The University Of Sheffield.

Modelling actin polymerisation and phagocytosis

James Edward Bradford

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Department of Physics and Astronomy
Faculty of Science
The University of Sheffield

February 2021

Abstract

Actin is a biopolymer abundant in most eukaryotic cells. Actin possesses the ability to organise into many different and complex architectures, providing it with the mechanical properties needed to suit various functions. Actin filaments can generate forces through polymerisation making it a fascinating protein to study from a physics perspective. In this thesis we present work done in modelling actin filaments, and some of the processes it can undergo. We first present our 2D stochastic simulation of actin polymerisation, which uses both Monte Carlo and Brownian dynamics methods. We developed this to investigate actin networks, and actin force generation. We then use this tool coupled with analysis derived from our knowledge of actin, to investigate *in vitro* experimental actin systems. In doing this we were able to uncover details surrounding the effects of actin related proteins. We then apply our tool to investigate phagocytosis. Phagocytosis is a process whereby large particles are internalised by specialised cells known as phagocytes. This process forms a crucial component of the innate immune response to infection, as phagocytes form the first line of defence against invading pathogens. Actin is known to be necessary to generate forces on the cell membrane, deforming it to engulf the target. However some of the details surrounding the type and geometry of actin network necessary remain unclear. In our work into phagocytosis we aimed to address this by using our simulations to model actin networks generating forces on a membrane to engulf a simple target. Finally we present work done in modelling infections at the level of the immune system rather than at the single cell level. Here we analyse experimental data with the help of some mathematical modelling and simulations to investigate the growth dynamics of pathogens in the presence of phagocytes.

Acknowledgements

There are many people I'd like to thank. People that believed in me even when I doubted myself. People that made a little difference everyday.

First of all my supervisor Rhoda Hawkins. Her guidance throughout the PhD journey has been unwavering. Her support to help me keep going when I doubted myself has been incredible. For pushing me to think differently. She has made me into the physicist I am and helped change the way I think about the world. Thank you.

My second supervisor Simon Johnston. For introducing me to the wonders of biology, for his company, good humour and support. Thank you.

My office mates and fellow PhD students. Tash, for being someone I can chat to, sometimes moan to. She is one of the smartest people I will ever know, but she doesn't know it! Mon for making me smile. Jon for always being in an uplifting mood, for being so friendly. Helen for being a good friend and for making me laugh. Ian, for being someone to look up to and for being a great help at the start. Thank you all.

Being cross-disciplinary means I also get to make friends with biologists too! Kate for being welcoming and for her humour. Jaime for being so passionate about phagocytosis and his work, and for wanting to learn about physics. Everyone in the D31 lab who was welcoming to me, there are too many to name. Thank you.

Ellen Allwood and John Palmer, who I had the pleasure of working with. Both very patient and willing to listen to what us physicists have to offer. Thank you.

My many housemates over the years. In particular Roy and Charlotte, for making me laugh and putting up with me, for making a little difference everyday. Thank you.

My girlfriend Harriet, for believing in me. Thank you.

My family. My mother and sister for always being there when I need. My niece Violet for bringing much joy into my life. Thank you.

I also thank the University of Sheffield for my funding, and everyone who I ever worked with whilst teaching undergraduate maths, especially Matthew Mears.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter we will introduce the topic of research and any necessary background. We will first cover the basics of the cytoskeleton, actin and phagocytosis. We will then discuss recent modelling work relevant to our work presented in this thesis.

1.1 The actin cytoskeleton

1.1.1 Cytoskeleton overview

The cytoskeleton is a complex and dynamic system of protein filaments in the cytoplasm of all cells. The cytoskeleton is responsible for giving cells their shape, mechanical properties and the ability to undergo certain phenomena like motility and mitosis. In eukaryotic cells the cytoskeleton is made up of three components. These are actin filaments, microtubules and intermediate filaments [1].

Intermediate filaments are a group of protein filaments that are responsible for providing mechanical support to the cell and the nucleus.

Microtubules (MTs) are biopolymers made up of the protein tubulin. One microtubule is made up of many proto-filaments which form a long hollow structure. Microtubules can grow relatively long with lengths of tens of microns. They are responsible for positioning organelles in the cell and directing intracellular transport. Molecular motors such as dynein and kinesin are proteins that bind to MTs and move along them with a stepping motion. These molecular motors can transport cargo across the cell [2].

Our focus in this research project is the third component of the cytoskeleton, actin filaments. Actin filaments are responsible for a wide variety of cellular processes such as shape changes, motility and endocytosis (transporting external substances into the cell). To carry out these functions actin can organise into many different types of architectures that we will cover in this section.

1.1.2 Actin introduction

Actin is a globular protein which can exist in two different forms, the monomeric form known as Globular-actin (G-actin) and the polymeric form known as Filamentous-actin (F-actin). F-actin is a biopolymer and is semi-flexible on the length scale of the cell, with a persistence length l_p of around 17µm [3]. The persistence length is an intrinsic mechanical property of a polymer, and is basically the length over which it appears straight due to Brownian forces alone [4, 5]. Filaments shorter than l_p behave as rigid polymers whereas filaments longer than l_p bend due to thermal fluctuations alone. One might therefore assume that on the length scale of the cell all actin filaments can be treated as straight rigid rods, however other external forces can be acting on actin inside the cell, causing filaments to bend on scales much shorter than their persistence length.

G-actin molecules come together to form filaments in a process called nucleation. It is typically believed that three G-actin molecules come together to form an F-actin filament, which is therefore a trimer. Once two G-actin molecules have bound together forming a dimer, a third one is needed quickly as the dimer is unstable and readily dissociates back into G-actin. This process of nucleation can therefore be slow and is usually the rate-limiting step in the growth of F-actin networks.

Once nucleated the F-actin seed can then elongate by a process called polymerisation. In polymerisation single G-actin monomers bind to either end of the F-actin filament. F-actin filaments are polar, and the rate of individual polymerisation events differs between the two ends. The polymerisation rates at both ends are linearly dependent on the G-actin concentration of the medium the filament is in. The two ends are named 'barbed' and 'pointed'. The barbed end is the more dynamic end and polymerises around ten times faster than the pointed end [6].

The molecule ATP (adenosine triphosphate) provides energy to many different processes in the cell. This is achieved by ATP undergoing hydrolysis to ADP (adenosine diphosphate) and phosphate, this reaction releases energy. Each G-actin molecule can carry with it an ATP molecule which becomes hydrolysed after nucleation or polymerisation. When G-actin binds to a F-actin filament end the ATP is not immediately hydrolysed. This usually results in the monomers at the ends of the filaments being ATP-actin, whereas monomers in the middle are ADP-actin. [1].

The opposite of polymerisation, depolymerisation, can also occur. In depolymerisation G-actin monomers that have been polymerised into an F-actin filament can fall off at either end. Depolymerisation rates are flat rates, independent of local G-actin concentration [1].

As polymerisation is dependent on local G-actin concentration and depolymerisation is not, there exists a 'critical concentration' where the rates of polymerisation and depolymerisation are equal. In mathematical form this is

$$C_G^* = \frac{k_d^{\text{point}} + k_d^{\text{barb}}}{k_p^{\text{point}} + k_p^{\text{barb}}}, \tag{1.1}$$

where C_G^* is the critical concentration of G-actin, k_d^{point} and k_d^{barb} are the depolymerisation rates for the pointed and barbed end of actin respectively, and k_p^{point} and k_p^{barb} are the polymerisation rates for the pointed and barbed end respectively. At this concentration the length of any F-actin filaments remains constant and the system has reached a steady state. As typically $(k_p^{\text{barb}} - k_d^{\text{barb}}) > (k_p^{\text{point}} - k_d^{\text{point}})$, F-actin filaments treadmill at the critical concentration. Treadmilling is when the filaments grow at one end but shrink at the other, but such that they remain constant in length [1].

1.1.3 Actin binding proteins

What makes actin interesting goes beyond simple nucleation and elongation. Actin is an interesting protein because it has many associated accessory proteins that can interact with it and modulate its structure and its properties. We will describe the changes some of these proteins make with our focus being more on the mechanics rather than the biochemistry. It is however worth pointing out that there is an entire zoo of many actin-related proteins that affect almost every aspect of actin's dynamics and mechanics [7–9].

In Figure 1.1 we have highlighted some of the key processes F-actin can undergo by its interaction with accessory proteins, these are described below.

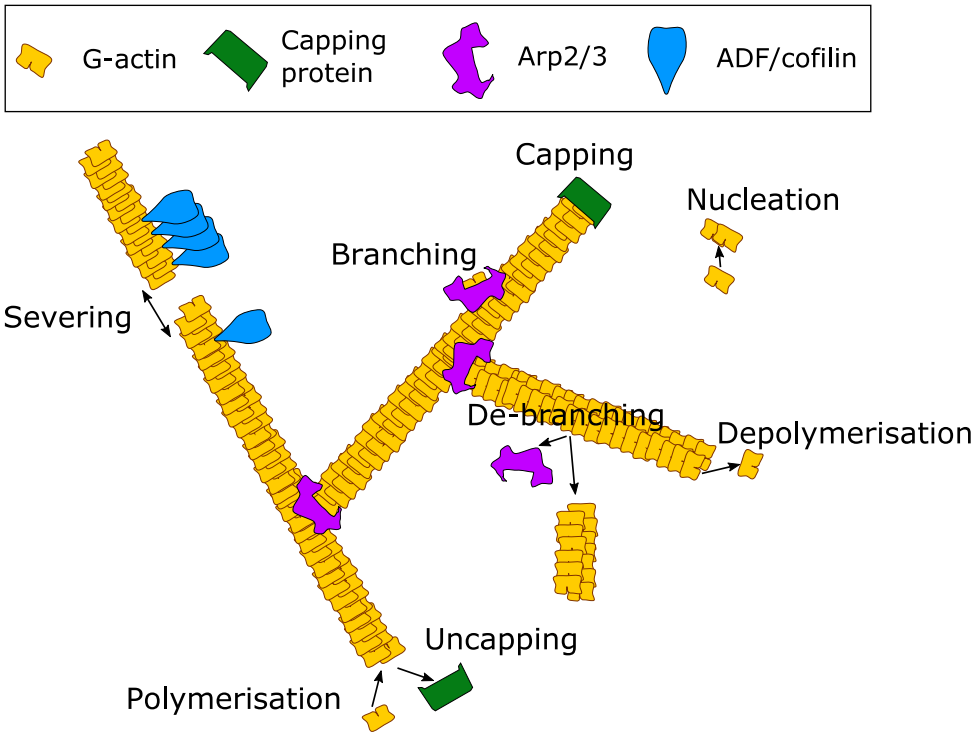


FIGURE 1.1: Cartoon highlighting some of the processes F-actin can undergo. F-actin filaments are made up of G-actin molecules (individual yellow shapes), that come together to nucleate a filament (top right) and then are added onto filaments to elongate them (polymerisation, bottom). G-actin molecules can fall off at filament ends (depolymerisation, right). Capping proteins (green shapes) can block filament ends from undergoing polymerisation and depolymerisation, and can bind (capping) and unbind (uncapping) to/from filament ends (top right, bottom respectively). Branches can be nucleated from the Arp2/3 complex (purple shape) that binds to the sides of F-actin filaments. Branches can disconnect from their mother, producing a free filament (debranching, middle). F-actin filaments can be broken, in a process called severing (top left), mediated by certain proteins such as ADF/cofilin (blue shape).

Arguably the most striking process F-actin can undergo is branching. This involves a complex of several proteins called Arp2/3 complex. Arp2/3 complex must first be activated by being bound by a nucleation promotion factor (NPF). Many NPFs exist and include Wiskott-Aldrich Syndrome Protein (WASP) and Las17 (which we cover in Chapter 3). Arp2/3 complex then binds to one (or possibly two) G-actin monomers [10–12] before binding to the side of a pre-existing F-actin filament [13]. The language of 'mother filament' and 'daughter filament' is used when referring to the pre-existing filament and the branch respectively. G-actin monomers can then polymerise onto the barbed end of this seed. The pointed end of the branch cannot polymerise or depolymerise, it is 'capped' (which we will cover later). Due to the orientation of Arp2/3 complex once attached to F-actin, the branch always grows at an angle of $\sim 70^{\circ}$ from the filament it is bound to [10, 14]. Since branches are simply F-actin filaments bound to the side of other F-actin filaments, impressive branched networks can quickly form in this dendritic way. Branches are 'metastable' and therefore can detach from their mother, this process can be promoted by certain proteins [15, 16]. Branching and de-branching are shown in Figure 1.1, with the Arp2/3 complex shown in purple.

Filament ends can be capped by a wide range of capping proteins. The capping protein simply binds to one of the ends of F-actin and blocks further polymerisation and depolymerisation. Capping can therefore drastically affect F-actin dynamics, even with low concentrations of capping protein [1]. Uncapping can also occur, where the bound capped protein falls off the end of the F-actin filament, allowing it to resume (de)polymerisation. Capping and uncapping are shown in our cartoon (Figure 1.1), with capping protein shown in green.

F-actin filaments can be broken into two filaments by a process called severing. Severing is promoted by certain proteins such as ADF/cofilin and can quickly dissemble large, branched actin networks [17]. ADF/cofilin decorates actin filaments and, given a high enough concentration on a filament, can reduce the persistence length of the local region down to as little as $\sim 2\mu m$, increasing its flexibility [18]. This therefore creates a boundary on the filament between this lower persistence length and the typical persistence length of actin. It is at these discontinuous flexibility boundaries that severing takes place [19]. A severing event is shown in Figure 1.1, with ADF/cofilin shown as a blue shape.

F-actin filaments can also join together forming a bundle of parallel (or anti-parallel) filaments. Parallel bundles of actin are structures made up of aligned actin filaments all pointing in the same direction. These are held together by proteins which bind together pairs of filaments. Examples of actin-bundling proteins are fascin, fimbrin and α-actinin [20].

Similar to bundles, F-actin filaments can form crosslinks together. Crosslinked actin filaments are connected together but are not orientated parallel/anti-parallel like a bundle, but rather are crossed over each other. The process of crosslinking filaments together is typically believed to be similar to bundle formation, but involve longer crosslinking proteins to form the structures seen. Also, the polymerisation dynamics are thought to also play a role in whether F-actin (in the presence of crosslinking proteins) forms bundles or crosslinks. Conditions promoting faster polymerisation has been suggested to favour crosslink formation over bundling [21].

1.1.4 In the cell

Different parts of the cell which are responsible for different functions, have vastly different structural organisations of actin. In cell motility, actin arranges itself into

a dense branched network at the leading edge, called the lamellipodium. Lamellipodia are thin, sheet like projections which help move the cell forward and are driven by branched actin polymerisation [22, 23]. Also important in cell motility are filopodia. Filopodia are thin, long, finger projections extending outwards from the cell. In filopodia, actin is arranged as unbranched bundles pointing outwards towards the membrane. They are responsible for sensing the local environment around the cell, migration and particle/pathogen capture. They are created by F-actin bundles polymerising against the membrane [24].

The cell cortex is a thin and crosslinked actin network that is bound to the membrane and is present in most animal cells [25]. The cortex's main function is the control of cell shape changes. It is a highly dynamic network, constantly undergoing turnover in order to regulate the material properties of the cell [26].

