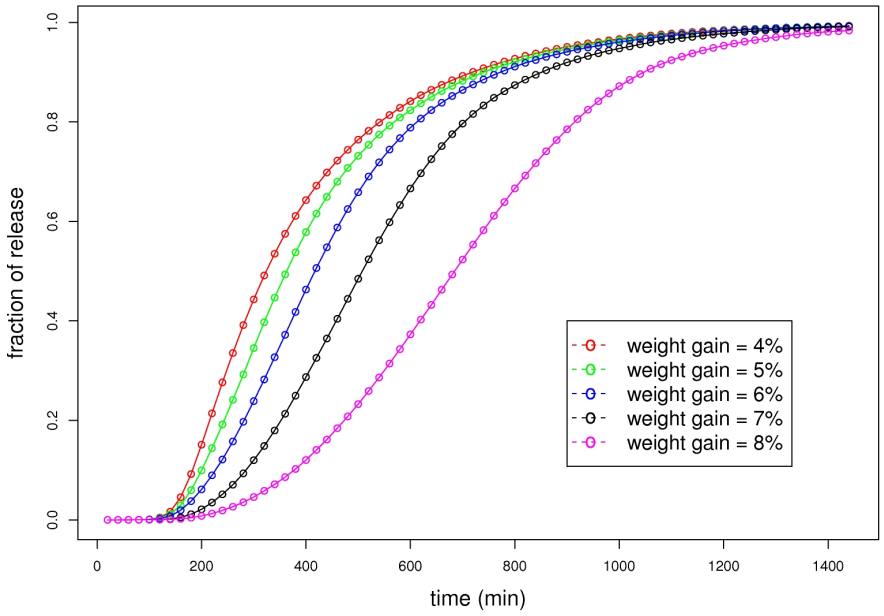


Figure 6.5: Release profiles as a function of the coating weight gain.

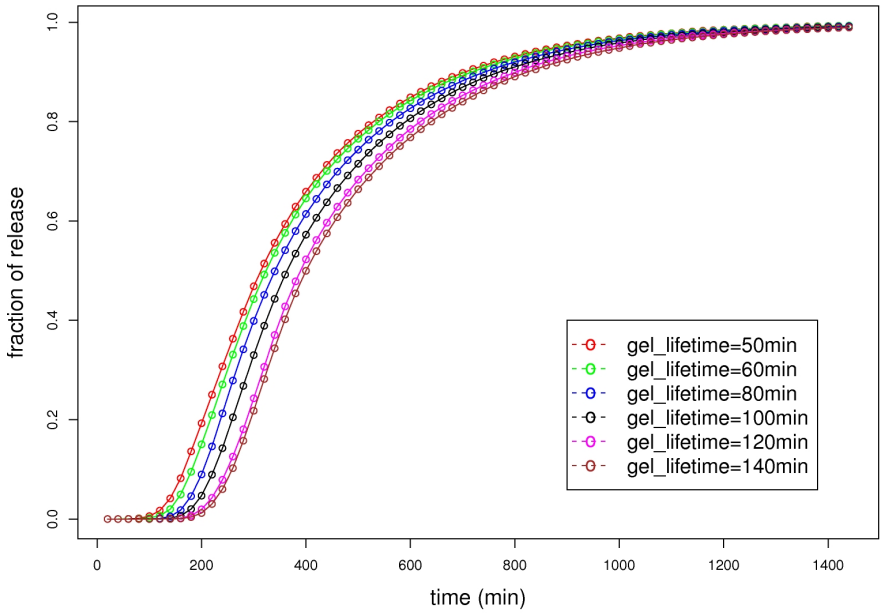


Figure 6.6: Release profiles as a function of the gelatine lifetime.

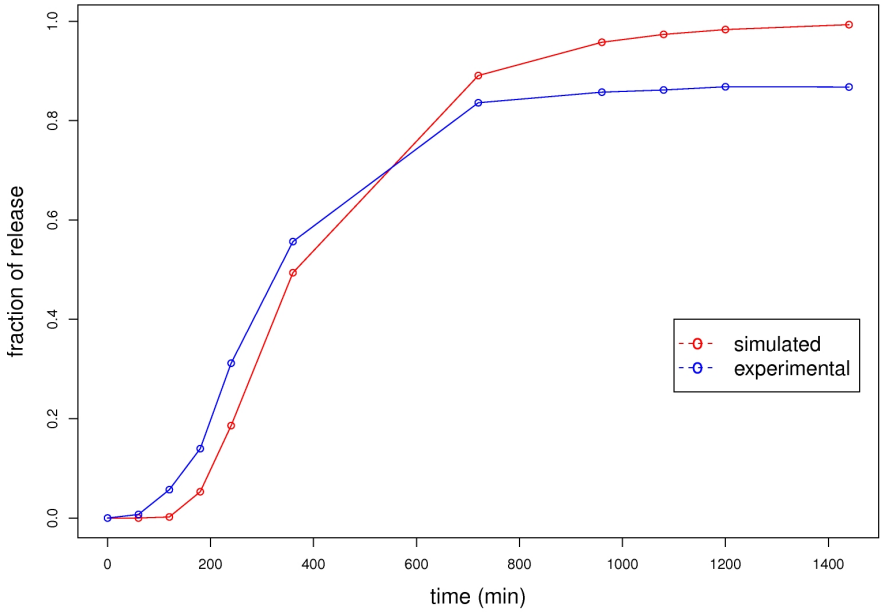


Figure 6.7: Simulated release profiles against experimental release profiles. Experimental data provided by Sigmoid Pharma Ltd.

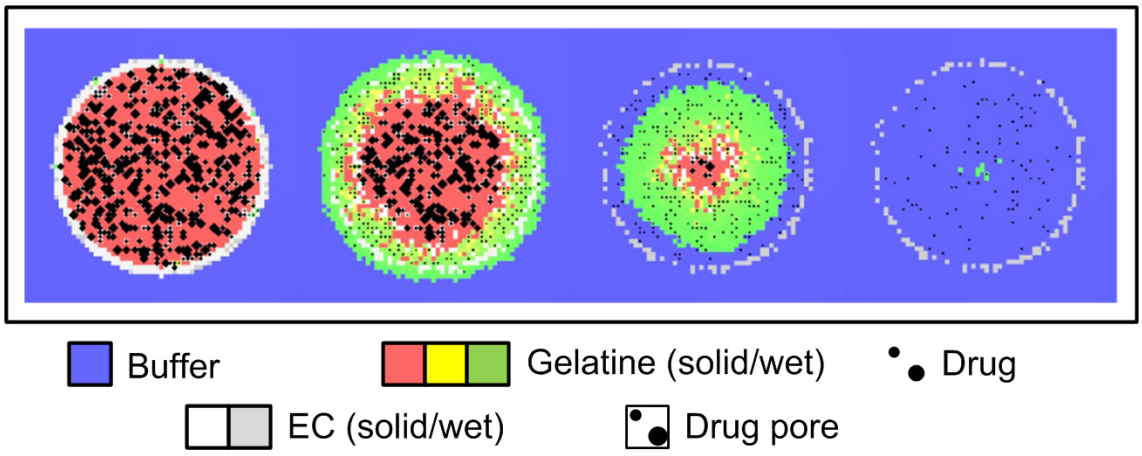


Figure 6.8: Visualisation of the SCDD model with relevant cell types. The spheres show progression from initial to the dissolved state. Green indicates the gel layer, and red the solid gel state. We can observe the "ghost shell" of the coating remaining in the end after core material has eroded completely.

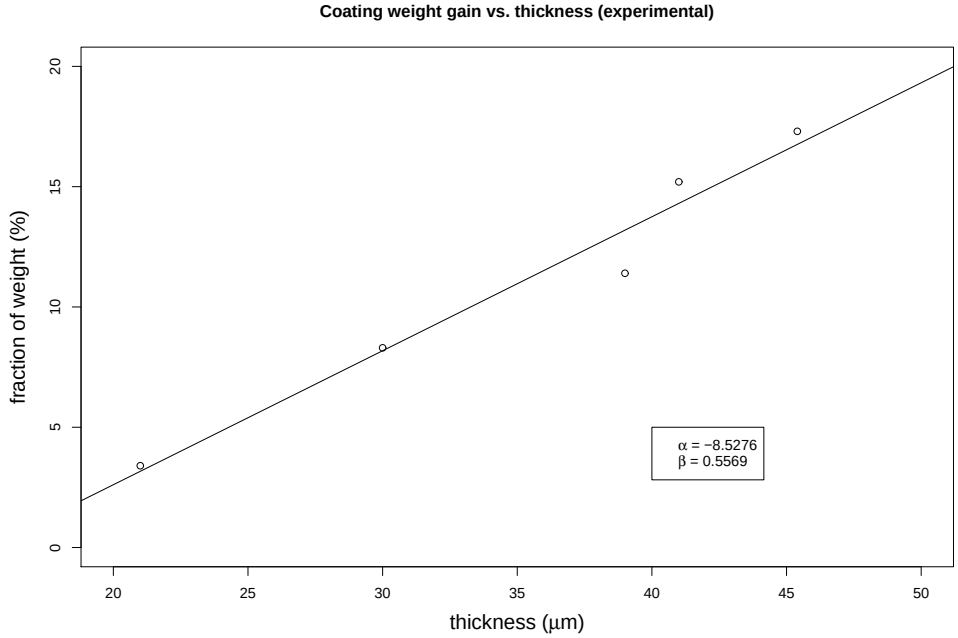


Figure 6.9: Linear regression curve used for estimating absolute thickness (in cm) from the available weight gain data. Experimentally available measurements and X-ray images show good linear correlation. This indicates that the approach is feasible for thicknesses $> 21\mu$m; (note that the position of the intercept indicates that the relation is non-linear somewhere below this point).

was derived from the relative weight gains by means of linear regression using known measurements and X-ray tomography images showing actual coating regions after the thermal processing. The regression curve used is plotted in Figure (6.9).

In order to analytically compare the level of correlation between simulation and the experiments we chose to employ the widely used similarity factor (f_2) equation as described by Shah et al. (1998), which considers two profiles to be similar on the basis of (i) overall profile similarity and (ii) similarity at every dissolution sampling point. Our measurements satisfied the conditions described and the factor was employed to analyse the statistical similarity between two given dissolution profiles. The results are accepted as similar if the value falls within the 10th percentile range (corresponding to values of f_2 between 50 - 100, with 50 indicating a 10% boundary and 100 a perfect match; correlated results fall somewhere within the range). The similarity factor is calculated using the equation:

$$f_2 = 50log\left\{100\left[1+\frac{1}{k}\sum_{i=1}^{k} w_i(y_{exp}-y_{sim})^2\right]^{0.5}\right\} \tag{6.1}$$

where *log* indicates the base-10 logarithm, k indicates the number of sample points taken, w_i the weight factor that takes into account the variability of samples at each timepoint, y_{exp} and y_{sim} the experimental and simulated release data at each timepoint respectively. The weight factor was calculated using the equation given by Moore and Flanner. (1996):

$$w_i = log\left\{100\left[1 + (y_{exp} - y_{sim})^2\right]^{0.5}\right\} \tag{6.2}$$

Simulation parameters were chosen based on direct experimental observations provided by Sigmoid Pharma. In situations where those observations were not available (or possible to obtain) we used available reference data from literature. Table 6.1 provides a summary of the parameter values used for validation with experimental data.

The main rate limiting parameter, the diffusion coefficient of Cyclosporine A was set at $2 \cdot 10^{-6}\ cm^2/s$, as indicated by the measurements made by Trammer et al. (2008), Park et al. (2013), which put the CyA diffusion speed within the range of $0.96 - 5 \cdot 10^{-6} cm^2/s$. The base diameter of the sphere was derived by averaging bead size data available from laboratory measurements, and gelatine lifetimes were chosen based on visual observations made in the Sigmoid lab. Probability of core swelling was kept to the low swelling region as observations showed that no significant expansion of the core occurred during the bead dissolution. Relative speed of the diffusion of water into the Surelease layer, (P_W), was derived using the measurements made by Ravindra et al. (2000) and Equation (4.20) and set to 0.16. Similar investigation was performed for the speed of water diffusion into the core layer (P_{WG}) using available gelatine/water interface data from Domenek et al. (2008) which reported the experimental value to be of the order of $10^8 cm^2/s$, resulting in relative speeds of order of 10^{-2} when compared with the main diffusion coefficient. The duration of the experiment was 24 hours, with comparative measurements made at 1, 2, 3, 4, 6, 12, 16, 18, 20 and 24 hours respectively.

Two parameters, which could not be derived either from reference data or experiments, were the coating diffusion probability (P_P) and the amount of entanglement (P_C). The entanglement rate could be estimated from available release data by looking at the "plateau" the release curves would attain as the CyA would no longer dissolve. The coating diffusion

Parameter	Simulation value	Experimental reference
Sphere diameter	1.65mm (1.5mm uncoated)	1.65mm (1.5mm uncoated)
D_f	$2 \cdot 10^{-6}$ cm^2/s	$0.96 \cdot 10^{-6}$ - $5 \cdot 10^{-6}$ cm^2/s
Gelatin λ	60 min	60 - 120 min
P_S	0.3	0 - 0.5
P_W	0.16	0.16
P_{WG}	0.01	0.01
P_P	0.3	N/A
P_C	0.1	N/A
Duration	24 h	24h

Table 6.1: Key SCDDa model parameters used for comparison with experimental data. Reference parameters were derived from batch data, experimental observations and literature. The coating permeability and probability of entanglement was estimated based on simulated model data ranges.

had to be estimated, using the comparison between available release, with sensitivity analysis data. It would be of interest to investigate this value *in vitro* by performing experiments on diffusion of CyA through Surelease films to further validate the estimated value. In addition to parameter values represented, it is also important to indicate that in the uncoated release scenario the dissolution media used was water instead of SDS, and therefore the maximum solubility of CyA was capped, as indicated by the experimental measurements.

The resulting simulation release curves for each batch are presented in Figures (6.10) to (6.13) respectively. Table 6.1 summarises the main batch parameters, while Table 6.2 gives similarity factors obtained for each curve. We can conclude that a very good match is obtained with the majority of the simulation points falling within the normative 10% variability range (i.e. having similarity factor values within the range of 50 to 100) indicate good agreement with the reference data. The only outlier is the FC021/09 batch, for witch the release curve showed anomalous behaviour and this might have been influenced by noise in the data (each batch had only two repeated data measurements). More measurements

Modelled batch	Achieved similarity factor
EXP11/085 (uncoated, 10% d.l.)	100 (<1%)
EXP12/111 (uncoated, 15% d.l.)	74 (<3%)
FC015/09 (15% Surelease, 10% d.l.)	61 (<6%)
FC008/09 (7.4% Surelease, 10% d.l.)	82.9 (<2%)
FC021/09 (4.6% Surelease, 10% d.l.)	40 (<15%)

Table 6.2: Values of the similarity factor (f_2) for different experimental modelling scenarios for SCDDa. Results show good match allowing for sample error. The only outlier was batch FC021/09 which fell outside 10% variability acceptable.

might establish a better baseline for comparisons and allow for better normalisation or identification of outliers, so that these could be allowed for in the comparison.

6.2.2 Sensitivity analysis

After obtaining satisfactory results for the SCDDa model (as described in the previous section), we investigated the model's ability to describe major phenomena influencing release, as well as to predict parameter changes required to obtain zero-order release, a desired quality of controlled release devices. For the purpose of classifying the results based on dominant release phenomena, we used the Korsmeyer-Peppas model described by Equation (2.6) to divide the results into specific release categories. The results obtained are described in detail in the following sections. Changes for each of the parameters were evaluated for three key release scenarios: (i) erosion dominant release; (ii) swelling dominant release and (iii) the equilibrium between the two which should lead to a constant gel layer driving the release profile. Each parameter change scenario, (except for the changes in coating thickness), was performed for both the cases of uncoated and coated beads in order to compare the two and help to understand the impact of coating on swellable beads. Additionally, the dynamics of dissolution behaviour was examined through the positions of two characteristic transition boundaries within the device (e.g. Figure (6.17): the swelling front, (where polymer moves from being in a static to dynamic state), and the erosion front, (where polymer starts dissolving). These are the best indicators of the spatial nature of the

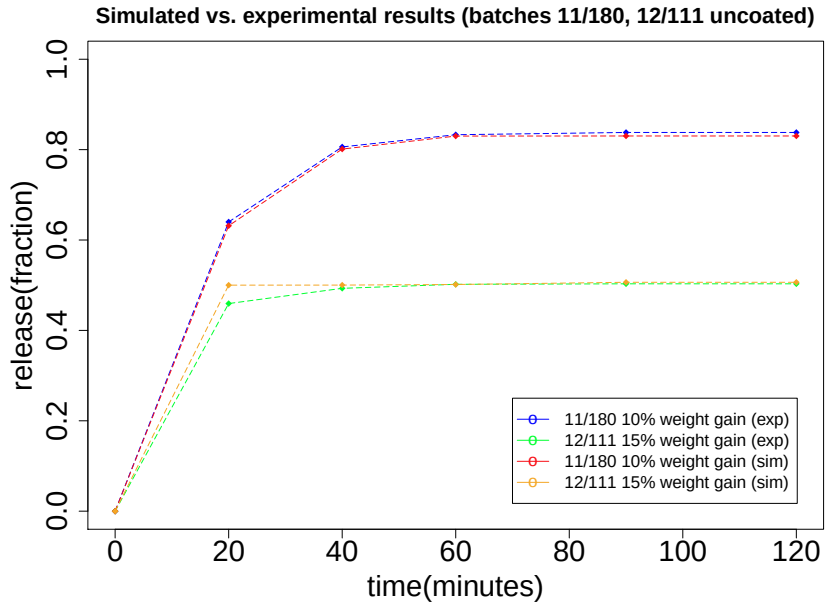


Figure 6.10: Experimental vs. simulated results for batches EXP11/180 (10% CyA loading) and EXP12/111 (15% CyA loading). Obtained similarity factor shows less than 1% difference in the obtained profiles. The maximum dissolution amount was capped as the dissolution medium used for the experiments was water instead of SDS.

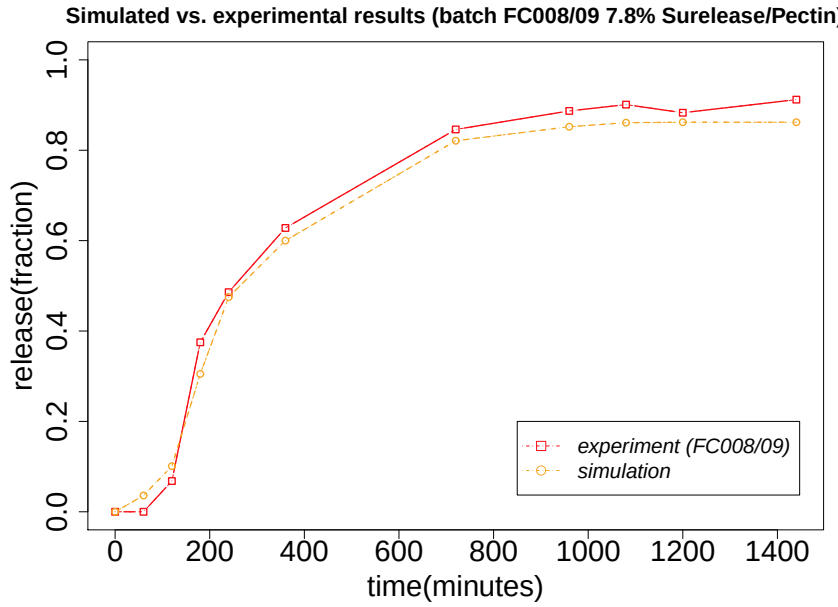


Figure 6.11: Experimental vs. simulated results for batch FC008/09 (7.4% Surelease coating and 10% CyA loading). Obtained similarity factor shows the results fall into the standard variability range (<10%).

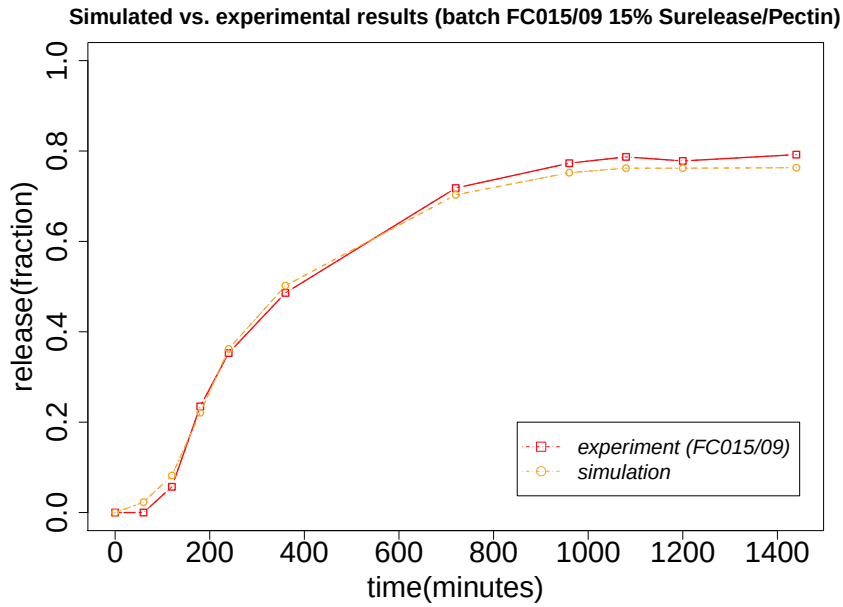


Figure 6.12: Experimental vs. simulated results for batch FC015/09 (15% Surelease coating and 10% CyA loading). Obtained similarity factor shows the results fall into the standard variability range (<6%).

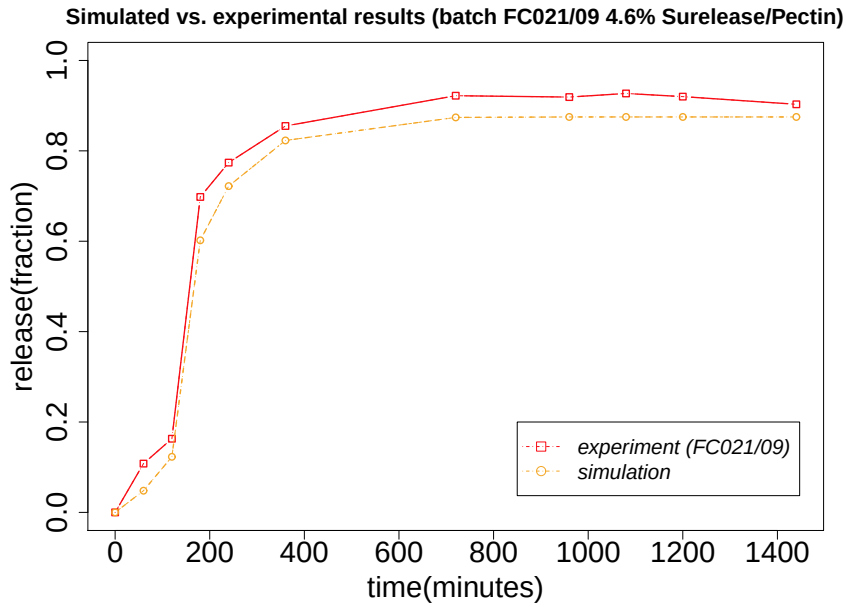


Figure 6.13: Experimental vs. simulated results for batch FC021/09 (4.9% Surelease coating and 10% CyA loading). Obtained similarity factor shows the results fall outside the standard variability range (<15%). We believe this anomalous behaviour is due to noisy data, and better results could be achieved by repeating experimental measurements in order to obtain smoothed results.

underlying phenomena, which cannot be observed directly or solely from release data.

Influence of ethylcellulose coating thickness

The first parameter investigated was the thickness of the ethylcellulose coating. The release curves were simulated for thicknesses of 0.04mm, 0.08mm and 0.16mm respectively. The results are presented in Figures (6.14) and (6.15). As expected the increase in thickness causes both a decrease in release rates, as well as larger offset until the release begins, as the surrounding buffer needs more time to penetrate the ethylcellulose matrix. Analysing the Peppas n factor for the three scenarios, thinner coatings indicated, as expected, larger diffusion influence, with the values of n decreasing correspondingly. Most n values were within the anomalous region, indicating the influence of both erosion and swelling on the release, however for thicker coatings n values indicated erosion controlled release. This suggests that heavier coatings inhibit swelling behaviour within the sphere.

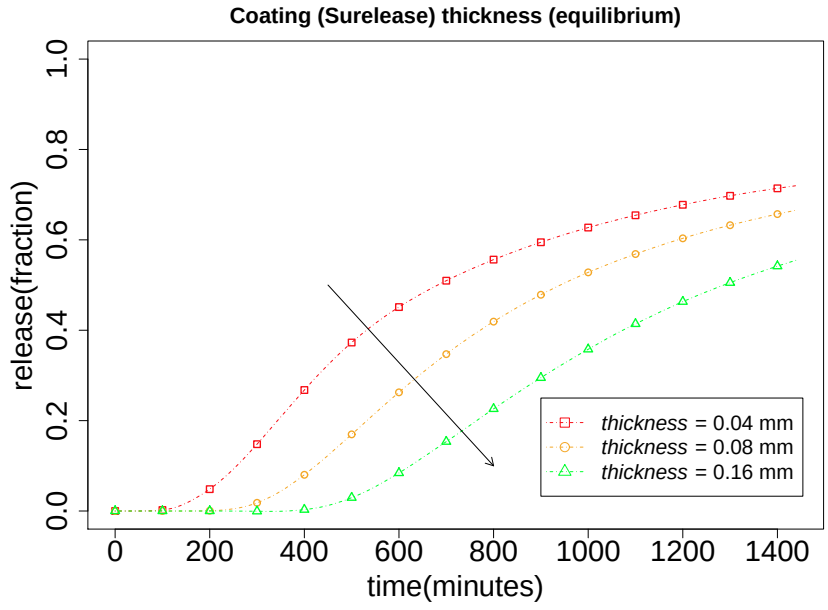


Figure 6.14: Influence of ethylcellulose coating thickness on resulting drug release curves in the equilibrium scenario. Thickness was varied from 0.04mm to 0.16mm and, as expected, caused an overall decrease in swelling rate. Release curves mostly indicated *anomalous* release with erosion and swelling equally controlling the release behaviour. Peppas n values obtained ranged from 0.7 to 1.29 indicating a zero-order release ($n = 0.85$) is possible to achieve.

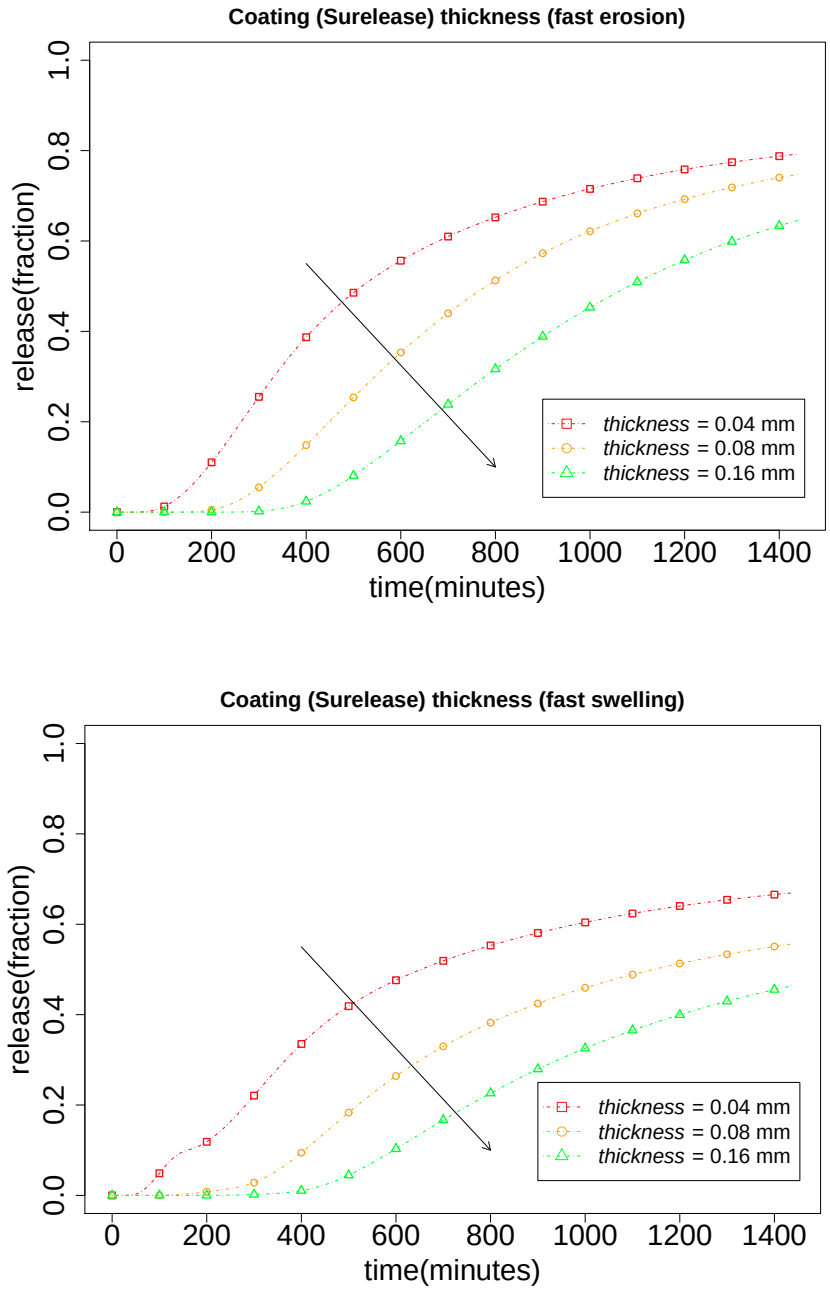


Figure 6.15: (top) Influence of ethylcellulose coating thickness on resulting drug release curves in the fast erosion scenario. Thickness was varied from 0.04mm to 0.16mm and, as expected, caused an overall decrease in swelling rate. Release curves mostly indicated *anomalous* release with erosion and swelling equally controlling the release behaviour. Peppas n values obtained ranged from 0.61 to 1.05. (bottom) Influence of ethylcellulose coating thickness on resulting drug release curves in the fast swelling scenario. Thickness was varied from 0.04mm to 0.16mm and led to overall decrease in swelling rate. The release curves mostly indicated *anomalous* release with erosion and swelling equally controlling the release behaviour and thinner coatings trending towards diffusion controlled, Fickian release. The Peppas n values obtained ranged from 0.57 to 1.04.

Influence of drug loading on release profiles

Another device design parameter that we considered was the amount of drug loading (in volume terms). Figures (6.16) to (6.26) show the effect of changing drug fractions from 0.2 to 0.8 across the three scenarios modelled, (namely fast erosion, fast swelling and equilibrium) and across coated and uncoated beads.

A major difference in how drug loading influences release is found for coated and uncoated beads. In the first place, obtained profiles show diffusion to be dominant release mechanism. Peppas n values range from 0.41 to 0.55 indicating Fickian influence at lower loading levels, and anomalous ones (> 0.55) indicating higher influence of swelling at upper drug loading levels. The results agree with those of (Laaksonen et al., 2009a) as higher amounts of drug by volume of the device lead to a smaller amount of gel polymers, and lower probability that a drug particle will be bound to the gel phase. This holds for both fast erosion and equilibrium scenarios, but in the case of the fast swelling scenario represented in Figure (6.25) we observe a behaviour which seems more consistent with that reported by Siepmann and Peppas (2001). The increased swelling enlarges the rubbery gel state trapping the drug particles for a longer time. The increased size of the gel front is visible in Figure (6.25).

For coated formulations, the reduced release is consistent across all of the three scenarios. As the presence of coating inhibits swelling, coated formulations are generally more erosion controlled, thus preventing the drug particles getting entangled in the wet polymer and thus diffusing out more quickly. Obtained Peppas numbers confirm this, with investigated release ranges not varying much from $n = 1.01$ to 1.16, indicating erosion controlled, case II release.

Effects of gelatine clustering and entanglement on release profiles

During the verification of the SCDDa model against experimental release data, one critical factor which negatively influenced the experimental profiles was partial core dissolution and entanglement of long, undissolved gelatine polymer chains with the ethylcellulose coating. This appears to have led to partial blockage of the pores formed during buffer penetration into the coating and large portions of drug ending up "encased" in the gel solid (glassy)

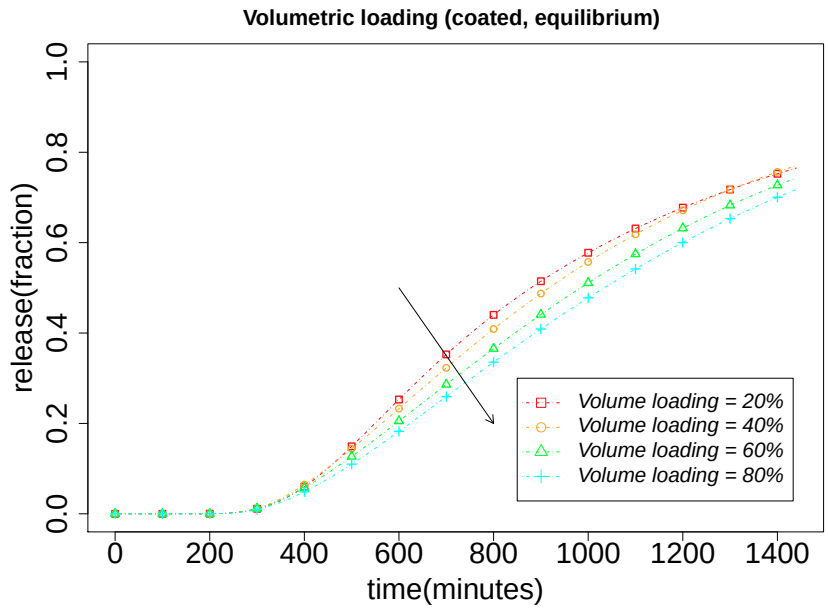


Figure 6.16: Influence of drug loading on release profiles from coated beads displaying combined erosion and swelling behaviour. The decrease in release rates with increase in drug amounts is attributed to faster core erosion and a smaller gel layer due to the coating inhibiting swelling.

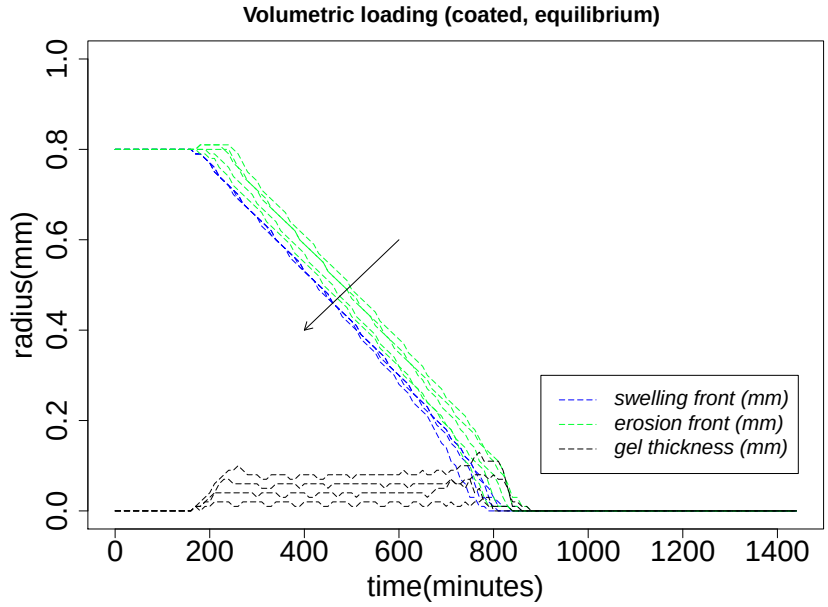


Figure 6.17: Influence of drug loading on swelling and erosion front changes for an equilibrium release scenario (coated beads). A decrease in gel layer thickness is caused by lack of a core material (due to a higher drug percentage). A constant gel layer thickness can be achieved by balancing the erosion and swelling influence leading to more stable and predictable release.