1.2 The innate immune system

Multicellular organisms need to be able to defend themselves against infection. We are under near constant attack from harmful pathogens including types of bacteria, fungi, viruses and parasites. Riding to our defence is the immune system. In vertebrates the immune system is made up of two groups; the innate immune system and the adaptive (or acquired) immune system.

The adaptive immune system involves acquired long-term immunity from specific pathogens. Two main cell types are important in adaptive immunity; T cells and B cells. B cells secrete antibodies which can interfere with pathogens, whereas T cells interact with pathogens, and can kill them directly. The adaptive immune system forms the basis of vaccination; providing in some cases lifelong immunity against certain pathogens and eradicating certain diseases [1].

However we are more interested in the second group; the innate immune system. The innate immune system is less sophisticated in some ways but still fascinating. It is more widespread in nature than the adaptive immune system, present in insects, fungi, plants and simple multicellular organisms [27]. Elements of the innate immune system include the skin, tears and mucus.

Specialised cells known as phagocytes form a crucial component of the innate immune response (they also play a role in the adaptive immune response). Phagocytes are cells that are able to carry out phagocytosis; a process whereby large particles (typically greater in diameter than 500nm) are internalised [28–30]. Examples of phagocytes include macrophages, neutrophils and dendritic cells. In the innate immune system pathogens are hunted, internalised and subsequently destroyed by phagocytes acting as the first line of defence against infection. Phagocytes also remove cellular debris. In humans, around 10 billion cells die each day [29], creating apoptotic bodies that need to be cleared by phagocytosis [31]. The different stages of phagocytosis are shown in Figure 1.2 and are explained below.

The first stage of phagocytosis is receptor engagement. Special receptors are present on the membrane of phagocytes and these engage with a large variety of related ligands on many different pathogen targets [29, 30]. Once receptors have bound to ligands on the target a complex biochemical signalling cascade is initiated. This triggers membrane remodelling and reorganisation of the underlying actin cytoskeleton. As many different receptor-ligand pairs exist, so do many different resultant biochemical signalling pathways, sometimes during a phagocytic event many can be working together in parallel. The broad result of these pathways and the one that takes our interest, is the promotion of actin polymerisation [29].

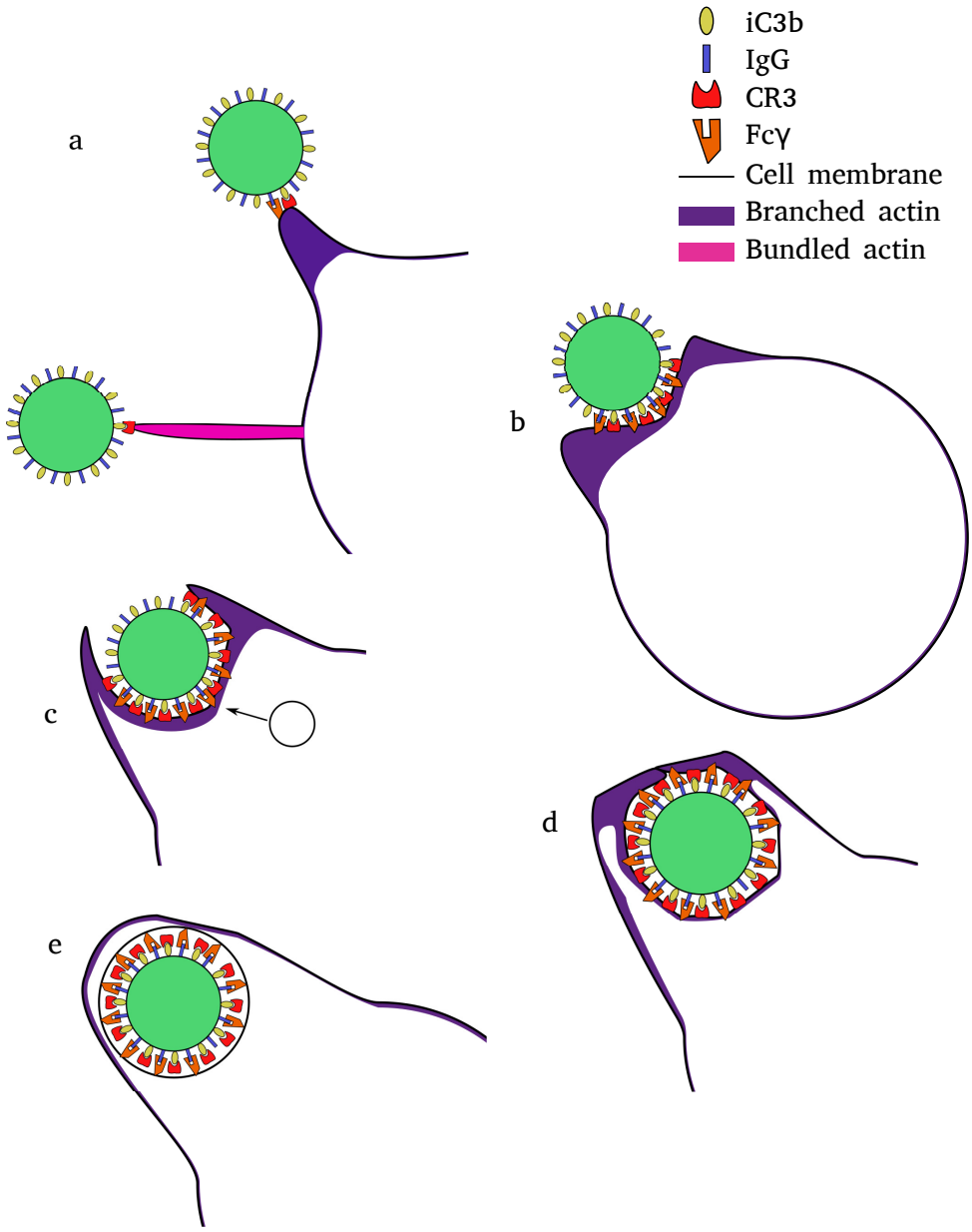


FIGURE 1.2: Different stages of phagocytosis. The phagocyte, shown with a black outline internalises the pathogen, shown in green. Two different examples of receptors are shown; Fcγ (orange shape) and CR3 (red shape), along with their cognate ligands on the target; IgG (blue shape) and iC3b (yellow shape). What is typically thought to be areas of branched and bundled actin are shown in purple and pink respectively. **(a)** Receptor engagement, the cell physically contacts the target and there is binding of receptors to ligands. **(b)** Formation of the phagocytic cup. **(c)** The cup is extended outwards, surround the target, extra membrane may be required, its delivery through exocytosis is shown. **(d)** Completion of engulfment, the membrane fuses at the apex of the target. **(e)** The target is internalised and the actin cleared.

In this early stage of phagocytosis actin related proteins such as Arp2/3 complex and formins (proteins that promote nucleation) are recruited to, and activated in, the region just below the bound target. Phagocytosis is heavily reliant on the presence of actin. Phagocytes which have been treated with actin inhibitors severely reduces uptake of targets [32]. Similarly phagocytes from patients suffering with Wiskott Aldritch Syndrome, a genetic disorder which affects an NPF of Arp2/3 complex, have impaired phagocytic function [33].

After receptor engagement, the next stage is formation of the phagocytic cup and then pushing this around the target. Actin polymerisation is the major component in force generation in phagocytosis, being key in both these processes.

Finally the membrane completes its journey around the target and fuses (joins together) at the apex. Thereby internalising the target by wrapping it. The target is contained within the cell within a vesicle called a phagosome. The cell can then deal with the target by fusing the phagosome with other vesicles containing acids and enzymes to ultimately destroy whatever it has internalised [29].

Phagocytosis is one form of a group of processes called endocytosis. Endocytosis is broad term simply referring to the internalisation of extracellular substances. As well as phagocytosis (the internalisation of large particles), there is also macropinocytosis (the uptake of fluid), and receptor-mediated endocytosis. The latter shares some similarities with phagocytosis. Both are reliant on receptors but receptor-mediated endocytosis tends to be a more passive process, with the membrane invaginating inwards and then pinching off, rather than protruding outwards as with phagocytosis [34].

It should be noted that phagocytosis comes in different flavours. We previously mentioned that many different receptor-ligand pairs exist and two examples are shown in Figure 1.2. Fcγ receptor mediated phagocytosis is the most well studied and is characterised by the phagocytic cup protruding outwards around the target. Another type of phagocytosis is complement receptor mediated phagocytosis. Here phagocytosis appears to be mechanistically similar to receptor-mediated endocytosis, with a membrane invagination and the target 'sinking' into the cell, rather than a phagocytic cup extending outwards to wrap it [35–37]. The work presented in this thesis concerns the more characteristic example of phagocytosis; Fcγ receptor phagocytosis.

1.2.1 Actin regulation in phagocytosis

From a physicist's perspective perhaps the most interesting part of phagocytosis is the actin force generation and the formation and extension of the cup around the target. Experimental methods which identify the localisation of actin and actin related proteins during phagocytosis have been developed to investigate actin's role.

Imaging of fluorescent actin in phagocytosis shows a build up of actin in the advancing cup which rapidly clears upon cup closure [38]. This build up of actin is co-localised with Arp2/3 complex [39] as an advancing ring that is present at the tips of the phagocytic cup [40]. WASP was discussed earlier as an activator of Arp2/3 complex. WASP is activated by a protein called Cdc42 [29]. Cdc42 has been also shown to be present only at the tip of the advancing phagocytic cup [38], suggesting that Arp2/3 complex is activated at the edge of the cup. Anti-capping (protection from capping) has been suggested to be important in actin during phagocytosis. Ena/Vasp proteins, which have been suggested to be anti-cappers [41–43], have been shown to be recruited to the cup. Phagocytic uptake has also been shown to be impaired upon their inhibition [44, 45].

Formins, which are involved in nucleation and enhancing polymerisation of actin, have been shown to be localised in phagocytic cups in complement receptor phagocytosis [37, 46, 47].

Depolymerisation of actin is also seen at the base of the phagocytic cup, where cofilin (a severing protein) is thought to play a role [48]. Cofilin is inhibited by the membrane bound phospholipid PIP2, which itself is depleted at the cup's base [29, 49]. This suggests that cofilin is inhibited only at the tip but free to sever actin at the base. Other depolymerisation promoting proteins are also thought to be present at the base [37, 48], which would be responsible for the clearance of actin at the base that has been reported [38].

We aim to complement this experimental work and investigate the relative importance of the roles of these regulatory proteins in this research project. We can probe what type and geometry of actin network is necessary.

1.3 Modelling the actin cytoskeleton

In this section we will provide information on common techniques used to model actin, followed by key examples of recent work in the literature.

1.3.1 Semi-flexible polymer physics

In section 1.1.2 we stated that actin is a semi-flexible polymer on the length scale of the cell. Semi-flexible polymers can be described well by the worm-like chain model [50]. In the worm like chain model the bending energy of a polymer is

$$U^b = \frac{\kappa_b}{2} \int_0^L \left(\frac{\partial^2 \boldsymbol{r}(s)}{\partial s^2} \right)^2 ds, \tag{1.2}$$

where κ_b is the bending modulus or bending rigidity of the polymer, L is the length of the polymer. $\boldsymbol{r}(s)$ is a position vector along the polymer and depends on s which is the arc length coordinate along the polymer and is what is integrated over. For more detail on the worm-like chain model a good resource is [5] and references therein.

The worm-like chain model can be implemented in a discrete way in a chain of points connected by links (typically springs or distance constraints). One way to solve for the motion of these points and links is through a method called 'overdamped Langevin dynamics', which goes by another name, 'Brownian dynamics'. In chapter 2 we will describe the Brownian dynamics method we use, so we will not provide much detail here, a good resource is [51] and references therein. It is important to highlight the 'overdamped' property of Brownian dynamics. This means that the inertial term in Langevin dynamics is neglected. To explain why this is valid we can introduce something called the Reynolds number.

The Reynolds number is defined as

$$\mathrm{Re} = \frac{\rho u L}{\eta}, \tag{1.3}$$

where ρ is the density of the given fluid ($\mathrm{kg\,m^{-3}}$), u is the flow speed of the fluid ($\mathrm{m\,s^{-1}}$), L is a characteristic length (m), and η is the (dynamic) viscosity of the fluid (Pa s). The systems we will cover in this thesis all happen in the limit of low Reynolds number ($\mathrm{Re} \ll 1$), as the length scales are very small. In this limit, viscous

forces dominate over inertial forces [52]. Therefore we can neglect the inertial term in Langevin dynamics.

1.3.2 Actin force generation

In chapters 2 and 4 we will cover work concerning force generation by actin polymerisation. Actin filaments can exert force on objects through polymerisation. For example, in section 1.1.4 we introduced lamellipodia, and that they are driven by actin polymerisation, but how is this actually carried out?

The Elastic Brownian Ratchet model [53, 54] is a model that describes how actin polymerisation can exert forces and push objects or barriers such as membranes. It was built on the original Brownian Ratchet model [55]. In the original model, actin filaments are arranged perpendicular to a barrier, which is described with a load force f_{bar}. The barrier is undergoing thermal fluctuations, and these are large enough that the gaps that open up between the barrier and the filament ends are such that polymerisation can occur. Once polymerisation has occured the barrier can then not move back to its original position. The velocity of the barrier is

$$v_{\text{bar}} = l_{\text{mon}} \left(k_p C_G e^{\frac{-f_{\text{bar}} l_{\text{mon}}}{k_B T}} - k_d \right), \tag{1.4}$$

where l_{mon} is the size of a G-actin monomer, and so is the move increment of a single polymerisation event. k_p and k_d are the polymerisation and depolymerisation rates of the actin filament end in contact with the barrier, C_G is the local G-actin concentration. If we set the velocity to equal zero, we can rearrange for the barrier's load force which is the stall force of a single actin filament,

$$f_{\text{stall}} = \frac{k_B T}{l_{\text{mon}}} \ln \left(\frac{k_p C_G}{k_d} \right). \tag{1.5}$$

Using typical values for parameters in equation 1.5 shows that actin filaments are capable of exerting around 7-9pN of force.

The Elastic Brownian Ratchet model built on this simple model by considering thermal fluctuations of the actin in addition to the barrier and by considering filaments orientated non-perpendicular to the barrier [53]. Actin filaments can arrange themselves in different configurations in order to exert force. In some cases this can amplify the force exerted beyond the stall force, however at a 'cost' to displacement (since energy has to be conserved) [56].

1.3.3 Actin computational models

In section 1.3.1 we introduced Brownian dynamics as a way to represent worm-like chains. This technique is therefore popular for simulating actin filaments [57, 58]. An extensive tool that uses Brownian dynamics and is open for use by the scientific community is Cytosim [59]. Cytosim has been used to investigate lots of different actin based phenomena [60–63].

Monte Carlo simulations are also popular for modelling actin. Monte Carlo (MC) simulations are stochastic simulations that use repeated sampling of random numbers to determine numerical results. They have been used to investigate the Elastic Brownian Ratchet model previously described, confirming the velocity of the wall decays exponentially with increasing load. A 'pushing catastrophe' was also explored, which is the phenomenon where the actin filaments start to bend at a certain stage

of growth and grow parallel to the surface, regardless of their initial angle of incidence [64]. MC simulations have been used to study branched actin networks growing against obstacles too [65, 66].