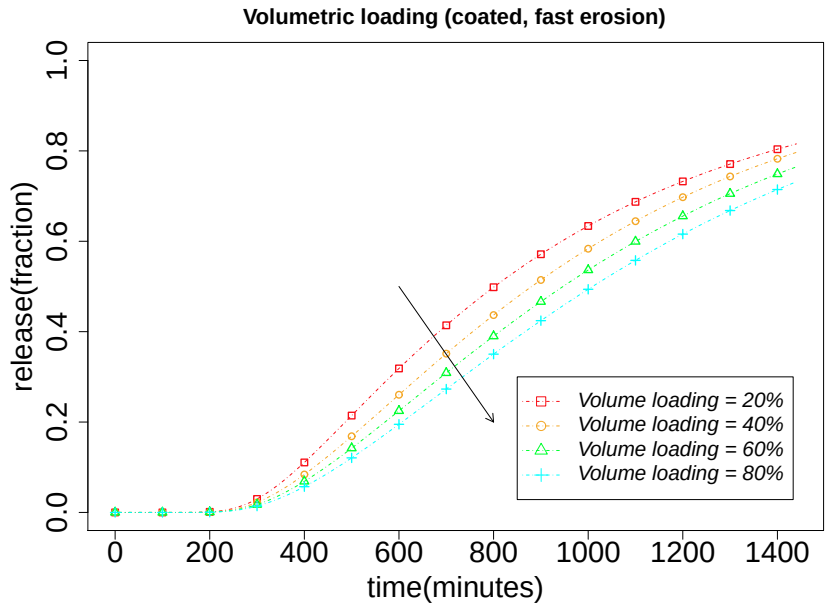


Figure 6.18: Influence of drug loading on release profiles from coated beads - displaying fast erosion behaviour. The decrease in release rates with increase in drug amounts is attributed to faster core erosion and a smaller gel layer due to the coating inhibiting swelling.

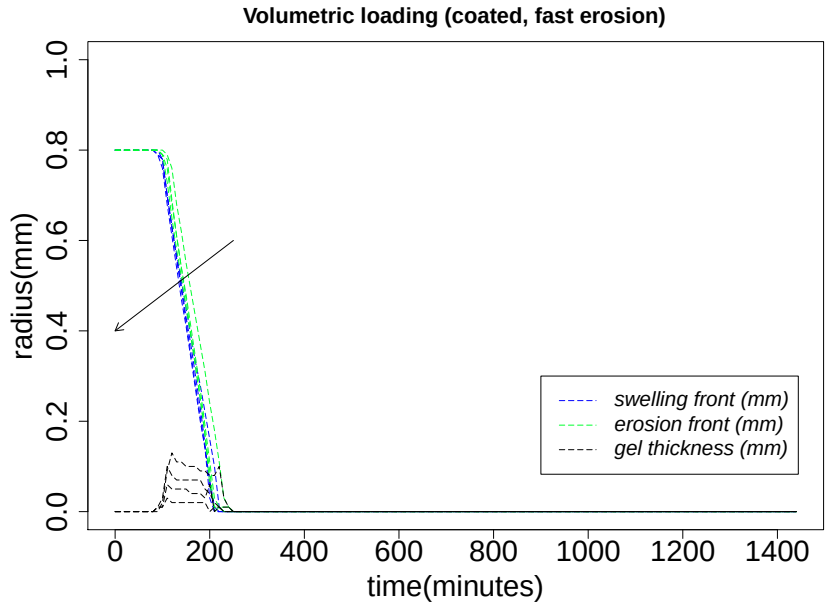


Figure 6.19: Influence of drug loading, on swelling and erosion front changes, for a fast erosion release scenario (coated beads). The decrease in gel layer thickness is caused by a lack of core material due to a higher drug percentage. A constant gel layer thickness is achieved, but the collapse of the core does not allow it to be maintained for a prolonged time due to lack of swelling.

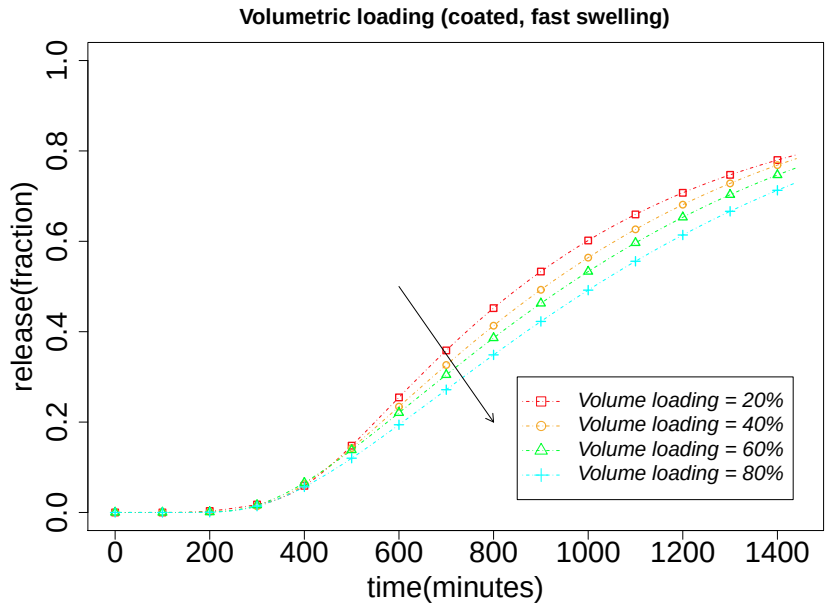


Figure 6.20: Influence of drug loading, on release profiles from coated beads, displaying fast swelling behaviour. The decrease in release rates with increase in drug amount is attributed to faster core erosion and smaller gel layer due to the coating inhibiting swelling.

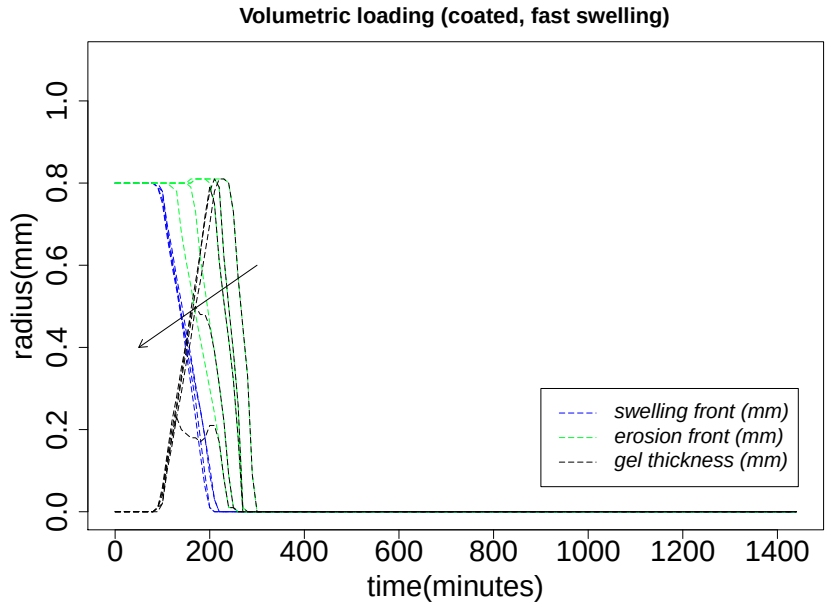


Figure 6.21: Influence of drug loading, on swelling and erosion front changes, for a fast swelling release scenario (coated beads). Decrease in gel layer thickness is caused by lack of core material, due to the higher drug percentage. The high variability in the wet gel phase, as shown generally leads to an unstable release.

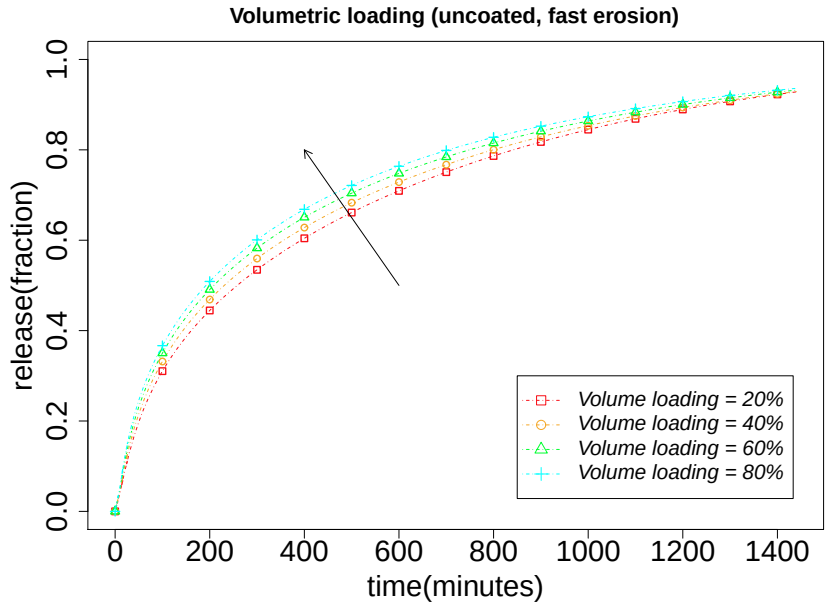


Figure 6.22: Influence of drug loading, on release profiles from uncoated beads, displaying fast erosion behaviour. The increase in release rates with increase in drug is attributed to smaller gel layer due to increased amount of drug solids. At the lower end of the investigated range (20% drug loading volume) the release tends more towards Fickian behaviour ($n = 0.45$).

phase. This phenomenon is difficult to observe *in vitro* and, mostly, it is the effects which are measured empirically. It was of interest, therefore, to simulate the correlation between the probability of its occurrence (P_C) and the resulting decrease in release curves.

We again ran sensitivity analysis simulations across all three release scenarios and Figures (6.27), (6.29) and (6.31) show release curve changes for each scenario, while Figures (6.28) and (6.32) indicate the amount of drug trapped in the solid gel state. Note that the released amount and the trapped drug amount are not mutually exclusive. There may be free moving drug particles in the solvent area within the coating that are also unable to diffuse out.

Comparing release curves, we can see that the apparent relationship is exponential to some degree in all cases. Highly erodible cores are the least impacted as, although a proportion of drug stays within the core, pore blocking is much reduced since the movement of core polymers is also significantly inhibited. Varying the parameter above 20% in this case produced almost linear reduction (roughly 10% per step), indicating that diffusion through

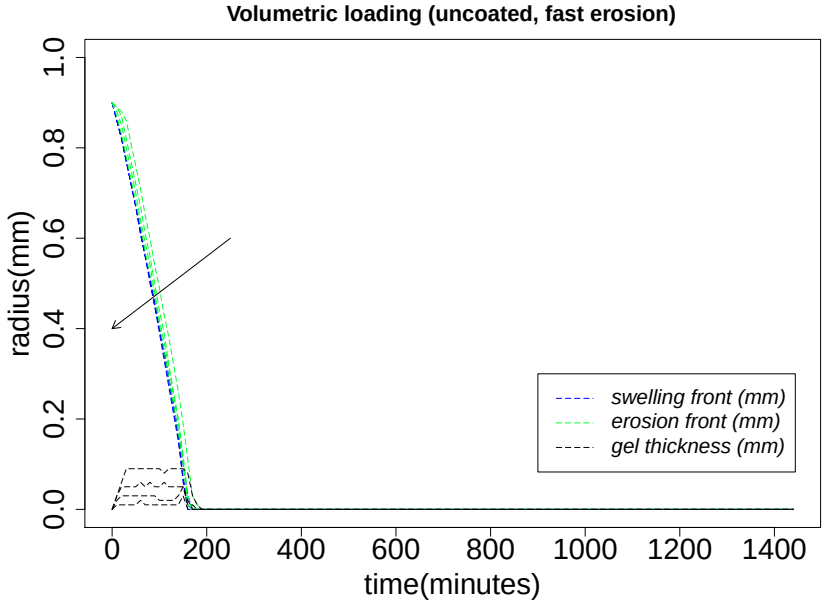


Figure 6.23: Influence of drug loading, on swelling and erosion front, changes for fast erosion release scenario (uncoated beads). The decrease in gel layer thickness is caused by the lack of core material, due to the higher drug percentage. A constant gel layer thickness is achieved, but the collapse of the core does not allow it to be maintained for a prolonged time due to lack of swelling.

the coating was not impacted and the majority of trapped drug was in the core. Observing scenarios where swelling of the core is a greater force driving the release, the inability of drug to diffuse through the coating can be shown to exponentially increase as the entanglement probability gets bigger. In the case of balanced dissolution, (Figure (6.27)), the difference in release between 10% and 20% entanglement probabilities is more pronounced, while in the case of highly swellable cores it is marked in the extreme. Therefore, the erodability of the core plays a major role in preventing negative impact on release.

6.2.3 Results for Variable rate swelling models

As described in the section 4.3.2, SCDDb model is based on the experimentally observed behaviour of gelatine to have a non-constant swelling rate, (decreasing either exponentially or sigmoidally). As the non-linearity constants of the equations (4.18) and (4.19) cannot be derived from the available experimental data, it is of interest to compare a set of simulated profiles for three cases: (i) constant swelling probability (Laaksonen et al., 2009a); (ii)

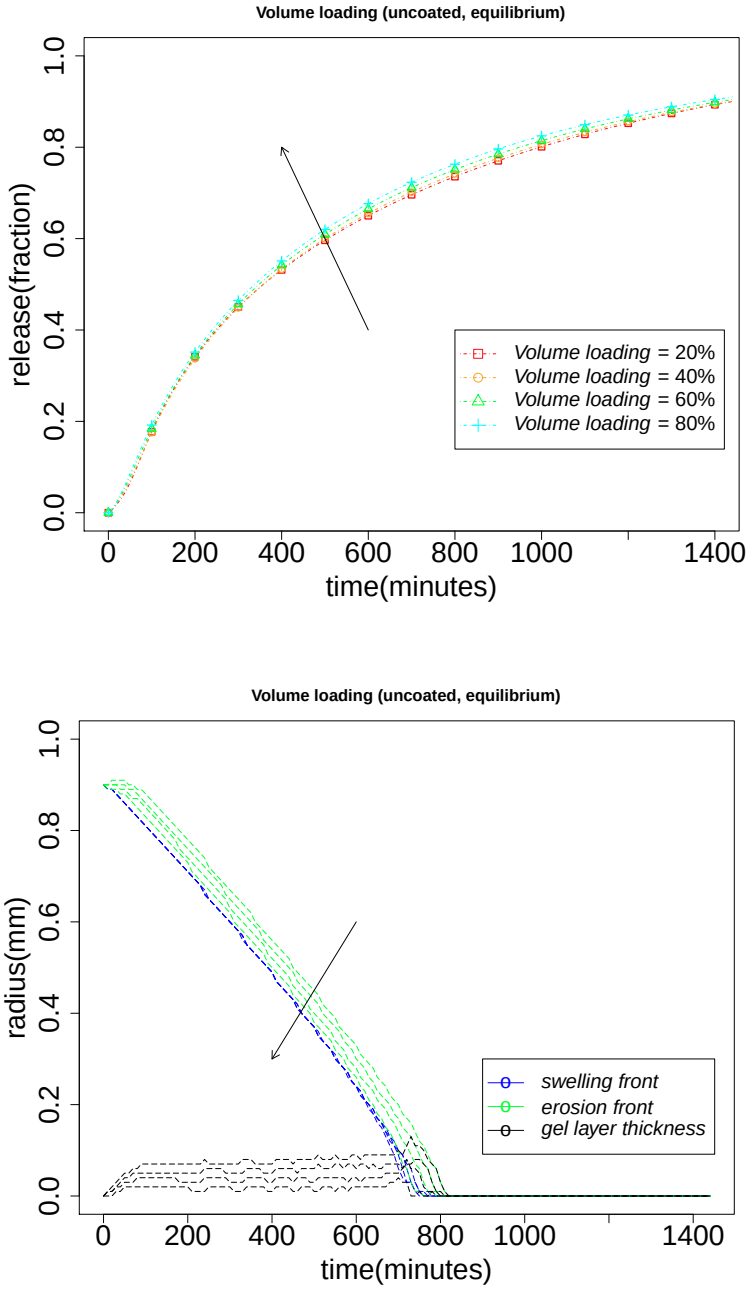


Figure 6.24: Influence of drug loading, on release and front changes, for an equal erosion and swelling release scenario (uncoated beads). (top) The increase in release rates with increase in drug is attributed to smaller gel layer due to increased amount of drug solids. (bottom) The decrease in gel layer thickness is ascribed to the lack of core material due to higher drug percentage. Equal influence of erosion and swelling results in constant gel layer thickness leading to stable release.

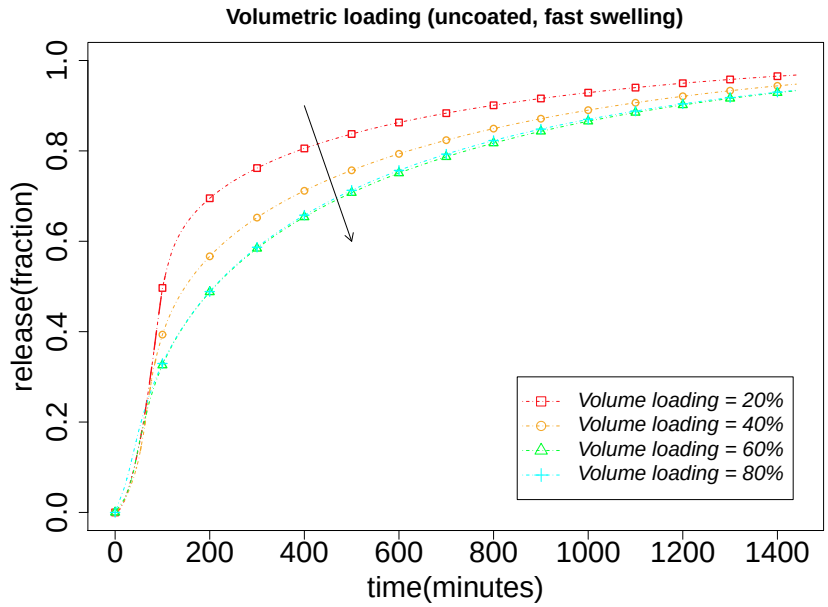


Figure 6.25: Influence of drug loading, on release profiles from uncoated beads, displaying fast swelling behaviour. The initial increase and subsequent rapid decrease in release rates with increased drug amounts is attributed to larger area of the gel layer due to larger swelling.

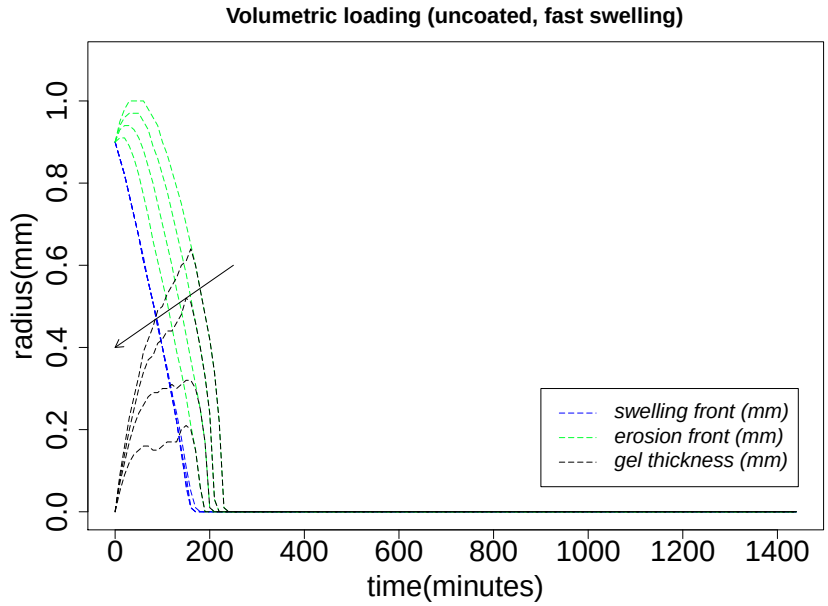


Figure 6.26: Influence of drug loading, on swelling and erosion front changes, for a fast swelling release scenario (uncoated beads). The decrease in gel layer thickness is ascribed to the lack of core material due to higher drug percentage. The high variability in the wet gel phase, as shown, generally leads to unstable release.

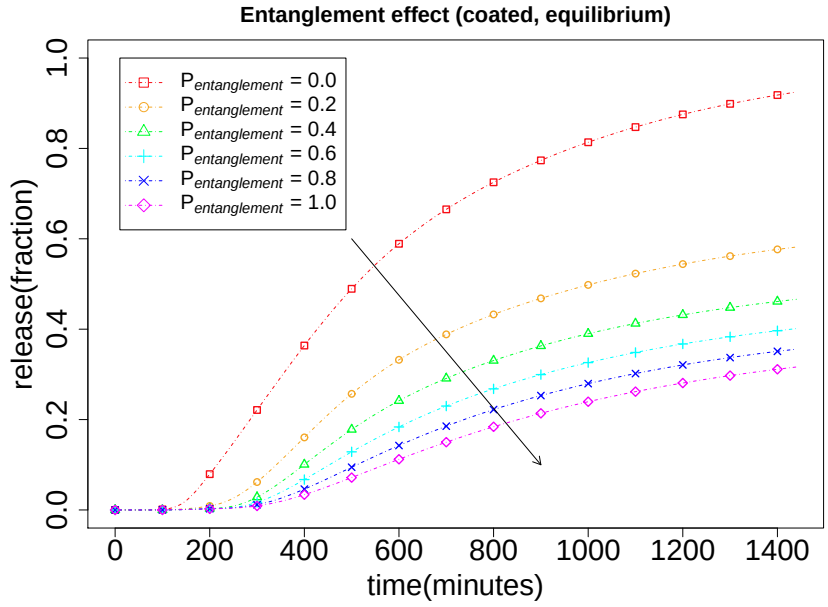


Figure 6.27: Influence of increasing entanglement probability ($P_{entanglement} = P_C$) for release curves in the equilibrium release scenario. These curves show an exponential decline due to the presence of swelling, which blocks the drug diffusion channels throughout the core.

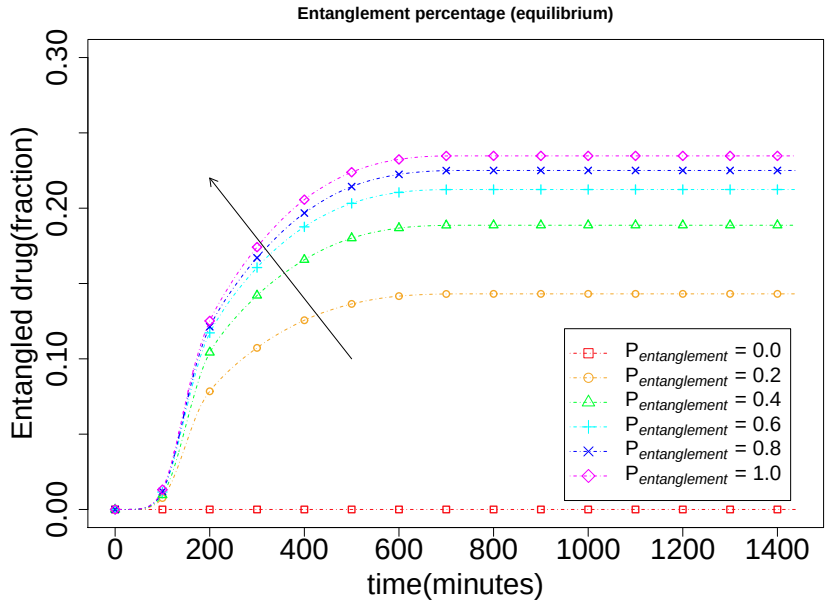


Figure 6.28: Influence of increasing entanglement probability ($P_{entanglement} = P_C$) with the amount of solid drug remaining in the core in the "equilibrium" release scenario. The amount of remaining drug shows an exponential increase due to the presence of swelling, which blocks the drug diffusion channels throughout the core.

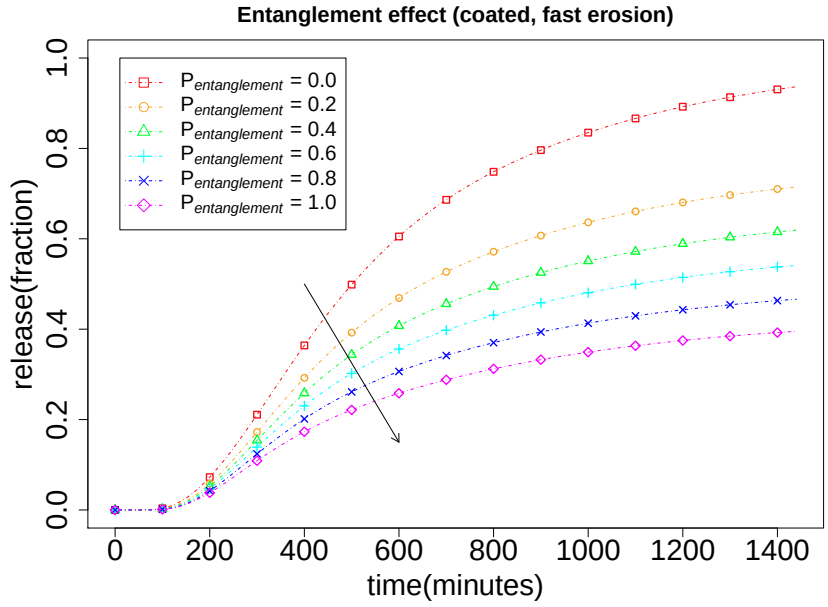


Figure 6.29: Influence of increasing entanglement probability ($P_{entanglement} = P_C$) for release curves in the fast erosion release scenario. The release curves show a linear decline due to lack of swelling with little impact on existing drug diffusion channels throughout the core.

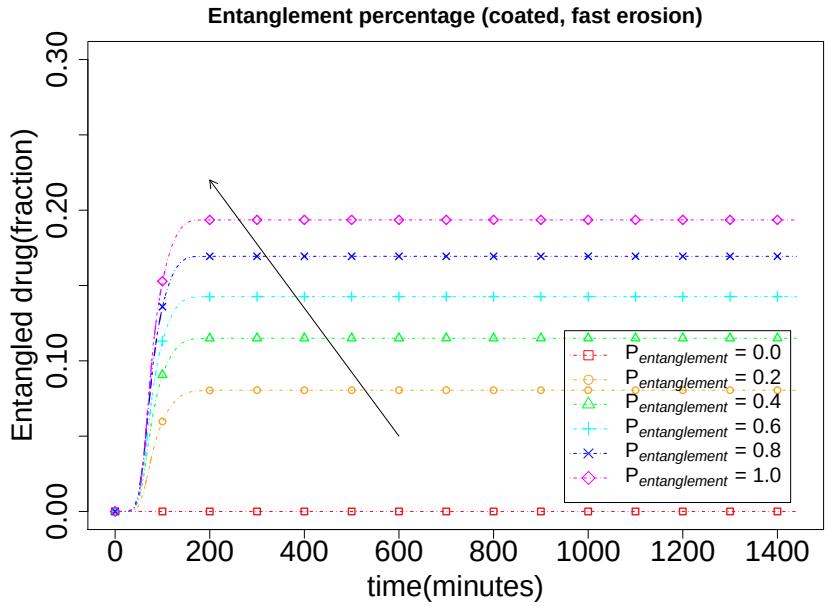


Figure 6.30: Influence of increasing entanglement probability ($P_{entanglement} = P_C$) with the amount of solid drug remaining in the core in the fast erosion release scenario. The difference in the amount of drug remaining shows that entanglement effect is not significant as the speed of erosion did not allow blocking or clustering phenomena to occur.

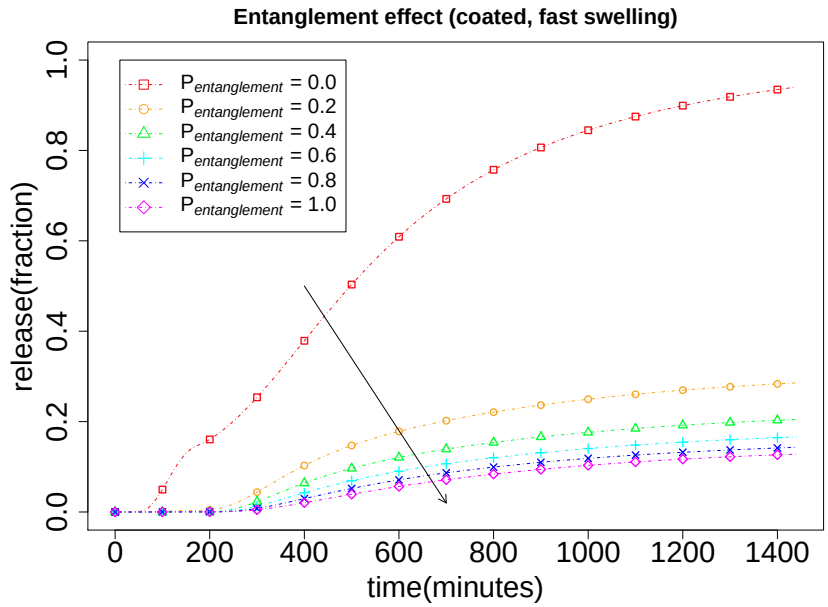


Figure 6.31: Influence of increasing entanglement probability ($P_{entanglement} = P_C$) with the amount of solid drug remaining in the core in the fast swelling release scenario. The amount of remaining drug shows an exponential increase due to the amount of swelling, which blocks the drug diffusion channels throughout the core.

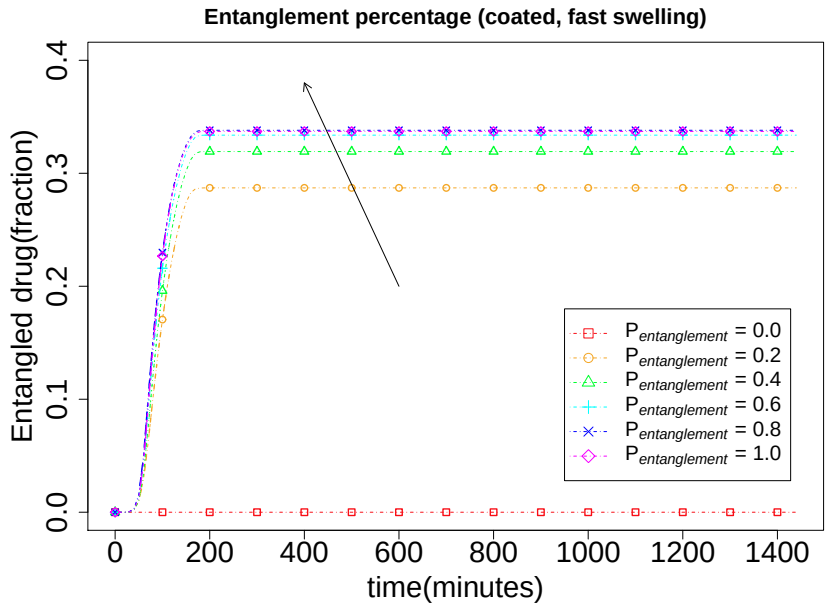


Figure 6.32: Influence of increasing entanglement probability with amount of solid drug remaining in the core in the fast swelling release scenario. The amount of remaining drug shows high exponential increase due to presence of swelling, which blocks the drug diffusion channels throughout the core.

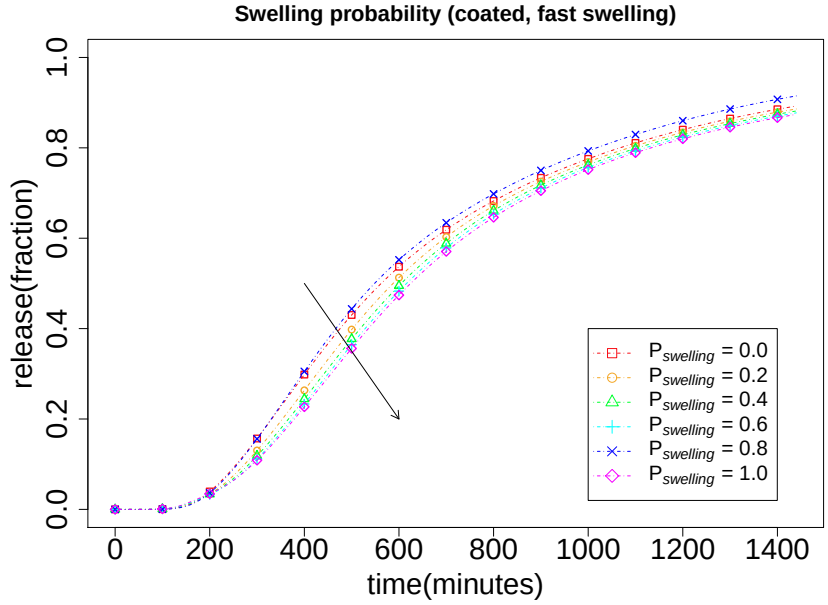


Figure 6.33: The effect of changing a swelling probability that is constant during the time of the simulation. Resulting profiles show a constant decrease with increased swelling.

exponentially decreasing swelling probability (Tanaka and Fillmore, 1979), (Li and Tanaka, 1990) and (iii) sigmoidally decreasing swelling probability (Singh and Weber, 1996).

Figures (6.34) and (6.35) show the results obtained for the case of a highly swellable scenario. Figure (6.30) shows the effect of varying a constant swelling probability (P_S) across the full value range [0, 1]. The resulting profiles show a uniform decrease with increased swelling, which seems in relatively poor agreement to what would be expected *in vitro* (Laaksonen et al., 2009a). This can be explained by the fact that swelling plays a major role at the start of the dissolution process, while later it can be less relevant as the amount of remaining material is low. In the case of exponential and sigmoidal probability change, the initial swelling rate is always high, it is only the dropoff rate that changes. This is apparent from Figures (6.34) and (6.35) where the difference is more pronounced at later stages of the release curve.

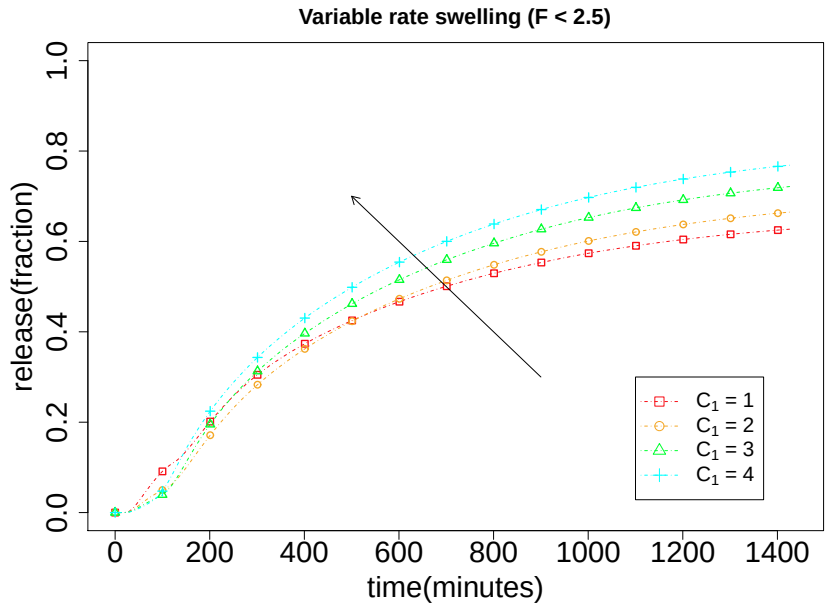


Figure 6.34: Effect of variable swelling probability on drug release curves in the case of exponentially decreasing P_S.