We want to highlight two pieces of work concerning actin filaments in lamellipodia protrusion as the techniques used are relevant to our work presented in this thesis. The first piece of work used a two dimensional MC model of a branched actin network growing under and exerting force on a model plasma membrane. The model included a zone of protection against capping situated within 5.4nm of the membrane. This resulted in branched filaments being orientated in the range of -35° to +35° to the membrane, as capping is faster than branching for filaments growing in other directions. The membrane moved with a speed $8\mu m\,min^{-1}$ and could change direction by 15° in around one minute after ∼12 generations of successive branching [67].

In a similar setup, lamellipod protrusion was investigated by using 3D Brownian dynamics simulations of rigid branched actin filaments growing and diffusing inside a box. The box represented the lamellipod with a load force on the 'front' wall pushing in and a smaller vertical force on the top wall pushing down, aiming to model surface tension. This work found branching to be important in generating forces, leading to decreased packing efficiency and a higher rate of protrusion than unbranched systems [68].

In chapter 2 we present the actin computational model we have developed, which is influenced from the ones cited here.

1.4 Modelling phagocytosis

Modelling phagocytosis is a fairly new area of study that has gained in popularity over the last decade or so. As discussed in section 1.2 phagocytosis is a complex process with many different components to consider. It is important then, that models of phagocytosis focus on as few ingredients as possible to maximise what can be learned. In this section we will highlight previous models which can broadly be split into the two following categories.

1.4.1 Receptor based models

There have been many recent models focusing on just the receptor binding component of phagocytosis, the adhesion between the membrane and the target. One early model reduces the system down to one dimension, where phagocytic receptors can diffuse along a one dimensional line representing the membrane. As new receptors move into the cup region from outside, they bind to the target pulling the cup up around it. This work was however modelling receptor mediated endocytosis rather than phagocytosis, but its core principles are transferable to phagocytosis. It found an optimal target size for engulfment time. This was explainable as small targets are highly curved and so the membrane must more tightly bend around them to form a closed cup. Larger targets simply need more membrane to wrap [69].

In work that followed, this simple model was expanded on to focus on phagocytosis, in particular the reported two stage process of engulfment from experiments of neutrophil phagocytosis. The two stages of engulfment reported were a relatively slow first stage followed by a much faster second stage with the transition between the two being at around half-engulfment. The passive diffusion of receptors from [69] was a good fit to the first slow stage of engulfment. The second stage required an 'active motion' for receptors, where receptors were driven by a constant velocity towards the cup edge. The model also included a somewhat crude role for signalling,

which initiates this second stage of receptor motion [70]. This model was extended to investigate the effects of different target shapes and their orientation with respect to the membrane [34].

Another receptor based model aimed to explain the observations that phagocytic cups seem to either stall before half-engulfment or continue to successfully engulf the entire target particle. A force-balance equation considering a pushing force which is proportional to a term representing the local concentration of F-actin, and a restoring force due to the membrane's resistance to shape change, was the basis of this work. This model was able to replicate the variation in progression and the presence of the observed 'mechanical bottleneck' at half engulfment [71].

1.4.2 Force and energy based models

Modelling phagocytosis using a mechanistic approach has also been addressed somewhat. Herant et al. developed models focusing on the forces required to explain the cup shape. These investigations suggest two interactions are required for successful phagocytosis. First, a repulsion between the cytoskeleton and the membrane at the cup edge that drives protrusion of the cup around the target. Second, an attractive, flattening force within the cup [72, 73].

Energy based models often use the famous Helfrich bending energy for membranes [74], this is

$$E_{\mathrm{b}} = \int_A \left(\frac{\kappa_b}{2} \left(K - c_0 \right)^2 + \bar{\kappa} K_G \right) dA. \tag{1.6}$$

where κ_b and $\bar{\kappa}$ are bending rigidities, K is the extrinsic curvature and K_G the Gaussian curvature, c_0 is the spontaneous curvature. The integration is over the surface area A of the membrane, for more information on this see [75].

This Helfrich bending energy has been used to model the cup shape by adding other terms to the energy like tension or terms considering receptors. One such model examined the effect these membrane parameters have on the shape of phagocytic cups. The work suggested that thin phagocytic cups require both a low surface tension and a strong constraint on the cell's volume [76].

1.4.3 Conclusion

Phagocytosis is a complicated process which therefore can be challenging to model. The brief review of previous work given here shows the necessary simplifications that have had to be made. These models have typically focused on receptors or the membrane. They have simplified or even neglected the cytoskeleton, ignoring the structure of the underlying network. They also tend to neglect the role of stochasticity.

We know that the cytoskeleton is very important in phagocytosis, as discussed in section 1.2. Our work aims to address these areas that have been lacking in modelling phagocytosis. We aim to complement these previous models by instead focusing on the underlying actin structure whilst simplifying other areas such as signalling and receptor dynamics. In chapter 4 we will cover our work into modelling phagocytosis.

We did not cover infection modelling in this introduction. Infection modelling aims to understand infections as a whole, by for example modelling populations of phagocytes and pathogens. We argue that these will be another useful technique in aiding our understanding of phagocytosis and infection in general. We will present some work into infection modelling in chapter 5.

Chapter 2

Simulation method

This chapter covers the technical details of the simulation model which I have developed during my PhD.

2.1 Introduction

One of the main goals of this PhD was to investigate actin based force generation in phagocytosis. To achieve this we built a computational tool from the ground up using aspects of previous tools and models described in the literature (covered in section 1.3). We built this to be fairly flexible and modular, allowing us to investigate actin dynamics as well as phagocytosis.

In this chapter we present our tool/model in detail. This is aimed to be quite general and some examples of phenomena we can investigate are given. In later chapters we will cover additional adjustments to the model we made to investigate specific phenomena. Phagocytosis related work is presented in chapter 4, and actin related work is presented in chapter 3.

2.1.1 Why develop our own?

Simulating the physics of actin is of course not a new area of study. We covered this area of study in chapter 1. At the start of this research project we looked at tools that we could use ourselves, the most extensive being Cytosim [59], which we covered in 1. However, Cytosim is suited to look mainly at filaments (e.g. actin and microtubules) and currently lacks the ability to include a membrane or membrane-like object. Recently Cytosim has been used to investigate endocytosis by representing the membrane invagination as an undeformable cylinder [61]. We wanted a tool that could look at actin filaments interacting with a deformable membrane.

There are some advantages to building a tool yourself, and some disadvantages.

Disadvantages include time taken to develop the basics which have been done before. We do not want to reinvent the wheel and have avoided this as much as possible, however some of this is inevitable. Another disadvantage is that it is arguably better for the community, which is largely cross-disciplinary, to have one unified tool that everyone can use, rather than lots of separate bits of code that each do one thing well.

Advantages include that there is no element (of interest) that is a 'black box', we understand the physics from close to first principles and can borrow methods used before from tools such as Cytosim. We have a thorough understanding of the code structure and therefore it is easy to build on the model, by adding components or improving areas. We argue that there should be some 'competition' between different tools, as there is often no single 'correct' way to model something.

I therefore chose to pursue my goal to build a tool myself.

2.1.2 Standard simulation methods

In this section we introduce three standard methods that are of relevance to this thesis. In section 2.2 we will present our method and explain the similarities and differences to the standard methods covered here.

2.1.2.1 Metropolis-Hastings algorithm

In general terms the Metropolis-Hastings algorithm is a Monte Carlo method that draws random samples from a probability distribution. It is a Markov process, meaning the next sample drawn depends only on the current sample [77]. At each iteration the next sample drawn is either accepted or rejected with some probability. If accepted, the algorithm proceeds to the next iteration with the new sample value. If rejected, the algorithm proceeds to the next iteration with the current sample.

Using this method for systems in physics usually concerns the Boltzmann distribution. Here the probabilities used to accept or reject a sample is based on the change in potential energy of the move. For example, let's consider the addition of one monomer on to the end of an actin filament. The probability of this move being accepted would be $\min(1, \exp(-\Delta E/k_{\mathrm{B}}T))$, given the energy cost of the polymerisation of ΔE. This is evaluated by drawing a random number $r \in [0, 1)$, if r is less than the probability the move is accepted, else it is rejected [78].

This example of an actin filament polymerising could be made complex by considering an obstacle in the way, perhaps another filament is blocking the polymerisation site. If we were to model the interactions between filaments as hard (in other words, no overlap between filaments is permitted) the polymerisation move would be rejected for any non-zero overlap.

The Metropolis-Hastings method is time independent and, strictly speaking, is valid for gaining static or equilibrium properties of systems. It is valid for systems in thermal equilibrium, since it is sampling an equilibrium distribution. When Monte Carlo algorithms like the Metropolis-Hastings method have been used to study time-dependent phenomena, difficulties arise when relating 'Monte Carlo time' (or steps) to real time [79]. For a method which is valid for dynamic systems see the next section 2.1.2.2, on the Gillespie (or dynamic Monte Carlo) method.

We borrow elements of the Metropolis-Hastings algorithm for simulating the actin polymerisation processes (described in chapter 2.3.4) but do not calculate potential energies, we instead use rates (as does the next method we describe in section 2.1.2.2). We use an approve-reject method for steric hindrance between filaments, described in section 2.3.3.

2.1.2.2 Gillespie algorithm

The Gillespie algorithm [80, 81] (also known as kinetic Monte Carlo or dynamic Monte Carlo) is a method used to simulate the time evolution of a system. The system is characterised by rate constants describing processes that can occur. For example in an actin system we could have the rate of barbed end polymerisation or the rate of capping.

Rather than using a fixed timestep to advance time and determine what processes have occurred in this time increment, the Gillespie algorithm determines the time at which the next process will occur. The timestep size is dynamic, and is a random variable sampled from a probability distribution, which takes into account stochasticity of the time between events. The timestep size is distributed according to the

exponential distribution from the assumption that the events are Poisson processes [80]. The derivation of this is shown in reference [82].

Once the time to the next process is determined, the process itself is chosen randomly based on the corresponding rate constants.

The Gillespie algorithm was originally developed for simulating the time evolution of chemical reactions [80], but as computer processing power has increased the algorithm is increasingly used to investigate more complex systems. A single simulation using the Gillespie algorithm gives one possible trajectory of the system, given the rate parameters used.

We use the Gillespie algorithm in work described in chapter 5, there we will discuss the technical details of the implementation.

2.1.2.3 Simulated dynamics

Simulated dynamics is a class of methods used to determine the motion of molecules or particles in a system. Simulated dynamics include molecular dynamics, Langevin dynamics and Brownian dynamics. Typically these methods use a fixed timestep length to advance time. The dynamics of the particles are determined by solving equations of motions taking into consideration forces acting on them in each timestep. Many different methods exist to do this such as the Euler method, Verlet method and Leap Frog method [78].

Molecular dynamics involves systems of moving molecules interacting through a interatomic potential (such as the well known Lennard-Jones potential) or a force field. Therefore every particle experiences a force due to all neighbouring particles. Simulations tend to be molecular scale or even atomistic, involving many particles but run for short timescales of picoseconds to microseconds [83].

Langevin dynamics is similar to molecular dynamics but includes frictional forces corresponding to a surrounding viscous medium and random forces, modelling Brownian forces. This allows Langevin dynamics to be used to model systems of larger lengthscales for longer timescales, as the solvent molecules are coarse grained out [51]. Hydrodynamical behaviour of the surrounding fluid is not typically captured by Langevin dynamics, and so momentum is not conserved [78].

Brownian dynamics is overdamped Langevin dynamics. The inertia component of Langevin dynamics is neglected given the assumption that the system is at low Reynolds number. Like a damped harmonic oscillator, once kicked it will relax back but not continue moving, nor vibrate back and forth. Each new kick is independent of the previous and so it is Markovian.

We use a Brownian dynamics method in the model described in this chapter, the technical details of our implementation is presented in section 2.3.2.3.

2.2 Overview of the model

The model should be considered a Monte Carlo model (similar to the Metropolis-Hastings and Gillespie methods described in section 2.1.2) since it involves the use of random number sampling. We also incorporate a Brownian dynamics method for the motion of actin filaments and membranes. The simulation code is written in C++, with visualisations using Python. We use third party libraries where available, these are Boost [84], Geometric tools [85] and Eigen [86]. The source code at the time of submitting this thesis is found online at `https://gitlab.com/JEBradford/actinmodel`.

The simulation progresses in time iteratively using a fixed discrete timestep Δt. During this time window many stochastic events can happen. We chose this method over an event driven timestep scheme, such as the Gillespie algorithm [80, 81], to reduce computational cost. We did initially experiment with the Gillespie algorithm, but having the much shorter timesteps coupled with the Brownian dynamics method increased running time of simulations significantly.

The main component of the simulation is actin filaments, with other components being membrane objects and exclusion zones/targets for investigating phagocytosis.

2.2.1 Random number generation

We use the Mersenne Twister algorithm [87] for the random number generation, in C++ this is the `mt19937_64` object. It generates integers in the range $[0, 2^{64} - 1]$ and the period is $2^{19937} - 1$. This random number generator (RNG) requires a seed to begin. In our method there are two ways of seeding; either supplying a seed manually, which allows previous runs to be repeated, or by generating a seed. Seeds are generated by using `random_device` which produces random numbers from the computer's entropy pool. In Linux this comes from sources such as mouse movements and keyboard presses. The seed used for a simulation is saved to a log file so that that exact simulation can be repeated if necessary.

Random numbers which follow certain distributions can then be generated by passing the output from `mt19937_64` (a 64 bit integer) to various distribution classes such as `normal_distribution` and `uniform_real_distribution`. The details of any distributions used will be described at the relevant parts in this chapter.

2.3 Actin

The component of the model with the greatest complexity is the actin. This section explains how actin is represented in the model, as well as covering how I simulate all the weird and wonderful biochemical events that it can undergo.

2.3.1 How filaments are represented

Actin filaments are modelled individually as a series of points in two-dimensional space. The points are connected by straight lines (subunits) which trace out the contour of the filament. A simple diagram of an actin filament in the model is shown in Figure 2.1. The minimum number of points a filament can have is four (two endpoints and two inner points). Points are added as the filament grows through polymerisation events, and removed if the filament shrinks through depolymerisation events. The parameter which controls how many points are in a filament is the desired subunit length, l_s, which is a constant. The use of l_s as well as the choice of value, will be discussed later.

The total length of a filament is

$$L = \sum_{i=0}^{N-2} l_i, \tag{2.1}$$

where N is the number of points in the filament, (therefore $N - 1$ is the number of subunits), and l_i is the length of the subunit between point i and point $i + 1$.

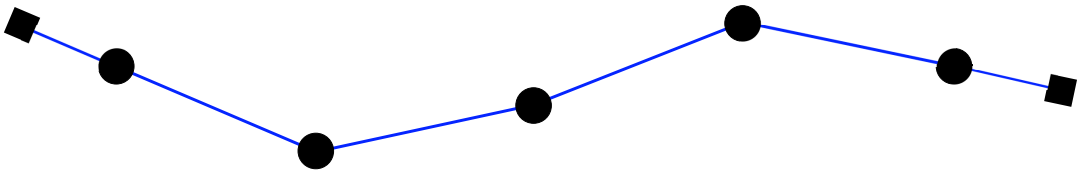


FIGURE 2.1: Basic representation of an actin filament (we will build on this later). This filament has seven points in total, five inner points (black circles) and the two endpoints (black squares). The points in the simulation have no size.

2.3.2 How motion of filaments is implemented

For motion, filaments can exist in three different forms; short rigid, long rigid and (semi-)flexible. These three different forms of filaments are shown in Figure 2.2. In following sections we will use language that we define here. In Figure 2.2 'rods' are shown as red rectangles, and represent the drag profile of either the whole filament (Figure 2.2A) or the 'inner point' corresponding to the rod (Figures 2.2B and C). 'Inner points' (black circles) are all the points that make up the filament except for the two 'end points' (black squares). 'Subunits' are straight lines which connect the points and are shown in Figure 2.2 as blue lines.

Short rigid filaments are those which have a total length, L, less than $2l_s$. Flexible filaments are those which are equal to or greater than $3l_s$ in length. Long rigid filaments have a length in between $2l_s$ and $3l_s$. This puts a limit on the value of l_s, as we can use the assumption that filaments are rigid if $L \ll l_p$ (where l_p is the persistence length of actin). Therefore $3l_s$ must be much less than the persistence length as this is the maximum length a rigid filament can be. The persistence length of actin has been measured to be 17µm [3]. We therefore suggest that the maximum l_s should be is 500nm. Actin filaments have a radius r of 3.5nm [1, 21] in the model.

All motion in the model is Brownian. We use the assumption that we are at low Reynolds number (Re $\ll 1$). The Reynolds number was defined in section 1.3.1.

2.3.2.1 Short rigid filaments

Short rigid filaments undergo motion as a random walk over the time of a single timestep. We define short as having a length, L, less than $2l_s$, which means the distance between the two middle points is less than l_s (see figure 2.2A). Translational motion in two dimensions and rotational motion in one dimension (the angle in the 2D plane) are considered separately. For each filament and for each timestep, the system calculates small steps along each of the two axes of the particular filament, parallel (along the long axis of the filament) ($\Delta a_{\parallel}$), and perpendicular ($\Delta a_{\perp}$), as well as a small step in angle ($\Delta\theta$). These steps are chosen randomly from a Gaussian distribution defined with a mean of 0 and a variance of $2D\Delta t$, where D is the diffusion coefficient for either parallel motion, perpendicular motion or rotational motion. D is calculated using the Einstein relation $D = \mu k_B T$ where μ is the relevant mobility for the particular direction of motion. In the regime of low Reynolds number the mobility is the inverse of the object's drag coefficient, which can be estimated for different shapes. We use Kirkwood's 'shish-kebab' model for rigid rods [88] where

$$\mu_{\parallel} = \frac{\ln(L/2r) + \gamma_{\parallel}}{2\pi\eta L}, \tag{2.2}$$

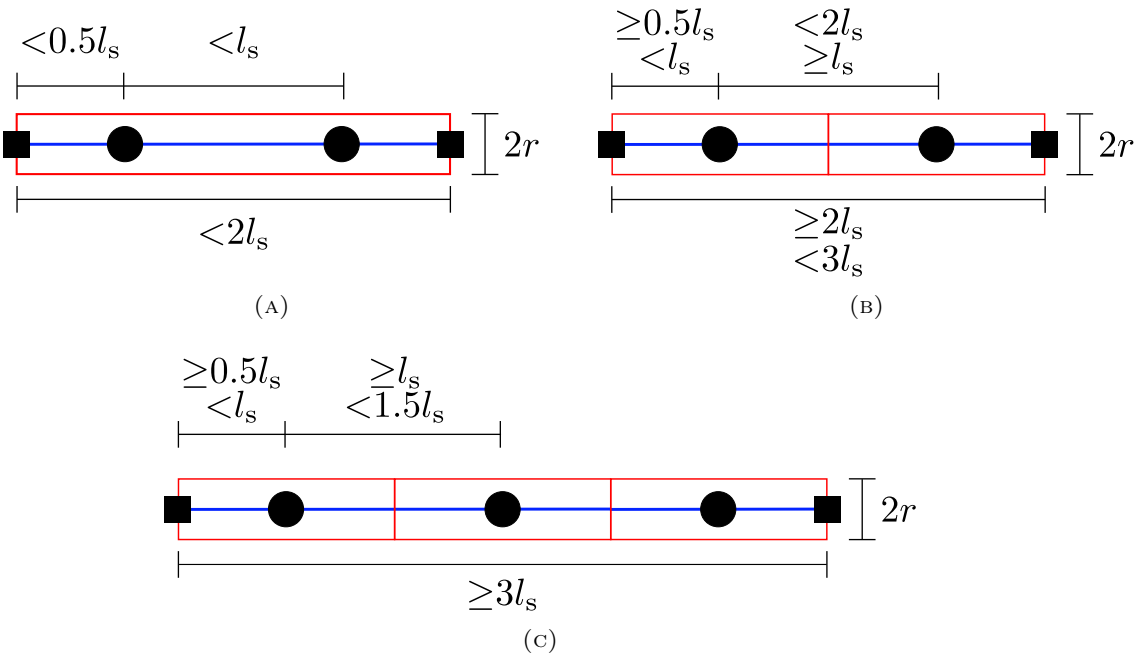


FIGURE 2.2: Three different types of actin filament where motion is concerned. Rods are shown as red rectangles. Inner points are shown as black circles. End points are shown as black squares. Subunits which connect the points are shown as blue lines. **(A)** Short rigid (which must have four points). **(B)** Long rigid (which must have four points). **(C)** Long flexible (which must have more than four points; this is an example with five points).

$$\mu_\perp = \frac{\ln(L/2r) + \gamma_\perp}{4\pi\eta L}, \tag{2.3}$$

and

$$\mu_\theta = \frac{3(\ln(L/2r)) + \gamma_\theta}{\pi\eta L^3}. \tag{2.4}$$

Where η is the viscosity of the fluid the rod is in, L is the length of the rod and $2r$ is the diameter of the rod. The gamma constants ($\gamma_\parallel$, $\gamma_\perp$ and γ_θ) are correction factors for parallel, perpendicular and rotational mobility respectively. These correction factors take into consideration end-effects [89]. We use the values proposed by [90] for $\gamma_\parallel$ and $\gamma_\perp$ (0.044 and 1.111 respectively) as these give very good agreement with experimental measurements [89]. We use the value of -0.8 for γ_θ taken from [88].

Equations 2.2 and 2.3 break down if $L < 2r\exp(-\gamma)$ since this would give a negative mobility, in our simulations this is never the case. The smallest filament possible is a trimer of three monomers together. Given the values used to describe actin, the length of a trimer is never less than $2r\exp(-\gamma)$.

Equation 2.4 would give a negative rotational mobility if $L < 2r\exp(-\gamma_\theta/3)$, and due to the value used for γ_θ being negative, this can occur in our simulation. If this is the case then γ_θ is set to equal 0, avoiding a negative mobility (L can never be less than $2r$).

2.3.2.2 Short rigid structures

Later in this chapter we will cover the branched and crosslinked actin in detail, for now it is worth noting that filaments can be connected to other filaments. This creates complexity when considering how these multi-filament structures move.

Let's consider the case where all filaments in a structure are less than $2l_s$ in length. Here the structure's motion is evaluated by fitting an ellipsoidal shape to it. Mobility constants are calculated for the fitted ellipsoid and the whole structure is moved together, experiencing drag as if it was the ellipsoid, using the random walk method previously described.

The method is as follows. First, a two-dimensional ellipse is fitted to 2D points generated along the whole structure. The points generated correspond to the centres of individual monomers along each filament in the structure. Every other monomer is chosen rather than every monomer, purely for computational reasons, reducing the data needed to fit by two. We define the length of an individual monomer as l_{mon}. So for a given filament in the structure, the first point is at the pointed end, the next is then $2l_{\text{mon}}$ along the filament, and so on. A confidence ellipse ($\chi^2 = 3.22$, corresponding to an 80% confidence interval) is generated which encloses ~80% of the generated points and therefore the structure. We chose a confidence ellipse rather than fitting an ellipse that encloses every point, to ignore outliers in the shape of the structure. Actin branched structures are not always close to ellipsoidal in shape, there could be an area where most of the actin is and then one long branch which diverts away, fitting a ellipse enclosing all the points to such a structure would result in a poor approximation to the drag of the structure. We chose an 80% confidence interval as this proved a good balance between capturing the core of the structure while avoiding outliers. Since we are in two dimensions the length of the semi-axes of the ellipsoid in the other third dimension is set to be equal to the radius of an actin filament r. This then turns our 2D ellipse into an ellipsoid. An example ellipsoid fit is shown in Figure 2.3, where the core of the structure is captured but the longer filaments extended outwards are neglected. The structure is refitted in future timesteps only if any part of it has changed, this is done to save computation time as refitting the same ellipsoid is unnecessary. The method for producing the confidence ellipsoids is not given here but is explained in [91] and for an alternate overview see [92].

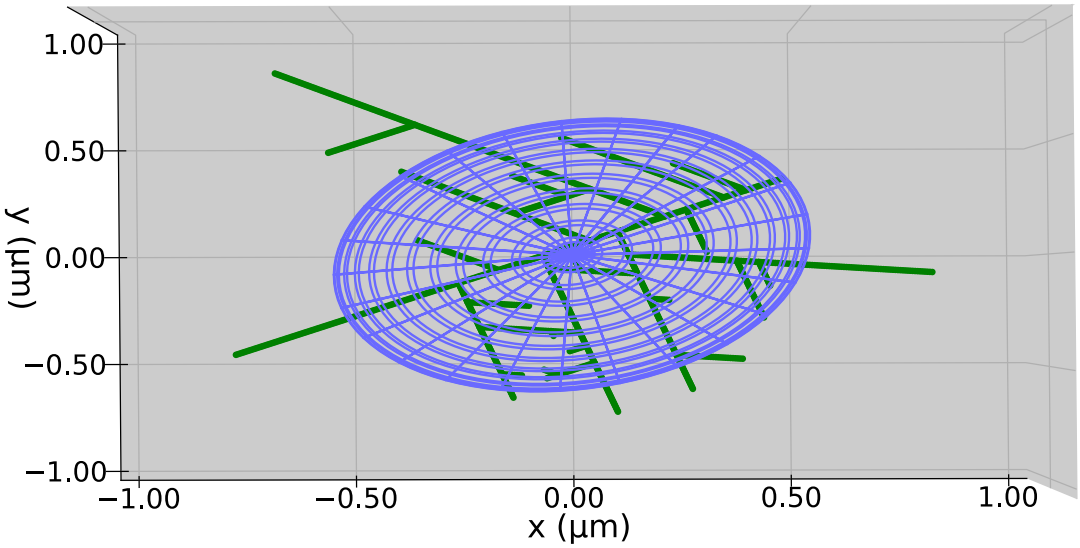


FIGURE 2.3: Example of our fitted ellipsoid, top down view. The green lines show the actin structure and the blue shape is the fitted ellipsoid. The ellipsoid encloses ~80% of the rigid branched structure. The length of the semi-axes of the third dimension (out of the paper) of the ellipsoid is set to be equal to r.

The mobility constants for an ellipsoid are calculated using Perrin's theory of Brownian motion of ellipsoids [93] and are given as

$$\mu_a = \frac{S + a^2 P}{16\pi\eta}, \tag{2.5}$$

$$\mu_b = \frac{S + b^2 Q}{16\pi\eta}, \tag{2.6}$$

$$\mu_\theta = \frac{3(a^2 P + b^2 Q)}{16\pi\eta(a^2 + b^2)}, \tag{2.7}$$

where

$$P = \int_0^\infty \frac{ds}{(a^2 + s)\sqrt{(a^2 + s)(b^2 + s)(c^2 + s)}}, \tag{2.8}$$

$$Q = \int_0^\infty \frac{ds}{(b^2 + s)\sqrt{(a^2 + s)(b^2 + s)(c^2 + s)}}, \tag{2.9}$$

and

$$S = \int_0^\infty \frac{ds}{\sqrt{(a^2 + s)(b^2 + s)(c^2 + s)}}. \tag{2.10}$$

a, b and c are the semi-axes of the ellipsoid, ($c = r$). μ_a and μ_b are the translational mobilities of the ellipsoid along the major and minor axis respectively, μ_θ is the rotational mobility of the ellipsoid. P, Q and S are elliptic integrals and are solved numerically using the Boost functions `ellint_rd` for P and Q and `ellint_rf` for S.

2.3.2.3 Longer rigid filaments and flexible filaments

For filaments with a length greater than or equal to $2l_\mathrm{s}$, we use a Brownian dynamics method adapted from that given in [94] and which is presented in full in this section.

Filaments are represented by a series of points, which include two endpoints and at least two inner points. In the algorithm presented here the inner points are connected to adjacent inner points by inextensible links that ensure our filaments are inextensible during motion. This method is commonly referred to as a 'bead-rod model' [94–97], the inner points of our filament being the 'beads' and the inextensible links between them the 'rods'. This is different to the 'bead-spring' model which model the links as springs [97]. We chose the 'bead-rod' model as we believe it is better suited for actin. Actin is inextensible at the magnitudes of force experienced in the cell [98]. A 'bead-spring' model would also require much shorter timesteps to evaluate the correct forces. We will not use the term 'bead-rod' as this will become confusing as it is incompatible with our own definitions of 'rods'.

Endpoints are not considered in the Brownian dynamics algorithm. This is because each inner point is treated as a midpoint of a small rod, shown as the three red rectangles in figures 2.2B and C. The number of inner points in a filament is therefore $N_\mathrm{inner} = N - 2$.

Let's consider a simple version of the Brownian dynamics algorithm [99]. Where during a single timestep the new position of the filament is calculated as

$$\boldsymbol{r}_{t+\Delta t} = \boldsymbol{r}_t + \frac{\Delta t}{k_B T}\mathbf{D}_t \boldsymbol{F}_t + \boldsymbol{\xi}_t, \tag{2.11}$$

where $\boldsymbol{r}_{t+\Delta t}$ is a set of position vectors for all the inner points in the filament at time $t + \Delta t$. $\mathbf{D}_t$ is the translational diffusion matrix evaluated at time t. $\boldsymbol{F}_t$ is a set of

total forces acting on the inner points evaluated at time t. $\boldsymbol{\xi}_t$ is a set of distance increments that are applied to each inner point, this is sometimes referred to as the Brownian term, Langevin term or Brownian noise. Therefore the left hand side is the new configuration of the filament and the first term on the right hand side is the old configuration of the filament. The second term on the right hand side is deterministic movements calculated from forces on the filament. As we covered earlier, $\frac{1}{k_B T}\mathbf{D}$ is the inverse of the drag, so this second term is the deterministic force response term. The final term is random movements which model Brownian motion.