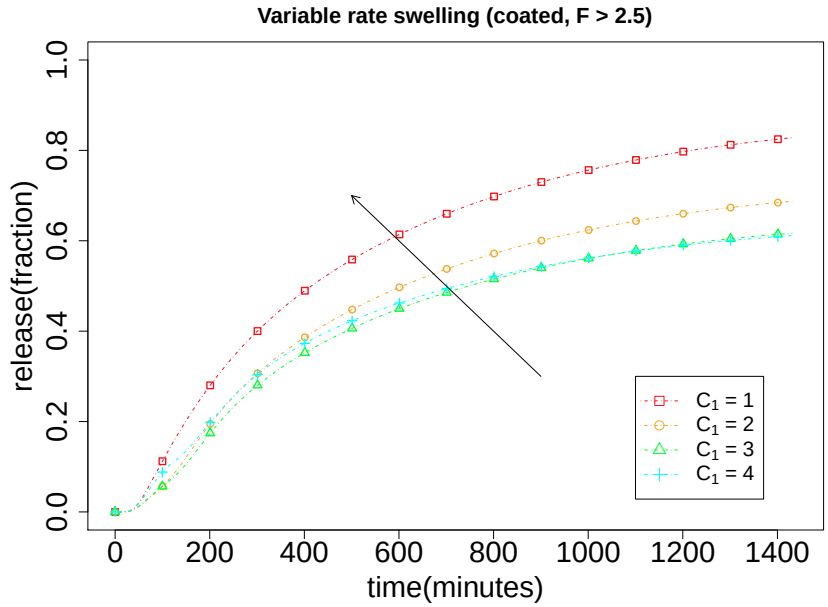


Figure 6.35: Effect of variable swelling probability on drug release curves in the case of sigmoidally decreasing P_S.

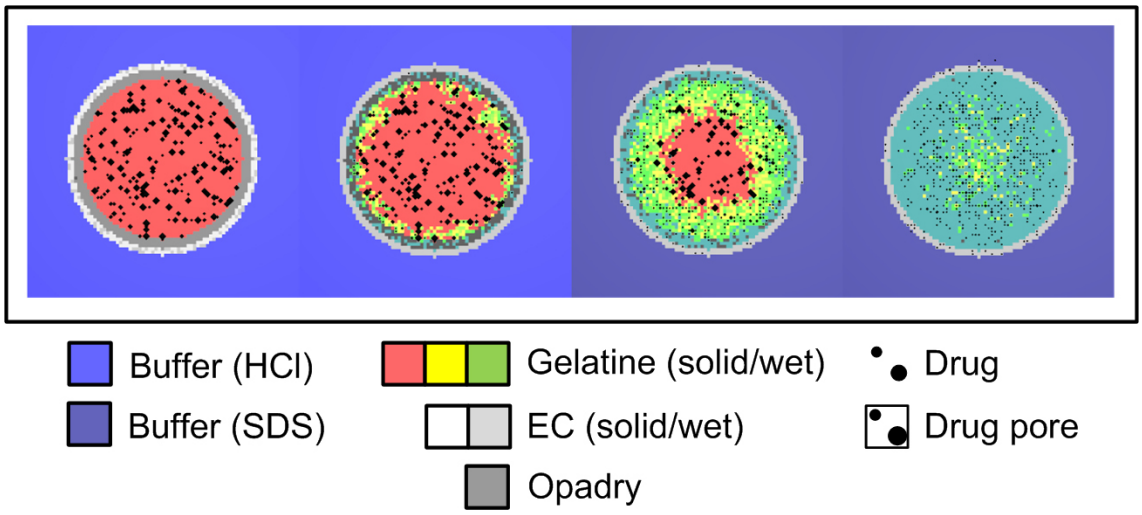


Figure 6.36: Visualisation of the MCES model with relevant cell types. The spheres show progression from initial to the dissolved state through different dissolution environments (HCl and SDS). Green indicates the gel layer, and red the solid gel state.

6.3 Simulations for model for multiple coated drug devices (MCES), Section (4.4)

6.3.1 Validation against experimental data

First MCES model runs, (Section 4.3), were performed for the purpose of comparing simulation results with available *in vitro* experimental data for Surelease/Opadry coated devices provided by Sigmoid Pharma. A set of simulation parameters, corresponding either to direct experimental observations or available literature, was chosen (Table 6.3). Then the respective thicknesses of Surelease/Pectin and Opadry layers were varied to match those values used in the lab experiments. Example visualisation of the performed simulations through different environments (HCl and SDS) is given in Figure (6.36).

The average diameter (d) of the modelled sphere was set to 1.8mm (including both coating layers of varied size). The diffusion coefficient (D_f) of Cyclosporine A was set at $2 \cdot 10^{-6}\ cm^2/s$, placing it within the range of $0.96 \cdot 10^{-6} - 5 \cdot 10^{-6}\ cm^2/s$ as measured by Trammer et al. (2008), Park et al. (2013). Gelatin λ factor, (Equation (4.9)), and Opadry lifetime were estimated from ranges observed during experiments. Based on improved understanding of the main release control phenomena, (obtained during development of

previous models), the swelling probability (P_S) was estimated to be in the low range, as swelling of the core, especially in double coated formulations was not observed to be marked. Relative rate of diffusion of water into the Surelease layer (P_W) was estimated based on measurements made by Ravindra et al. (2000) and the CyA diffusion coefficient (Equation (4.20)) and found to be around 0.16. A similar derivation was performed for the case of water diffusion into the core layer (P_{WG}) using available gelatine/water interaction data from Domenek et al. (2008) which gave an experimental value of approximately $10^{-8}\ cm^2/s$, resulting in relative speeds of approximately 10^{-2} when compared with the main diffusion coefficient. Probability of the clustering/blocking phenomenon was kept at zero, under the assumption that presence of Opadry inhibits this process from occurring. Finally, as the *in vitro* solution was changed after two hours from the *acidic* (HCl) to the *enteric* (SDS) environment, we have kept the solubility of CyA low in HCl, (as reinforced by experimental observations). The only parameter, where direct experimental values could not be obtained, and where controlled, wet-lab investigation would be desirable was relative diffusion of Cyclosporine A through the Surelease membrane. This had to be estimated by comparing available release curves and sensitivity analysis data. The value chosen agreed with most of the experimental curve profiles used, but it would be of interest to validate this in isolated *in vitro* experiments on Surelease slabs. The duration of the simulation was 24 hours, corresponding to experiment, with control values sampled at 1, 2, 4, 6, 12 and 24 hours respectively. The values of the simulation parameters are summarised in Table 6.3.

For analytical comparison we used the similarity factor of Equation (6.1) with standard weight factor as for Equation (6.2).

The obtained results are plotted in Figures (6.37) to (6.42) for each batch respectively, and Table 6.4 summarises the similarity factors achieved. As shown, a very good match was seen for different variations in thicknesses of both layers with similarity factors staying in the range of 2% - 3% and not exceeding 8%. The only outlier was the batch using relatively low amounts of Opadry (2.7%). As indicated from its graph (6.41) slower release was observed, which seemed to be due to some clustering of gelatine occurring within the bead, (even though in our models we assume this value to be zero), as the Opadry levels used were not significant to inhibit this phenomenon fully. Overall, however, a majority of

Parameter	Simulation value	Experimental reference
Sphere diameter	1.8 mm	1.8 mm
D_f	$2 \cdot 10^{-6}$ cm^2/s	$0.96 \cdot 10^{-6}$ - $5 \cdot 10^{-6}$ cm^2/s
Gelatin λ	60 min	60 - 120 min
Opadry(c) lifetime	120 min	30 - 120 min
P_S	0.3	0 - 0.5
P_W	0.16	0.16
P_{WG}	0.01	0.01
P_P	0.2	N/A
P_C	0.0	0.0
HCl solubility	0.0	0.0
SDS solubility	1.0	1.0
Duration	24 h	24h

Table 6.3: MCES model parameters used for comparison with experimental data. The reference parameters were derived from batch data, experimental observations and literature. The coating permeability was estimated based on simulated model data ranges.

Modelled batch	Achieved similarity factor
11-085 (10% Opadry, 11% Surelease, 10% d.l.)	73.46 (3%)
11-086 (10% Opadry, 17% Surelease, 10% d.l.)	86.01 (2%)
11-115 (10% Opadry, 23% Surelease, 10% d.l.)	72.01 (3%)
11-227 (6.3% Opadry, 10.3% Surelease, 10% d.l.)	82.9 (2%)
11-241 (2.7% Opadry, 12.1% Surelease, 10% d.l.)	46.17 (12%)
12-119 (10% Opadry, 11% Surelease, 15% d.l.)	54.96 (8%)

Table 6.4: Values of the similarity factor (f_2) for different modelling scenarios. Results show a good match within the limits of sample variability.

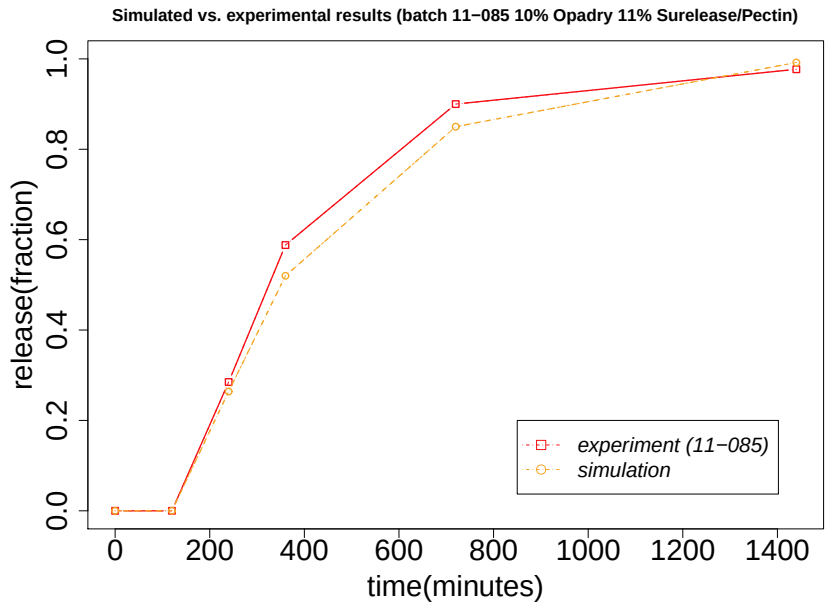


Figure 6.37: Experimental vs. simulated results for batch 11-085 with 10% of Opadry and 11% of Surelease/Pectin layer weight gains respectively. The achieved similarity factor was 73.46 (indicating less than 3% difference in release profiles).

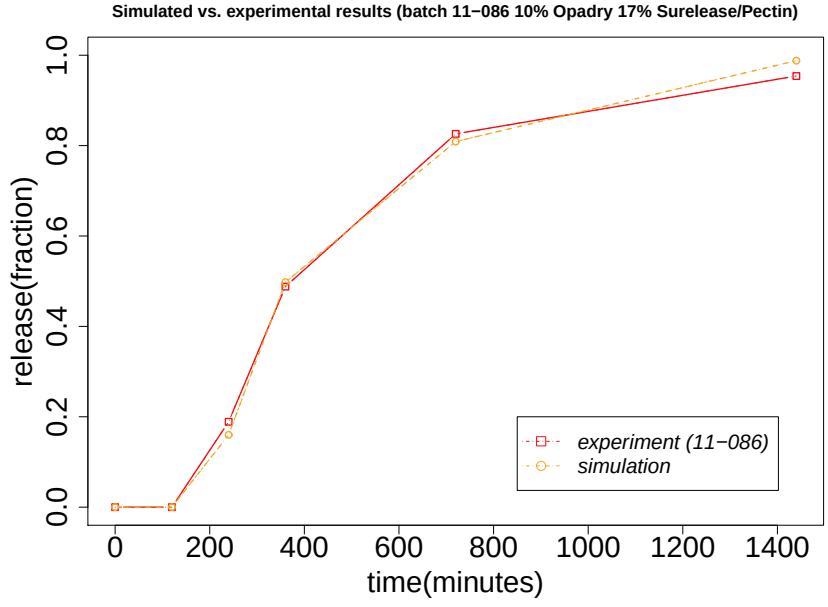


Figure 6.38: Experimental vs. simulated results for batch 11-086 with 10% of Opadry and 17% of Surelease/Pectin layer weight gains respectively. The achieved similarity factor was 86.01 (indicating less than 2% difference in release profiles).

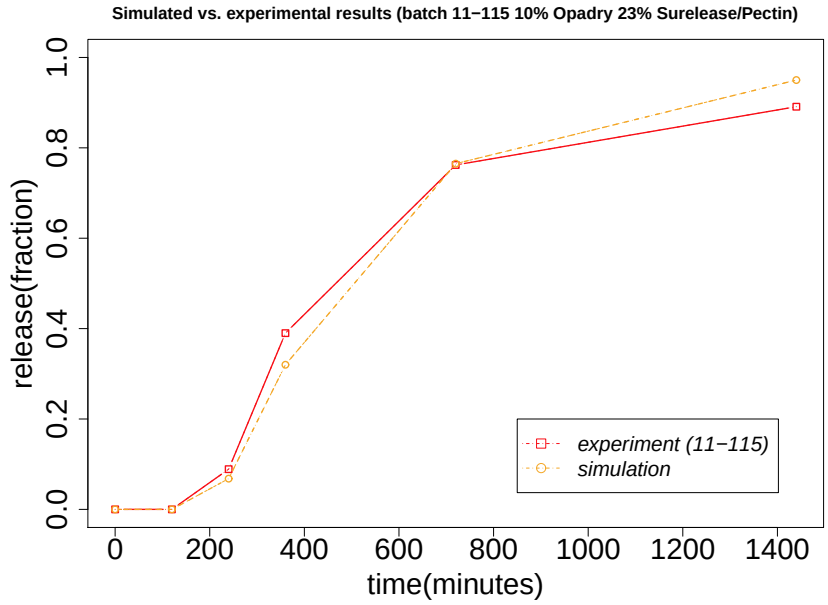


Figure 6.39: Experimental vs. simulated results for batch 11-115 with 6.3% of Opadry and 12.1% of Surelease/Pectin layer weight gains respectively. The achieved similarity factor was 72.01 (indicating less than 3% difference in release profiles).

simulation results were in good agreement with experiment, suggesting a large degree of confidence in the models' ability to react correctly to main design parameter changes.

6.3.2 Sensitivity analysis

Following validation of the MCES model as useful for modelling controlled release from double coated spheres in Section 6.3.1, we investigated its ability to predict factors influencing zero- or near zero-order release in several dissolution scenarios. We chose to analyse the effect of the MCSE model parameters in three key cases: (i) where erosion/diffusion is the dominant release phenomenon; (ii) where swelling of the polymer core is the main phenomenon and (iii) where both phenomena are contributing equally. All three scenarios are examined over a range of both design and internal parameters.

Influence of ethylcellulose/pectin coating thickness

The first parameter investigated was the thickness of the outer, ethylcellulose/pectin, layer which is a major factor influencing release behaviour.

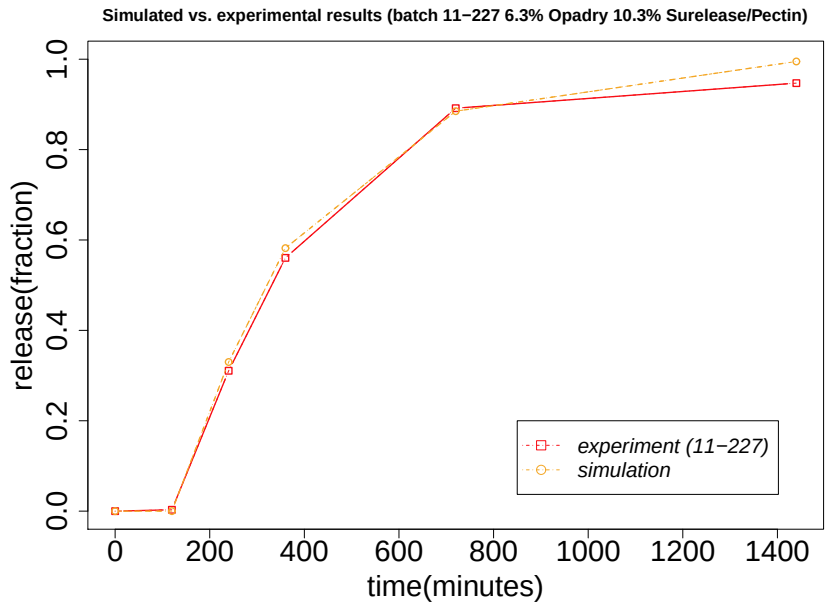


Figure 6.40: Experimental vs. simulated results for batch 11-227 with 10% of Opadry and 23% of Surelease/Pectin layer weight gains respectively. The achieved similarity factor was 82.09 (indicating less than 2% difference in release profiles).

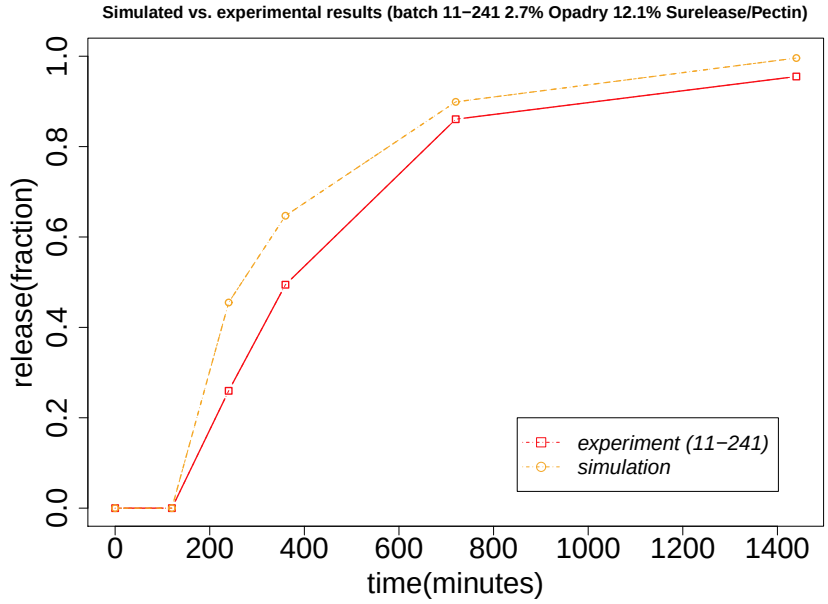


Figure 6.41: Experimental vs. simulated results for batch 11-241 with 2.7% of Opadry and 23% of Surelease/Pectin layer weight gains respectively. The achieved similarity factor was only 46.17 (12% match). We explain this by the fact that release was significantly slower when compared to other experiments and reached the plateau at around 90% indicating some clustering effects were occurring due to low concentrations of Opadry.

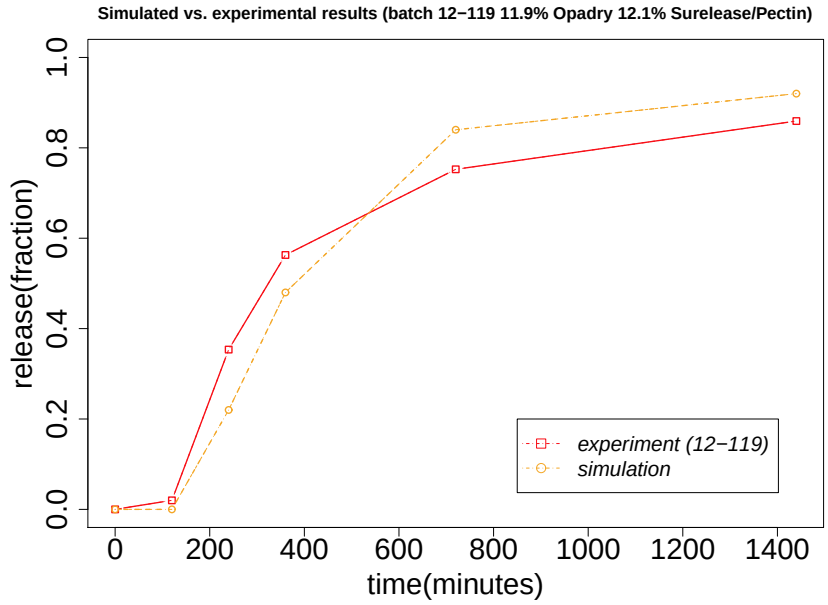


Figure 6.42: Experimental vs. simulated results for batch 12-119 with 10% of Opadry and 11% of Surelease/Pectin layer weight gains respectively. The drug loading in this case was 15% of mass. Achieved similarity factor was 54.96 (less than 8% difference).

Figure (6.43) shows the influence of changing the thickness for the case of fast eroding spheres. As expected, increase in thickness causes a decrease in release rate greater than that observed in the SCDD model. It also increases the initial delay as water needs more time to penetrate ethylcellulose polymeric chains, degrade the Opadry coat and only then penetrate the gelatine core. Examining values of Peppas n factor, this decreases with thickness increase and indicates strong Case II, erosion controlled release, consistent with the expectations of the used parameter set. Best values ($n = 0.86$) were obtained for small coating thicknesses indicating that thinner coatings produce curves consistent with zero-order release ($n = 0.85$). The n factor further increases as the coating thickness increases going further into the region of erosion controlled release.

Figure (6.44) represents corresponding dissolution radii change of the core sphere, indicating that the core radii dynamics are not affected by thickness change. The delay factor caused by the thicker coatings is visible as it also delays the start of the swelling process. The results are consistent with the erodible device with constant gel layer thickness, although, (as indicated by the Peppas factor), erosion forces do play a major role in release.

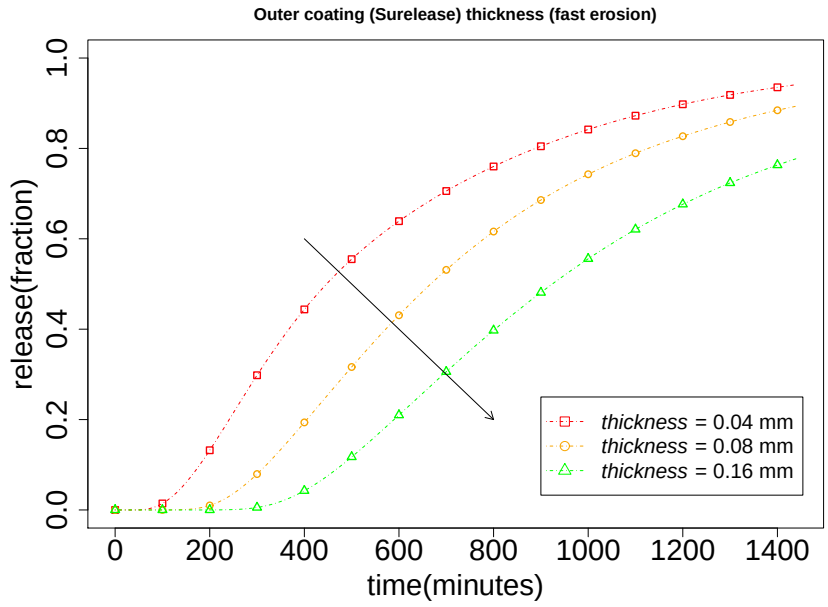


Figure 6.43: Influence of outer coating (Surelease) thickness on resulting drug release curves. The thickness was varied from 0.04 mm to 0.16 mm and causes a decrease in overall release. The profiles closer to zero-order release were better correlated with smaller coating thicknesses with 0.04 mm giving Peppas n values of approx 0.86.

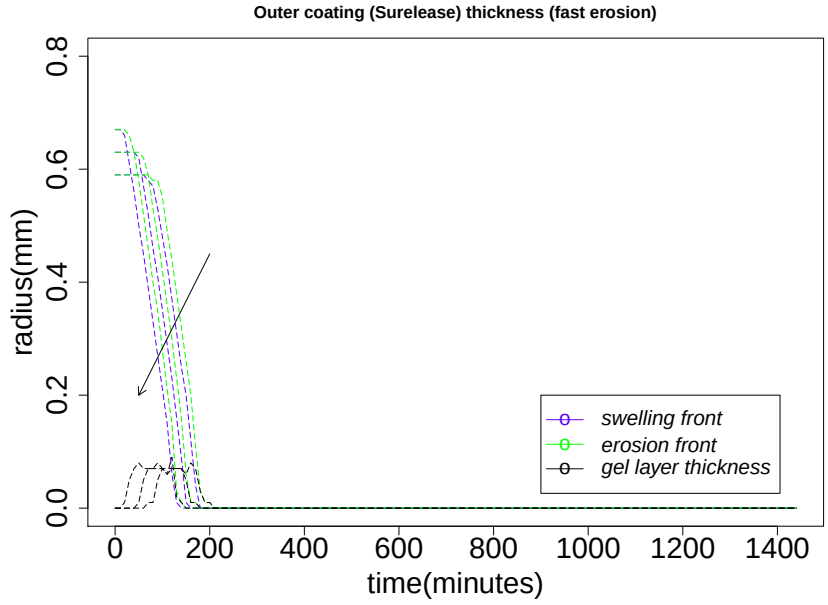


Figure 6.44: Influence of outer coating (Surelease) thickness on resulting core radii. The delay in the start of the swelling process is visible from the picture. The gel layer thickness was constant indicating good zero-order release potential. However a rapid erosion dynamic in the model was the main driver of release.

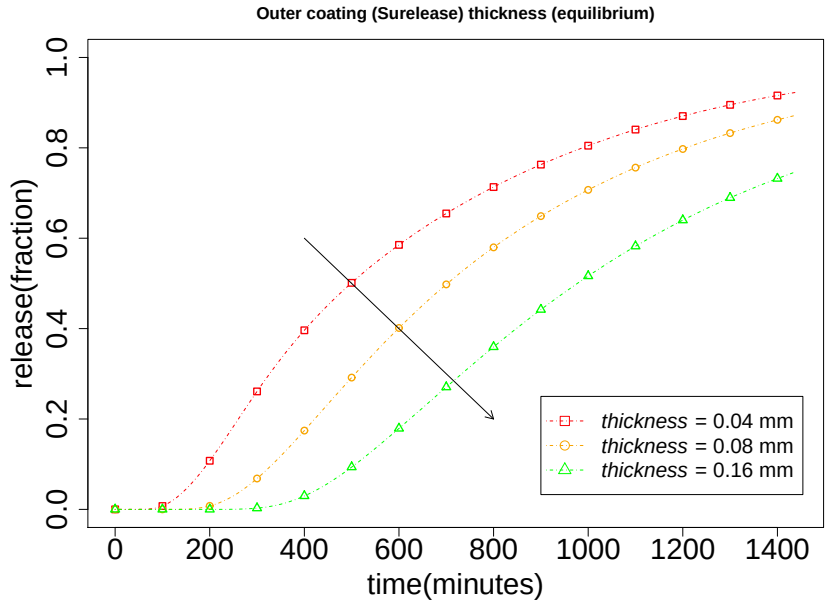


Figure 6.45: Influence of outer coating (Surelease) thickness on resulting drug release curves. The thickness was varied from 0.04 mm to 0.16 mm and causes a decrease in overall release. However, this shows better potential for zero-order release than the rapidly erodible device. Profiles closer to zero-order release were again better correlated with smaller coating thicknesses with 0.04 mm giving Peppas n values of 0.85, indicating zero order release was achieved.

Figures (6.45) and (6.46) analyse the same coating variation, but this time for a scenario where swelling and erosion forces are approximately equal. We observe similar effects for drug release rates and for thinner coatings (due to more constant release in the desired area), with good n values obtained (0.85) indicating zero-order release from the sphere. Radii behaviour shows more stable core degradation, with larger but still not significant swelling effect. The gel layer thickness was constant throughout the release, showing that balancing erosion and swelling phenomena in the core material helps to achieve more stable release, which is in agreement with results obtained in Laaksonen et al. (2009a).

Finally, we looked at the other end of the spectrum by simulating the coating influence in a "fast swelling" scenario. It is important to note here that rapid swelling will also lead to rapid subsequent erosion (as polymer chains will disentangle fast and thus undergo increased exposure to the surrounding buffer much more quickly, leading to their degradation). This is also reported by Laaksonen et al. (2009a) However, the initial swelling will influence the release profile differently from erosion. Figures (6.47) and (6.48) present resulting release

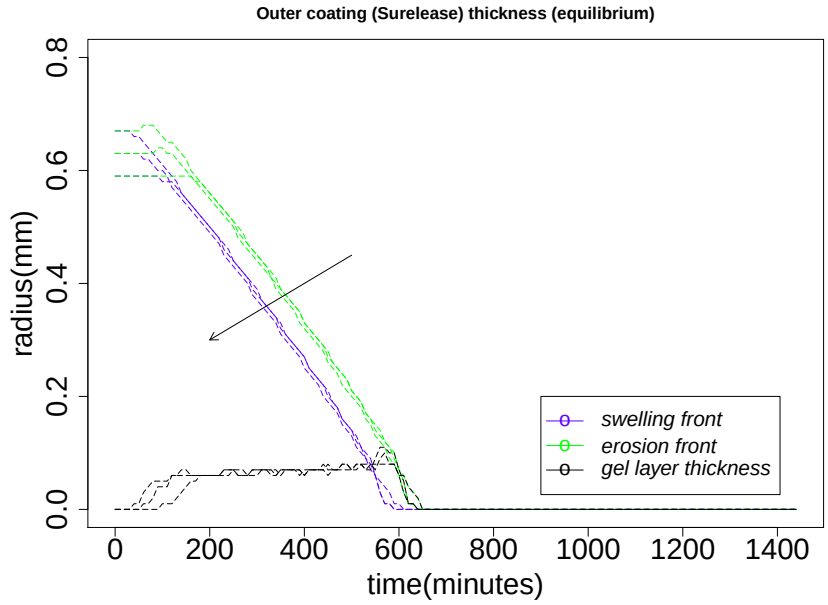


Figure 6.46: Influence of outer coating (Surelease) thickness on resulting core radii for the "equilibrium" scenario. Gel layer thickness was constant, indicating good zero-order release potential, with erosion and swelling dynamics balancing each other.

behaviour. The release is faster in general as the swelling polymer chains are carrying the drug particles out of the sphere. Once the core material is spent, a similar residual release profile is exhibited across all values of the parameter. The best n factor obtained is 0.87, slightly worse than for the equilibrium scenario in terms of obtaining constant release. The resulting radii in Figure (6.48) show the expansion/collapse behaviour of the quickly swellable polymer.

Influence of Opadry® coating thickness

Figures (6.49) - (6.54) show release and dissolution radii profiles when varying the thickness of the inner, Opadry coating, across all three dissolution scenarios. Although smaller, a significant effect on release exists, with thinner coatings again achieving a better value of Peppas n factor (0.92). It is worth pointing out that the variation in Opadry thickness does not take into account the possible occurrence of the clustering/blocking effect and we assume that the smallest coating thickness used is enough to allow complete dissolution of the gelatine core. The results obtained show that the release is still in the erosion-dominated

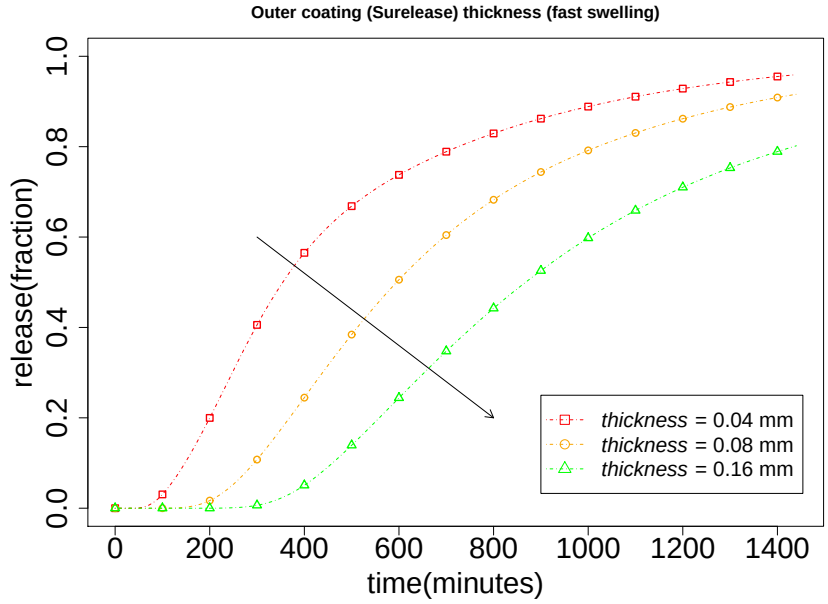


Figure 6.47: Influence of outer coating (Surelease) thickness on resulting drug release curves for "fast swelling" cores. The thickness was varied from 0.04 mm to 0.16 mm and led to decrease in overall release. Profiles closer to zero-order release were slightly worse than the "equilibrium" scenario with the best Peppas n of 0.87.

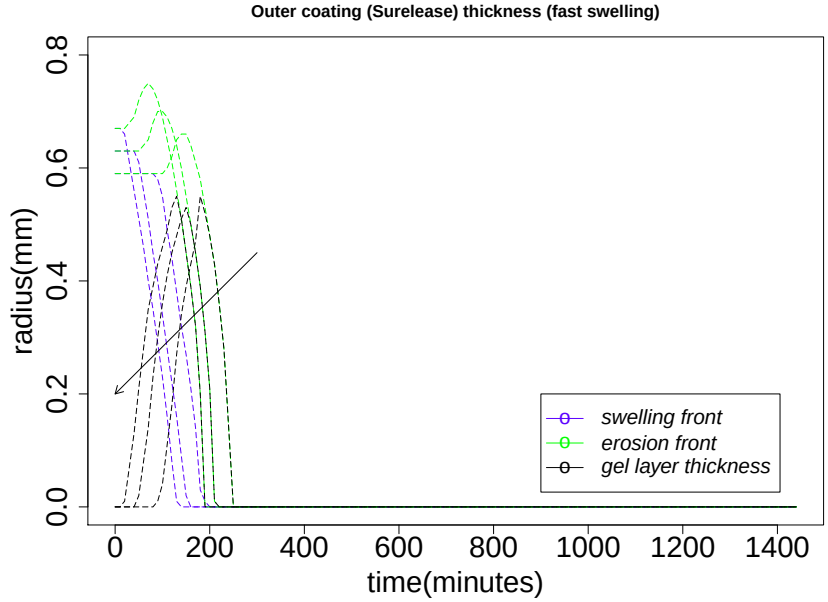


Figure 6.48: Influence of outer coating (Surelease) thickness on resulting core radii for the fast swelling scenario. Profiles illustrate a quick expansion/collapse of the swelling polymer.