The translation diffusion matrix, $\mathbf{D}_t$ is made up of 2x2 blocks along its diagonal which are calculated as

$$\mathbf{D}_{ii} = k_B T(\mu_{\parallel,i}(\tilde{\boldsymbol{u}}_i \tilde{\boldsymbol{u}}_i^{\mathsf{T}}) + \mu_{\perp,i}(\mathbf{I} - \tilde{\boldsymbol{u}}_i \tilde{\boldsymbol{u}}_i^{\mathsf{T}})), \tag{2.12}$$

where we have omitted the subscript for the time t, to use subscripts to denote points in the filament. $i = [1, N_{\text{inner}}]$ and is the inner point number (all points go from 0 to $N-1$, inner points go from 1 to $N-2$). $\mu_{\parallel,i}$ and $\mu_{\perp,i}$ are modified from equations 2.2 and 2.3 respectively. However instead of being the mobilities for the whole filament, they are now the mobilities for the rod that the inner point i is the midpoint of. $\mu_{\parallel,i}$ and $\mu_{\perp,i}$ are given below.

$$\mu_{\parallel,i} = \frac{\ln(L/2r) + \gamma_{\parallel}}{2\pi\eta s_i}, \tag{2.13}$$

$$\mu_{\perp,i} = \frac{\ln(L/2r) + \gamma_{\perp}}{4\pi\eta s_i}. \tag{2.14}$$

In equations 2.13 and 2.14 the L in the denominator (from equations 2.2 and 2.3) is replaced with the length of the rod the inner point represents, s_i. For rods not at either end of the filament s_i is always equal to l_{s}, the desired subunit length. For end rods s_1 and $s_{N_{\text{inner}}}$ are equal to $2l_0$ and $2l_{N-2}$ respectively. The L in the natural logarithm remains the total length of the filament, which results in the sum of the all the drag coefficients (reciprocals of mobilitys) for each inner point equalling the drag coefficient of the whole filament (if the filament is straight) [59, 96].

$\tilde{\boldsymbol{u}}_i$ is a tangent unit vector of point i and is calculated as

$$\tilde{\boldsymbol{u}}_i = \frac{(\boldsymbol{u}_i + \boldsymbol{u}_{i-1})}{\|\boldsymbol{u}_i + \boldsymbol{u}_{i-1}\|}, \tag{2.15}$$

where $\boldsymbol{u}_i$ is the unit vector of the subunit connecting point i and point $i+1$. For the first inner point ($i = 1$) $\tilde{\boldsymbol{u}}_1 = \boldsymbol{u}_1$ and for the final inner point ($i = N-2$ or $i = N_{\text{inner}}$) $\tilde{\boldsymbol{u}}_{N-2} = \boldsymbol{u}_{N-3}$. This is because $\boldsymbol{u}_1 = \boldsymbol{u}_0$ and $\boldsymbol{u}_{N-3} = \boldsymbol{u}_{N-2}$. As we are using equations 2.13 and 2.14, we are incorporating anisotropic drag coefficients and so each inner point is treated as the centre of a small rod. This has previously been used before in [96]. The operation $\tilde{\boldsymbol{u}}_i \tilde{\boldsymbol{u}}_i^{\mathsf{T}}$ in equation 2.12 is the outer vector product sometimes alternatively indicated as $\tilde{\boldsymbol{u}}_i \otimes \tilde{\boldsymbol{u}}_i$.

$\boldsymbol{F}_t$ is a list of N_{inner} vectors containing the total forces acting on each point. In this section just the forces on the inner points due to bending rigidity of the actin ($\boldsymbol{F}^{\boldsymbol{b}}$) are covered, other additional forces will be considered later. Again, we have omitted the time subscript to make space for point subscript notation in following equations. In chapter 1 the worm like chain (WLC) model of a polymer was introduced. The bending forces on a worm like chain can be calculated by taking the first derivative of the bending energy

$$\boldsymbol{F}^{\boldsymbol{b}} = -\nabla U^b. \tag{2.16}$$

The expressions for calculating the discrete bending forces on the inner points are found in the appendix of [100], but are also presented here. We need to introduce another quantity first, the actual subunit length d_i. Previously we have used l_i to mean the subunit length, however this is the rest subunit length - the length that the subunit should be. The Brownian dynamics method will not produce new filament configurations where the distance between the points is exactly the same from timestep to timestep, the discrepancy will be very small, but non-zero. Therefore we have the actual subunit length, defined as

$$d_i = \|\boldsymbol{r}_i - \boldsymbol{r}_{i+1}\|, \tag{2.17}$$

which is just the distance between the points.

The bending force acting on the first inner point is

$$\boldsymbol{F^b}_1 = -\frac{\kappa_b}{l_s}\left[\frac{\boldsymbol{u}_2 - \boldsymbol{u}_1}{d_1}\right], \tag{2.18}$$

where κ_b is the bending modulus of our actin filament (equivalent to $l_P k_B T$).

The force acting on the second inner point is

$$\boldsymbol{F^b}_2 = -\frac{\kappa_b}{l_s}\left[\frac{\boldsymbol{u}_3 - \boldsymbol{u}_2}{d_2} - (\boldsymbol{u}_2 - \boldsymbol{u}_1)\left(\frac{1}{d_1} + \frac{1}{d_2}\right)\right]. \tag{2.19}$$

If there are only 3 inner points in the filament this reduces down to

$$\boldsymbol{F^b}_2 = \frac{\kappa_b}{l_s}\left[(\boldsymbol{u}_2 - \boldsymbol{u}_1)\left(\frac{1}{d_1} + \frac{1}{d_2}\right)\right]. \tag{2.20}$$

For any inner point i where $3 < i < N_{\text{inner}} - 1$, the force is

$$\boldsymbol{F^b}_i = -\frac{\kappa_b}{l_s}\left[\frac{\boldsymbol{u}_{i-1} - \boldsymbol{u}_{i-2}}{d_{i-1}} - (\boldsymbol{u}_i - \boldsymbol{u}_{i-1})\left(\frac{1}{d_{i-1}} + \frac{1}{d_i}\right) + \frac{\boldsymbol{u}_{i+1} - \boldsymbol{u}_i}{d_i}\right]. \tag{2.21}$$

Finally the forces acting on the last two inner points ($i = N_{\text{inner}} - 1$ and $i = N_{\text{inner}}$) are going to be the same form as equations 2.18 and 2.19 and are given below for clarity

$$\boldsymbol{F^b}_{N_{\text{inner}}-1} = -\frac{\kappa_b}{l_s}\left[\frac{\boldsymbol{u}_{N_{\text{inner}}-2} - \boldsymbol{u}_{N_{\text{inner}}-3}}{d_{N_{\text{inner}}-2}} - (\boldsymbol{u}_{N_{\text{inner}}-1} - \boldsymbol{u}_{N_{\text{inner}}-2})\left(\frac{1}{d_{N_{\text{inner}}-2}} + \frac{1}{d_{N_{\text{inner}}-1}}\right)\right], \tag{2.22}$$

$$\boldsymbol{F^b}_{N_{\text{inner}}} = -\frac{\kappa_b}{l_s}\left[\frac{\boldsymbol{u}_{N_{\text{inner}}-1} - \boldsymbol{u}_{N_{\text{inner}}-2}}{d_{N_{\text{inner}}-1}}\right]. \tag{2.23}$$

If the filament is a 'long rigid' filament (less than $3l_s$ in length), it is represented by only two inner points and therefore cannot bend, $\boldsymbol{F^b}$ is therefore set to 0. We use the Brownian dynamics method for these long rigid filaments rather than the random walk method used for short rigid filaments, to include other contributions to $\boldsymbol{F}_t$ which will be discussed later on.

$\boldsymbol{\xi}_t$ is the Brownian term and is a set of distance increment vectors - one for each inner point. To explain how this is generated let's first consider this vector for a rod i in a coordinate system of the rod (so the x axis is along the x component of $\tilde{\boldsymbol{u}}_i$ and the y axis is therefore along the y component of $\tilde{\boldsymbol{u}}_i$). $\boldsymbol{\xi}'^i$ (the prime denotes this alternate

reference frame) is then a 2D vector made up of components equal to

$$\xi_{\parallel,i} = q_1\sqrt{2k_BT\mu_{\parallel,i}\Delta t}, \tag{2.24}$$

and

$$\xi_{\perp,i} = q_2\sqrt{2k_BT\mu_{\perp,i}\Delta t}, \tag{2.25}$$

where q_1 and q_2 are random numbers $q \sim N(0,1)$. To get $\boldsymbol{\xi}_i$ we must apply a rotation to $\tilde{\boldsymbol{u}}_i$, moving into the Cartesian frame.

$$\boldsymbol{\xi}_i = \begin{bmatrix} \tilde{u}_{i,x} & -\tilde{u}_{i,y} \\ \tilde{u}_{i,y} & \tilde{u}_{i,x} \end{bmatrix} \begin{bmatrix} \xi_{\parallel,i} \\ \xi_{\perp,i} \end{bmatrix} \tag{2.26}$$

$\boldsymbol{\xi}_t$ is then the list of these vectors, one for each inner point.

Equation 2.11 is missing something, it does not contain the distance constraints between each inner point. The forces that produce these constraints must be included. The method for determining the constraint forces is given below. We follow the description provided in [94], and start with a set of $N_{\text{inner}} - 1$ constraint equations

$$g_i = d_i - l_i = 0, \tag{2.27}$$

where g_i is the constraint between point i and point $i+1$. We define a matrix $\mathbf{B}_t$ which contains these $N_{\text{inner}} - 1$ constraints as

$$\mathbf{B}_t = \left\{ \frac{\partial g_i}{\partial \mathbf{r}}, i = [1, N_{\text{inner}} - 1] \right\}. \tag{2.28}$$

so $\mathbf{B}_t$ has $N_{\text{inner}} - 1$ rows (for each constraint) and $2N_{\text{inner}}$ columns (for x and y components of each inner point) and is upper bidiagonal. An example for a filament with three inner points is given below

$$\mathbf{B}_{\boldsymbol{N_{\text{inner}}=3}} = \begin{bmatrix} \frac{r_{x,1}-r_{x,2}}{d_1} & \frac{r_{y,1}-r_{y,2}}{d_1} & \frac{r_{x,2}-r_{x,1}}{d_1} & \frac{r_{y,2}-r_{y,1}}{d_1} & 0 & 0 \\ 0 & 0 & \frac{r_{x,2}-r_{x,3}}{d_2} & \frac{r_{y,2}-r_{y,3}}{d_2} & \frac{r_{x,3}-r_{x,2}}{d_2} & \frac{r_{y,3}-r_{y,2}}{d_2} \end{bmatrix} \tag{2.29}$$

Adding this in, equation 2.11 becomes

$$\boldsymbol{r}_{t+\Delta t} = \boldsymbol{r}_t + \frac{\Delta t}{k_BT}\mathbf{D}_t(\boldsymbol{F}_t + \mathbf{B}_t^{\mathsf{T}}\boldsymbol{\lambda}_t) + \boldsymbol{\xi}_t, \tag{2.30}$$

where $\boldsymbol{\lambda}_t$ is the Lagrange multiplier vector, this is shown in [94] to be

$$\boldsymbol{\lambda}_t = -(\mathbf{B}_t\mathbf{D}_t\mathbf{B}_t^{\mathsf{T}})^{-1}\mathbf{B}_t\left(\mathbf{D}_t\mathbf{F}_t + \frac{k_BT}{\Delta t}\boldsymbol{\xi}_t\right), \tag{2.31}$$

which means we can rewrite equation 2.30 as

$$\boldsymbol{r}_{t+\Delta t} = \boldsymbol{r}_t + (\mathbf{I} - \mathbf{T}_t\mathbf{B}_t)\left(\frac{\Delta t}{k_BT}\mathbf{D}_t\boldsymbol{F}_t + \boldsymbol{\xi}_t\right). \tag{2.32}$$

$(\mathbf{I} - \mathbf{T}_t\mathbf{B}_t)$ is a projection matrix which enforces the distance constraints, where $\mathbf{T}_t = \mathbf{D}_t\mathbf{B}_t^{\mathsf{T}}(\mathbf{B}_t\mathbf{D}_t\mathbf{B}_t^{\mathsf{T}})^{-1}$ and can be approximated as $\mathbf{T}_t \approx \mathbf{B}_t^{\mathsf{T}}(\mathbf{B}_t\mathbf{B}_t^{\mathsf{T}})^{-1}$ which is what we use. As mentioned in [101], equation 2.32 is unstable and an extra term of $-\mathbf{T}_t(\mathbf{B}_t\boldsymbol{r}_t - \boldsymbol{p}_t)$ is needed, where $\boldsymbol{p}_t$ is a vector containing the $N_{\text{inner}} - 1$ rest lengths between inner points. This gives the final equation to generate a new contour of the inner points of

our filament as

$$\boldsymbol{r}_{t+\Delta t} = (\mathbf{I} - \mathbf{T}_t\mathbf{B}_t)\left(\boldsymbol{r}_t + \frac{\Delta t}{k_B T}\mathbf{D}_t\boldsymbol{F}_t + \boldsymbol{\xi}_t\right) + \mathbf{T}_t\boldsymbol{p}_t. \tag{2.33}$$

This particular implementation is known as LINCS (LINear Constraint Solver) which was first described in [101] and has since been incorporated into GROMACS [102], the well known molecular dynamics package.

We use equation 2.33 to generate the new inner points of our flexible filament but we also need to move the two endpoints. This is simply done by projecting the endpoints out along the same direction as the two end inner point subunits. In Figure 2.2C if the three inner points (black circles) move, then the two endpoints (black squares) are then projected out so that the two end subunits are in the same direction as the two penultimate subunits. The new pointed end point is therefore

$$\boldsymbol{r}_0 = \boldsymbol{r}_1 - d_0\boldsymbol{u}_1, \tag{2.34}$$

and the new barbed end point is

$$\boldsymbol{r}_{N-1} = \boldsymbol{r}_{N-2} + d_{N-2}\boldsymbol{u}_{N-3}. \tag{2.35}$$

The reason we do not use this method for our short rigid filaments is due to timestep limitations, this will be discussed later in section 2.6.1.

2.3.2.4 Long rigid and Flexible structures

In section 2.3.2.2 we looked at how structures of branched and/or crosslinked actin can move together, but only in the case where all filaments in the structure were short rigid. This section will cover how structures move when some of the filaments in it are longer than $2l_\mathrm{s}$ and others are not, and when all the filaments in the structure are longer than $2l_\mathrm{s}$.

Short rigid filaments are connected to other filaments (short, long rigid and flexible) by fixed connections. Each filament in the structure has its motion evaluated one by one. If the filament is not short rigid it is moved using the method described in section 2.3.2.3. Any short rigid filaments that are connected are simply moved with the longer filament, this assumes the short rigid filament does not have a significant effect on the motion of the flexible filament it is connected to, as the long filament moves as if the short rigid filament was not there.

Longer filaments are connected to other longer filaments with springs, which respond to forces with equal and opposite restoring forces on the two connecting filaments. Connections could have been modelled using additional constraints and including these in $\mathbf{B}_t$ in equation 2.28, this is a potential improvement for the future. Springs were chosen for simplicity. The additional forces from the springs are added to the bending forces in $\boldsymbol{F}_t$ in equation 2.33. Springs can join branches to mothers and crosslinked filaments together. These are covered in turn here.

With branches the branchpoint is modelled both as a linear spring to maintain attachment of the branch to the mother and an angular (torsion) spring to maintain the characteristic angle (θ_br) between the mother and branch.