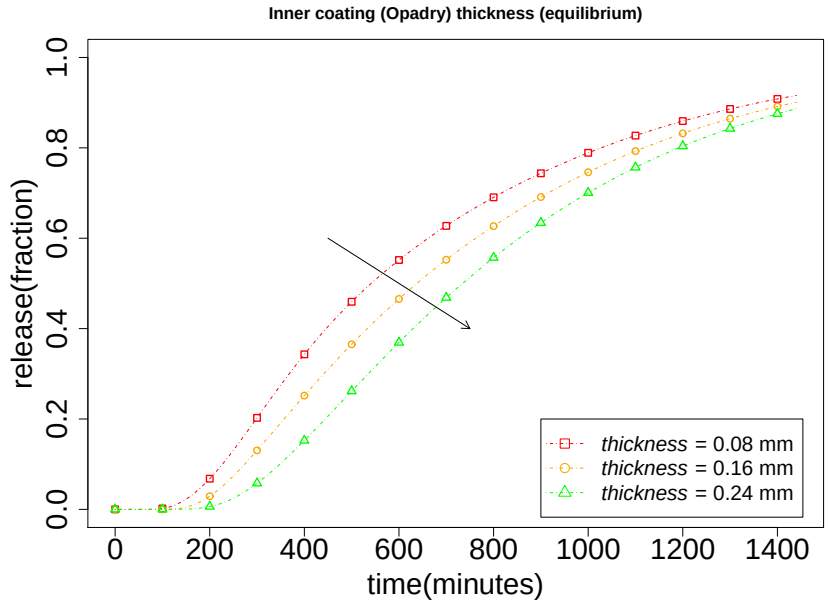


Figure 6.49: Influence of inner coating (Opadry) thickness on the resulting drug release curves for cores, where swelling and erosion are in equilibrium. The thickness was varied from 0.08 mm to 0.24 mm and led to somewhat smaller decrease in overall release when compared to effect of outer coating variations. All profiles show erosion-controlled behaviour with the best Peppas $n = 0.92$.

zone.

Diffusion of drug through polymer coating

Here we look at the influence of the P_P parameter which controls the relative diffusion speed of drug packets through the outer coating. Figures (6.55) to (6.57) present simulation results corresponding to varying P_P in range from 0.2 to 0.8. Although the relative difference between release curves is small, their categorisation according to the Peppas n factor shows the increase in the contribution of diffusion to the release, moving the profiles from erosion-controlled into the anomalous phase. The value of n ranges from 0.83 (for fast erosion scenario) to 0.61 (for fast swelling scenario). The erosion and swelling radii are not impacted by changes in diffusivity of the coating. Although previously reported work deals with the same phenomena in the core, rather than coating, the underlying mechanism and the results obtained are similar to those of Narasimhan (2001) which investigated the effect of drug diffusion on the release profiles.

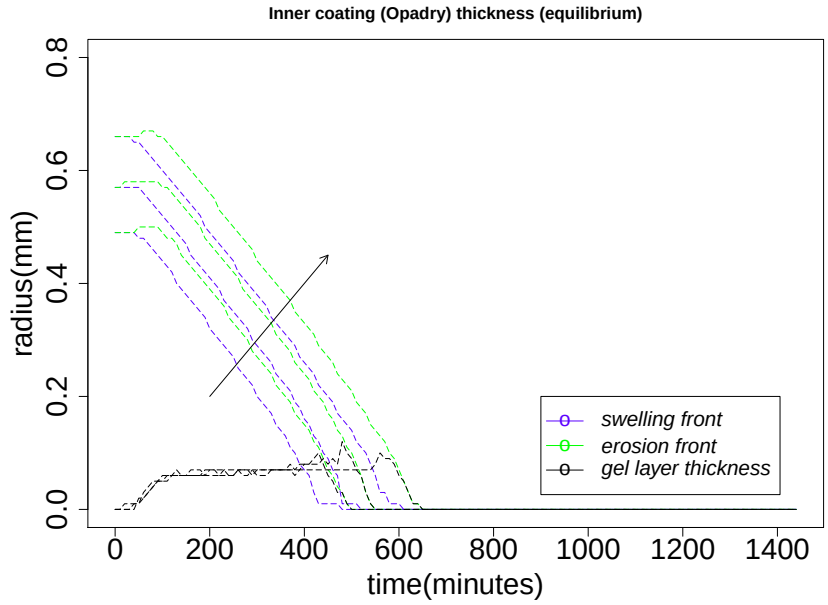


Figure 6.50: Influence of inner coating (Opadry) thickness on resulting core radii for the "equilibrium" scenario. The profiles illustrate a constant gel layer thickness throughout the release.

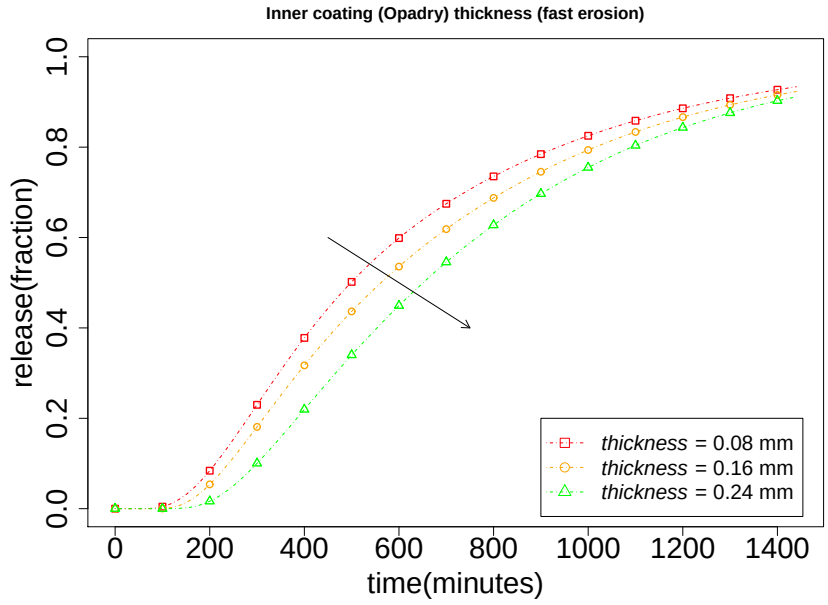


Figure 6.51: Influence of inner coating (Opadry) thickness on resulting drug release curves for cores where erosion is the dominant force. The thickness was varied from 0.08 mm to 0.24 mm and all profiles show strong erosion-controlled behaviour with the best Peppas n = 0.94.

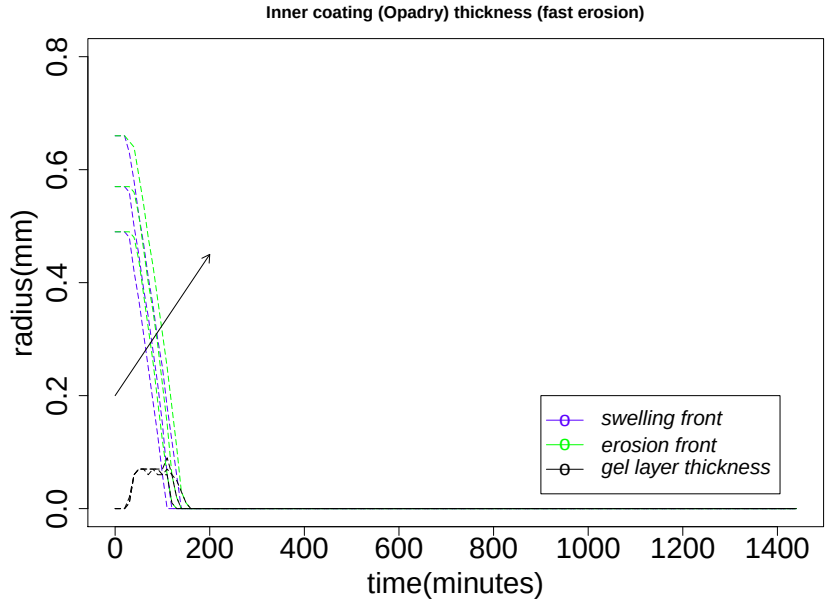


Figure 6.52: Influence of inner coating (Opadry) thickness on resulting core radii for the "fast erosion" scenario. The profiles illustrate a constant gel layer thickness although the core collapse is also rapid, resulting in a somewhat stronger erosion controlled behaviour.

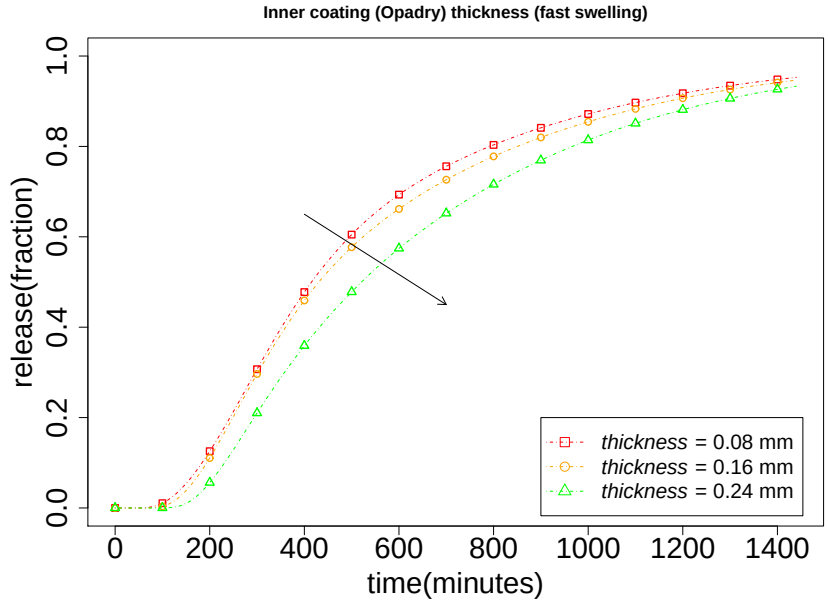


Figure 6.53: Influence of inner coating (Opadry) thickness on the resulting drug release curves for cores where swelling is strong. Thickness was varied from 0.08 mm to 0.24 mm. The profiles still show strong erosion-controlled behaviour with the best Peppas $n = 0.93$.

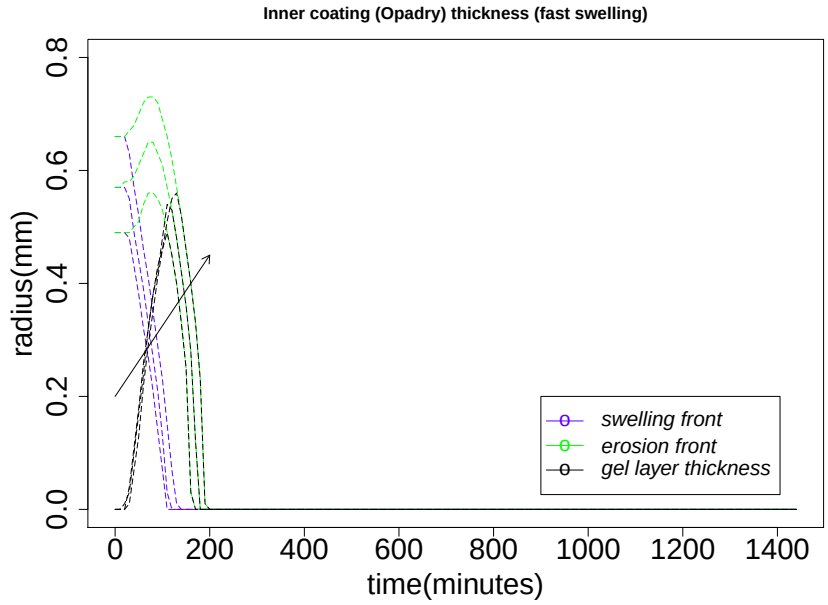


Figure 6.54: Influence of inner coating (Opadry) thickness on the resulting core radii for the "fast swelling" scenario. The profiles illustrate an expanding gel layer, causing fast core collapse.

As the inner coating is not permeable to drug and collapses quickly during bead dissolution it is not of particular importance.

Influence of drug loading to release profiles

Finally, we examine the effect of drug loading on the release profiles from double coated devices. We varied the loading from 20% to 80% of the mass of the sphere for the three distinct scenarios and the results are shown in Figures (6.58) to (6.63). The results agree with observations reported by Siepmann and Peppas (2001), where increasing the drug volume caused a decrease in release curves due to larger dominance of the drug solid phase. The influence of drug volume has the biggest impact on the release of swellable devices, but, for the purpose of achieving zero-order release, lower concentrations are desirable and lead to the best results (values of n are 0.85, 0.82 and 0.87 for fast erosion, equilibrium and fast swelling scenarios, respectively). Higher drug concentrations lead to less polymer material in the core, and hence increased erosion, as evidenced by both the increase in Peppas factor and the radii changes in Figures (6.59), (6.61) and (6.63).

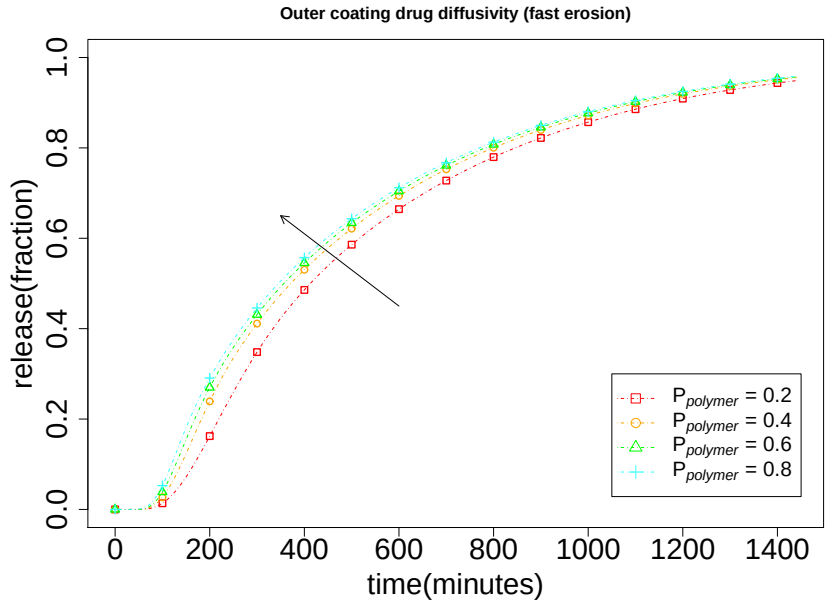


Figure 6.55: Influence of P_P changes on the release profiles in the "fast erosion" scenario. P_P was varied from 0.2 - 0.8 with 0.2 step. Results show an increase in dissolution speed with profiles moving through the anomalous region ($0.66 \leq n \leq 0.83$) towards a more Fickian, diffusion controlled release.

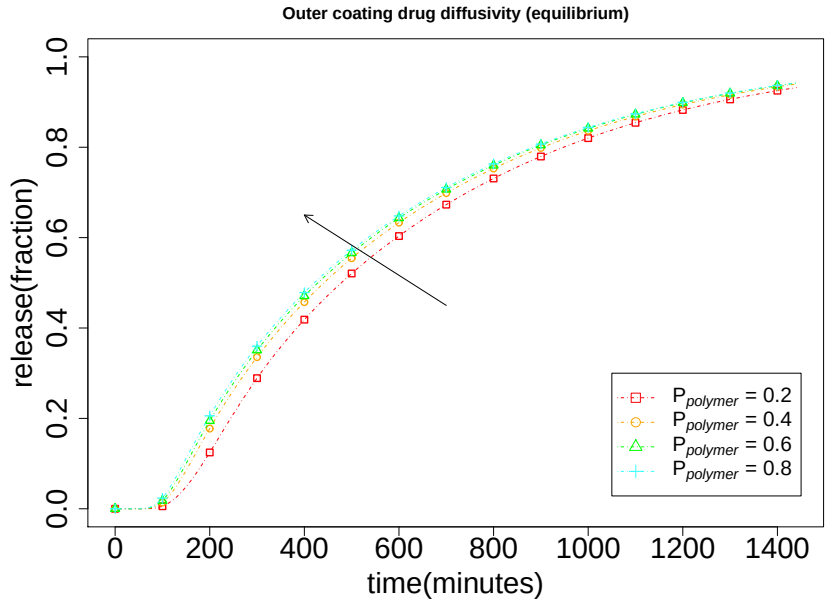


Figure 6.56: Influence of P_P changes on the release profiles in the "equilibrium" scenario. P_P was varied from 0.2 - 0.8 with 0.2 step. Results show an increase in dissolution speed with profiles moving through the anomalous region ($0.69 \leq n \leq 0.78$) towards a more Fickian, diffusion controlled release.

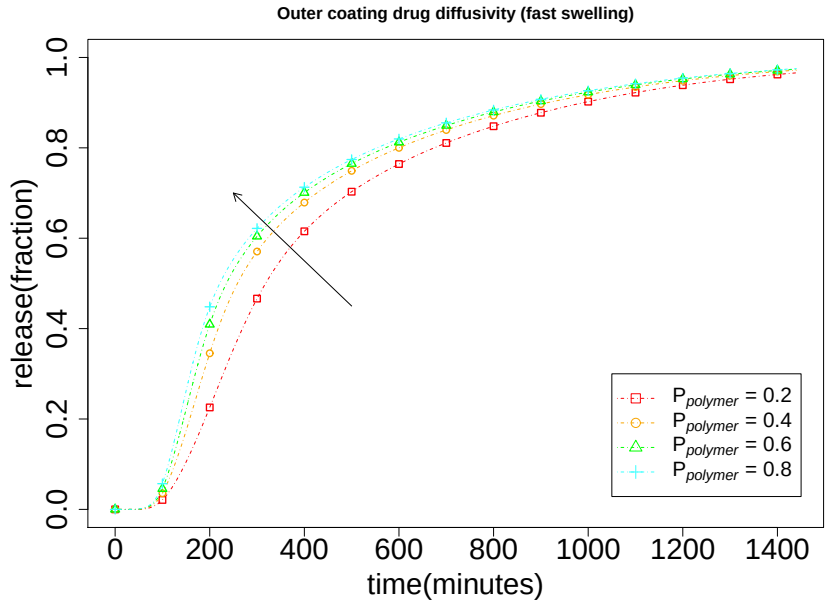


Figure 6.57: Influence of P_P changes on the release profiles in the "fast swelling" scenario. P_P was varied from 0.2 - 0.8 with 0.2 step. Results show an increase in dissolution speed with profiles moving through the anomalous region ($0.61 \leq n \leq 0.82$) towards a more Fickian, diffusion controlled release.

The results of the analysis are summarised in Table (6.5). For the general case model usage, in terms of understanding the parameter influence described here, it is important to note the direction of change in n rather than its absolute value, as the Peppas factor is influenced by many key parameters of the model (for example drug diffusion coefficient as the main rate limiting factor).

Influence of dissolution environment on release profiles

Lastly, we investigated the influence of certain properties of the dissolution environment itself in order to understand how factors such as drug solubility and temperature differences can be modelled.

Figure (6.64) shows the effects of applying solubility and dissolution rate limitations to drug release from the modelled device, (Equations (4.22) and (4.21)). We applied the two together as they describe related phenomena. As shown, we can effectively model the gradual drop in solubility as the drug approaches the fixed solubility rate. This enables the "cap" value to be included in the model once the solubilisation effect of a particular

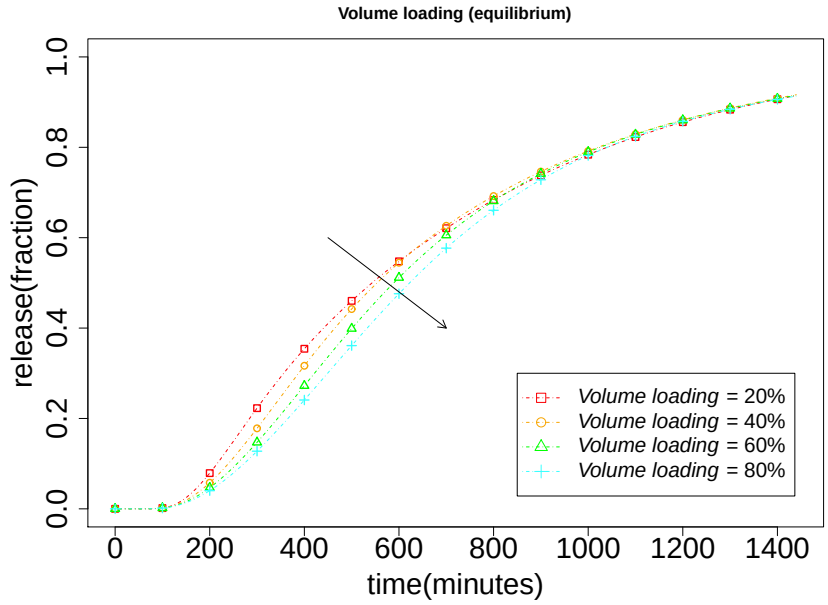


Figure 6.58: Influence of drug loading levels on release curves for the "equilibrium" scenario. The loading was varied from 20% to 80% with 20% steps. The Peppas factor ranges from anomalous behaviour (0.82) for 20% to case II erosion-controlled behaviour (1.1) for 80% consistent with the increase in solid drug state and decrease in polymeric material in the core.

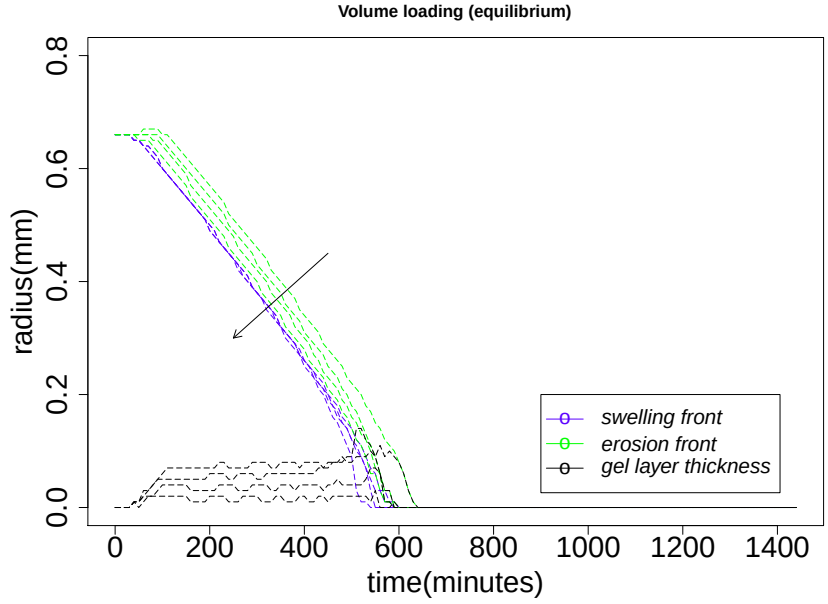


Figure 6.59: Influence of drug loading levels on the main dissolution fronts for "equilibrium" scenario for levels of loading ranging from 20% to 80%. The decrease in front size is directly related to decrease in available polymeric material in the core.

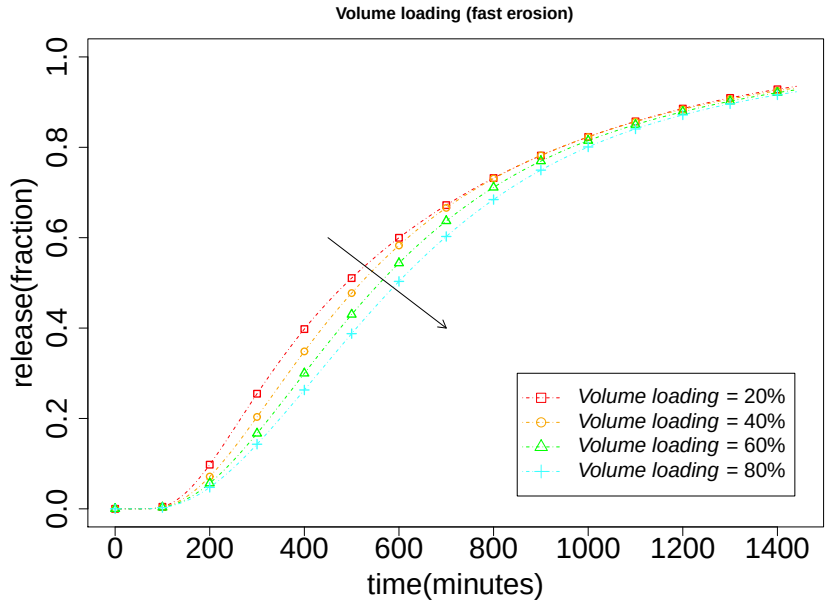


Figure 6.60: Influence of drug loading levels on the release curves for the "fast erosion" scenario. The loading was varied from 20% to 80% with 20% steps. The Peppas factor ranges from zero-order release (0.85) for 20% to case II erosion-controlled behaviour (1.16) for 80% consistent with the increase in solid drug state and decrease in polymeric material in the core.

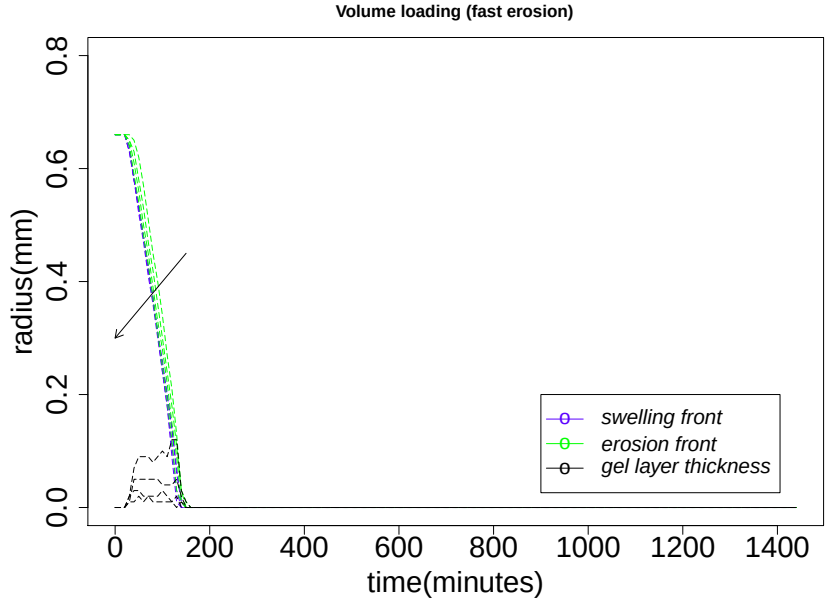


Figure 6.61: Influence of drug loading levels on main dissolution fronts for the "fast erosion" scenario for levels of loading ranging from 20% to 80%. The decrease in front size is directly related to decrease in available polymeric material in the core.

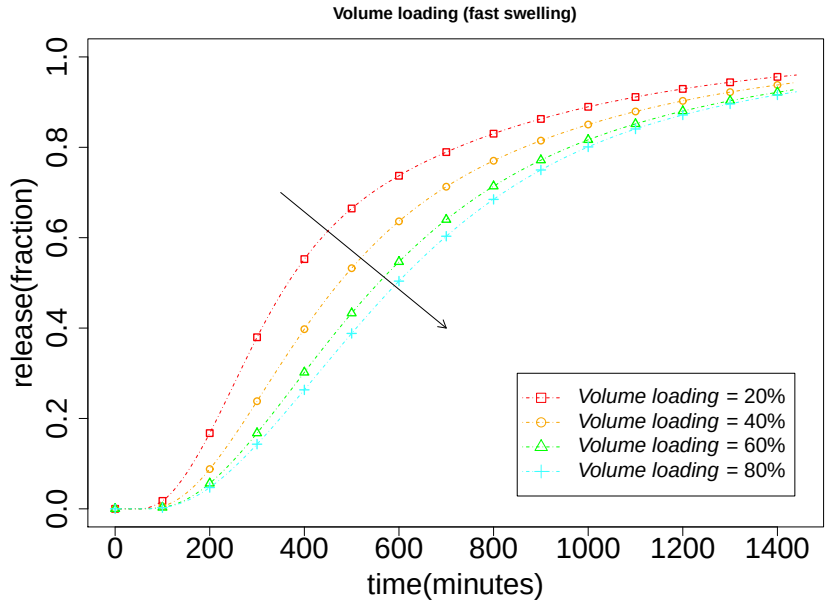


Figure 6.62: Influence of drug loading levels on release curves for the "fast swelling" scenario. The loading was varied from 20% to 80% with 20% steps. The Peppas factor ranges from near constant release behaviour (0.87) for 20% to case II erosion-controlled behaviour (1.09) for 80% consistent with the increase in solid drug state and decrease in polymeric material in the core.

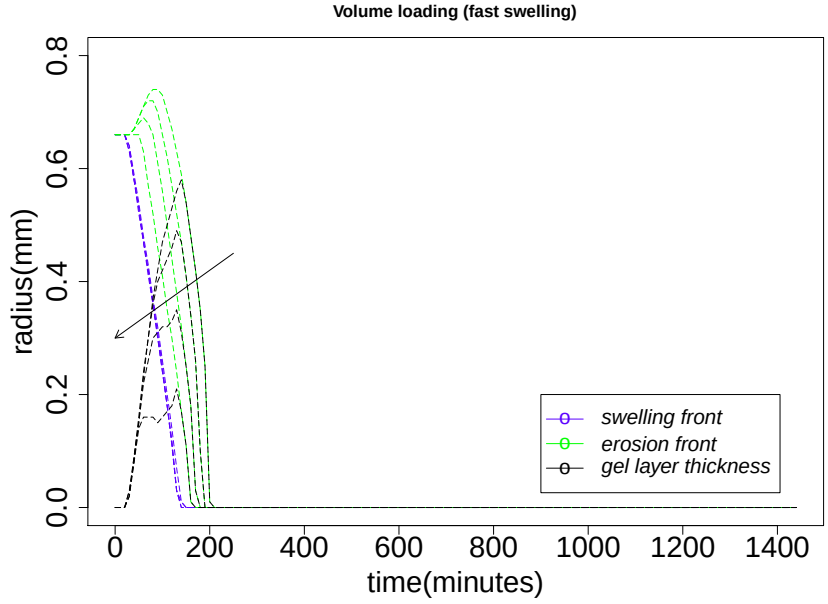


Figure 6.63: Influence of drug loading levels on main dissolution fronts for "fast swelling" scenario for levels of loading ranging from 20% to 80%. The decrease in front size is directly related to decrease in available polymeric material in the core.

Modelled scenario	**Peppas n range**	**Release phenomena**
Thickness of the outer coating	0.85 - 1.19	Erosion dominant over swelling with increasing values of the parameter.
Thickness of the inner coating	0.94 - 1.12	Erosion dominant with increasing values of the parameter. Smaller influence when compared to outer coating.
Coating diffusivity	0.61 - 0.83	Diffusion becoming a dominant process with increasing values of the parameter. Extreme values lead to Fickian behaviour.
Drug loading	0.82 - 1.16	Erosion of the core driving the release as core material is replaced by the drug. Swelling more pronounced for lower values.

Table 6.5: Summary of the parameter sensitivity analysis presented in this section. Results show that model is able to simulate a wide range of dominant release phenomena.

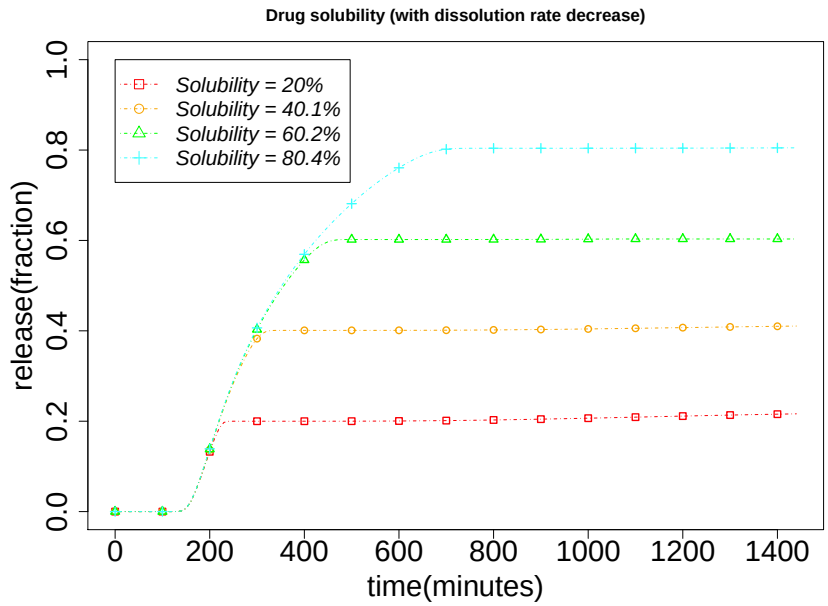


Figure 6.64: Influence of various drug solubility levels on release curves. Drug release reaches a plateau at expected levels, with a slow decline in drug dissolution rates.

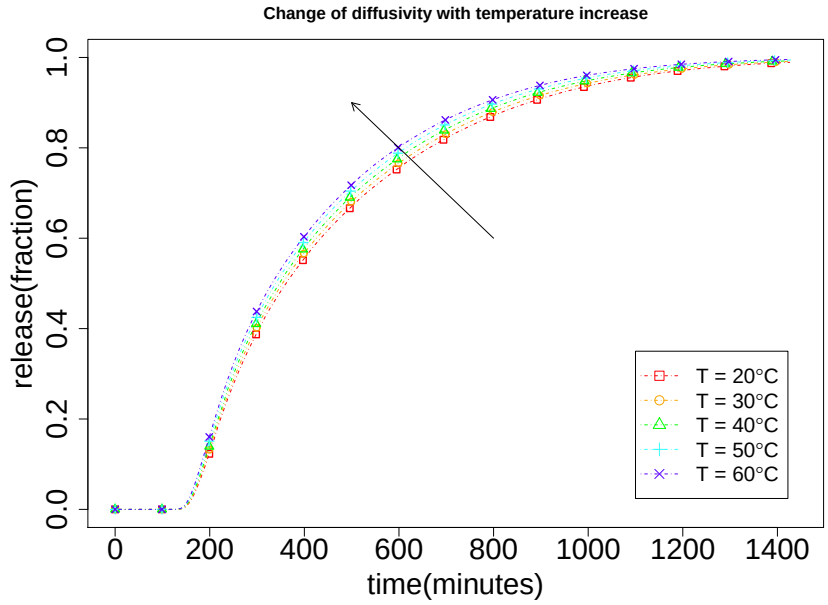


Figure 6.65: Examining the effect of changes of environmental temperature on drug diffusion speeds shows negligible differences.

environment on the drug is known.

Additionally, Figure (6.65) shows the influence of environmental temperature on the release curves, correctly predicting increased release due to increased diffusion rate. In the case of the modelled drug (Cyclosporine A) the differences are small and have little effect, but drugs with lower diffusion rates might be expected to have a larger shift.

6.4 ACA experimental results, Section (4.6)

For each of the described update mechanisms, simulations investigated the following:

- The shape of the release curve during a 24hr period, (characteristic of GI tract transition time for the drug) (Figure (6.66) left)
- The radii of two main reaction fronts (Figure (6.66) right).
- The device composite structure changes during characteristic stages of the dissolution, (visualisation of details) (Figure (6.67)).

In Figures (6.66a1), (6.66a2) and (6.67a) we show the results for *synchronous* updates, (used as a basis for relative comparison with all subsequent ACA methods). Comparing with

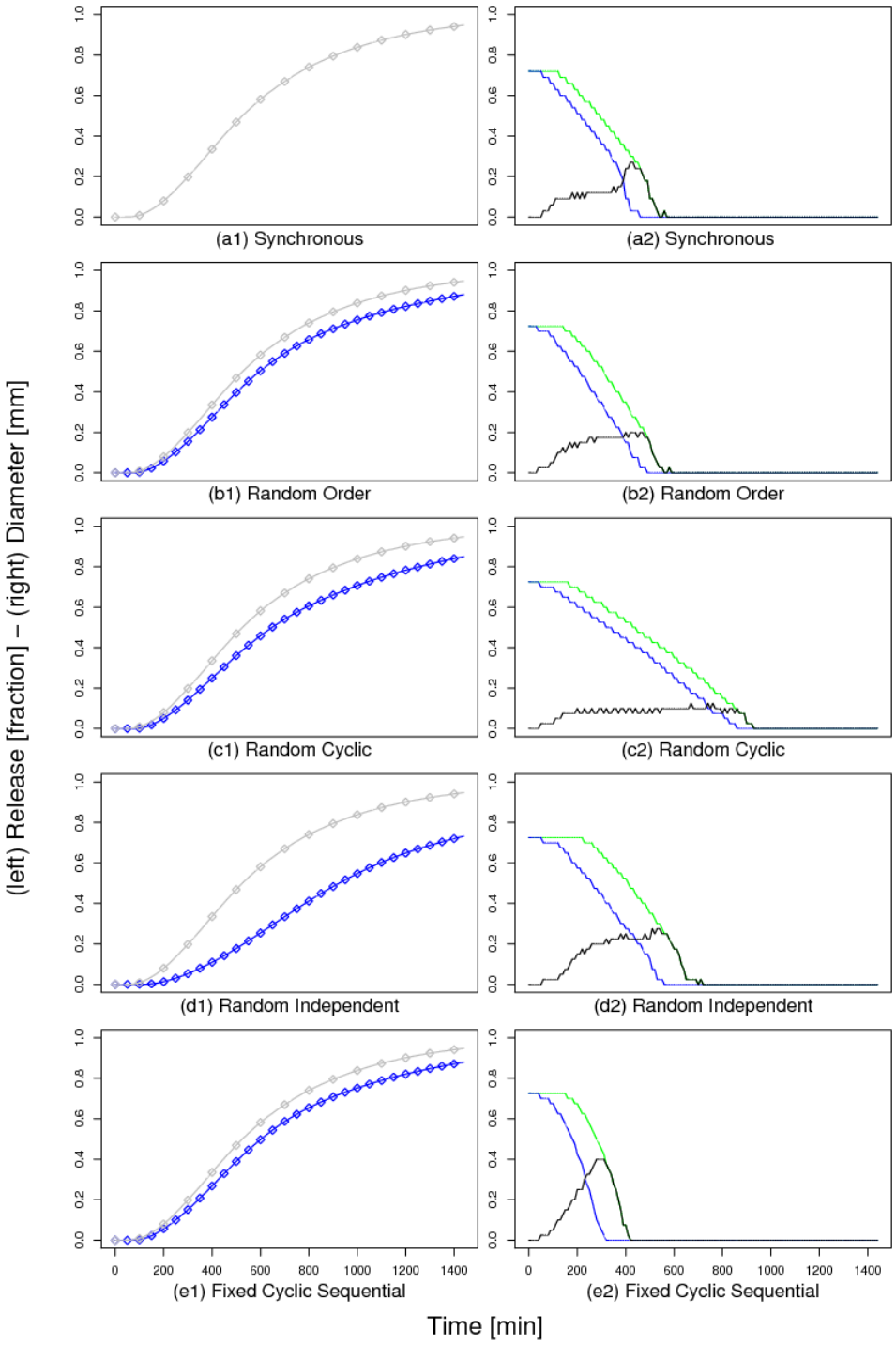


Figure 6.66: (left) Drug release curves during a 24hr simulation period for different update methods (blue - release curve, grey - synchronous reference value); (right) Bead radii changes over time (green - swelling front, blue - erosion front, black - core radius).

random order updates, (Figures 6.66b1, 6.66b2, 6.67b), we find a good match, with negligible release curve difference, (indicating that the methods are effectively interchangeable e.g. where synchronous update is deemed more appropriate, (for specified structure for example (Bezbradica et al., 2012)).

Figures (6.66c1/d1), (6.66c2/d2) and (6.67c/d) show possible alternatives to the asynchronous *random order method*. It can be seen that the *random independent* selection (Figure (6.66)d1) produces release curves, which are significantly shifted with respect to expectation, although radii behaviour is similar, in the sense that polymer transitions occur at the same rate. Features, which give rise to this discrepancy, can be observed in model visualisations, where large *drug clusters* (black diamonds) occur as a consequence of some cells being updated more often than others, (Figure (6.67d)). *Random cyclic updating*, on the other hand, produces release curves which are qualitatively similar to those expected, (Figures (6.66c1), (6.66c2) and (6.67c)), although radii decrease dynamics are much slower. Finally, Figures (6.66e1), (6.66e2) and (6.67e), show results obtained using sequential matrix sweeps. This approach is not recommended due to the significant *bias*, which can be observed in the visualisation, leading to highly non-realistic radii dynamics.

It is also important to note that our results do not display the variance associated with many CA models when switching from asynchronous to synchronous updates, (Fatès et al., 2006; Sloot et al., 1999). As the stochastic model is both slow changing, with usual probability values used «1, and highly symmetrical, due to the spherical device geometry, results obtained do not show highly anomalous results. This is in line with theorems presented in Toffoli and Margolus (1987) relating to equivalence between slow changing synchronous and asynchronous update processes in CA models.

As a final assessment, we examined the overall performance in terms of simulation length using different ACA mechanisms in thread-level execution context. As expected, sequential algorithms were the fastest, as these could utilise the CPU memory cache better. The random cyclic variants also perform well, which might make them an optimal choice for the scenario where the simulation efficiency is needed, as opposed to state update realism. The worst performers are the random order algorithms which are not able to leverage the processor cache, due to constantly changing order of memory access. However, these offer

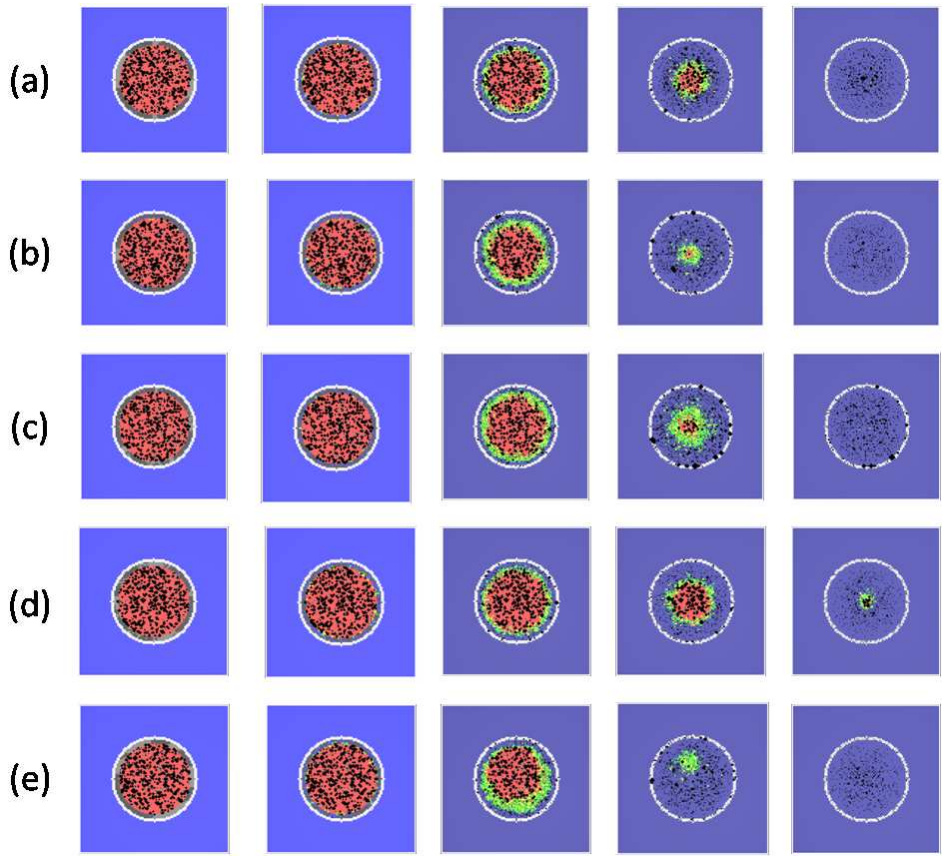


Figure 6.67: Model visualisation for 10, 30, 150, 400 and 700 minute intervals, respectively: (a) synchronous; (b) random order; (c) random cyclic; (d) random independent; (e) fixed cyclic sequential.

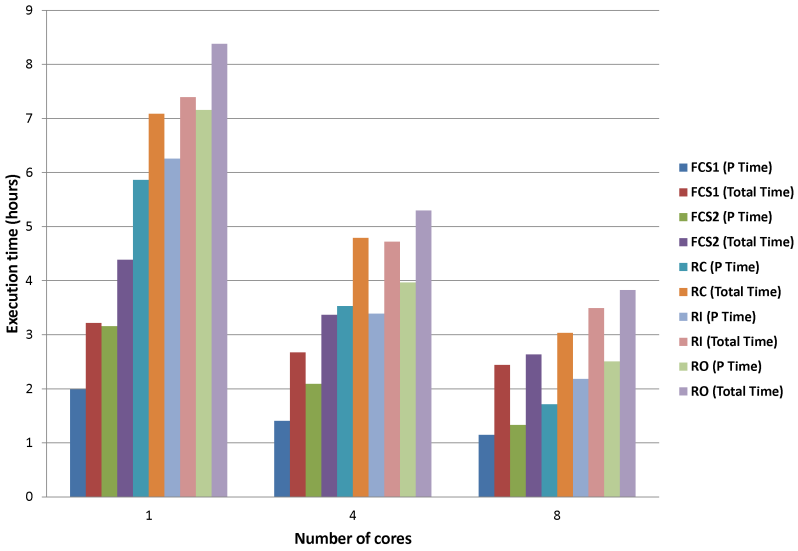


Figure 6.68: Comparison of parallel and total simulation times for different synchronous and asynchronous update mechanisms. The examined mechanisms are, in order, FCS1 - Fixed Cyclic Sequential (1st variant); FCS2 - Fixed Cyclic Sequential (2nd variant); RC - Random Cyclic; RI - Random independent; RO - Random Order.

the best simulation realism and precision when compared to the synchronous variant, so the pros and cons of each should be weighed when making the decision. Figure (6.68) shows the performance gains associated with switching between different asynchronous update methods. We can observe that methods which utilise the CPU caching better due to the inherent order of updates generally perform better. Figure (5.7) (middle) shows the performance profile for synchronous updates when switching from a thread-level to process-level parallelisation model (Bezbradica et al., 2012). Although synchronous updates do not perform to the same level as asynchronous ones, they do not have a parallelisation limit, and thus, ultimately, can be scaled to any number of nodes allowed by the model size.

6.5 Summary

In this chapter we presented the results and accompanying analysis of simulation runs performed for each of the developed models (described in Chapter 4). For each model, two kinds of simulations were run. First type were the ones intended to validate to what degree the models were able to simulate *in vitro* dissolution of an actual pharmaceutical device. The second type focused more on the model outputs for different kinds of erosion and swelling scenarios.

Results obtained during the experimental validation stage are promising and show that probabilistic approach is viable for pharmaceutical modelling methods. The results obtained by the ECDD model represented an initial "gauge" which was used to further refine the cellular automata rules and derive more complex models (SCDD and MCES) whose match with the experimental data was more satisfactory.

Running sensitivity analysis on all models, on the other hand, demonstrated the *versatility* of the underlying modelling mechanism. We are able to simulate different intrinsic polymer behaviours with only a change in the parameter set and add incremental components, such as additional coatings to the modelled device. Utilising standard release curve classification methodologies we are able to profile our results and get an insight into the *in vitro* behaviour based on the simulated data.

Both investigations also emphasised the need for other methods as means to optimise values of the input parameters, further explored through application of Inverse Monte Carlo

methods, Chapter 7.

Finally, at the end of the presented chapter we digressed a bit into the computational problematic of the *asynchronous vs. synchronous* approach to iterating over the simulation space. We showed that choosing the proper iteration algorithm has a significant impact on the obtained results and the simulation realism, and has to be taken into account during model development. The type of asynchronicity chosen was not only shown to have an influence to the release curves and dissolution fronts, but also on the overall model performance. Our findings show that one of the most appropriate solutions is *random order asynchronous*, but a synchronous solution is viable as well. In this regard, model visualisation provides valuable additional insight on structural behaviour and dissolution mechanisms, which is not readily apparent from working with standard release curve data alone, or which is inaccessible from supplementary experiment. These findings could be useful for future modelling scenarios where it may be necessary to switch from one update mechanism to another, e.g. for large-scale optimisation, or in response to the need to describe component interactions in various tailored solutions.

Chapter 7

Inverse Monte Carlo

We have so far covered development and usage aspects of probabilistic drug dissolution models in two main stages: *model development* - in which fundamental physico-chemical laws are translated into model behavioural rules and *model prediction* - in which a constructed model is used to understand the physical system behaviour under novel parameter combinations. In between the two, however, there is an essential phase of *model calibration*, where the initial *base mapping* is established between the experimental results and the starting set of model parameters that will be tweaked in the prediction phase, (Figure (7.1)). We now focus on this calibration phase, in which it is essential to understand how the structure of the model relates to experimental data, as this forms the basis of the later predictive analysis.

In our examination of the suitability and application of Monte Carlo (MC) techniques to modelling of drug dissolution system phenomena we have so far mostly paid attention to the "*direct*" question. That is to say, starting from a set of parameters as the basis for generating different probability distributions applied in the model space, we have measured the resulting drug release curves. However, the values of the constituent parameters of those sets can only be *estimated* from available experimental data. Noise is often an issue and some parameters values are difficult to obtain without costly experimental analysis. Also, various models include parameters, the purpose of which is simplification of model realisation rather than direct one-to-one mapping with the physical world.

In the calibration phase, we turn the initial problem around by asking the *inverse*

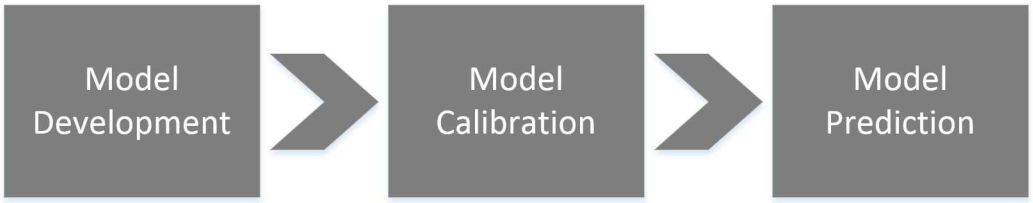


Figure 7.1: Successive stages of the modelling process. (1) During model development behavioural rules are established to represent the physical processes occurring within the modelled drug device. (2) During the calibration phase, certain (non-design) parameters, values for which cannot be obtained from experiment, are examined and indicative ranges investigated. (3) Finally, when both the rule set and base parameter values are known, the model can be used to predict drug release curve changes for changes in design parameters.

question: starting with the experimental data given and assuming certain error margins, what is the *optimal set* of model parameters that would describe the experimentally observed system behaviour. The inverse problem is always posed within a certain set of constraints, (well-known experimental conditions), so that the problem space can be constrained, in order not to end up answering an extremely general question.

7.1 Applications of inverse modelling

IMC methods are a variation on the standard Metropolis-Hastings algorithm (Metropolis et al., 1953), a Markov Chain Monte Carlo (MCMC) method that calculates random sequences from a probability distribution, where direct sampling is not applicable (i.e. the exact probability distribution function is not known). The aim of the method is to produce a structural model which is consistent with one or more sets of experimental data (within acceptable errors), and subject to a set of constraints. Its purpose is not to answer the question of model *correctness*, as that cannot be known in the general case, but rather to gauge the *usefulness* of the model in understanding the relationship between the structure and some physical property (McGreevy, 2001).

It is important to emphasise here, before reviewing the specific inverse modelling methods of interest to finding our solution, the difference between Inverse Monte Carlo and standard data fitting methods (Mosegaard and Sambridge, 2002). The idea is to start from

a *known* experimental model and *derive* the possible values of unknown input parameters. In doing this, we must explore the parameter space. The best parameter set will, of course, give the best match to the experimental results, but that is of secondary importance to some sense. The value of parameters found in this way is they can be used for future prediction, rather than to evaluate the direct model in itself!

7.2 Application of inverse methods for elucidation of unknown parameter values; Sampling from multi-variate distributions

In this chapter, we apply the inverse Monte Carlo (IMC) algorithms to the solution of the inverse problem posed at the beginning of this chapter. We present an approach based on Metropolis-Hastings and Gibbs algorithms for sampling from the model parameter space. As a preliminary, we explore possible approaches that help illustrate issues that can arise during inverse modelling.

7.2.1 Sampling from joint parameter distributions

Simulated Annealing

At the core of inverse algorithms, (incorporating a general Bayesian approach), is *sampling* from an unknown joint distribution of the parameter space. For a single-dimensional Normal distribution, methods such as *adaptive rejection sampling* can be used, (Robert and Casella, 2011), although MC methods in general could be considered to be too costly for single dimensional applications, where more direct numerical methods can be applied (Mosegaard and Sambridge, 2002).

However, for multi-variate parameter problems, MC numerical methods give a considerably better convergence rate, (order of $N^{-1/2}$ as opposed to $N^{-1/M}$, where N is the number of iterations and M the number of dimensions in the problem space).

We present here several viable approaches to solving the inverse sampling problem in drug dissolution models and outline the advantages and disadvantages of each. The methods generally build on elements of each other, so represent a natural evolution in application to

problems of increasing complexity, and we present them in this corresponding sequence.

The first approach utilises simulated annealing for finding the global optimum of a release function in large, discrete search space. Each point of the search space (s), analogous to a state of the physical system is considered, together with the function $E(s)$ to be minimised.

At each step, the heuristic would consider a neighbouring state s' of the current state, and decide between moving the system or remaining in the current state depending on the resulting change to $E(s)$. The idea of the algorithm is to be able to accept a non-greedy step for $E(s)$ with a given probability, so that the algorithm does not get trapped in a local optimum and is able to converge ultimately to a global optimum.

We mark the probability of a transition between neighbouring states using an *acceptance probability function* $P(e, e', T)$ that depends on the energies $e = E(s)$ and $e' = E(s')$ of states s and s'. T represents the globally varying parameter, (typically called *temperature* in early applications of the algorithm). The acceptance probability function must tend to zero when T tends to zero. Therefore, for sufficiently small values of T the system will increasingly favour moves that reduce energy values. The temperature T thus plays a crucial role in controlling the evolution of the state s of the system depending on its sensitivity to variation of system energies. For large T, the evolution of s is sensitive to coarser energy variations, while it is sensitive to finer energy variations when T is small. Also, in order for the algorithm to converge, T has to decrease at each step following an *annealing schedule* which has to end with $T = 0$.

The Metropolis-Hastings algorithm (M-H)

We have already mentioned the Metropolis-Hastings algorithm as one way of performing the sampling operation from multi-dimensional distributions. The algorithm can be used to draw samples from a desired probability distribution $P(\bar{X})$, where $\bar{X}$ represents the parameter vector, as long as the function proportional to density is known. The resulting set of samples constitutes a Markov chain whose distribution, after sufficient length of time, matches that of the target. In most Bayesian sampling models, the density proportionality constant, also known as the *normalisation factor* needs to be known, thus this category

of algorithms offers a significant advantage by allowing a sample to be generated without needing to know the proportionality constant. The algorithm works as follows:

1. An arbitrary probability density $Q(x'|x_t)$ is chosen, also known as the *proposal density* or *jumping distribution*. This is used to sample a *proposal* value x', given a known sample value x_t. The proposal distribution must be a symmetric one, with the most commonly used being the Normal and Uniform distributions. The proposal density is used to explore the probability space using a random walk, thus generating a Markov chain.

2. Once the proposed new sample is drawn, we calculate the acceptance ratio $a = P(x')/P(x_t)$. If $a \geq 1$ we accept the proposed value and if $a < 1$ then we accept it with probability a. That is, we generate a uniformly distributed random number from $r = U(0, 1)$ and accept the sample if $r \leq a$. This is done in order to avoid the problem of the search algorithm being trapped in local minima.

3. The algorithm should settle down after a certain number of iterations, after which only small oscillations around the candidate value are performed.

However, the Metropolis-Hastings algorithm can suffer from a number of disadvantages to which we need to pay attention when analysing the results:

1. The samples suffer from the so called *correlation problem*. In this, nearby samples will be correlated with each other and will not correctly reflect the desired distribution. In order to avoid this, we can take every nth sample. We can also increase the jumping width (the variance of the proposal distribution) but we have to avoid the likelihood of sample rejection.

2. It takes some time for the samples to converge to the desired distribution, especially if the starting point is in the low density region. Thus, a *burn-in* period is necessary, where a number of initial samples are discarded.

For problems with a larger number of dimensions, it can be difficult to properly set the proposal density and define the jumping width. An alternative approach is to use Gibbs sampling, a variant of the original Metropolis-Hastings algorithm.

Gibbs Sampling

The Gibbs sampler generates an instance from the distribution of each variable in turn, conditional on the current values of the other variables. This sequence of samples constitutes a Markov chain converging to the desired joint distribution after a certain burn-in period, making it ideal for Bayesian applications as they are typically specified as a collection of conditional distributions.

The steps of a Gibbs sampling algorithm can be specified as follows:

- Suppose we want to generate samples from vector $\bar{X} = \{x_1, x_2, ..., x_n\}$ that follows a joint distribution $p(x_1, x_2, ..., x_n)$, and denote such a sample in i-th iteration as $\bar{X}^{(i)} = \{x_1^{(i)}, x_2^{(i)}, ...x_n^{(i)}\}$
- We assume (using e.g. known experimental data or system constraints) an initial value of the parameter vector $\bar{X}_{(0)}$
- For each sample $i = \{1, 2, ...k\}$ we sample each variable $x_j^{(i)}$ from the distribution conditional on the value of all other variables. The associated probability is thus: $p(x_j^{(i)}|x_1^{(i)}, ..., x_{j-1}^{(i)}, x_{j+1}^{(i)}, ..., x_n^{(i)})$

The resulting samples approximate to the desired joint distribution of all the variables. In addition, we can also approximate the marginal distribution of any subset of variables by simply examining the samples for that subset and the expected value of any variable can be approximated by averaging over all the samples.

The initial values for $\bar{X}_{(0)}$ can either be assumed from experimental data, set randomly (using a simple uniform distribution for example) or obtained from some other algorithm, such as Expectation-Maximisation (Dempster et al., 1977).

As with Metropolis-Hastings, it may be necessary to apply burn-in periods and thinning[1] to ensure good convergence and avoid correlation between the nearby samples.

[1] Thinning represents selecting every nth sample, where n is taken from the empirical observations of the inverse process. This allows ignoring small oscillations in obtained acceptance results.

7.3 Bayesian credibility intervals

In Bayesian inference, credibility intervals are the analogue of confidence intervals in frequentist probability[2]. However, unlike the confidence interval, which is defined as an estimated range of values which is likely to include an unknown population parameter, credibility intervals tell us how likely it is for the parameter to reside within a given contiguous interval $[a, b]$. The tightness of the interval (in both definitions) is denoted by value α, such that (1-α)% of the probability mass is within the interval. One usual measure is a 95% credibility interval, describing the 95% chance that the real value is within the given boundaries.

There are two main conventions for determining interval boundaries:

- We choose the shortest possible interval which encloses (1-α)% of the posterior mass. This is called a "highest posterior density" or HPD credibility interval.
- We choose the interval boundaries so that an equal amount of the probability mass lies on either side of the interval. This is called a "symmetric" credibility interval.

7.4 IMC Algorithm Description

In combination with direct MC methods, our IMC algorithm depends on several key properties. First, direct MC model inputs are structured into vectors of individual parameter values:

$$\bar{X} = \{X_1, X_2, ..., X_n\} \tag{7.1}$$

Where each X_i corresponds to one of the fundamental Direct MC model parameters, (e.g. λ factor describing polymer lifetime or various diffusion probabilities).

Starting from these parameters there exists a measurable model output Y for the given parameter vector $\bar{X}$, (the simulated release curve) and there exists a corresponding

[2]Frequentist probability considers a definition of a random event as an absolute in terms of occurrences during a controlled experiment. Bayesian methods on the other hand deal with probability of an event relative to another that might have previously occurred.

experimental measure to which the simulated output can be compared, within some margin of error (usually experimental batch data with standard measurement error):

$$Y = F(\bar{X}) + \Theta(\bar{X}) \tag{7.2}$$

where $F(\bar{X})$ denotes the *deterministic* part of the functional dependency between the system input and output and $\Theta(\bar{X})$ is the *stochastic* part resulting from uncertainties and noise in the physical model, (Barat, 2006)

A set of *constraints* is placed on the possible values of the input vectors to limit the parameter space. These constraints are made based on understanding of the physical system. A constraint can also, for example, discard input vectors which produce experimental data outside of the acceptable error limits without performing the minimisation at all. In our DDS case, the constraints come from experimental data that it is possible to measure: most importantly this includes the major diffusion coefficient of the drug, coating thicknesses and properties of constituent polymers.

An initial configuration of the model $\bar{X}_{init}$ (selection of the initial parameter vector) is made. This selection can be random, but it is useful to start with a reasonable estimate in order to avoid long algorithm convergence rates or the *isolated space* problem. In principle, IMC modelling is independent of the starting configuration, so this requirement is not a strict one. However, in practice, the convergence time can be drastically shortened except for the simplest of cases. Also, we make here an important assumption that any point in our parameter space is reachable from any other point by repeatedly sampling from *posterior* distributions of individual parameters.

For each parameter in the input vector, there exists a means of generating a new sample based on the value of the previous one, thus creating a Markov chain. This can be achieved by selecting an initial *proposal density* or *jumping distribution*, $Q((\breve{X})|X_t)$, usually a Gaussian, based on the value of the initial parameter $X_t \in \bar{X}_{init}$, and then modifying that density if the new parameter value gives better agreement with the experimental data than was previously available. The adjustment is achieved by use of an appropriate *sampler*, e.g. with the Gibbs, (Geman and Geman, 1984) or a Particle filter method, (Doucet et al., 2001).

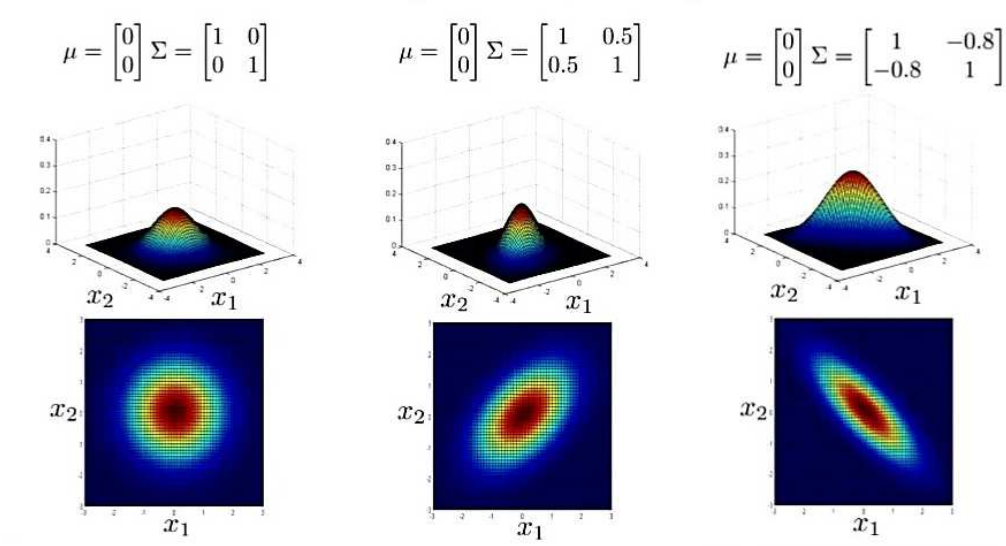


Figure 7.2: An illustration of bi-variate MVN for different values of μ and $\mathbf{\Sigma}$. Adapted from (Ng, 2013).

In our case, since the values of the model parameters have an expected value that can be assumed to lie in a certain narrow range, using a Gaussian as a prior might present itself as a natural choice. In context of the full parameter vector sampling however, this translates into sampling from a multi-variate distribution, so we will use the Multi-Variate Normal (MVN), and its truncated form as the source distribution.

The MVN distribution represents a generalisation of the Gaussian distribution to higher dimensions and can be used to represent k-valued random vectors whose every linear combination has a univariate normal distribution. It is often used to describe any set of real-valued random variables centered around a mean value (Figure (7.2)). In our case this would be any likely set of parameters which describes a system with the same or very similar release behaviour compared to the experimental one. The spread of these "parameter points" describes the stability of the system solution and is represented through the deviation of the MVN. The MVN can be written using the following notation:

$$\mathbf{X} \sim \mathcal{N}_k(\mu, \mathbf{\Sigma}) \tag{7.3}$$

where $\mathbf{X}$ is the k-dimensional parameter vector taking values from R^n, μ is the k-

dimensional mean vector:

$$\mu = [E[X_1], E[X_2], ..., E[X_k]] \tag{7.4}$$

and $\mathbf{\Sigma}$ is a $k \times k$ positive, semi-definite covariance matrix

$$\mathbf{\Sigma} = [Cov[X_i, X_j]], i = 1, 2, ..., k; j = 1, 2, ..., k \tag{7.5}$$

It can be said that a variable $X \in R^n$ is multivariate normal if any linear combination of X is univariate normal, i.e. $\mathbf{aX} = \Sigma_{i=1}^{n} a_i X_i, \forall \mathbf{a} \in R_n$ is normally distributed.

We make an assumption here that the set of parameters in our model is independent (and indeed, this is a consequence of the way in which the direct model has been constructed, since each one of the constituent parameters describes an independent property of the model - a single physico-chemical process), and we can generally assume that each of these parameters follows a normal distribution centered on the theoretically precise value and with a variation that represents the natural variation effects of the polymeric structure.

If we assume then that possible values for each of the model parameters X_i are normally distributed with $N(\mu, \sigma^2)$, the combined parameter vector of our model $\mathbf{X}$ is a multi-variate normal distribution with $\mathcal{N}(\mu, \mathbf{\Sigma})$, where $\mu = (\mu_1, \mu_2, ..., \mu_n)$ and the covariance matrix is given by:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & & 0 \\ & \sigma_2^2 & \\ & & \ddots \\ 0 & & \sigma_n^2 \end{pmatrix} \tag{7.6}$$

where σ_i^2 represents the variance of individual distribution components.

In addition, as each of the model parameters is bounded by some limit (a, b), representing either a physical constraint, (for example the polymer degradation lifetime always larger than zero), or a mathematical constraint, (probabilities always drawn from $[0, 1]$ interval), parameter sets are not from the MVN distribution itself, but rather from its truncated, uni- and multivariate, version. Although more efficient methods for sampling from truncated

distributions including specialised MCMC and Gibbs sampling have been proposed in the literature (Kotecha and Djuric, 1999) we have chosen to use a simpler *rejection sampling* method as the number of variable parameters is not too large. However, if better performance is desired or the parameter set is large, specialised sampling methods are an option.

Finally, as we are using Gaussians to define priors of individual parameters, we need to know their mean and standard deviations. As those are unknown (and essentially, explored using the sampling algorithm itself), we assume an adaptive approach.

We start from a reasonable estimate of μ and σ for every parameter. As the sampling algorithm should converge to the optimal values, the estimate can essentially be anything with good coverage of the valid parameter range $[a, b]$, but choosing estimates for μ based on expected subranges of $[a, b]$ would yield faster convergence. Based on the mean of the prior distribution, the measure of difference to the experimental results and the spread of the accepted parameter values, we can calculate the μ and σ of the posterior distribution. Based on these, we can define the transition of the inverse algorithm from iteration i to iteration $i + 1$:

$$\mathcal{N}^{(k)}\left(\mu_i, \mathbf{\Sigma}_i\right) \rightarrow \mathbf{X}_i^{(k)} \rightarrow F\left(\mathbf{X}_i^{(k)}\right) \rightarrow \chi_i^2 \rightarrow \mathcal{A}\left(\chi_i^2\right) \rightarrow \mathcal{N}^{(k)}\left(\mu_{i+1}, \mathbf{\Sigma}_{i+1}\right) \tag{7.7}$$

Where $\mathcal{A}(\chi_i^2)$ is an *acceptance function* the result of which is used to accept or reject a particular input vector and χ^2 is the measure of fitness between the simulated and experimental results:

$$\chi^2 = \frac{\Sigma\left(y_{obs} - F(\mathbf{X})\right)^2}{\sigma^2} \tag{7.8}$$

With all of the above defined, we can run the IMC algorithm in several steps:

1. Define the initial input parameter vector - $\bar{X}_{init}$,
2. Pass the vector through a Gibbs sampler to obtain the next proposal sample,
3. Run the direct simulation with the proposed sample and quantify the simulation output based on the indicator of χ^2 value for the agreement between simulated and experimental values at key measurement points,

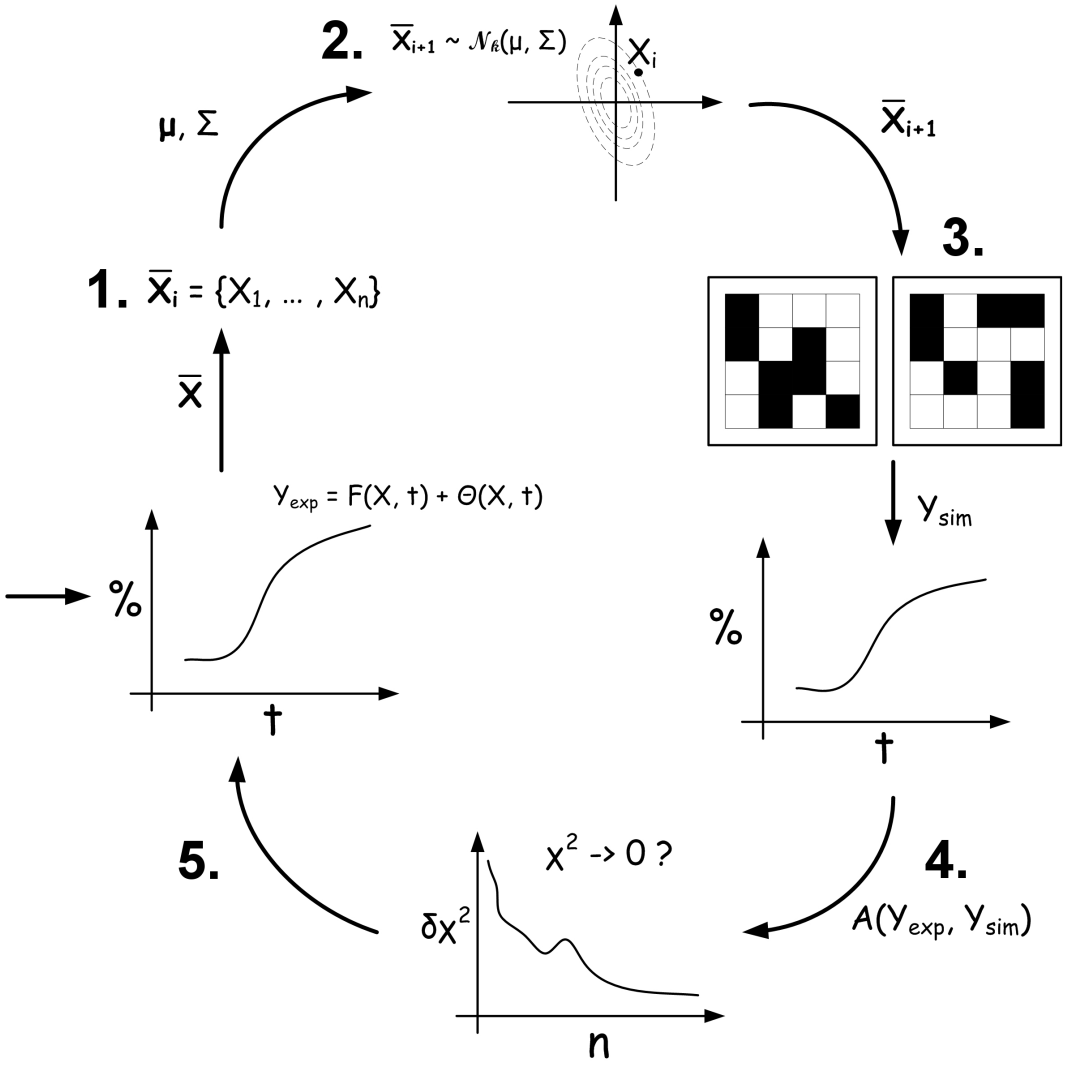


Figure 7.3: Graphical representation of the main stages of Inverse Monte Carlo algorithm, used for deduction of unknown parameter sets. Arrows indicate the flow of information between different steps.