With the linear spring, one end is situated on the mother filament in a position corresponding to the relevant 'branched' monomer (see later in section 2.3.4.4) and the other end is fixed to the pointed end of the branch. The spring therefore connects the two with a zero rest length. When evaluating the movement of the mother filament, the force is distributed on the two end points of the subunit that the branch is attached to

by using an interpolation coefficient $\alpha \in [0, 1)$ as in [59]. This interpolation coefficient is the relative distance the branchpoint is along the subunit. The total restoring force of the linear spring acting on the mother filament is

$$\boldsymbol{F}_{\mathbf{tot}}^{\mathbf{lin\text{-}br}} = -k_{\mathrm{br}} \left(B - P \right), \tag{2.36}$$

and is simply Hooke's law. The branchpoint on the mother is B and the pointed end of the branch is P. k_{br} is the branch spring constant. We experimented with the value of k_{br} to ensure branches remained connected to their mothers. We found a value of $5{\times}10^{-5}\mathrm{N\,m^{-1}}$ to be suitable. This force is then distributed linearly on the two points i and $i+1$ (that make up subunit i on the mother) by

$$\boldsymbol{F}_{i}^{\mathbf{lin\text{-}br}} = (1 - \alpha)\, \boldsymbol{F}_{\mathbf{tot}}^{\mathbf{lin\text{-}br}}, \tag{2.37}$$

and

$$\boldsymbol{F}_{i+1}^{\mathbf{lin\text{-}br}} = \alpha \boldsymbol{F}_{\mathbf{tot}}^{\mathbf{lin\text{-}br}}. \tag{2.38}$$

If the branchpoint lies on an end subunit, then all the force is applied to the end inner point. When evaluating the movement of the branch, equation 2.36 is simply multiplied by -1 and all the force is applied to the first inner point on the branch.

With the torsion spring the resting angle between the mother and branch is $\pm\theta_{\mathrm{br}}$. When evaluating the movement of the mother filament a torque is calculated on the subunit the branch is attached to and this is converted into equal and opposite forces on the two points that make up the ends of this subunit. The following equation gives these forces and can be derived from equation 1.2. The full derivation is given in Appendix A but it is stated here as

$$\boldsymbol{F}_{i}^{\mathbf{ang\text{-}br}} = -\boldsymbol{F}_{i+1}^{\mathbf{ang\text{-}br}} = k_{\theta_{\mathrm{br}}} \Delta\theta \frac{\hat{\boldsymbol{n}}_i}{d_i}. \tag{2.39}$$

In equation 2.39 the index i refers to the ith subunit on the mother (which the branch is attached to) connecting the two points i and $i+1$. $\boldsymbol{F}_{i}^{\mathbf{ang\text{-}br}}$ is the force acting on point i due to the presence of the torsion spring, $k_{\theta_{\mathrm{br}}}$ is the torsion spring constant, $\Delta\theta$ is the difference between the measured branch angle and the resting angle of $\pm\theta_{\mathrm{br}}$, and $\hat{\boldsymbol{n}}_i$ is the unit normal vector from subunit i pointing towards the branch. We experimented with values of the torsion spring constant $k_{\theta_{\mathrm{br}}}$ to maintain the angle at $\pm\theta_{\mathrm{br}}$ between mother and daughter. We found the value of $5{\times}10^{-18}\mathrm{N\,m\,rad^{-1}}$ to be suitable. When considering the branch, the forces are always applied to the first two inner points. So we use equation 2.39 with the overall sign flipped, i is equal to one (corresponding to the first inner point) and $\hat{\boldsymbol{n}}_1$ is now from the branch's first inner point subunit pointing towards the mother. Torsion and linear springs have previously been used to simulate the motion of actin branches [60, 61].

After each filament in a particular structure has been evaluated, the whole structure is moved as an ellipse using the method described in 2.3.2.2. This is done to cover global movement of the whole structure. In cases where the mother filament is rigid and a branch is flexible the above whole structure motion is carried out and the branch moves according to section 2.3.2.3 considering these extra forces from the mother filament.

Crosslinks work in a similar way to branches, they are modelled as a linear spring but without a torsional spring. If the two filaments which belong to a crosslink are

flexible the additional force is

$$\boldsymbol{F}_{\textbf{cl-1}} = -\boldsymbol{F}_{\textbf{cl-2}} = -\kappa_{\mathrm{cl}} \left(1 - \frac{d_{\mathrm{rest}}}{\|C_1 - C_2\|}\right)(C_1 - C_2), \tag{2.40}$$

where κ_{cl} is the stiffness of the crosslink spring, d_{rest} is the rest distance of the spring and is equal to the crosslinking distance plus the diameter of an actin filament, C_1 and C_2 are the coordinates of the crosslink points on filaments 1 and 2 respectively. We take the value of κ_{cl} to be $5{\times}10^{-5}\mathrm{N\,m^{-1}}$, the same as with branch springs.

Rigid filaments behave differently. If a rigid filament has just one crosslink associated with it, it will rotate about its crosslink point. If this filament does not have any branches, this rotation is carried out using the same method as that described in section 2.3.2.1, using the mobility constant given in equation 2.4. If the filament does have branches, an ellipse is fitted to the actin and branches but not any actin on the other side of the crosslink. Equation 2.7 is used to determine the rotation of this structure. If there is more than one crosslink associated with a filament, it cannot rotate in this local sense. However in both these two cases the whole structure is still translated and rotated using the ellipse fitting method described in section 2.3.2.2.

2.3.3 Steric hindrance

Actin filaments cannot interpenetrate each other and modelling this is essential to capture the physics of actin based force generation. This is modelled in the simplest way through a 'hard-core' interaction, where any step in the simulation (for example motion as discussed in section 2.3.2) that results in intersections are rejected. Hard-core steric interactions have been used before in modelling actin filament networks [65, 68]. There are some allowed exceptions to this rule, as well as unallowed exceptions that can happen under rare circumstances, these will be covered later. We call this steric hindrance.

After any step that may cause a filament to intersect with another, a distance d_{steric} is calculated between the filament and other filaments. If $d_{\mathrm{steric}} \leq 2r_{\mathrm{steric}}$ then the step is rejected otherwise it is approved. r_{steric} is the steric radius of filament, which is larger than its physical radius. This is to account for short-ranged repulsive forces between filaments. Actin filaments carry a negative charge [103], coulomb interactions will favour filaments repulsing each other. So, sterically, our filaments interact as if they were made up of rods (one for each subunit connecting each point) with a circular cap on each end, having a diameter of r_{steric}. Our filaments have a steric radius r_{steric} of 7.5nm [104]. Previous modelling work has also used an 'effective fibre diameter' which is larger than the radius of an actin filament, [60]. Although in this case, the interaction was soft; producing a repulsive force if two actin filaments are at a distance less than this effective diameter.

2.3.3.1 The steric grid

Distance measurements to check for intersections are done using the functions for line segment to line segment distance queries in the Geometric tools library [85]. Whilst these functions are well optimised, we want to avoid doing them unnecessarily, as this will have a negative impact on the run time of simulations. If done in the simplest way, after each potentially steric hindrance breaking move, we would calculate distances from all the subunits in the filament that has just undergone the change, with all the subunits of every single other filament in the simulation (as well as some within the same filament to check for it crossing itself). Therefore these checks are roughly an

operation of order N_{sys}^2 with N_{sys} being the total number of points in the system. It is a waste of compute to measure distances between filaments that are much further away from each other than the diameter of a filament. So to overcome this problem, and reduce the number of distance queries, we partitioned space up into a square grid which we name the 'steric grid'.

The steric grid has a finite size and this must be such that no filaments exit the space covered by the grid during the simulation. Upon initialisation of the simulation, the width and height of individual square cells (s_{cell}) in the grid should be set equal to $2r_{\text{steric}}$. However, we obviously need an integer number of cells to fit inside the grid, therefore the size of the cells is actually

$$s_{\text{cell}} = s_{\text{grid}} / \left\lfloor \frac{s_{\text{grid}}}{2r_{\text{steric}}} \right\rfloor, \tag{2.41}$$

where s_{grid} is the size (width or height - since it is a square) of the grid. So the floor of the fraction on the right would give the number of cells across, therefore ensuring $s_{\text{cell}} \geq 2r_{\text{steric}}$.

The simulation then keeps track of which actin filament subunits occupy which cells. Each cell has a list of which actin filament subunits occupy it, and each actin filament subunit has a list of all the steric grid cells it occupies. These lists are updated when necessary, for example after a filament has moved. It is important to stress that the filaments are not restricted to move on discrete sites - they still move in continuous space. The grid can be thought of as an overlay to roughly track where everything is at a given time. A diagram representing our steric grid is given in Figure 2.4. Here, actin filaments are shown as blue points and lines, occupying many of the one hundred cells that make up the grid. In a true simulation the grid would be much larger and each cell would be much smaller relative to the length of the actin filaments.

Since filament subunits tend to be much larger than the cell size, we developed a robust method to determine which cells a subunit occupies. Each subunit is simply described as two points, and so it is simple to calculate which cells these two points are in, but what about the subunit connecting the two? One might think you can use a fast algorithm like Bresenham's line algorithm [105] or Xiaolin Wu's line algorithm [106] as these are often used in computer graphics to trace out lines using pixels. However these are strictly discrete algorithms and do not take into account where in the cell the endpoints lie. We did indeed first try a modified version of Bresenham's line algorithm [107], but through testing, this was not accurate enough. Simulations using this algorithm produced discrepancies with ones using the baseline $O(N_{\text{sys}}^2)$ method - as a small amount of steric hindrance breaking moves were incorrectly allowed. So instead, we developed our own method.

Figure 2.5 is a flow chart detailing the algorithm described below. Some boxes contain questions, solid green arrows from these boxes show the flow given a 'Yes' answer, dashed red lines give the flow for a 'No' answer. In addition to Figure 2.5, to help explain our method, I will refer to Figure 2.4 using examples.

The method is described as follows. Once we have the cells the two endpoints lie in we first determine the general direction from the first point to the second by calculating the difference in columns: $\Delta x_{\text{cell}} = c_2 - c_1$ and rows: $\Delta y_{\text{cell}} = r_2 - r_1$, where c is the column and r is the row. If Δx_{cell} and Δy_{cell} are both zero, then both points lie in the same cell. If only Δx_{cell} is zero, then any cells in between the two endpoints are simply in the same column. If only Δy_{cell} is zero, then any cells in between the two endpoints are simply in the same row. Using an example, looking at the leftmost actin filament in Figure 2.4, and its first subunit in the top left. The first

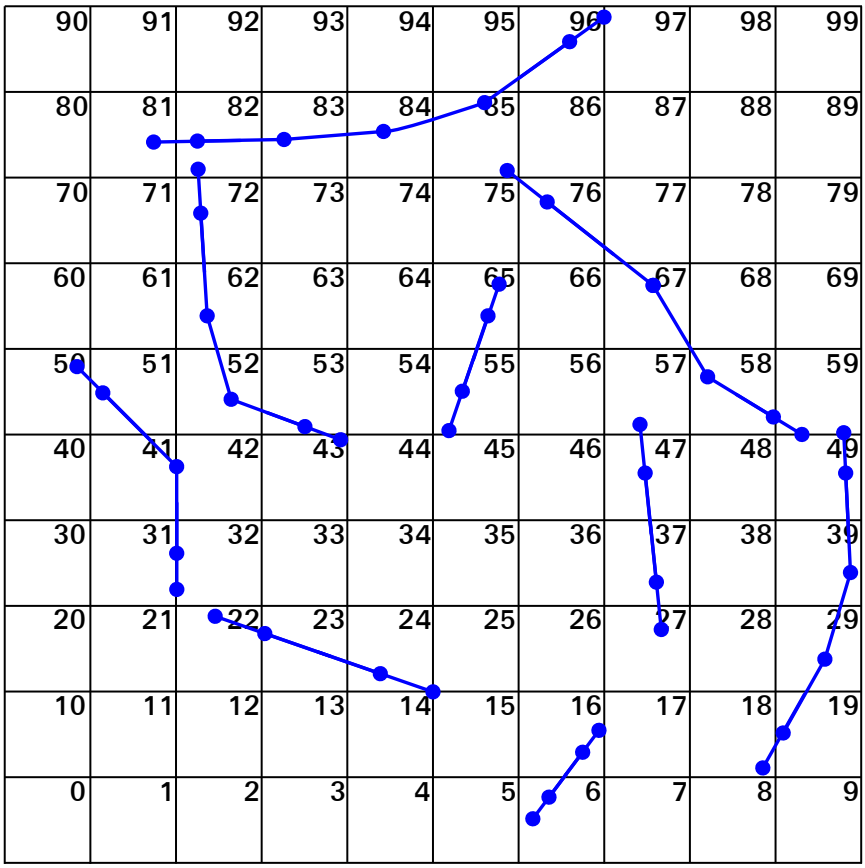


FIGURE 2.4: Representation of our steric grid with 100 cells. Actin filaments are represented as blue points and lines. In an actual simulation, the cell size would be much smaller than this.

point sits in cell 50 and the second point sits in cell 51. Therefore in this example $\Delta x_{\text{cell}} = 1$ and $\Delta y_{\text{cell}} = 0$. We therefore know that there are no cells in between the two cells that the two points lie in, they are in adjacent cells. If the second point was in cell 52, we would know that cell 51 is occupied.

However if these three conditions are not the case then it is non-trivial which cells the subunit occupies. Our solution to this involves the system checking an intersection between the actin subunit and either the right side (if Δx_{cell} is positive) or the left side (if Δx_{cell} is negative) of the cell the first point is in. If there is an intersection then this means the filament occupies the adjacent cell in the next column (right for a positive Δx_{cell}, left for a negative Δx_{cell}). If there is not an intersection then this means the filament occupies the adjacent cell in the next row (above for positive Δy_{cell}, below for negative Δy_{cell}). The process is repeated with the next cell until it reaches the same cell as the second point, therefore the method traverses the subunit and identifies all the cells it sits in. Intersections are done using the segment-segment intersection query in the Geometric tools library [85].

A good example to illustrate the previous point is the subunit which two points lie in cells 67 and 58 in Figure 2.4. Here it is unclear if the subunit occupies cell 57 or 68. If we take the point in cell 58 to be the first point then Δx_{cell} is negative. Therefore an interesection check is made between the subunit and the left side of cell 58. If there is an intersection, the subunit occupies cell 57, if not, the subunit occupies cell 68.

After every potential steric breaking event, the simulation first updates the steric grid just for the filament that has just performed the event, then runs the checks. However now, for each subunit we need to check, we need only run distance queries against other subunits that occupy the same and adjacent cells to the subunit we are checking. Since subunits in cells more than one cell away could not intersect, as the cell size is at least the steric diameter of a filament. Using this grid method rather than the brute force $O(N_{\text{sys}}^2)$ method produces speedups of up to two orders of magnitude. Similar methods have been developed to address the same problem [59, 65] and similar speedup factors were observed [65].

2.3.4 Biochemical events

As mentioned previously, a simulation advances in time with a fixed timestep Δt. A range of biochemical processes can happen to each filament during this window, and the details of which are covered in this section. In general these events are simulated using a Monte Carlo method previously described [65, 68, 82]. In this method, the rates of the biochemical processes measured in *in vitro* experiments are used. Generally speaking, the probability of an event of rate k occurring within a time Δt is $1 - \exp(-k\Delta t)$, and is always less than one. In the simulation described here we use the approximation of $k\Delta t$ for the probability, which is valid for $k\Delta t \ll 1$. The random number generator, using the Mersenne Twister algorithm, is called producing a random number $q \in [0, 1)$, the process will therefore occur if $q \leq k\Delta t$.

It is important to note that although all these biochemical processes can be looked at together, we tend to investigate as few as possible in turn, or make assumptions based on the system we are investigating, reducing the number of parameters.