4. Use an *acceptance function* $A(\mathbf{X}, \chi^2)$ that will set the next sample as the current sample and modify the mean and standard deviation of the input parameters if the χ^2 is the best one obtained so far,

5. Repeat the procedure until the set of acceptable input values converges to a narrow set that satisfies all the constraints. The resulting mean and standard deviation will then describe the possible range of interest for the given parameter.

The main stages of the algorithm are illustrated in Figure (7.3).

7.4.1 Algorithm optimisations

The Gibbs sampler usually requires from several hundred to several thousand iterations in order to converge to the desired distribution. Also, a certain number of initial samples is usually discarded as the *burn in* period. This number of forward simulations can place a significant performance penalty that makes minimisation of running time of each iteration of primary importance if convergence time is to be reduced.

As the runtime of forward simulation is a constant, there are two possible approaches to optimisation of the total convergence time. One approach is to increase the number of forward simulations run in each iteration of the algorithm and the other is to reduce the sample space. We focus on the former optimisation and reduction of the sample space using a simplified approach based on sensitivity of the output to parameter variations. We also outline the possible algorithm improvements for sample space reduction using the neighbourhood algorithm in Section 7.6.

Increasing the number of forward iterations is achieved by parallelising the sampling and forward simulation steps of the IMC algorithm. Instead of taking one sample at a time, we draw n distinct samples from the same prior and run a simulation for each at the same time. At the end, we can accept the best sample as the next one and modify the mean and standard deviation of the priors as already described.

Further optimisation can be achieved by examining the sensitivity of the release curve for different simulation parameters (e.g. using a standard sensitivity analysis approach). As it is common for small variations in parameter values to produce nearly identical results, then if absolute precision of those values is of secondary importance we can reduce the sample space by *discretising* it to some acceptable precision before sampling occurs. This will drastically reduce the sample size and allow us to perform an exhaustive search in realistic time. The discretisation and loss of precision could result in somewhat larger standard deviation and thus less precise mean estimates of the parameter values, but these are still useful for the purposes of modelling.

7.5 Simulations and Results

In order to validate the approach presented above in a practical experimental setting, we ran the inverse simulation of our double coated model described in Chapter 4 with comparison function operating against a known *in vitro* sample. The sample represented averaged data from 6 batches of double coated beads. The coating polymers used were *Surelease*® mixed with Pectin - X% of total weight gain - serving as a release control mechanism, and *Opadry*®, (Colorcon Inc, 2009), - 10% of total weight gain - used to prevent polymer blocking reaction between the gelatinous core and the top coat. A bead contained 10.8% of mass drug loading of the active substance - Cyclosporine A. An average diameter measured in the batch was 1.65mm. The *in vitro* dissolution environment consisted of two different solutions: 0.1 HCl concentration for the first 2 hours, and 0.75 SDS enzyme concentration from 2 to 24 hours.

We varied four simulation parameters:

- The gelatin λ factor, indicating the mean lifetime value of core cells,
- The probability of water diffusion into polymeric chains of Surelease/Pectin outer coating (P_W),
- The diffusion of water into core layer (P_{WG}),
- The diffusivity of Cyclosporine A through the outer layer (P_P).

With the exception of the λ factor, which is known to lie within certain limits, the rest are probabilities whose values are derived from the diffusion coefficients of various polymers (according to rules similar to Equation (4.20)), and, as indicated in the Chapter 4, Section 4.4, can be at best guessed using rather general data available from the literature. Therefore, they make good candidates for inverse modelling. The λ factor was selected in order to understand how inverse search applies to parameters with known limits.

The set of parameters that were varied, along with their initial mean, deviation, boundaries and sampling precision used, is given together in Table 7.1.

The inverse simulations were performed under a set of known constraints used for other parameter values, either specified as part of the experiment or derived from known

Parameter	**Init. μ**	**Init. σ**	**Left bound**	**Right bound**	**Precision range**
λ factor	100	50	50	200	10^0
P_W	0.05	0.02	0.001	0.1	10^{-3} - 10^{-2}
P_{WG}	0.05	0.02	0.001	0.1	10^{-3} - 10^{-2}
P_P	0.5	0.2	0.01	1	10^{-2} - 10^{-1}

Table 7.1: The inverse model parameter space. The initial mean, standard deviation, bounds and precision of each investigated parameter are given. The truncated normal distribution, defined by {μ, σ, *left bound*, *right bound*}, was used as the initial sampling prior for each parameter.

literature. The list of constraints used, along with their values is given in Table 7.2. All simulations were run with a cell matrix precision of 0.03 mm per cell in order to enable faster forward simulation time. This has to be taken into account when analysing the results, as according to Equation (4.20) smaller precision will lead to generally larger probability values (as diffusion coefficients values are inversely proportional to the dimension of individual cells). Deriving probabilities from values obtained at coarser precision for these of finer precision simulations is straightforward, as the resulting probability along a direction of drug movement can be obtained by multiplying the individual cell probabilities together. Some accumulation of error is possible, as indicated in a similar case, (Barat, 2006). Of course, coarser precisions do not allow for simulations involving small variations of drug coatings, (something that would be a preferred target of investigation, originally flagged for the direct model). However, since those parameters are used as constraints, then, as long as the required thickness can be modelled at the given precision, the ability to vary the drug coating parameter so finely is not needed.

Volumetric loading was derived from mass loading using a process, similar to that described in Chapter 4, Section 4.4, and was found to be $\approx 20\%$. Also in line with the double coated model data described, the diffusion coefficient of CyA was kept at a standard value of $10^{-6} cm^2/s$. The mean Opadry lifetime was assigned to be 30 minutes, based on experimental *in vitro* observations. Swelling probability was chosen so that the swelling of gelatine core is one of the more dominant processes affecting the release. Finally, the blocking and clustering effects of gelatine, observed to significantly impact release rates toward the latter end of the simulation have been removed, as the dissolving Opadry

Parameter	Value
cell size	0.03 mm
volumetric loading	20%
mass loading	10.8%
D_f	$10^{-6} cm^2/s$
Opadry® lifetime	30 min
P_S	0.5
P_C	0.00
P_D	1.00
surelease/pectin coat	0.06 mm
Opadry coat	0.06 mm

Table 7.2: Physical constraints for the inverse simulations. The coating thicknesses, Opadry lifetime and volumetric and mass loadings were derived from existing batch data. The swelling probability was set to enable controlled behaviour, influenced by both swelling and erosion, as observed in the Direct MC experiments in Section 6.3. The probability of gelatin blocking/clustering due to the low solubility inside the core is zero as dissolved Opadry® significantly enhances gelatin solubility. Finally, the diffusion was set to be the fastest process.

coat causes better and more complete degradation of gelatin within the bead. Finally, we assumed the CyA diffusion to be the fastest process in the bead, and therefore used it as a reference value for measuring other relative diffusivities using the equations defined in Chapter 4, Sections 4.3, 4.4.

We first examined the convergence properties of the Gibbs sampler used. Figure (7.4) shows the change of χ^2 value over approximately 1400 iterations, while Figures (7.5) and (7.6) show, respectively, the comparison of experimental and simulated curves during different convergence periods and χ^2 convergence patters for different input parameter ranges. Regression analysis using a locally weighted smoothing line (LOESS method) was plotted against the data to show the convergence behaviour. Each point represents the χ^2 value obtained for each forward simulation, with 10 sequential points representing each parallel batch of iterations. The optimal values selected in each batch are visible as the lowest points of the graph showing the optimal χ^2 change. As expected, the algorithm takes some time to converge. The burn-in period took around 800 iterations, after which the sampling stabilised around values of χ^2 between 70 and 80. Noting the vertical distribution of points during the inverse simulation we can see how the sampling distributions are getting narrower producing sample parameter vectors with decreasing standard deviations. At around the 400th iteration there was a brief example of the sampler entering the local

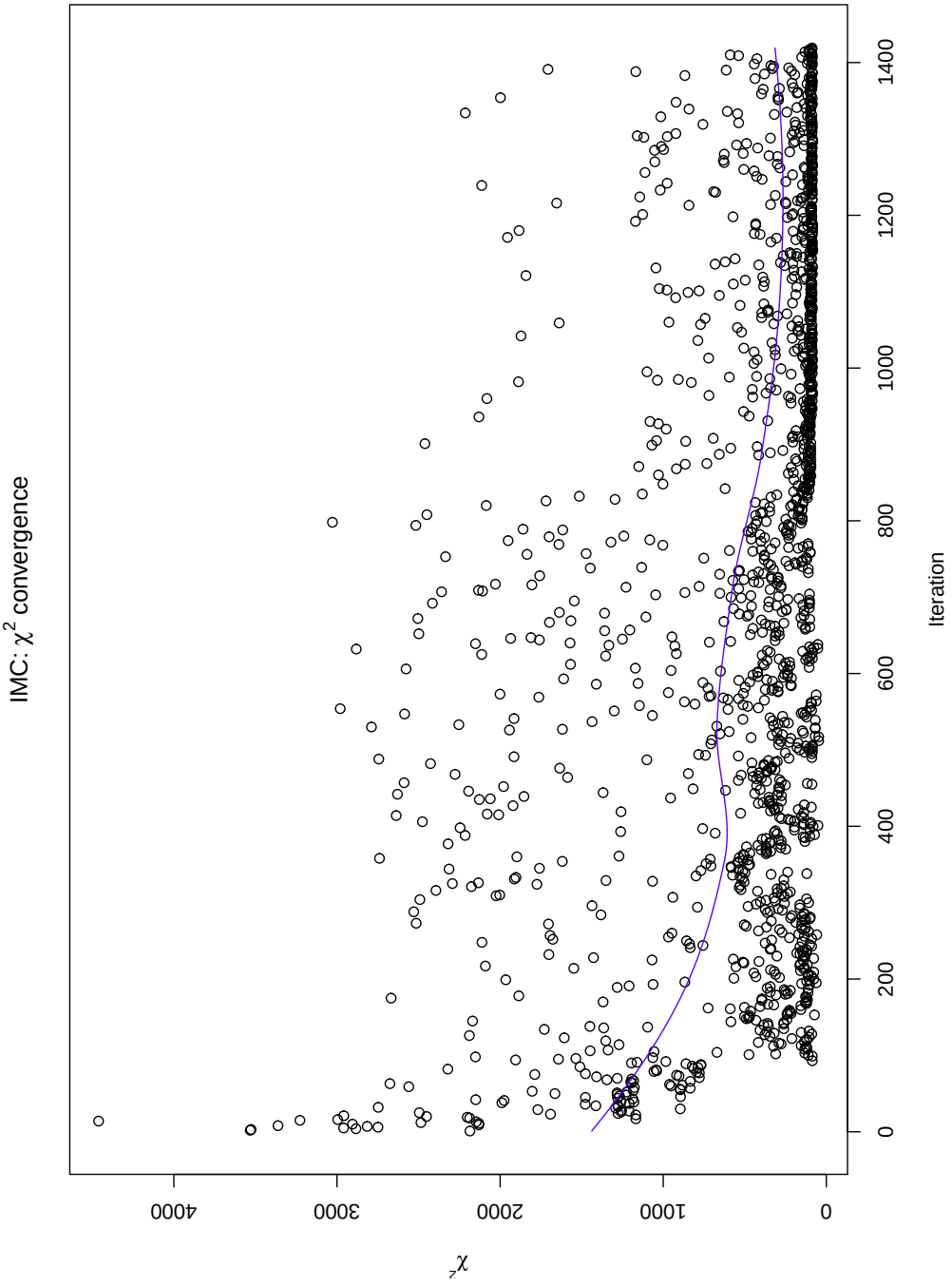


Figure 7.4: Changes of chi-square value indicating the correspondence between simulated and experimental values in each iteration. The convergence trend is indicated using a LOESS* smoothing line (blue). Selected chi-square values - the best ones in a given run - are the lowest points of the graph. The first dip in the smoothing line indicates a point where the Gibbs sampler exited from a local minimum.

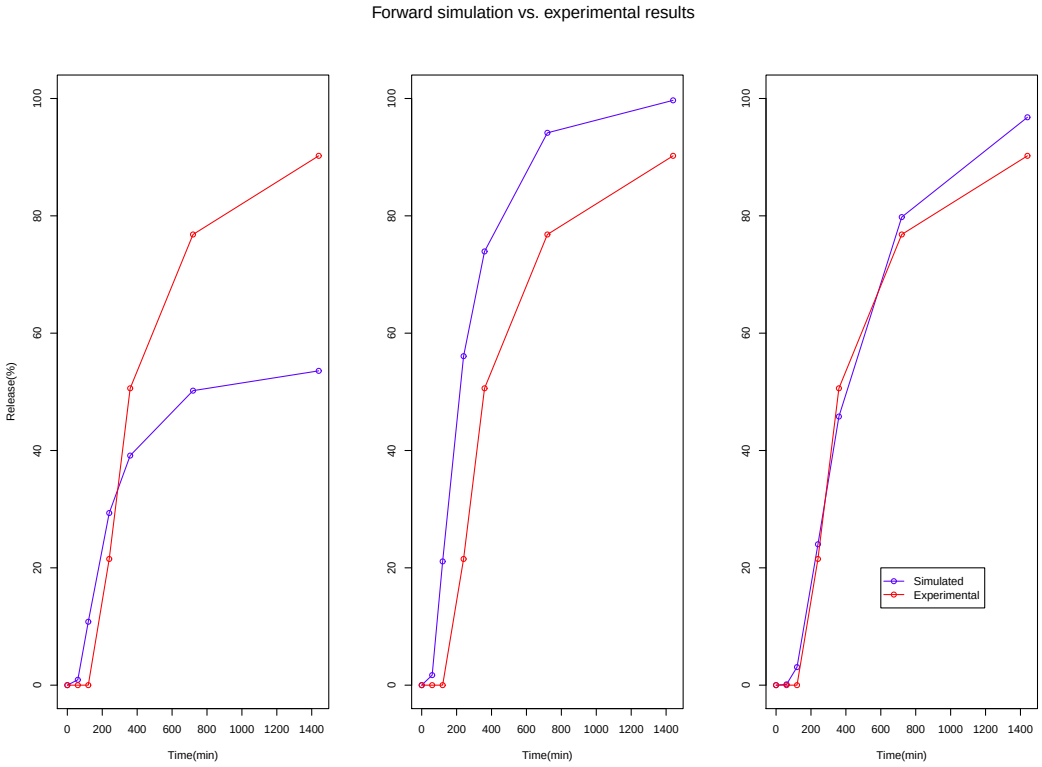


Figure 7.5: Comparison of selected samples of forward simulations during different convergence periods. (Left) Initial configuration; (Centre) After burn in period (Right) After stabilisation.

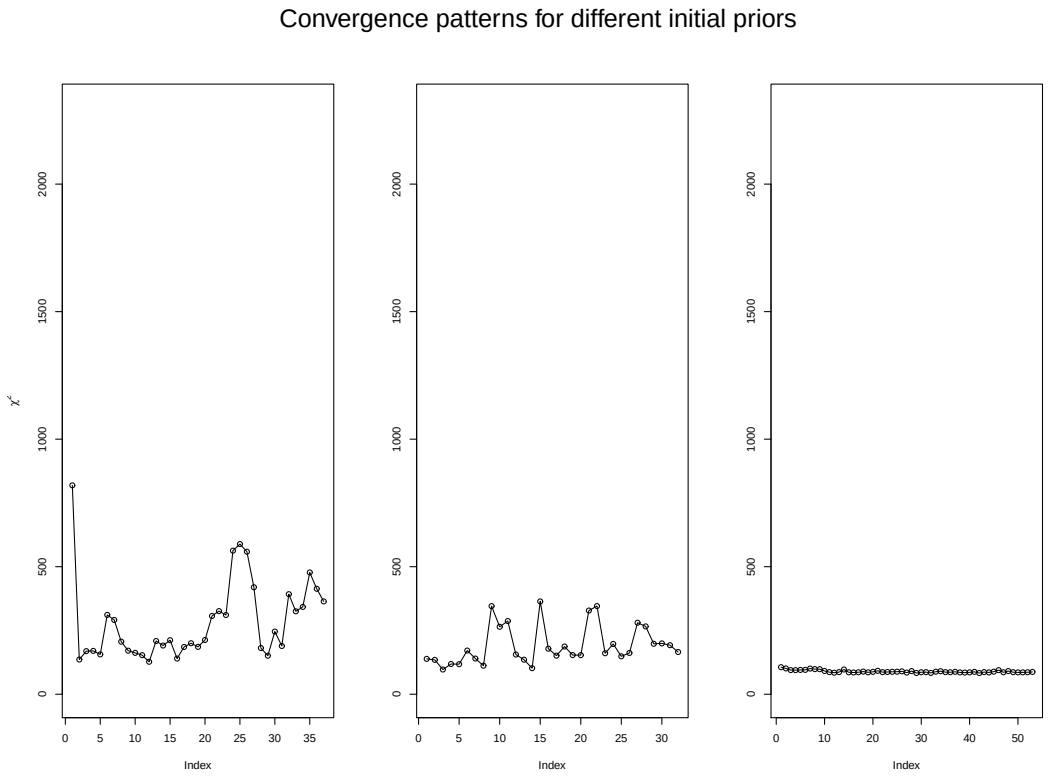


Figure 7.6: Convergence patterns for different input parameter ranges.

minimum, and exiting it to continue converging. This shows the recovery advantage of using the acceptance functions that take locally minimal χ^2 values. Although convergence to zero is ideal, we have to take into account the quality of the proposed constraints, which will never completely reflect the actual values in the experiment, i.e. the variance never reaches zero. With respect to data match for the simulation, using the converged vector of parameter values against the experimental data, we observe this to be very good, (Figure (7.5)). In the same figure, we demonstrate the same trend on selected samples of forward simulations. The first comparison represents the release curve, which matches poorly with experimental data from which the sampling algorithm started. At the end of the burn-in period, slightly better agreement is achieved. Finally, after convergence has stabilised the resulting release curve fluctuates only slightly around the optimal value found.

In order to understand how parameter convergence to best fit values with observed experimental data depended on the initial priors used for their sampling, we analysed the parameter variation rate for priors of different sizes. As the sampling of parameter values

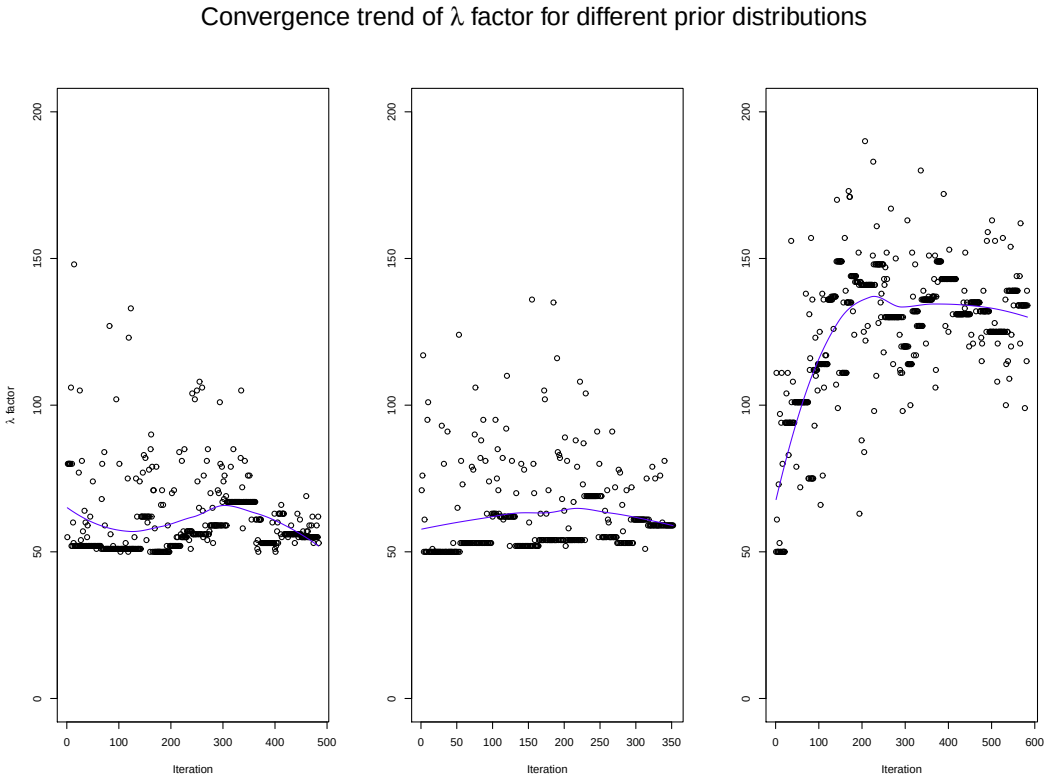


Figure 7.7: Comparison of the variation of λ sample (dependent on choice of priors) with different bounds and parameter ranges. (left) Medium-sized initial prior using truncated normal $\{\mu = 80,\ \sigma = 50,\ a = 50,\ b = 150\}$ (centre) Smaller-sized initial prior using truncated normal $\{\mu = 50,\ \sigma = 30,\ a = 50,\ b = 140\}$ (right) Larger initial prior using truncated normal$\{\mu = 100,\ \sigma = 50,\ a = 50,\ b = 200\}$.

at iteration k is dependent on χ^2 convergence at iteration $k-1$, which itself is influenced by the entire set of parameter values changed up to that iteration, we cannot base the analysis of parameter value correctness solely on comparison with resulting χ^2 values, but must take into account also the value variations within the sample parameter space. Thus, small variations of the sampled parameter value can indicate one of two things: (1) the parameter has converged around the optimal value or (2) the initial prior is too restricted and the sampling algorithm cannot escape the local minimum. To understand which of these apply, we need to look at the corresponding convergence of the acceptance ratio. Case (2) is illustrated in Figures (7.7)(a) and (7.7)(b). If we look at the respective χ^2 square convergence patterns, (Figure (7.7)) overall, we see that convergence is clearly established for the parameter range in (c) but is lacking for (a) and (b). This, along with the range of samples explored in (c) gives a better indication that the parameter value found is suitable.

Finally, we examined *credibility intervals* of the resulting MCMC chain for the λ factor values. Figure (7.7)(c) shows the resulting upper and lower bounds of the 95% credibility interval in each sampling iteration, displayed as highest precision density (HPD) intervals. The HPD intervals give the smallest credibility range and, although they are not invariant under changes of the parameter value, we are usually only interested in the intervals of the last few samples, (which indicate that the optimal parameter value has been found).

The next parameter examined was *water permeability* through the outer membrane. Again, we used three intervals of decreasing size. Figure (7.9) shows the convergence patterns for three different intervals. In contrast to the convergence of the λ factor, it is much more difficult to establish a reasonable convergence pattern in this case. There is no clear grouping of the results around a common cluster of values and the changes in the mean are frequent. The standard deviation remains constant without decreasing further. However, if we consider the results of the sensitivity analysis in Chapter 6, this pattern of results can be inferred from the fact that the model output is not very sensitive to changes in coating water permeability. Thus, the Gibbs sampler would often accept changes in $P_{[W]}$ as those would result in χ^2 values not far away from the previous ones.

Another parameter that shows good convergence properties is the *diffusivity* of CyA in the outer coating (P_P). Figure (7.10) shows the convergence for the three size intervals

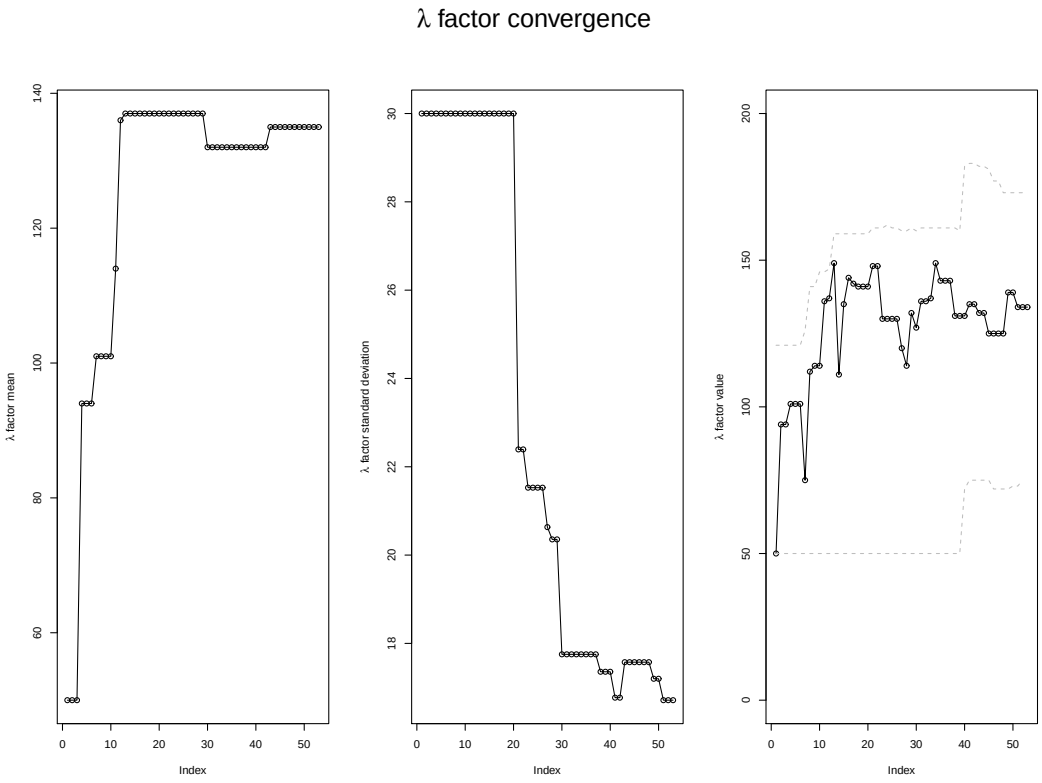


Figure 7.8: Convergence of mean and standard deviation of prior distribution used by the Gibbs sampler, with 95% credibility intervals for λ values for case (c) of Figure 7.7. Reduced standard deviation constrains possible optimal values of the parameter. High probability density (HPD) credibility intervals were calculated for those MCMC samples, generated using the resulting priors.

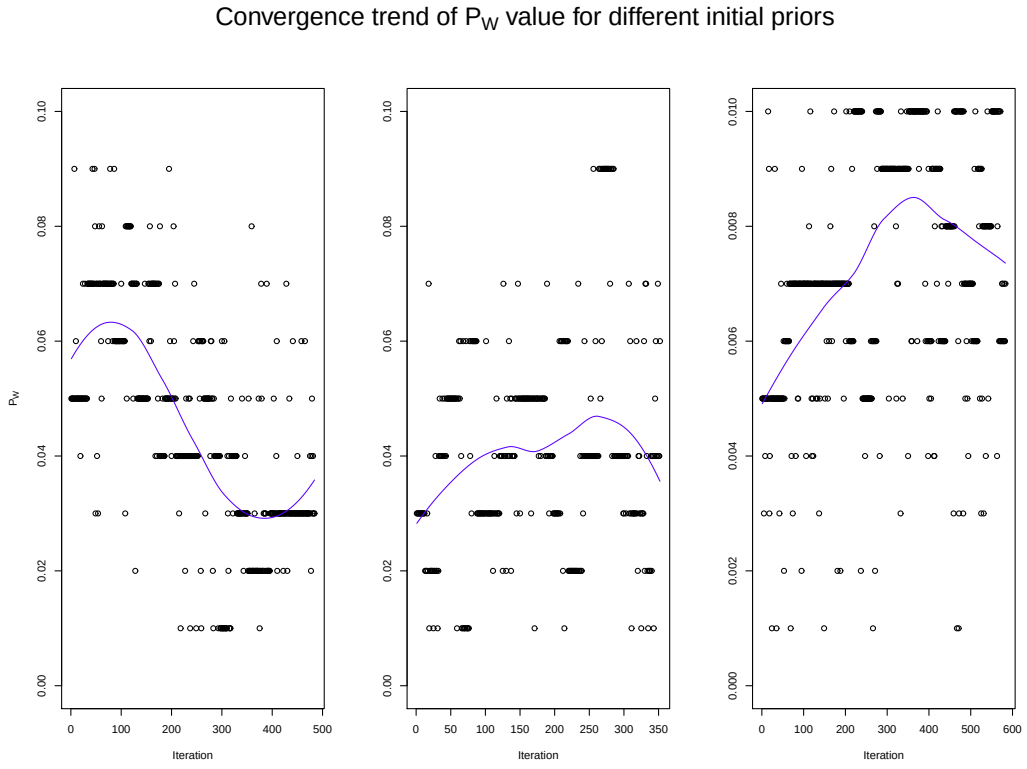


Figure 7.9: Convergence pattern of P_W values for three different initial priors. (left) Large-sized initial prior using truncated normal $\{\mu = 0.05,\ \sigma = 0.02,\ a = 0.1,\ b = 0.9\}$ (centre) Medium-sized initial prior using truncated normal $\{\mu = 0.03,\ \sigma = 0.02,\ a = 0.01,\ b = 0.06\}$ (right) Small-sized initial prior using truncated normal $\{\mu = 0.005,\ \sigma = 0.002,\ a = 0.001,\ b = 0.01\}$ Convergence pattern is not present for any of the used priors, and parameter variation cannot be correlated with χ^2 convergence trends.

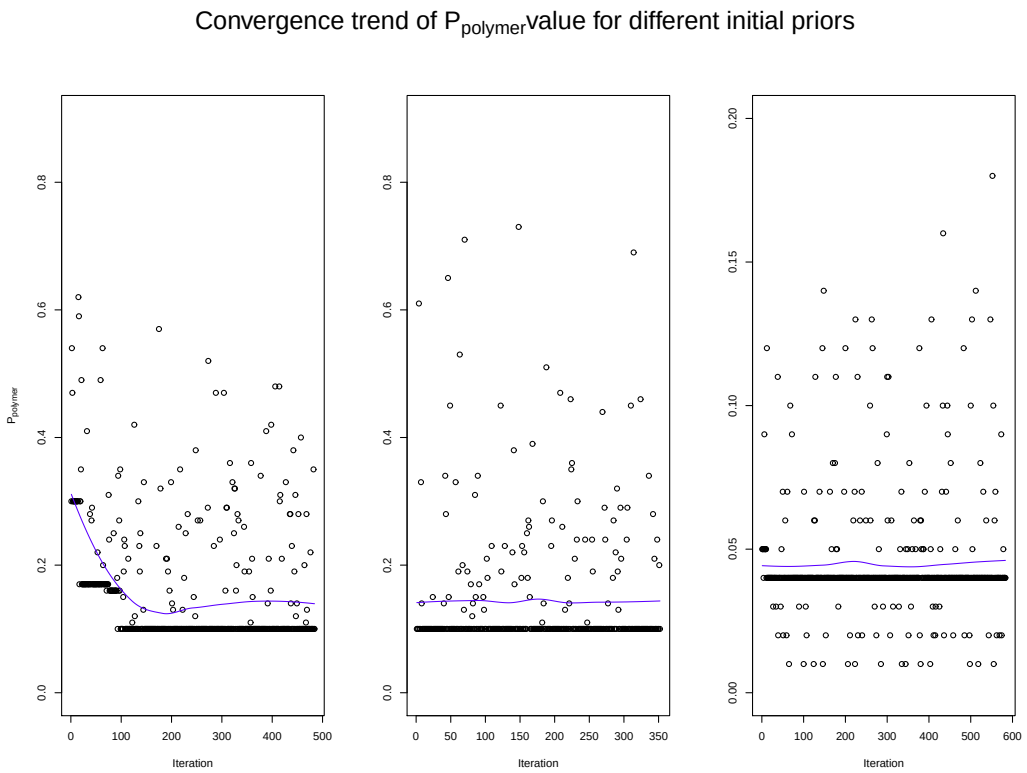


Figure 7.10: Convergence pattern of P_P values for three different initial priors. (a) Large-sized initial prior using truncated normal $\{\mu = 0.3, \sigma = 0.3, a = 0.1, b = 0.9\}$ (centre) Medium-sized initial prior using truncated normal $\{\mu = 0.1, \sigma = 0.1, a = 0.1, b = 0.8\}$ (right) Small-sized initial prior using truncated normal $\{\mu = 0.05, \sigma = 0.03, a = 0, b = 0.2\}$ Convergence is improved with each sampling prior and corresponds to that of χ^2.

mentioned before. The first two show convergence all the way to the lower boundary, indicating that the lower sampling boundary can be lowered, and the precision of sampling increased. Using a smaller sampling interval from [0, 2] shows good convergence around the 0.04 mark and a good match with the respective χ^2 convergence patterns.

Finally, the convergence intervals for *diffusivity of water into the gelatin core* (P_{WG}) are shown in Figure (7.11). Again, good convergence is established with narrowing intervals, and in line with the overall convergence of the release curve.

Overall, we can say that we note that relatively good candidate values for three of the four parameters investigated were found. In the case of P_W we can see that the relatively low model sensitivity to changes of the parameter does not give us sufficient confidence in its values.

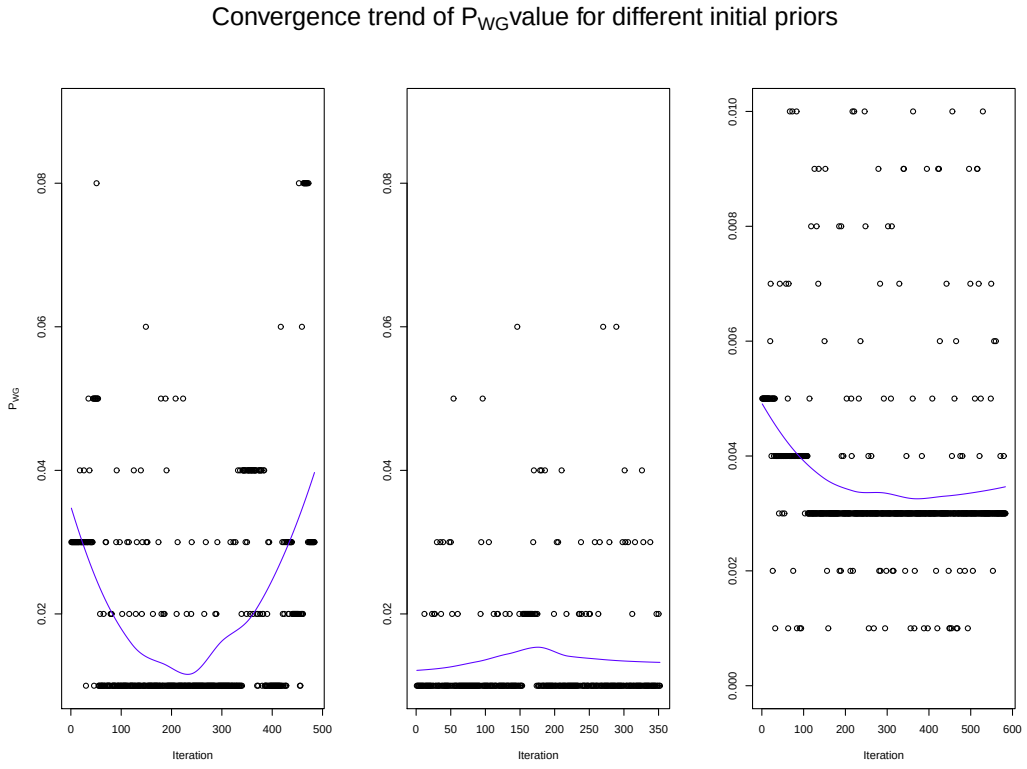


Figure 7.11: Convergence pattern of P_{WG} values for three different initial priors. (left) Large-sized initial prior using truncated normal $\{\mu = 0.05, \sigma = 0.02, a = 0.01, b = 0.09\}$ (centre) Medium-sized initial prior using truncated normal $\{\mu = 0.02, \sigma = 0.01, a = 0.01, b = 0.06\}$ (right) Small-sized initial prior using truncated normal $\{\mu = 0.005, \sigma = 0.002, a = 0.001, b = 0.1\}$ Convergence is improved with each sampling prior and corresponds to that of χ^2.

7.6 Extensions: Optimisation of parameter space search using Neighbourhood algorithm

7.6.1 The Neighbourhood algorithm (NA)

The IMC model presented so far depends on running the forward model for each sampling of the parameter vector. If the forward model can be evaluated using an analytical expression this does not present a problem. However, in the majority of models which solve the forward problem stochastically by iterating many times over properties of a large discrete space, (such as in Cellular Automata simulations) this results in a significant runtime overhead, which impairs the effectiveness of the inverse approach.

Sambridge (1999) developed an inversion algorithm for multidimensional parameter spaces that saves significant time by utilising the existing knowledge on *data misfits* from previously run simulations. The search algorithm uses what is known as Voronoi cells to derive the search in parameter space, (Voronoi, 1908). Given a set of points (*"control points"*) in a parameter space, Voronoi cells represent nearest neighbour regions of the space, each containing one of the control points, with all other points within a region defined as having a smaller distance norm to one control point than to all the others.

Formally, this can be represented by the following. Let $P = \{X_1, X_2, ...X_{n_p}\}$ mark a set of points in d-dimensional parameter space of interest, where $2 \leq n_p \leq \infty$ and $X_i \neq X_j$ for $i \neq j$. The Voronoi cell around the control point X_i is given by:

$$V(X_i) = \{X|||X - X_i|| \leq ||X - X_j|| for j \neq i, (i, j = 1, 2, ..., n_p)\} \tag{7.9}$$

The Voronoi cells created in this way are used to approximate (i.e. interpolate) the misfit function[3] across the entire parameter space. The algorithm takes the following simplified form:

- Create the approximation of the misfit surface from the n_p previous models for which the forward simulation has been run,

[3] *Misfit function* can be considered as analogue of the "goodness of fit" function presented earlier in the work. Larger values would indicate poorer match with the experimental results.

- Utilise the resulting approximation, (instead of running forward modelling again), together with the chosen search algorithm, (simulated annealing, M-H or Gibbs, for example), to generate the next n_s samples,
- Combine the real and approximate results and repeat.

The misfit function is set to a constant value within each of the Voronoi cells (NA-surface). Therefore in order to find the value of the misfit function at any given point, we only need to look up the Voronoi cell to which it belongs.

Integration of NA with M-H and Gibbs samplers

Let the sampling density function be $S(X)$. Then, as noted earlier, the M-H algorithm performs a move from point A to point B along one of the d parameter axes, according to the probability distribution $q(X_B|X_A)$, and the move is accepted if

$$r < min\left[1, \frac{S(X_B)q(X_B|X_A)}{S(X_A)q(X_A|X_B)}\right] \tag{7.10}$$

As the q is usually chosen to be symmetric (so $q(x|y) = q(y|x)$), we can simplify the acceptance criterion to

$$r < min\left[1, \frac{S(X_B)}{S(X_A)}\right] \tag{7.11}$$

The acceptance/rejection step requires the misfit function to be evaluated at point B, requiring one solution to the forward problem along each of the parameter axes. However, a large number of parameter permutations will be rejected leading to the large "wasted" computational cost.

The M-H method can be integrated directly into step (2) of the described "idealised" algorithm. At each stage, the M-H method is used to draw n_S statistically independent samples from the current NA-misfit surface, thus using it as an approximation for the actual misfit function, and to avoid solving the forward problem.

If we denote the computation time of forward modelling by T_{FM}, that of the neighbourhood search as T_{NN}, and the number of iterations as I_T, we can represent the cost of

generating samples using the NA-surfaces as:

$$T_{NA} = T_{FM} + T_{NN} I_T d \tag{7.12}$$

On the other hand, the cost of using forward modelling each time is:

$$T_{misfit} = T_{FM} I_T d \tag{7.13}$$

Leading to a cost reduction ratio of

$$\frac{T_{NA}}{T_{misfit}} = \frac{1}{I_T d} + \frac{T_{NN}}{T_{FM}} \tag{7.14}$$

As in almost all relevant cases we expect $I_T d \gg 1$ and $T_{FM} \gg T_{NN}$ the cost saving is significant, and we can generate many more samples from the desired joint distribution for the same computational cost.

The usefulness of this approach depends on the nature of the forward problem (its cost) and the sample size of n_S. Larger sample sizes lead to existing NA-surfaces being exploited for longer. On the other hand, as n_S is decreased the NA-surfaces are updated more rapidly requiring more forward solutions to be computed.

In the case of applying NA algorithm as a step in the Gibbs sampler, the cost reduction is even greater:

$$\frac{T_{NA}}{T_{misfit}} = \frac{1}{n_a d} + \frac{T_{NN}}{T_{FM}} \tag{7.15}$$

with n_a corresponding to the number of parameter samples generated along one of the d axes.

7.7 Summary and Conclusion

In this chapter we have described the systematic development and examples of inverse modelling algorithms, and we have designed and evaluated a possible approach to elucidating unknown parameter values, based on analysing the results of hundreds of forward simulations. We have shown that using the Gibbs sampler and Metropolis-Hastings algorithm as the

basis for inverse methods gives good parameter estimates and can converge in reasonable time, (which may yet be improved further). This makes these methods very useful in the context of drug modelling since in practice, the actual value of many parameters, (usually derived from theory) are subject to considerable experimental error. We have established good convergence for several of the investigated parameters, giving good credibility intervals on true parameter value.

Chapter 8

Concluding discussion and future work

8.1 Summary and Conclusions

Although modelling of DDS has been extensively studied, there is growing need for improved models, which are able to accommodate more complex formulations as well as help expand theoretical knowledge in the real world applications. The results from this research have so far successfully facilitated comparison of simulated and experimental release curves, allowing us to determine key parameters and their values for the novel drug formulation, introduced by Sigmoid Pharma, and to reproduce qualitative release behaviour patterns observed in experimental systems *in vitro*.

Initial models predicted the essential biphasic, sigmoidal nature of the release curves for coated beads, and provided good insights into mechanisms affecting controlled drug release. Model organisation allows independent investigation not only of superposition of the phenomena of interest but also their individual contribution to release rates, supporting the case for using probabilistic modelling methods in controlled drug delivery. Most importantly, we have now introduced parallelisation in order to facilitate processing of large amounts of data.

The models also allow us to look into the role of coating in controlled release, which is one device design aspect less well addressed in the literature so far, yet widely used in the

pharmaceutical industry.

Main findings

- The first three chapters of the thesis provided an in-depth assessment of the current state of DDS modelling and outlined particular advantages of using *stochasticity* as an integral part of drug dissolution modelling. The main methodologies that can be utilised in this context were presented. Fundamental formulae used for all three main modelling groups were discussed. In addition, status and limitations of existing models were outlined. This led to identification of questions that need to be addressed in new models, serving as the motivation for development of the further models presented in this thesis.

- Chapters 4 and 5 presented a new integrative modelling process and framework, flexible enough to accommodate not only the simulation of all major dissolution phenomena, but also straightforward at very little "cost". These are based on having a unified, common, meta model which describes a basic set of rules intrinsic across derived models. We showed the advantage of this approach by deriving three specific models that address all main release phenomena, (erosion, swelling and diffusion) with very good results when compared to experimental *in vitro* data. We have shown that the influence of particular Cellular Automata update mechanism plays an important role in the realism achieved in describing problem dynamics. Finally, we have shown that adaptation of the framework for high performance execution has a major impact in reducing the simulation time and allowing greater scalability for multiple simultaneous solutions, or, equivalently, increasing spatial resolution at which the simulations are performed. The HPC solution developed is adaptable to the size of the available parallel infrastructure.

- Results presented in Chapter 6 validated the developed models against experimental data and showed a very good match, with the majority of the results falling within acceptable sampling range. Additionally, with models increasing in complexity, the differences between simulation and experiment were shown to decrease demonstrating that the progressive approach, towards incorporating additional details, was sound.

- Finally, in Chapter 7 we introduced a novel application of Inverse Monte Carlo methods to the field. We demonstrated a feasible approach for reverse engineering the unknown model parameter values from generally noisy data by using a combination of Gibbs sampling and Metropolis-Hastings algorithms. The approach gave good results with reasonable time to convergence. It is clear that IMC methods can be very useful in real world applications as direct models often suffer from trying to determine many simulation parameters of the *in vitro* experiment from incomplete, sparse or noisy data.

8.2 Future Work

The models developed here have permitted investigation of the role of coating in controlled release, which is one of the device design aspects least understood so far in the literature, yet widely used in the pharmaceutical industry. A number of follow-on projects is clearly indicated.

Controlled vs. instant release for drugs of similar profiles

To evaluate model validity, simulation results should be compared to independent experimental data (Seidenberger et al., 2011). The goal of such an auxiliary project would be to understand how factors such as excipient choice, erosion/swelling mechanisms, drug loading, different media, drug solubility and so on influence instant vs. controlled release formulations. The question is whether we can use knowledge from one drug to predict release aspects of another? For example, certain dissolution phenomena can be better studied in isolation. Looking at instant release tablets will improve our knowledge of what is happening in the first few hours of release. To our knowledge there are no models which currently investigate these very different release profiles.

The influence of fragmentation and percolation on drug release from polydisperse matrix systems

Fragmentation processes and percolation of drug particle networks have been little modelled to date. Probabilistic modelling of this aspect of release would provide more realistic simulation of drug release from devices, where underlying dissolution phenomena are aggregated, rather than occurring at an individual cell level. This is of considerable importance for

pharmaceutical investigation.

The influence of Type IV dissolution apparatus on drug release from matrix controlled release tablets

As mention in the thesis, the most widely used dissolution apparatus is the USP Type II, (The United States Pharmacopeial Convention, 2007). However, its hydrodynamic conditions are far from ideal. It was shown that there are highly variable shear distributions and velocity gradients throughout the vessel, particularly in the area immediately surrounding the device. The Type IV apparatus works on a "flow through method", where the device is held in a fixed position with the dissolution medium pumped through the device holder. This makes the Type IV apparatus highly applicable for poorly soluble drugs and for detecting changes in pH in the dissolution media, (Greco et al., 2011), (Shiko et al., 2011). A collaboration with Pharmaceutical Department from University College Cork (UCC) is planned with the idea of adapting the meta-model so that main features of Type IV can be simulated.

Sample space minimisation in Inverse Monte Carlo methods using neighbourhood algorithm

In Chapter 7, we described one possible way to further optimise the speed of the inverse algorithm. As shown in Figure (7.4) it takes a relatively large number of iterations to pass the $burn - in$ period and stabilise around the optimal value. Considering that for high precision simulations runtime of individual forward simulation is significant, so that the entire iteration set is relatively slow, (even if we consider parallelisation of individual runs), reducing the parameter space becomes the primary way of speeding up the inverse simulation. We can apply the neighbourhood algorithm and divide our space into Voronoi cells to that end. This, and other means of sample space reduction would be an interesting subject for future work in this area.

Pseudo-random number generation algorithms

Similar to the pattern observed when analysing different asynchronous update mechanisms, it would be useful to investigate the influence of various pseudo-random number generation algorithms on the properties of large populations of different cell types. As random number generation forms the basis of the Monte Carlo approach used in the models, any bias initial state generation patterns on the resulting models should be understood.

Model interface unification

The ideal, of course, would be integration of all described models into one unified general-purpose system. This would facilitate use of the models as prediction tools through a single interface, adapted for pharmaceutical scientists to investigate various dissolution scenarios.

References

Adachi, S., Lee, J., Peper, F. and Umeo, H. 2008. Kaleidoscope of life: A 24-neighbourhood outer-totalistic cellular automaton. *Physica D: Nonlinear Phenomena*, 237(6), pp.800 – 817.

Ahmed, I.S. 2005. Effect of Simulated Gastrointestinal Conditions on Drug Release from Pectin/Ethylcellulose as Film Coating for Drug Delivery to the Colon. *Drug Development and Industrial Pharmacy*, 31(4-5), pp.465 – 470.

Alba, E., Giacobini, M., Tomassini, M. and Romero, S. 2002. Comparing Synchronous and Asynchronous Cellular Genetic Algorithms *IN:* Guervós, J., Adamidis, P., Beyer, H.G., Schwefel, H.P. and Fernández-Villacañas, J.L. (eds.) *Parallel Problem Solving from Nature PPSN VII*, *Lecture Notes in Computer Science* 2439. Springer Berlin / Heidelberg, pp.601 – 610.

An, L. 2012. Modeling human decisions in coupled human and natural systems: Review of agent-based models. *Ecological Modelling*, 229(0), pp.25 – 36.

Arifin, D.Y., Lee, L.Y. and Wang, C.H. 2006. Mathematical modeling and simulation of drug release from microspheres: Implications to drug delivery systems. *Advanced Drug Delivery Reviews*, 58(12-13), pp.1274 – 1325.

Atyabi, F., Vahabzadeh, R. and Dinarvand, R. 2004. Preparation of Ethylcellulose Coated Gelatin Microspheres as a Multiparticulate Colonic Delivery System for 5-Aminosalicilic Acid. *Iranian Journal of Pharmaceutical Research*, 2, pp.81 – 86.

Avdeef, A., Tsinman, K., Tsinman, O., Sun, N. and Voloboy, D. 2009. Miniaturization

of Powder Dissolution Measurement and Estimation of Particle Size. *Chemistry and Biodiversity*, 6(11), pp.1796 – 1811.

Bader, D. 2012. Accelerating Drug Discovery. *Scientific Computing World*, 123(2012), pp.30 – 31.

Baetens, J., der Weeën, P.V. and Baets, B.D. 2012. Effect of asynchronous updating on the stability of cellular automata. *Chaos, Solitons & Fractals*, 45(4), pp.383 – 394.

Bandini, S., Bonomi, A. and Vizzari, G. 2010. What Do We Mean by Asynchronous CA? A Reflection on Types and Effects of Asynchronicity *IN:* Bandini, S., Manzoni, S., Umeo, H. and Vizzari, G. (eds.) *Cellular Automata*, *Lecture Notes in Computer Science* 6350. Springer Berlin / Heidelberg, pp.385 – 394.

Bandman, O. 2006a. Parallel Simulation of Asynchronous Cellular Automata Evolution *IN:* El Yacoubi, S., Chopard, B. and Bandini, S. (eds.) *Cellular Automata*, *Lecture Notes in Computer Science* 4173. Springer Berlin / Heidelberg, pp.41 – 47.

Bandman, O. 2006b. Synchronous versus asynchronous cellular automata for simulating nano-systems kinetics. *Bulletin of the Novosibirsk Computer Center, Computer Science*, 25, pp.1 – 12.

Barat, A. 2006. Probabilistic models for drug dissolution. *Ph.D. thesis, Faculty of Engineering and Computing, School of Computing, Dublin City University.*

Barat, A., Crane, M. and Ruskin, H.J. 2008. Quantitative multi-agent models for simulating protein release from PLGA bioerodible nano- and microspheres. *Journal of Pharmaceutical and Biomedical Analysis*, 48(2), pp.361 – 368.

Barat, A., Ruskin, H.J. and Crane, M. 2006a. Probabilistic methods for drug dissolution. Part 2. Modelling a soluble binary drug delivery system dissolving *in vitro*. *Simulation Modelling Practice and Theory*, 14(7), pp.857 – 873.

Barat, A., Ruskin, H.J. and Crane, M. 2006b. Probabilistic models for drug dissolution. Part 1. Review of Monte Carlo and stochastic cellular automata approaches. *Simulation Modelling Practice and Theory*, 14(7), pp.843 – 856.

Batycky, R.P., Hanes, J., Langer, R. and Edwards, D.A. 1997. A theoretical model of erosion and macromolecular drug release from biodegrading microspheres. *Journal of Pharmaceutical Sciences*, 86(12), pp.1464 – 1477.

Bezbradica, M., Crane, M. and Ruskin, H.J. 2012. Parallelisation strategies for large scale cellular automata frameworks in pharmaceutical modelling *IN: 2012 International Conference on High Performance Computing and Simulation (HPCS)*. pp.223 –230.

Bezbradica, M., Ruskin, H.J. and Crane, M. 2011. Modelling Drug Coatings: A parallel Cellular Automata Model of Ethylcellulose-coated Microspheres *IN: International Conference on Bioscience, Biochemistry and Bioinformatics IPCBEE*, 5. IACSIT Press, Singapore, pp.419 – 424.

Bosch, R.A. 2000. Maximum density stable patterns in variants of Conway's game of Life. *Operations Research Letters*, 27(1), pp.7 – 11.

Braido, D. 2011. *Modeling and simulation of dissolution and erosion of porous solids*. Master's thesis, Graduate School - New Brunswick.

Burguillo, J.C. 2013. Playing with complexity: From cellular evolutionary algorithms with coalitions to self-organizing maps. *Computers & Mathematics with Applications*, (0), pp.In Press, Corrected Proof.

Burstedde, C., Klauck, K., Schadschneider, S. and Zittartz, J. 2001. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3-4), pp.507 – 525.

Casas, M., Strusi, O.L., Jimènez-Castellanos, M.R. and Colombo, P. 2010. Tapioca starch graft copolymers and Dome Matrix® modules assembling technology. I. Effect of module shape on drug release. *European Journal of Pharmaceutics and Biopharmaceutics*, 75(1), pp.42 – 47.

Chirico, S., Dalmoro, A., Lamberti, G., Russo, G. and Titomanlio, G. 2007. Analysis and modeling of swelling and erosion behavior for pure HPMC tablet. *Journal of Controlled Release*, 122(2), pp.181 – 188.

Colombo, P., Bettini, R., Santi, P. and Peppas, N.A. 2000. Swellable matrices for controlled drug delivery: gel-layer behaviour, mechanisms and optimal performance. *Pharmaceutical Science & Technology Today*, 3(6), pp.198 – 204.

Colorcon Inc 2009. *Opadry(R) II High Performance Film Coating Systems.* Product Information Sheet, URL: `https://www.colorcon.com/products/coatings/immediate-release/opadry-II/Literature`.

Cooney, D.O. 1972. Effect of geometry on the dissolution of pharmaceutical tablets and other solids: Surface detachment kinetics controlling. *AIChE Journal*, 18(2), pp.446 – 449.

Cornforth, D., Green, D.G. and Newth, D. 2005. Ordered asynchronous processes in multi-agent systems. *Physica D: Nonlinear Phenomena*, 204(1-2), pp.70 – 82.

Costa, P. and Lobo, J.M.S. 2001. Modeling and comparison of dissolution profiles. *European Journal of Pharmaceutical Sciences*, 13(2), pp.123 – 133.

Coulter, I. 2010. Pharmaceutical Cyclosporin Compositions. *Patent Application. US 20100203120*, (URL: `http://www.freepatentsonline.com/y2010/0203120.html`).

Council of Europe 2007. *The European Pharmacopoeia.* Strasbourg: Council of Europe, 7th edition.

Crane, M., Crane, L., Healy, A.M., Corrigan, O.I., Gallagher, K.M. and McCarthy, L.G. 2004a. A Pohlhausen solution for the mass flux from a multi-layered compact in the USP drug dissolution apparatus. *Simulation Modelling Practice and Theory*, 12(6), pp.397 – 411.

Crane, M., Hurley, N.J., Crane, L., Healy, A.M., Corrigan, O.I., Gallagher, K.M. and McCarthy, L.G. 2004b. Simulation of the USP drug delivery problem using CFD: experimental, numerical and mathematical aspects. *Simulation Modelling Practice and Theory*, 12(2), pp.147 – 158.

Crank, J. 1975. *Mathematics of Diffusion.* Clarendon: Oxford.

Cuppok, Y., Muschert, S., Marucci, M., Hjaertstam, J., Siepmann, F., Axelsson, A. and Siepmann, J. 2011. Drug release mechanisms from Kollicoat SR:Eudragit NE coated pellets. *International Journal of Pharmaceutics*, 409(1-2), pp.30 – 37.

Davies, M.E. 1998. Nonlinear noise reduction through Monte Carlo sampling. *Chaos*, 8, pp.775 – 781.

de Almeida, L.P., Simões, S., Brito, P., Portugal, A. and Figueiredo, M. 1997. Modeling dissolution of sparingly soluble multisized powders. *Journal of Pharmaceutical Sciences*, 86(6), pp.726 – 732.

Dempster, A.P., Laird, N.M. and Rubin, D.B. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1), pp.1 – 38.

Désérable, D., Dupont, P., Hellou, M. and Kamali-Bernard, S. 2011. Cellular Automata in Complex Matter. *Complex Systems*, 20(1), pp.67 – 91.

Dokoumetzidis, A. and Macheras, P. 2006. A century of dissolution research: From Noyes and Whitney to the Biopharmaceutics Classification System. *International Journal of Pharmaceutics*, 321(1-2), pp.1 – 11.

Domenek, S., Petit, E., Ducept, F., Mezdour, S., Brambati, N., Ridoux, C., Guedj, S. and Michon, C. 2008. Influence of concentration and ionic strength on the adsorption kinetics of gelatin at the air/water interface. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 331(1-2), pp.48 – 55.

Doucet, A., de Freitas, N. and Gordon, N. 2001. *Sequential Monte Carlo Methods in Practice*. Springer.

Dowd, M. and Meyer, R. 2003. A Bayesian approach to the ecosystem inverse problem. *Ecological Modelling*, 168(1-2), pp.39 – 55.

Ensslin, S., Moll, K.P., Metz, H., Otz, M. and Mäder, K. 2009. Modulating pH-independent release from coated pellets: Effect of coating composition on solubilization processes and

drug release. *European Journal of Pharmaceutics and Biopharmaceutics*, 72(1), pp.111 – 118.

Fatès, N., Thierry, E., Morvan, M. and Schabanel, N. 2006. Fully asynchronous behavior of double-quiescent elementary cellular automata. *Theoretical Computer Science*, 362(1-3), pp.1 – 16.

Gardner, M. 1970. Mathematical games: The fantastic combinations of John Conway's new solitaire game Life. *Scientific American*, 223, pp.120 – 123.

Gehrke, S.H. and Cussler, E.L. 1989. Mass transfer in pH-sensitive hydrogels. *Chemical Engineering Science*, 44(3), pp.559 – 566.

Geman, S. and Geman, D. 1984. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6), pp.721 – 741.

Geraghty, M. 2004. Investigation of ibuprofen release from ethylcellulose matrix compacts. *Ph.D. thesis, Trinity College, University of Dublin.*

Göpferich, A. 1996. Mechanisms of polymer degradation and erosion. *Biomaterials*, 17(2), pp.103 – 114.

Göpferich, A. 1997a. Bioerodible implants with programmable drug release. *Journal of Controlled Release*, 44(2-3), pp.271 – 281.

Göpferich, A. 1997b. Erosion of composite polymer matrices. *Biomaterials*, 18(5), pp.397 – 403.

Göpferich, A., Karydas, D. and Langer, R. 1995. Predicting drug release from cylindric polyanhydride matrix discs. *European Journal of Pharmaceutics and Biopharmaceutics*, 42(2), pp.81 – 87.

Göpferich, A. and Langer, R. 1993. Modeling of Polymer Erosion. *Macromolecules*, 26, pp.4105 – 4112.

Göpferich, A. and Langer, R. 1995a. Modeling monomer release from bioerodible polymers. *Journal of Controlled Release*, 33(1), pp.55 – 69.

Göpferich, A. and Langer, R. 1995b. Modeling of polymer erosion in three dimensions: rotationally symmetric devices. *American Institute of Chemical Engineers Journal*, 41(10), pp.2292 – 2299.

Grassi, M., Colombo, I. and Lapasin, R. 2000. Drug release from an ensemble of swellable crosslinked polymer particles. *Journal of Controlled Release*, 68(1), pp.97 – 113.

Grassi, M. and Grassi, G. 2005. Mathematical Modelling and Controlled Drug Delivery: Matrix Systems. *Current Drug Delivery*, 2(1), pp.97 – 116.

Greco, K., Bergman, T.L. and Bogner, R. 2011. Design and characterization of a laminar flow-through dissolution apparatus: Comparison of hydrodynamic conditions to those of common dissolution techniques. *Pharmaceutical Development and Technology*, 16(1), pp.75 – 87.

Harland, R.S., Gazzaniga, A., Sangalli, M.E., Colombo, P. and Peppas, N.A. 1988. Drug/Polymer Matrix Swelling and Dissolution. *Pharmaceutical Research*, 5(8), pp.488 – 494.

Healy, A.M. and Corrigan, O.I. 1996. The influence of excipient particle size, solubility and acid strength on the dissolution of an acidic drug from two-component compacts. *International Journal of Pharmaceutics*, 143(2), pp.211 – 221.

Heller, J. and Baker, R.W. 1980. Theory and Practice of Controlled Drug Delivery from Bioerodible Polymers *IN:* Baker, R. (ed.) *Controlled Release of Bioactive Materials.* Academic Press, pp.1 – 17.

Higuchi, T. 1960. Physical Chemical Analysis of Percutaneous Absorption Process from Creams and Ointments. *Journal of the Society of Cosmetic Chemists*, 11, pp.85 – 97.

Higuchi, T. 1961. Rate of release of medicaments from ointment bases containing drugs in suspension. *Journal of Pharmaceutical Sciences*, 50(10), pp.874 – 875.

Higuchi, W.I. and Hiestand, E.N. 1963. Dissolution rates of finely divided drug powders I. Effect of a distribution of particle sizes in a diffusion-controlled process. *Journal of Pharmaceutical Sciences*, 52(1), pp.67 – 71.

Hixson, A.W. and Crowell, J.H. 1931. Dependence of reaction velocity upon surface and agitation. *Industrial and Engineering Chemistry*, 23(8), pp.923 – 931.

Hoekstra, A., Falcone, J.L., Caiazzo, A. and Chopard, B. 2008. Multi-scale Modeling with Cellular Automata: The Complex Automata Approach *IN:* Umeo, H., Morishita, S., Nishinari, K., Komatsuzaki, T. and Bandini, S. (eds.) *Cellular Automata*, *Lecture Notes in Computer Science* 5191. Springer Berlin Heidelberg, pp.192 – 199.

Holland, J.H. and Miller, J.H. 1991. Artificial adaptive agents in economic theory. *American Economic Review*, 81(2), pp.365 – 370.

Hopfenberg, H.B. 1976. *Controlled release from erodible slabs, cylinders, and spheres.* Controlled Release Polymeric Formulations, ACS Symposium Series, Vol 33. American Chemical Society, Washington.

Hughes, H.P.N., Clegg, C.W., Robinson, M.A. and Crowder, R.M. 2012. Agent-based modelling and simulation: The potential contribution to organizational psychology. *Journal of Occupational and Organizational Psychology*, 85(3), pp.487 – 502.

Jennings, N.R. and Sycara, K. 1998. A Roadmap of Agent Research and Development.

Jin, H., Jespersen, D., Mehrotra, P., Biswas, R., Huang, L. and Chapman, B. 2011. High performance computing using MPI and OpenMP on multi-core parallel systems. *Parallel Computing*, 37(9), pp.562 – 575.

Ju, R.T.C., Nixon, P.R. and Patel, M.V. 1995a. Drug release from hydrophilic matrices. 1. New scaling laws for predicting polymer and drug release based on the polymer disentanglement concentration and the diffusion layer. *Journal of Pharmaceutical Sciences*, 84(12), pp.1455 – 1463.

Ju, R.T.C., Nixon, P.R., Patel, M.V. and Tong, D.M. 1995b. Drug release from hydrophilic matrices. 2. A mathematical model based on the polymer disentanglement concentration and the diffusion layer. *Journal of Pharmaceutical Sciences*, 84(12), pp.1464 – 1477.

Kalampokis, A., Argyrakis, P. and Macheras, P. 1999a. A heterogeneous tube model of

intestinal drug absorption based on probabilistic concepts. *Pharmaceutical Research*, 16(11), pp.1764 – 1769.

Kalampokis, A., A., Argyrakis, P. and Macheras, P. 1999b. Heterogeneous Tube Model for the Study of Small Intestinal Transit Flow. *Pharmaceutical Research*, 16(1), pp.87 – 91.

Kalgin, K.V. 2008. Parallel Simulation of Asynchronous Cellular Automata evolution. *Bulletin of the Novosibirsk Computer Center, series Computer Science*, 27, pp.55 – 62.

Kandalla, K., Subramoni, H., Vishnu, A. and Panda, D. 2010. Designing topology-aware collective communication algorithms for large scale InfiniBand clusters: Case studies with Scatter and Gather *IN: Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*. pp.1 – 8.

Kaunisto, E., Marucci, M., Borgquist, P. and Axelsson, A. 2011. Mechanistic modelling of drug release from polymer-coated and swelling and dissolving polymer matrix systems. *International Journal of Pharmaceutics*, 418(1), pp.54 – 77.

Keilis-Borok, V.I. and Yanovskaja, T.B. 1967. Inverse Problems of Seismology (Structural Review). *Geophysical Journal of the Royal Astronomical Society*, 13(1-3), pp.223 – 234.

Kim, B. 2009. Prevention of falls during stairway descent in older adults. *Applied Ergonomics*, 40(3), pp.348 – 352.

Kimber, J.A., Kazarian, S.G. and Štěpánek, F. 2011a. A fast algorithm for mass transfer on an unstructured grid formed by DEM particles. *Powder Technology*, 214(3), pp.415 – 422.

Kimber, J.A., Kazarian, S.G. and Štěpánek, F. 2011b. Microstructure-based mathematical modelling and spectroscopic imaging of tablet dissolution. *Computers & Chemical Engineering*, 35(7), pp.1328 – 1339.

Kimber, J.A., Kazarian, S.G. and Štěpánek, F. 2012. Modelling of pharmaceutical tablet swelling and dissolution using discrete element method. *Chemical Engineering Science*, 69(1), pp.394 – 403.

Klepko, V.V. and Mel'nichenko, Y.B. 1995. Kinetics and equilibrium swelling of gelatine gels. *Polymer*, 36(26), pp.5057 – 5059.

Korcek, P., Sekanina, L. and Fucik, O. 2011. A scalable cellular automata based microscopic traffic simulation *IN: Intelligent Vehicles Symposium (IV), 2011 IEEE*. pp.13 –18.

Korsmeyer, R.W., Gurny, K., Doelker, E., Buri, P. and Peppas, N.A. 1983. Mechanisms of solute release from porous hydrophilic polymers. *International Journal of Pharmaceutics*, 15(1), pp.25 – 35.

Korsmeyer, R.W., Von Meerwall, E. and Peppas, N.A. 1986. Solute and penetrant diffusion in swellable polymers. II. Verification of theoretical models. *Journal of Polymer Science Part B: Polymer Physics*, 24(2), pp.409 – 434.

Kosmidis, K., Argyrakis, P. and Macheras, P. 2003a. Fractal kinetics in drug release from finite fractal matrices. *Journal of Chemical Physics*, 119(12), pp.6373 – 6377.

Kosmidis, K. and Macheras, P. 2007. Monte Carlo simulations for the study of drug release from matrices with high and low diffusivity areas. *International Journal of Pharmaceutics*, 343(1-2), pp.166 – 172.

Kosmidis, K. and Macheras, P. 2008. Monte Carlo simulations of drug release from matrices with periodic layers of high and low diffusivity. *International Journal of Pharmaceutics*, 354(1-2), pp.111 – 116.

Kosmidis, K., Rinaki, E., Argyrakis, P. and Macheras, P. 2003b. Analysis of Case II drug transport with radial and axial release from cylinders. *International Journal of Pharmaceutics*, 254(2), pp.183 – 188.

Kotecha, J.H. and Djuric, P.M. 1999. Gibbs sampling approach for generation of truncated multivariate Gaussian random variables *IN: Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, 3. pp.1757 –1760.

Kreye, F., Siepmann, F. and Siepmann, J. 2011. Drug release mechanisms of compressed lipid implants. *International Journal of Pharmaceutics*, 404(1-2), pp.27 – 35.

Laaksonen, H., Hirvonen, J. and Laaksonen, T. 2009a. Cellular automata model for swelling-controlled drug release. *International Journal of Pharmaceutics*, 380(1-2), pp.25 – 32.

Laaksonen, T.J., Laaksonen, H.M., Hirvonen, J.T. and Murtomäki, L. 2009b. Cellular automata model for drug release from binary matrix and reservoir polymeric devices. *Biomaterials*, 30(10), pp.1978 – 1987.

Langer, R. and Peppas, N. 1983. Chemical and Physical Structure of Polymers as Carriers for Controlled Release of Bioactive Agents: A Review. *Journal of Macromolecular Science, Part C: Polymer Reviews*, 23(1), pp.61 – 126.

Lao, L.L., Peppas, N.A., Boey, F.Y.C. and Venkatraman, S.S. 2011. Modeling of drug release from bulk-degrading polymers. *International Journal of Pharmaceutics*, 418(1), pp.28 – 41.

Lee, P.I. 1980. Diffusional release of a solute from a polymeric matrix - approximate analytical solutions. *Journal of Membrane Science*, 7(3), pp.255 – 275.

Lee, P.I. and Peppas, N.A. 1987. Prediction of polymer dissolution in swellable controlled-release systems. *Journal of Controlled Release*, 6(1), pp.207 – 215.

Lee, S.J.E. and Chakraborty, A.K. 2002. Sequence dependence of polymer dynamics in quenched disordered media: weak attraction facilitates transport. *Journal of Chemical Physics*, 117(23), pp.10869 – 10876.

Leon, P.A., Basurto, R., Martinez, G.J. and Seck-Tuoh-Mora, J.C. 2011. Complex dynamics in a hexagonal cellular automaton *IN: High Performance Computing and Simulation (HPCS), 2011 International Conference on.* pp.750 – 756.

Leopold, C.S. 1999. Coated dosage forms for colon-specific drug delivery. *Pharmaceutical Science & Technology Today*, 2(5), pp.197 – 204.

Li, D., Hohne, D., Bortz, D., Bull, J. and Younger, J. 2007. Modeling bacterial clearance from the bloodstream using computational fluid dynamics and Monte Carlo simulation. *Journal of Critical Care*, 22(4), pp.344 – 344.

Li, Y. and Tanaka, T. 1990. Kinetics of swelling and shrinking of gels. *Journal of Chemical Physics*, 92(2), p.1365.

Liu, L., Fishman, M.L. and Hicks, K.B. 2007. Pectin in controlled drug delivery - a review. *Cellulose*, 14(1), pp.15 – 24.