2.3.4.1 Nucleation

Nucleation is the production of new (linear) filaments. When a filament is nucleated it has a length of $3l_{\text{mon}}$ where l_{mon} is the length of a single monomer. We always take l_{mon} to be equal to 2.7nm [55]. The nucleated filament is therefore representing a trimer

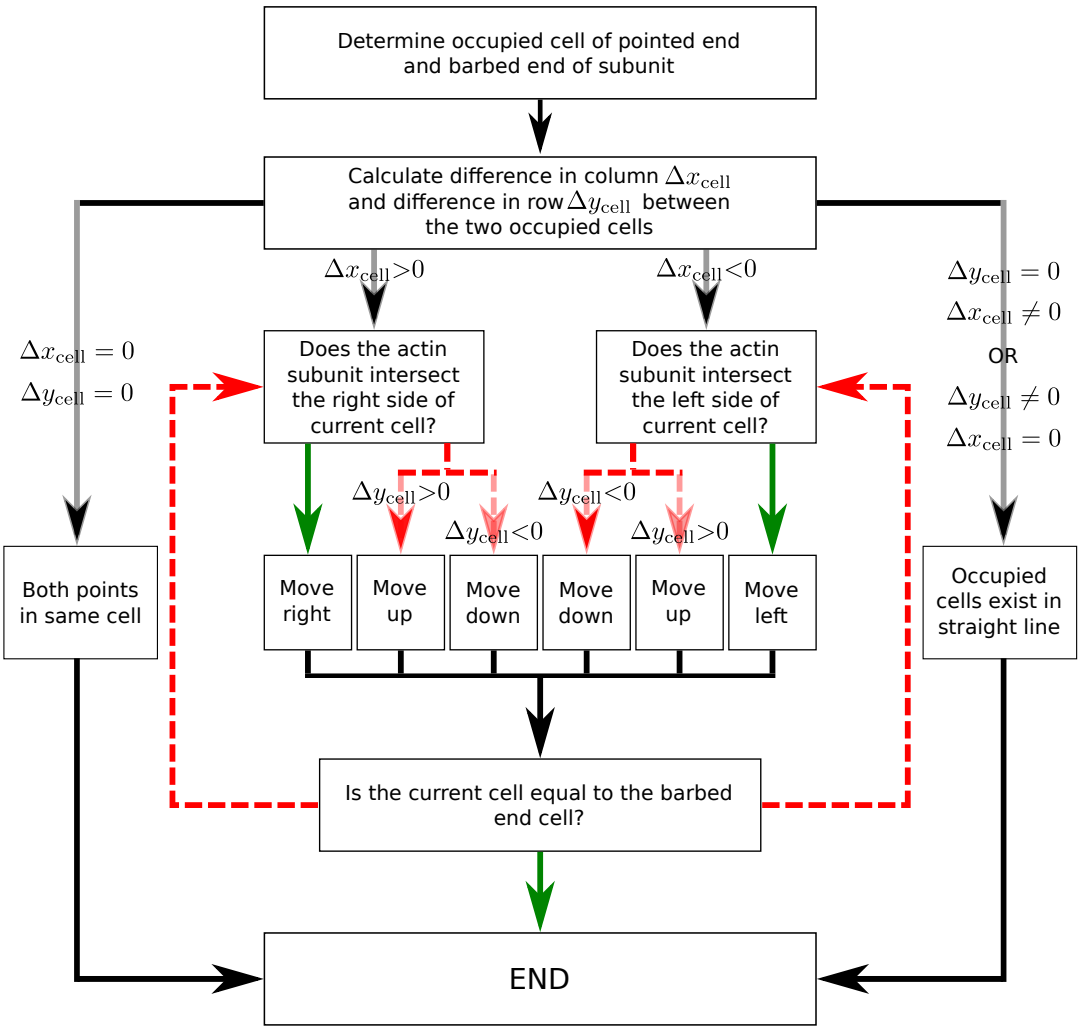


FIGURE 2.5: Flow chart describing the algorithm used for finding which cells a given actin filament occupies. Solid green arrows depict the flow from a 'Yes' answer from the question, whereas dashed red arrows depict the flow from a 'No' answer.

of three monomers. Nucleation occurs with a rate density per G-actin concentration to the power of three, k_{nuc} which therefore has units of $\mu\text{m}^{-2}\,\mu\text{M}^{-3}\,\text{s}^{-1}$. In order for nucleation to occur we must have a region in 2D space defined in which our filaments can appear, spatial regions will be discussed later in section 2.3.5. The nucleation probability in a given timestep with length Δt is

$$P_{\text{nuc}} = k_{\text{nuc}} A_{\text{nuc}} C_{\text{G}}^3 \Delta t, \tag{2.42}$$

where A_{nuc} is the area of the nucleation region and C_{G} is the concentration of G-actin.

Each timestep a maximum of one nucleation event can occur per region. If a nucleation event occurs, the nucleated filament is placed in a position in the region according to a uniform distribution, and is given a random orientation. The filament is initialised with four points connected by three subunits with lengths of $3l_{\text{mon}}/4$, $3l_{\text{mon}}/2$, and $3l_{\text{mon}}/4$.

We also include an alternative to nucleated filaments having a random orientation from a uniform distribution. Trimers can instead have their direction based on a Gaussian distribution. This was of interest to investigate filament orientation in force generation phenomena like in lamellipodia [108] or phagocytosis. Whether orientation needed to be somewhat forced (perhaps from proteins bound in a membrane) or filaments orientate themselves, with respect to the leading edge or phagocytic cup, emergently.

After the filament is created it must be checked if it intersects any other object in the simulation, if it does then the nucleation event is deemed to have 'failed' and the filament is immediately removed.

Figure 2.6 shows snapshots from a simulation where nucleation takes place. The time of each snapshot is shown in the top right of each subfigure. The blue square in the centre is the nucleation region and has an area of $1\mu\text{m}^2$. Trimers appear in this region according to a nucleation rate density of $1\mu\text{m}^{-2}\,\mu\text{M}^{-3}\,\text{s}^{-1}$, the G-actin concentration is fixed at $1\mu\text{M}$. The trimers undergo Brownian motion in a fluid of viscosity 1Pa s and temperature 300K. The trimers are shown as short blue rectangles and have been made larger in this visualisation so they can be clearly seen (they are not to scale).

2.3.4.2 Polymerisation

Polymerisation is the addition of a single monomer on either end of an actin filament. In our model we distinguish between barbed end polymerisation and pointed end polymerisation. When a filament undergoes polymerisation at either end its length increases by l_{mon}. How this length increase is distributed along the filament depends on which end has polymerised and what type of filament it is. This is shown in Figure 2.7 and is explained here. In Figure 2.7 green arrows depict polymerisation events, and green points are points that have moved in the polymerisation event. For barbed end polymerisation the barbed end point moves by a distance of l_{mon} along the unit vector of the barbed end subunit, $\boldsymbol{u}_{N-2}$. For pointed end polymerisation the pointed end point moves by a distance of l_{mon} along the reverse of the unit vector of the pointed end subunit $(-\boldsymbol{u}_0)$.

If it is a short rigid or long rigid filament ($L < 3l_s$) the length increment is shared amongst all the rods. In Figure 2.7, rods are shown as red rectangles. Therefore the adjacent point to the polymerising end moves by a distance of $\pm 3l_{\text{mon}}/4$ along the unit vector of the filament and the next point along moves by $\pm l_{\text{mon}}/4$. This means the end subunit has grown by $l_{\text{mon}}/4$, the penultimate subunit by $l_{\text{mon}}/2$ and the next

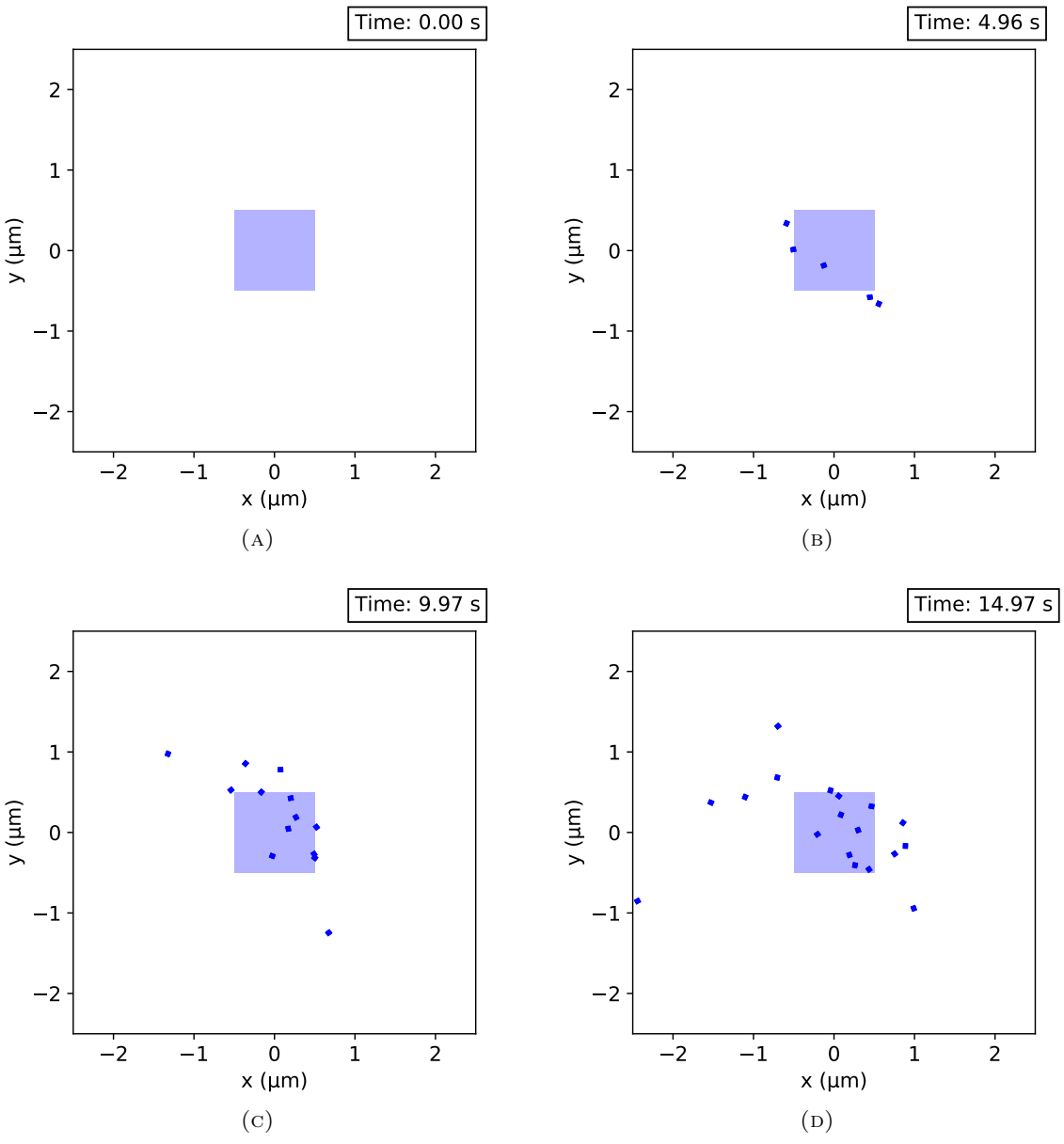


FIGURE 2.6: Snapshots of a simulation showing nucleation of actin trimers (small blue rectangles, not to scale) in a nucleation region (blue square). Time is shown in the top right: **(A)** the initial frame at 0 seconds, **(B)** 5 seconds, **(C)** 10 seconds and **(D)** the final frame at 15 seconds.

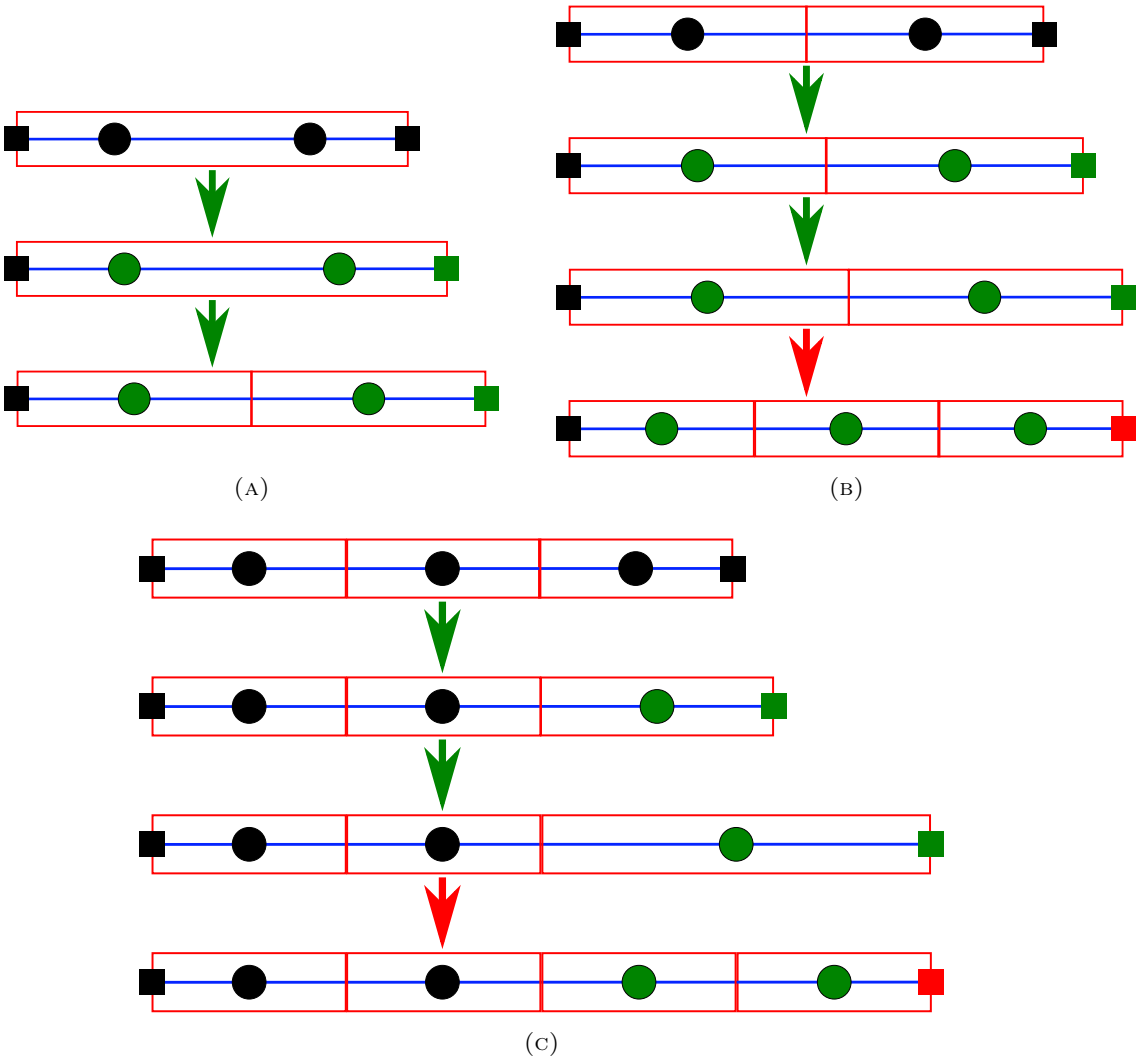


FIGURE 2.7: Diagram showing polymerisation and addition of points. The filament progresses as you move down each diagram. Green arrows represent polymerisation events that happen in time. Red arrows represent point addition events that are triggered immediately. Green points are points that have moved from the previous. Red points are points that have been added from the previous. Black points are either initial points (in the uppermost filament) or points that have not moved. All polymerisation shown happens at the right hand side end. **(A)** Polymerisation of a short rigid filament, that becomes a long rigid filament in the third iteration. **(B)** Polymerisation and point addition of a long rigid filament, which becomes a flexible filament in the final iteration. **(C)** Polymerisation and point addition of a flexible filament. Here you can see the growth of just the final rod, then it splitting into two.

subunit by $l_{\text{mon}}/4$. Subunits are shown as blue lines in Figure 2.7. This is shown in Figure 2.7A.