Loney, N.W. and Susarla, R. 2009. Mathematical Modeling of Drug Release from spherical Drug Particles: Analysis of the Effect of Absorption Rate on Drug Release. *Chemical Product and Process Modeling*, 4(10).

Maggi, L., Torre, M.L., Giunchedi, P. and Conte, U. 1996. Supramicellar solutions of sodium dodecyl sulphate as dissolution media to study the *in vitro* release characteristics of sustained-release formulations containing an insoluble drug: Nifedipine. *International Journal of Pharmaceutics*, 135(1-2), pp.73 – 79.

Margenstern, M. 2011. Bacteria inspired patterns grown with hyperbolic cellular automata *IN: High Performance Computing and Simulation (HPCS), 2011 International Conference on.* pp.757 –763.

Martínez, L., Villalobos, R., Sánchez, M., Cruz, J., Ganem, A. and Melgoza, L.M. 2009. Monte Carlo simulations for the study of drug release from cylindrical matrix systems with an inert nucleus. *International Journal of Pharmaceutics*, 369(1-2), pp.38 – 46.

Martínez-Ruvalcaba, A., Sánchez-Díaz, J.C., Becerra, F., Cruz-Barba, L.E. and González-Álvarez, A. 2009. Swelling characterization and drug delivery kinetics of polyacrylamide-co-itaconic acid/chitosan hydrogels. *eXPRESS Polymer Letters*, 3(1), pp.25 – 32.

Marucci, M., Ragnarsson, G., Nilsson, B. and Axelsson, A. 2010. Osmotic pumping release from ethyl-hydroxypropyl-cellulose-coated pellets: A new mechanistic model. *Journal of Controlled Release*, 142(1), pp.53 – 60.

Massaioli, F., Castiglione, F. and Bernaschi, M. 2005. OpenMP parallelization of agent-based models. *Parallel Computing*, 31(10-12), pp.1066 – 1081.

Matsumoto, M. and Nishimura, T. 1998. Mersenne twister: a 623-dimensionally equidis-

tributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8, pp.3 – 30.

McGreevy, R.L. 2001. Reverse Monte Carlo modelling. *Journal of Physics: Condensed Matter*, 13(46), pp.R877 – R913.

McGreevy, R.L. and Pusztai, L. 1988. Reverse Monte Carlo Simulation: A New Technique for the Determination of Disordered Structures. *Molecular Simulation*, 1(6), pp.359 – 367.

McMahon, N. 2008. The Mechanics of Drug Dissolution. *Ph.D. thesis, Faculty of Engineering and Computing, School of Computing, Dublin City University.*

McMahon, N., Crane, M., Ruskin, H.J. and Crane, L. 2007. The importance of boundary conditions in the simulation of dissolution in the USP dissolution apparatus. *Simulation Modelling Practice and Theory*, 15(3), pp.247 – 255.

McMahon, N.M., Crane, M., Ruskin, H.J. and Crane, L. 2003. The Mechanics of Drug Dissolution. *Proceedings of Applied Maths and Mechanics*, 3, pp.392 – 393.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. 1953. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6), pp.1087 – 1092.

Mitchell, T.M. 1997. *Machine Learning.* Mc-Graw Hill.

Moore, J.W. and Flanner., H.H. 1996. Mathematical comparison of curves with an emphasis on *in-vitro* dissolution profiles. *Pharmaceutical Technology*, 20(6), pp.64 – 74.

Mosegaard, K. and Sambridge, M. 2002. Monte Carlo analysis of inverse problems. *Inverse Problems*, 18(3), pp.R29 – R54.

Muschert, S., Siepmann, F., Leclercq, B., Carlin, B. and Siepmann, J. 2009. Prediction of drug release from ethylcellulose coated pellets. *Journal of Controlled Release*, 135(1), pp.71 – 79.

Nakasato, N. 2009. Oct-tree Method on GPU, [Online]. URL: `http://arxiv.org/abs/0909.0541v1`.

Narasimhan, B. 2001. Mathematical models describing polymer dissolution: consequences for drug delivery. *Advanced Drug Delivery Reviews*, 48(2-3), pp.195 – 210.

Narasimhan, B. and Peppas, N.A. 1997. Molecular analysis of drug delivery systems controlled by dissolution of the polymer carrier. *Journal of Pharmaceutical Sciences*, 86(3), pp.297 – 304.

Ng, A. 2013. Multivariate Gaussian Distribution, Homepage of Andrew Ng, Stanford University. *Online homepage available from:* `http://ai.stanford.edu/~ang/`. *Access date: 27/02/2013.*

Ninagawa, S., Yoneda, M. and Hirose, S. 1998. $1/f$ fluctuation in the "Game of Life". *Physica D: Nonlinear Phenomena*, 118(1-2), pp.49 – 52.

Noyes, A.A. and Whitney, W.R. 1897. The rate of solution of solid substances in their own solutions. *Journal of the American Chemical Society*, 19, pp.930 – 934.

Ojanen, T., Sijtsema, J.J., Hawley, P.H. and Little, T.D. 2010. Intrinsic and extrinsic motivation in early adolescents' friendship development: friendship selection, influence, and prospective friendship quality. *Journal of Adolescence*, 33(6), pp.837 – 51.

Papadopoulou, V., Kosmidis, K., Vlachou, M. and Macheras, P. 2006. On the use of the Weibull function for the discernment of drug release mechanisms. *International Journal of Pharmaceutics*, 309(1-2), pp.44 – 50.

Park, M.J., Balakrishnan, P. and Yang, S.G. 2013. Polymeric nanocapsules with SEDDS oil-core for the controlled and enhanced oral absorption of cyclosporine. *International Journal of Pharmaceutics*, 441(1-2), pp.757 – 764.

Patanarapeelert, K., Frank, T. and Tang, I. 2011. From a cellular automaton model of tumor-immune interactions to its macroscopic dynamical equation: A drift-diffusion data analysis approach. *Mathematical and Computer Modelling*, 53(1-2), pp.122 – 130.

Peppas, N.A. and Langer, R. 1994. New Challenges in Biomaterials. *Science*, 263(5154), pp.1715 – 1720.

Perrin, D. 2008. Multi-layered model of individual HIV infection progression and mechanisms of phenotypical expression. *Ph.D. thesis, Faculty of Engineering and Computing, School of Computing, Dublin City University.*

Perrin, D., Bezbradica, M., Crane, M., Ruskin, H.J. and Duhamel, C. 2012. High-Performance Computing for Data Analytics *IN: Distributed Simulation and Real Time Applications (DS-RT), 2012 IEEE/ACM 16th International Symposium on.* pp.234 – 242.

Pope, B.J., Fitch, B.G., Pitman, M.C., Rice, J.J. and Reumann, M. 2011. Performance of hybrid programming models for multiscale cardiac simulations: preparing for petascale computation. *IEEE Transactions on Biomedical Engineering*, 58(10), pp.2965 – 2969.

Ramtoola, Z. and Corrigan, O.I. 1987. Dissolution characteristics of benzoic acid and salicylic acid mixtures in reactive media. *Drug Development and Industrial Pharmacy*, 13, pp.9 – 11.

Ravindra, R., Sridhar, S., Khan, A. and Rao, A. 2000. Pervaporation of water, hydrazine and monomethylhydrazine using ethylcellulose membranes. *Polymer*, 41(8), pp.2795 – 2806.

Ritger, P.L. and Peppas, N.A. 1987. A simple equation for description of solute release I. Fickian and non-Fickian release from non-swellable devices in the form of slabs, spheres, cylinders or discs. *Journal of Controlled Release*, 5(1), pp.23 – 36.

Robert, C. and Casella, G. 2011. A Short History of Markov Chain Monte Carlo: Subjective Recollections from Incomplete Data. *Statistical Science*, 26(1), pp.102 – 115.

Rodríguez-González, A., Torres-Niño, J., Hernández-Chan, G., Jiménez-Domingo, E. and Alvarez-Rodríguez, J.M. 2012. Using agents to parallelize a medical reasoning system based on ontologies and description logics as an application case. *Expert Systems with Applications*, 39(18), pp.13085 – 13092.

Rothstein, S.N., Federspiel, W.J. and Little, S.R. 2009. A unified mathematical model for the prediction of controlled release from surface and bulk eroding polymer matrices. *Biomaterials*, 30(8), pp.1657 – 1664.

Sackett, C.K. and Narasimhan, B. 2011. Mathematical modeling of polymer erosion: Consequences for drug delivery. *International Journal of Pharmaceutics*, 418(1), pp.104 – 114.

Sadeghi, F., Ford, J.L. and Rajabi-Siahboomi, A. 2003. The influence of drug type on the release profiles from Surelease-coated pellets. *International Journal of Pharmaceutics*, 254(2), pp.123 – 135.

Sambridge, M. 1999. Geophysical inversion with a neighbourhood algorithm-I. Searching a parameter space. *Geophysical Journal International*, 138(2), pp.479 – 494.

Schatz, M.C., Langmead, B. and Salzberg, S.L. 2010. Cloud Computing and the DNA Data Race. *Nature Biotechnology*, 28(7), pp.691 – 693.

Schönfisch, B. and de Roos, A. 1999. Synchronous and asynchronous updating in cellular automata. *Biosystems*, 51(3), pp.123 – 143.

Seidenberger, T., Siepmann, J., Bley, H., Maeder, K. and Siepmann, F. 2011. Simultaneous controlled vitamin release from multiparticulates: Theory and experiment. *International Journal of Pharmaceutics*, 412(1-2), pp.68 – 76.

Shah, V., Tsong, Y., Sathe, P. and Liu, J.P. 1998. *In Vitro* Dissolution Profile Comparison-Statistics and Analysis of the Similarity Factor, f2. *Pharmaceutical Research*, 15, pp.889–896.

Shiko, G., Gladden, L.F., Sederman, A.J., Connolly, P.C. and Butler, J.M. 2011. MRI Studies of the Hydrodynamics in a USP 4 Dissolution Testing Cell. *Journal of Pharmaceutical Sciences*, 100(3), pp.976 – 991.

Siepmann, J., Faisant, N. and Benoit, J.P. 2002. A New Mathematical Model Quantifying Drug Release from Bioerodible Microparticles Using Monte Carlo Simulations. *Pharmaceutical Research*, 19(12), pp.1885 – 1893.

Siepmann, J. and Göpferich, A. 2001. Mathematical modeling of bioerodible, polymeric drug delivery systems. *Advanced Drug Delivery Reviews*, 48(2-3), pp.229 – 247.

Siepmann, J. and Peppas, N.A. 2001. Modeling of drug release from delivery systems based on hydroxypropyl methylcellulose (HPMC). *Advanced Drug Delivery Reviews*, 48(2-3), pp.139 – 157.

Siepmann, J. and Peppas, N.A. 2011. Higuchi equation: Derivation, applications, use and misuse. *International Journal of Pharmaceutics*, 418(1), pp.6 – 12.

Siepmann, J. and Siepmann, F. 2008. Mathematical modeling of drug delivery. *International Journal of Pharmaceutics*, 364(2), pp.328 – 343.

Siepmann, J. and Siepmann, F. 2012a. Modeling of diffusion controlled drug delivery. *Journal of Controlled Release*, 161(2), pp.351 – 362.

Siepmann, J. and Siepmann, F. 2012b. Swelling Controlled Drug Delivery Systems *IN:* Siepmann, J., Siegel, R.A. and Rathbone, M.J. (eds.) *Fundamentals and Applications of Controlled Release Drug Delivery*, Advances in Delivery Science and Technology. Springer US, pp.153 – 170.

Silberschatz, A., Galvin, P.B. and Gagne, G. 2012. *Operating System Concepts*. Wiley.

Simões, S., de Almeida, L. and Figueiredo, M. 1996. Testing the applicability of classical diffusional models to polydisperse systems. *International Journal of Pharmaceutics*, 139(1-2), pp.169 – 176.

Singh, J. and Weber, M.E. 1996. Kinetics of one-dimensional gel swelling and collapse for large volume change. *Chemical Engineering Science*, 51(19), pp.4499 – 4508.

Sloot, P.M.A., Kaandorpa, J.A., Hoekstra, A.G. and Overeinder, B.J. 1999. Distributed Simulation with Cellular Automata: Architecture and Applications *IN:* Pavelka, J., Tel, G. and Bartošek, M. (eds.) *SOFSEM'99: Theory and Practice of Informatics*, *Lecture Notes in Computer Science* 1725. Springer Berlin Heidelberg, pp.203 – 248.

Socco, L. and Boiero, D. 2008. Improved Monte Carlo inversion of surface wave data. *Geophysical Prospecting*, 56, pp.357 – 371.

Sopasakis, A. 2004. Stochastic noise approach to traffic flow modeling. *Physica A: Statistical Mechanics and its Applications*, 342(3-4), pp.741 – 754.

Stevenson, A., Du, Y.L. and Afrit, M.E. 2011. High-performance computing on gamer PCs. *Ars Technica*.

Tanaka, T. and Fillmore, D. 1979. Kinetics of swelling of gels. *Journal of Chemical Physics*, 70(3), pp.1214 – 1218.

Tezuka, S., Murata, H., Tanaka, S. and Yumae, S. 2005. Monte Carlo grid for financial risk management. *Future Generation Computer Systems*, 21(5), pp.811 – 821.

Thakur, R., Gropp, W. and Toonen, B. 2004. Minimizing Synchronization Overhead in the Implementation of MPI One-Sided Communication *IN:* Kranzlmüller, D., Kacsuk, P. and Dongarra, J. (eds.) *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, *Lecture Notes in Computer Science* 3241. Springer Berlin / Heidelberg, pp.57 – 67.

The United States Pharmacopeial Convention 2007. *United States Pharmacopeia and National Formulary (USP 30-NF 25)*, 2. Rockville, MD.

Toffoli, T. and Margolus, N. 1987. *Cellular Automata Machines: A New Environment for Modeling*. MIT Press, Cambridge MA.

Trammer, B., Amann, A., Haltner-Ukomadu, E., Tillmanns, S., Keller, M. and Högger, P. 2008. Comparative permeability and diffusion kinetics of cyclosporine A liposomes and propylene glycol solution from human lung tissue into human blood ex vivo. *European Journal of Pharmaceutics and Biopharmaceutics*, 70(3), pp.758 – 764.

Ulubayram, K., Eroglu, I. and Hasirci, N. 2002. Gelatin Microspheres and Sponges for Delivery of Macromolecules. *Journal of Biomaterials Application*, 16, pp.227 – 241.

Valsecchi, A., Vanneschi, L. and Mauri, G. 2010. A Study on the Automatic Generation of Asynchronous Cellular Automata Rules by Means of Genetic Algorithms *IN:* Bandini, S., Manzoni, S., Umeo, H. and Vizzari, G. (eds.) *Cellular Automata*, *Lecture Notes in Computer Science* 6350. Springer Berlin / Heidelberg, pp.429 – 438.

Vasic, J. and Ruskin, H.J. 2012. Cellular automata simulation of traffic including cars and bicycles. *Physica A: Statistical Mechanics and its Applications*, 391(8), pp.2720 – 2729.

Verhoeven, E., Siepmann, F., De Beer, T., Van Loo, D., Van den Mooter, G., Remon, J.P., Siepmann, J. and Vervaet, C. 2009. Modeling drug release from hot-melt extruded mini-matrices with constant and non-constant diffusivities. *European Journal of Pharmaceutics and Biopharmaceutics*, 73(2), pp.292 – 301.

Villalobos, R., Domínguez, A., Ganem, A., Vidales, A.M. and Cordero, S. 2009. One-dimensional drug release from finite Menger sponges: In silico simulation. *Chaos, Solitons & Fractals*, 42(5), pp.2875 – 2884.

von Burkersroda, F., Schedl, L. and Göpferich, A. 2002. Why degradable polymers undergo surface erosion or bulk erosion. *Biomaterials*, 23(21), pp.4221 – 4231.

von Neumann, J. 1966. *Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana.

Voorhees, B. 1990. Nearest neighbor cellular automata over Z2 with periodic boundary conditions. *Physica D: Nonlinear Phenomena*, 45(1-3), pp.26 – 35.

Voronoi, G. 1908. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1908(134), pp.198 – 287.

Voutilainen, A., Kolehmainen, V. and Kaipio, J.P. 2001. Statistical inversion of aerosol size measurement data. *Inverse Problems in Science and Engineering*, 9(1), pp.67 – 94.

Vrentas, J.S., Jarzebski, C.M. and Duda, J.L. 1975. A Deborah number for diffusion in polymer-solvent systems. *AIChE Journal*, 21(5), pp.894 – 901.

Wen, W. 2008. A dynamic and automatic traffic light control expert system for solving the road congestion problem. *Expert Systems with Applications*, 34(4), pp.2370 – 2381.

Wooldridge, M. and Jennings, N.R. 1995. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10, pp.115 – 152.

Xiao, X., Shao, S., Ding, Y., Huang, Z., Chen, X. and Chou, K.C. 2005. Using cellular automata to generate image representation for biological sequences. *Amino Acids*, 28, pp.29 – 35.

Zygourakis, K. 1990. Development and temporal evolution of erosion fronts in bioerodible controlled release devices. *Chemical Engineering Science*, 45(8), pp.2359 – 2366.

Zygourakis, K. and Markenscoff, P.A. 1996. Computer-aided design of bioerodible devices with optimal release characteristics: a cellular automata approach. *Biomaterials*, 17(2), pp.125 – 135.

Appendix A

Framework and individual model implementation details

The framework and models have been implemented using an object-oriented paradigm, with model classes representing an extended version of the ones offered by the modelling framework. The framework defines a set of common modules, including the definition of the main Cellular Automata matrix, the parallel algorithms used for its traversal, the visualiser used for graphical display of the state of each iteration and the configuration block responsible for reading the simulation parameters from a file. Unified Modelling Language (UML) class diagram of the framework and one of the derived models (MCES) is represented in Figure A.1. Note that only the most important class methods and attributes have been displayed for brevity purposes. What follows is a more detailed description of each class:

CA Matrix

The **CAMatrix** is the *abstract* class holding the parameters and methods relevant to describing the Cellular Automata space (matrix) itself. Besides the spatial (width, height, depth, number of cells) and temporal (number of iterations, current iteration, global time) attributes it defines several key simulation methods:

- **Initialise()** is a *virtual* method responsible for assigning the initial state of each cell in the matrix. It must be derived by any class implementing the specific device model

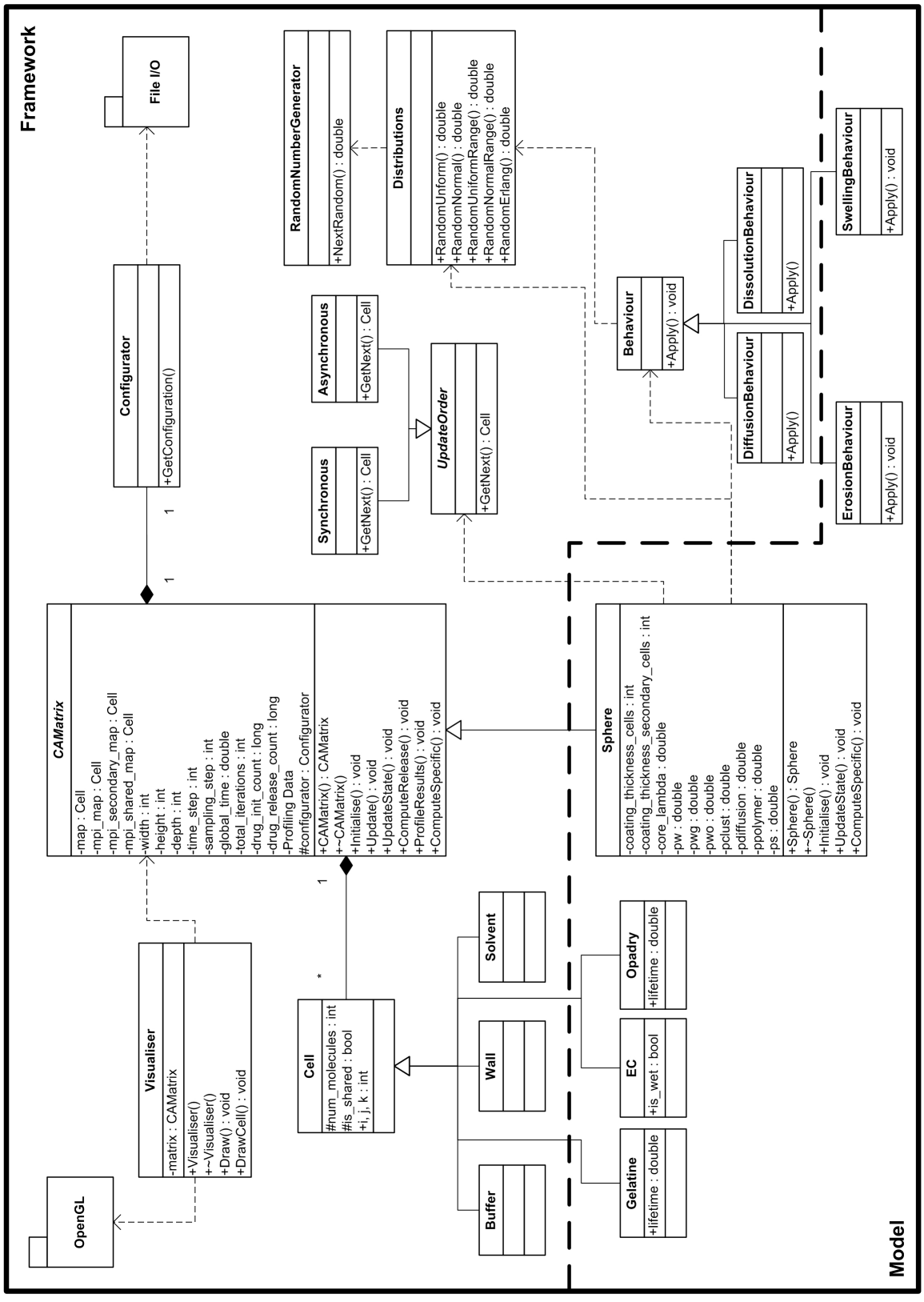


Figure A.1: UML diagram of the developed code. Thick dashed line shows the division between the classes and methods provided by the framework and meta-model and the ones derived by the individual models.

and geometry.

- **Update()** is a method called during each iteration by either the *Visualiser* class (if the graphical representation mode is enabled) or the main program. It uses the *UpdateOrder* to traverse the CA matrix space and calls the virtual **UpdateState()** function defined by derived geometry classes. The **Update()** function also handles the *scatter-gather* logic of the MPI parallelisation and the division of matrix space across individual threads within the single process using OpenMP.

- **ComputeRelease()** is a function that calculates the amount of released drug within the matrix and calls the virtual method **ComputeSpecific()** implemented by the derived models. **ComputeSpecific()** can then calculate model outputs specific to the drug/geometry being modelled (e.g. dissolution fronts, coating porosity, gel blocking amount, etc.)

- **ProfileResults()** gathers profiling parameters using several timers dispersed across the code, such as the duration of single iteration, durations of iterations in each MPI process and the amount of time spent in parallel and sequential code.

Sphere

Sphere is a class derived from **CAMatrix()** describing the specific device properties, including its geometry. It defines the main model parameters (swelling probability, clustering probability, probability of diffusion of water through ethylcellulose etc.) and is responsible for setting up the initial state. It implements the *virtual* methods of **Initialise()**, **UpdateState()** and **ComputeSpecific()** to add the logic specific to the model itself.

Visualiser

The **Visualiser** class is responsible for displaying the graphical representation of the **CAMatrix** on the screen using OpenGL libraries. It does so by traversing the cross section of the matrix and drawing each cell depending on the type of material, its lifetime and number of drug packets present.

Configurator

The **Configurator** class is responsible for parsing the configuration file with the simulation parameters and making those available to other classes.

Cell and derived classes

Cell class represents the basic building block of the CA matrix. It contains information about a particular cell, including the type of material (through specific subclasses) and the amount of drug present. It also contains several attributes describing the position of the cell within the matrix and whether it is shared between several processes for the purpose of MPI parallelisation.

UpdateOrder

UpdateOrder implements the specific cell traversal algorithm, which can be either synchronous (each cell in order) or asynchronous (random cells depending on the ACA algorithm used). It has a single *virtual* method **GetNext()** which is responsible for returning the next cell to update.

Distributions

Distributions is a *static* class that provides several random number generation methods used for sampling from different distributions (uniform, Gaussian, Erlang, etc.)

Behaviour and derived classes

Behaviour is a *virtual* class providing a single *virtual* method **Apply()** which is used to calculate the next state of the given cell as the traversal algorithm updates it. The derived classes implement the method for specific behaviour desired (i.e. erosion, swelling, diffusion, dissolution and so on).

Typical iteration algorithm

The simulation starts by initialising the MPI and OpenMP runtimes and setting up the initial CA matrix state by calling the **Initialise()** method of the **Sphere** object. Depending on

whether visualisation is enabled, the **Update()** method is set up to be called by the OpenGL runtime before rendering each frame of the animation to the screen, or, if the visualisation is disabled (in HPC cluster execution context for example), it is called sequentially as many times as there are iterations. The **Update()** method itself distributes the matrix state across all of the initiated MPI processes, which in turn execute an OpenMP parallel loop over the individual sections received (thus dividing the space further across threads). Each thread invokes the **UpdateSpecific()** method on the cell obtained via the **UpdateOrder** object. This, in turn, applies all of the desired behaviours on the cell, setting its final state.

After the simulation goes through all cells, the main MPI process collects the resulting state, performs release data calculations and writes the output into a file that can be later plotted using any available tabular processing program (such as *R*). In case of cluster executions, the plotting is performed automatically as part of the execution script thus producing readily-available graphs once the simulation finishes.

Appendix B

List of abbreviations

ABM Agent Based Modelling

ACA Asynchronous Cellular Automata

API Application Programming Interface

CA Cellular Automata

CyA Cyclosporine A

DDS Drug Delivery System

EC Ethylcellulose

EC/P Ethycellulose/Pectin

EXP Experiment

GI Gastro-intestinal

HPC High-Performance Computing

HPMC HydroxyPropyl MethylCellulose

IMC Inverse Monte Carlo

LOESS Locally Weighted Scatterplot Smoothing

MC Monte Carlo

MCMC Markov Chain Monte Carlo

M-H Metropolis Hastings algorithm

MPI Message Passing Interface

MVN MultiVariate Normal distribution

NA Neighbourhood Algorithm

N-W Noyes-Whitney equation

PLA Polylactide

PLGA Polylactic-co-glycolic Acid

SA Simulated Annealing

SCA Synchronous Cellular Automata

SEM Scanning Electron Microscope

SMP Symmetric MultiProcessing

UML Universal Modelling Language

USP United States Pharmacopeia

Appendix C

Glossary

- **Biodegradable** - capable of being decomposed by living organisms.
- **Coating weight gain** - amount of device mass increase (in %) after application of the coating layer.
- **EXP11/085, EXP11/086, EXP11/115, EXP11/180, EXP11/227, EXP12/111, EXP12/119, FC008/09, FC015/09, FC021/09** - Laboratory designations of *in vitro* experiments performed by Sigmoid Pharma. These experiments were used for comparison with simulation results.
- ***In silico*** - via a computer simulation.
- ***In vivo*** - within a living organism.
- ***In vitro*** - in an artificial environment outside the living organism.
- **Locally Weighted Scatterplot Smoothing** - a set of regression methods combining multiple regression models in a k-nearest-neighbour meta-model. Used for plotting a smooth curve through a set of data points.
- **Message Passing Interface** - a computing standard for implementing parallelisation across processes using message transfers. There are various implementations of the standard, including OpenMPI and MPICH.
- **Opadry®** - a complete film coating system that combines polymer, plasticiser and pigment in a dry concentrate. Developed by Colorcon®.

- **OpenMP** - application programming interface that provides shared memory multiprocessing. It is commonly used on SMP architectures.
- **Parenteral** - administration of pharmaceutics by intravenous routes.
- **Process-level parallelism** - code parallelisation methodology using heavy-weight *processes* usually executing on different computers (*nodes*). Technologies such as different MPI implementations are commonly used to achieve process-level parallelism.
- **Surelease** - aqueous coating system utilising ethylcellulose as a rate controlling polymer. Developed by Colorcon®.
- **Symmetric MultiProcessing** - hardware architecture where two or more processors are connected to a single shared main memory. This allows the code executing on each processor to access a common shared memory space.
- **Thread-level parallelism** - code parallelisation methodology using light-weight *threads* (or *strands* of operation) to execute sections of the code concurrently. Technologies such as OpenMP are commonly used to achieve thread-level parallelism.
- **Transdermal** - administration of pharmaceutics through the skin.
- **Volumetric loading** - the volume of a drug relative to the volume of the device. A more suitable description for the amount of drug for the case of spatial simulations, as the mass loading does not give information on the amount of space consumed by the drug.

Appendix D

List of publications

1. **Perrin, D., Bezbradica, M., Crane, M., Ruskin, H. J. and Duhamel, C. High-Performance Computing for Data Analytics, *2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, pp.234 - 242, Dublin, October 2012.**

   URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6365069`

   Abstract One of the main challenges in data analytics is that discovering structures and patterns in complex datasets is a computer-intensive task. Recent advances in high-performance computing provide part of the solution. Multicore systems are now more affordable and more accessible. In this paper, we investigate how this can be used to develop more advanced methods for data analytics. We focus on two specific areas: model-driven analysis and data mining using optimisation techniques.

2. **Bezbradica, M., Crane, M., and Ruskin, H. J. Parallelisation strategies for large scale cellular automata frameworks in pharmaceutical modelling, *2012 International Conference on High Performance Computing and Simulation (HPCS)*, pp.223 - 230, Madrid, July 2012.**

   URL: `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6266916`

   - Selected among best six papers of HPCS 2012. Nominated for the HPCS 2012 Outstanding Paper Award

Abstract Cellular Automata (CA) properties facilitate the detail required for the bottom-up approach to modelling and simulation of a broad range of physico-chemical reactions. In pharmaceutical applications, CA models use a combination of discrete-event rules based on probabilistic distributions and fundamental physical laws to predict the behaviour of active substances (drug molecules) and structural changes in Drug Dissolution Systems (DDS) over time. Several models of this type have been described so far in the scientific literature. Yet practical applications are lacking in the context of large-scale, high-precision, high-fidelity simulations. The key obstacle to parallelisation of such models is not only the amount of data involved, but also the fact that many of these models incorporate agent-like behaviour within the CA framework in order to describe pharmaceutical components. This makes communication across process boundaries expensive. In this paper, we apply different parallelisation strategies to a large scale CA framework, used to model coated drug spheres. We use two parallel-computing application programming interfaces (APIs), namely OpenMP and MPI, to partition the simulation space. We analyse the applicability of each API to the problem individually, as well as in the hybrid solution. We examine speedup potential and overhead for local and global communication for simulation speed and solution scalability. For these types of problems, our results show that performance is much improved for appropriate combinations of parallelisation solutions.

3. **Bezbradica, M., Ruskin, H. J., and Crane, M. Applications of High Performance Algorithms to Large Scale Cellular Automata Frameworks used in Pharmaceutical Modelling. Submitted by invitation for journal *Concurrency and Computation: Practice & Experience*. Expected to appear in the 4th Quarter, 2013 (Tentative).**

Abstract Background: Cellular automata (CA) frameworks have been gaining momentum as a promising tool in the field of *in-silico* Drug Dissolution System (DDS) modelling. This is due to their inherent properties which facilitate the detail required for a bottom-up approach to modelling as well as their ability to support simulations of a broad range of physico-chemical reactions. In pharmaceutical applications, CA

models use a combination of discrete-event rules, based on probabilistic distributions and fundamental physical laws, in order to predict the behaviour of active substances and structural changes in a DDS over time.

Several models of this type have been successfully described so far in the scientific literature. Nevertheless, practical applications are lacking in the context of large-scale, high-precision, high-fidelity simulations requiring significant amount of data and computing resources to execute. In the present high performance computing era, such resources are readily available, but algorithmic solutions in the field tend to lag behind. The key obstacle to parallelisation of such models is not only the amount of data involved, but also the fact that many of these models incorporate agent-like behaviour of pharmaceutical components within the CA framework, which makes communication across process boundaries computationally expensive.

Methods: In this paper, we consider different parallelisation strategies for a CA framework used in modelling coated drug spheres. Here, one of the main factors influencing drug dissolution is sensitivity to the physical thickness of the spherical coating. This is orders of magnitude smaller compared to the core sphere of drug, so that large 3D models are needed to accommodate the measurement range. Two parallel-computing application programming interfaces (APIs), namely OpenMP and MPI, are used to partition the simulation space.

Results: We analyse individual applicability of each API to the problem, as well as in the hybrid form. Speed-up potential and overhead for local and global communication are also explored for simulation execution time and solution scalability. For these types of problems, results indicate that performance can be greatly improved for appropriate combinations of parallelisation strategies. This finding has important implications for the scope of such models, enabling incorporation of diverse interactions at small-scale within the global framework.

4. **Bezbradica, M., Ruskin, H. J., and Crane, M. Modelling Drug Coatings: A parallel cellular automata model of ethylcellulose-coated microspheres,** ***Proceedings of The International Conference on Bioscience, Biochemistry***

***and Bioinformatics, ICBBB 2011* 5, pp.419 - 424, Singapore, February 2011.**

URL: `http://www.etlibrary.org/?m=fbook&a=details&aid=2923`

Abstract Pharmaceutical companies today face a growing demand for more complex drug designs. In the past few decades, a number of probabilistic models have been developed, for the purpose of giving a better insight into the microscopic features of these complex designs. Of particular interest are those models that simulate controlled release systems to provide targeted dose delivery. Controlled release is achieved by using polymers with different dissolution characteristics. We present here a model of a drug delivery system based on Monte Carlo methods, as a framework for Cellular Automata mobility, and show a solution for model enhancement through parallelisation. The objective is the high performance simulation of targeted drug release in the gastro-intestinal tract, from a capsule composed of ethylcellulose coated microspheres. The overall aim is to understand the importance of various molecular effects with respect to system evolution over time. Important underlying mechanisms of the process, such as erosion and diffusion, are described.

5. **Bezbradica, M., Ruskin, H. J. and Crane, M. Probabilistic models for dissolution of ethylcellulose coated microspheres, in *Proceedings of The European Simulation and Modelling conference, ESM'2010*, pp.408 - 412, Hasselt, Belgium, October, 2010.**

   The book edited by Janssens, G. K., Ramaekers, K. and Caris, A. ISBN 978-90-77381-57-1

   Abstract In the last few decades, a number of probabilistic models for drug delivery have been developed. Of particular interest are those that model controlled release systems to provide targeted dose delivery. Controlled release is achieved by using polymers with different dissolution characteristics. We present here a model based on Monte Carlo and Cellular Automata approaches, for simulating drug release from coated microspheres in the gastro-intestinal tract. Controlled release is obtained using ethylcellulose as the coating polymer. Modelling features, such as the drug and

coating dissolution are nontrivial, since material is non-homogenously dispersed and the dissolution exhibits complex behaviour. Important underlying mechanisms of the process, such as erosion, are described here.

6. **Bezbradica M., Crane M. and Ruskin H. J. Probabilistic modelling of drug dissolution, *ERCIM news*, 84, pp.42 - 43. January, 2011.**

   URL: `http://ercim-news.ercim.eu/en84/research-and-innovation/`

   Abstract The project at Dublin City University aims to establish new computational models for drug dissolution systems, specifically for sustained release drugs used in the gastro-intestinal tract. It seeks to improve existing algorithms and develop improved ones to predict the release curves of active substances. High-performance computing and cellular automata are used to investigate probabilistic drug dissolution phenomena. The project is an ongoing collaboration with industrial partner, Sigmoid Pharma Ltd.