If the filament is flexible then the length increment is applied only to the end rod. Therefore the adjacent point to the polymerising end point (the midpoint of the end rod) moves along its unit vector by $\pm l_{\text{mon}}/2$, meaning the penultimate end subunit has grown by $l_{\text{mon}}/2$ and also the end subunit has grown by $l_{\text{mon}}/2$. This is shown in Figure 2.7C.

As filaments get longer more points are needed to trace them out. In Figure 2.7 the addition of points is shown by red arrows. New points are shown in red. The first new point is added when the filament grows such that it becomes flexible ($L \geq 3l_s$), this is shown in figure 2.7B in the process marked by the final, red arrow. After the polymerisation event which increases the length beyond $3l_s$, an extra point is added and all the points apart from the non-polymerising endpoint are adjusted. The non-polymerised end inner point is placed $l_s/2$ from the non-polymerised end point, and the next point is placed l_s from the previous point. The next point is placed at a distance of

$$l_{\text{adj}} = \frac{L - l_s}{2} \tag{2.43}$$

from the previous, and the final point (polymerising end point) is placed at a distance of

$$l_{\text{end}} = \frac{L}{2} - l_s \tag{2.44}$$

from the previous. Although the filament will be straight immediately after this point addition, it is considered flexible, and is free to undergo bending fluctuations.

The addition of points to growing flexible filaments is described here and shown in Figure 2.7C, marked by the final, red arrow. The flexible filament grows just by elongating its last two subunits (elongating the end rod). An additional point is added when the sum of the lengths of these two end subunits is greater than or equal to $2.5l_s$ (which is also when the length of the end rod is greater than or equal to $2l_s$). When this happens the final three subunits are adjusted so their lengths measure l_s, l_{adj} and l_{end} respectively with l_{end} and l_{adj} defined as

$$l_{\text{end}} = \frac{l_{N-2} + l_{N-3} - (3/2)l_s}{2}, \tag{2.45}$$

and

$$l_{\text{adj}} = l_{\text{end}} + \frac{l_s}{2}. \tag{2.46}$$

Another way to think about it, is that this splits the end rod into two rods. The resultant penultimate rod has a length of l_s, and the new end rod has a length equal to the remaining length. So for example, if the lengths of the two end subunits is exactly $2.5l_s$ (meaning the length of the end rod is exactly $2l_s$), this process results in three subunits with lengths of l_s, l_s and $0.5l_s$ (and therefore two rods both of length l_s). This is shown in figure 2.7C. A similar method was used in [109] to allow the use of a Brownian dynamics method with growing actin filaments.

Polymerisation occurs with a rate per concentration of G-actin, k_p and differs for the barbed end (k_p^{barb}) and pointed end (k_p^{point}). Values for polymerisation rates are taken from *in vitro* work [6]. The barbed end polymerisation probability in a single timestep is therefore

$$P_p^{\text{barb}} = k_p^{\text{barb}} C_{\text{G}} \Delta t, \tag{2.47}$$

and the pointed end polymerisation probability is similarly

$$P_p^{\text{pointed}} = k_p^{\text{point}} C_{\text{G}} \Delta t. \tag{2.48}$$

Polymerisation is limited to a maximum of one per end per filament per timestep.

Figure 2.8 shows snapshots of a simulation showing polymerisation of a single filament. The blue square off-centre is the nucleation region where the trimer was nucleated initially at the start of the simulation. The G-actin concentration is 1µM. The barbed end polymerisation rate k^{barb} is $12\mu\text{M}^{-1}\,\text{s}^{-1}$ and the pointed end polymerisation rate k^{point} is $1.3\mu\text{M}^{-1}\,\text{s}^{-1}$. The filament undergoes Brownian motion in a fluid of viscosity 0.1Pa s and temperature 300K. The desired subunit length l_s is 250nm. These values used for the polymerisation rates, viscosity, temperature and desired subunit length will be used in all future simulations unless stated otherwise. The length of the actin filament is to scale but the width is not. In all future simulation snapshot figures this scaling will be the case.

After a polymerisation event the subunits that have changed (all for rigid filaments, and just the final two for flexible filaments) are checked against neighbouring objects in the steric grid for any intersections. If any intersection has occured, the polymerisation move is reversed.

2.3.4.3 Depolymerisation

Depolymerisation can be thought of as the opposite of polymerisation, it is the removal of a single monomer off of either end of an actin filament. The details of how the lengths of subunits are shortened given a depolymerisation event is simply the opposite of a polymerisation event discussed in the previous section. Points are removed when the lengths of the final two subunits become less than $1.5l_s$.

If a depolymerisation event reduces the filament's length such that it becomes rigid, the filament needs to straighten. The unit vector describing the line between the first point and the last point is calculated and all the subunit's unit vectors are set to equal this. Points are then updated following these unit vectors, from the depolymerising end. After straightening the filament, the filament is checked against neighbouring objects in the steric grid for any intersections. If an intersection has occured, the depolymerisation and straightening move is reversed.

If a trimer undergoes a depolymerisation event it is 'dissociated', which is the opposite of nucleation, it is removed from the system.

Depolymerisation occurs with a flat rate k_d, which differs for the barbed end (k_d^{barb}) and pointed end (k_d^{point}). Values for depolymerisation rates are taken from *in vitro* work [6]. The barbed end depolymerisation probability in a single timestep is therefore

$$P_d^{\text{barb}} = k_d^{\text{barb}} \Delta t, \tag{2.49}$$

and the pointed end depolymerisation probability is similarly

$$P_d^{\text{point}} = k_d^{\text{point}} \Delta t. \tag{2.50}$$

Depolymerisation is limited to a maximum of one per end per filament per timestep.

2.3.4.4 Branching

Branching is the production of new filaments but unlike nucleation these are connected to preexisting filaments. When a branched filament is created it has a length of $2l_{\text{mon}}$, its pointed end is located inside the mother filament. The angle of the branched

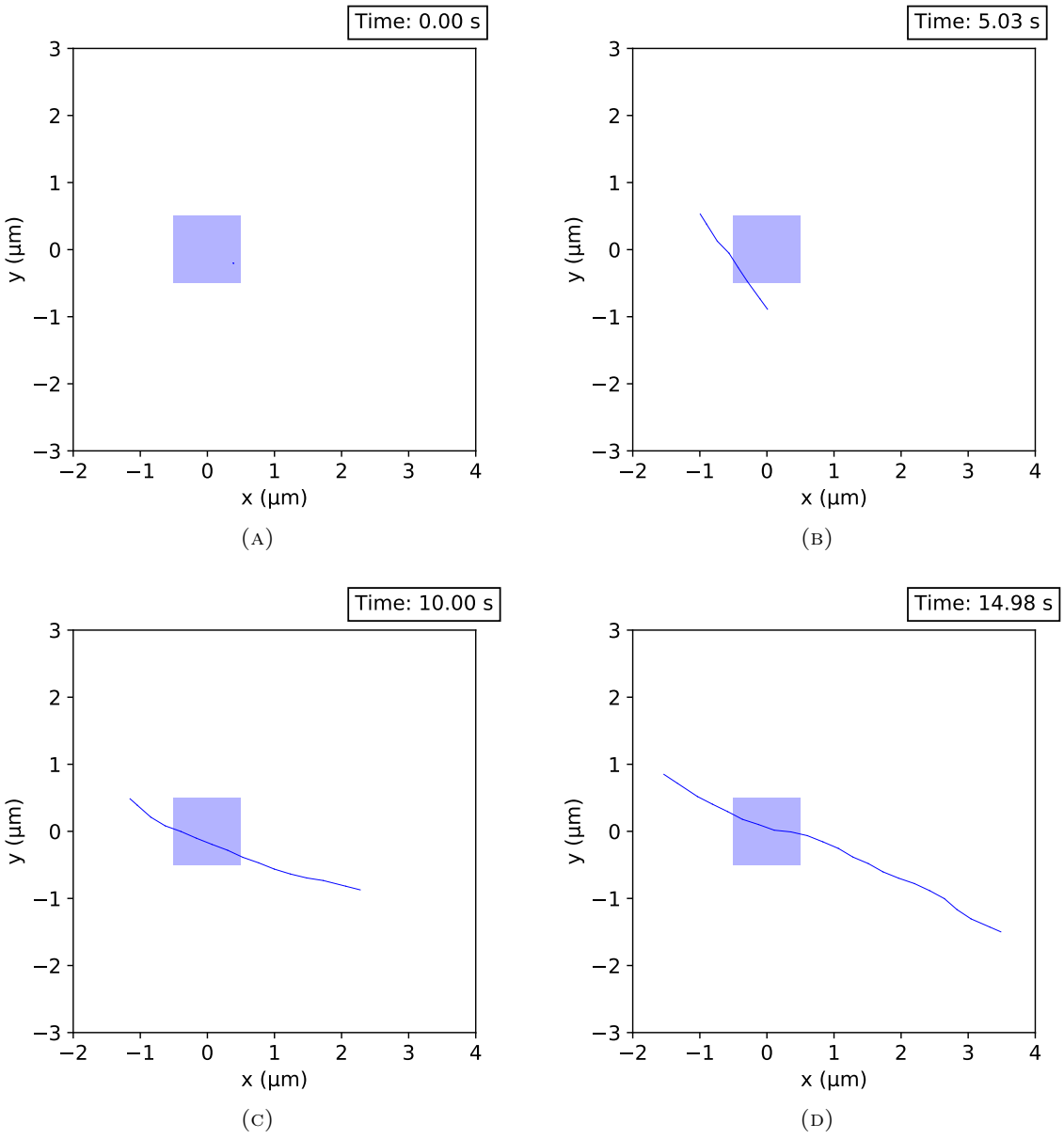
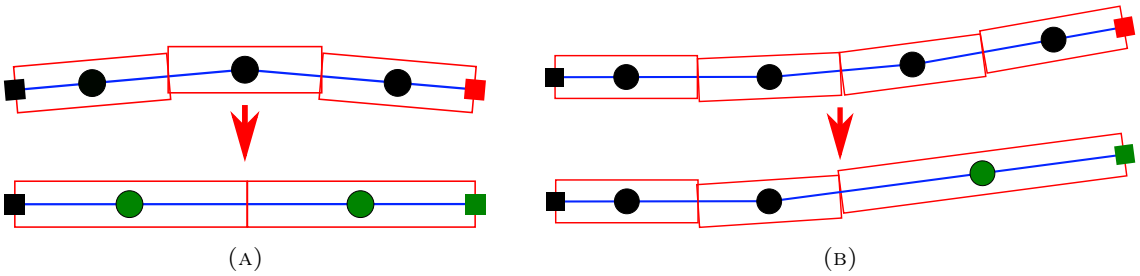


FIGURE 2.8: Snapshots of a simulation showing polymerisation of a single filament. A trimer is placed initially in the nucleation region (blue square), it then undergoes polymerisation at both ends. Time is shown in the top right. **(A)** The initial frame at 0 seconds, **(B)** 5 seconds, **(C)** 10 seconds and **(D)** the final frame at 15seconds. The filament's length is to scale but it's width is not.



(A) (B)

FIGURE 2.9: Diagram showing removal of points. **(A)** shows the removal of a point and the transition from a flexible filament to a long rigid filament, showing straightening of the filament. **(B)** shows the removal of a point from a flexible filament, this shows the straightening of the end.

filament from the relevant subunit of the mother is ±1.2 radians (corresponding to the characteristic branch angle reported in experiment [10]). A random number is drawn to determine which side of the mother the branch extends from.

For each monomer on a filament that is available to branch off (we will cover availability later), branching occurs with a rate per concentration of Arp2/3 complex per concentration of G-actin squared, k_{branch}. The branching rate in this form has been used before [110, 111], and we take the value of k_{branch} to be 5.3×10^{-4} $\mu\text{M}^{-3}\,\text{s}^{-1}$ from [110]. Each timestep the number of available monomers for each filament is calculated which gives a probability of creating a branch off that filament of

$$P_{\text{branch}} = N_{\text{avail}} k_{\text{branch}} C_G^2 C_A \Delta t, \tag{2.51}$$

where N_{avail} is the number of available monomers in the filament that can form a branch, C_A is the concentration of Arp2/3 complex.

N_{avail} is determined based on what is already attached to the filament and spacing parameters d_{br} which is the minimum distance between branches along a filament, and d_{cl} which is the minimum distance between crosslinks along a filament (crosslinks will be covered later in section 2.3.4.7). Branches form at discrete sites on the mother which correspond to the middle of each monomer along the filament. Branches can not be created on a monomer within a distance of d_{br} from a pre-existing branch or from the pointed end of the filament. This is shown in Figure 2.10 below. As branches form along the discrete sites, this distance has to correspond to an integer number of monomers, and it is defined as the number of monomers that sit between two adjacent branches. This branch spacing restriction is included because it would be expected that there is a physical steric hindrance effect from the Arp2/3 complex molecule, preventing two from binding close together. Arp2/3 complex molecules are 10-20nm in length (along the actin filament) [14, 112]. Another argument for a minimum branch spacing was made by [66] based on the observation that Arp2/3 complex binding is curvature sensitive [113], meaning there has to be some distance for the filament to curve slightly to allow for another Arp2/3 complex molecule to bind. Branch spacing has been measured in lamellipodia using electron tomography and shown to be frequently separated by multiples of 36nm, which corresponds to the helical repeat of actin. Branches that were more closely separated than this were found but were relatively rare and their existence was explained by the branches lying in different planes [23]. Since we are in one plane we therefore use a value of 13 monomers for d_{br} ($13l_{\text{mon}} = 35.1$nm). Therefore, a minimum of 13 monomers has to sit between branches, and between the pointed end of the filament and the branch

closest to the pointed end. If separation is defined as between the middle of one branch to the middle of the adjacent branch then using a d_{br} of 13 monomers results in a minimum branch separation of 37.8nm ($14l_{\text{mon}}$).

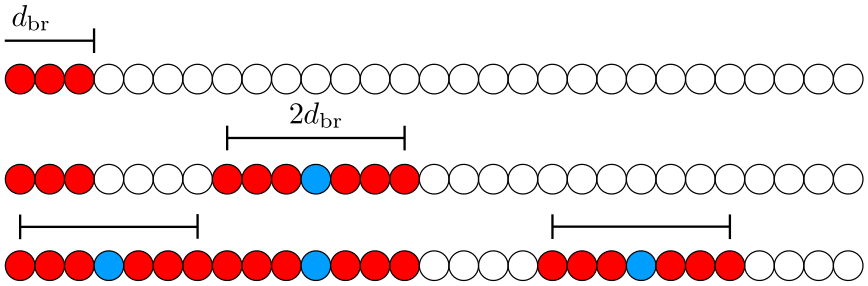


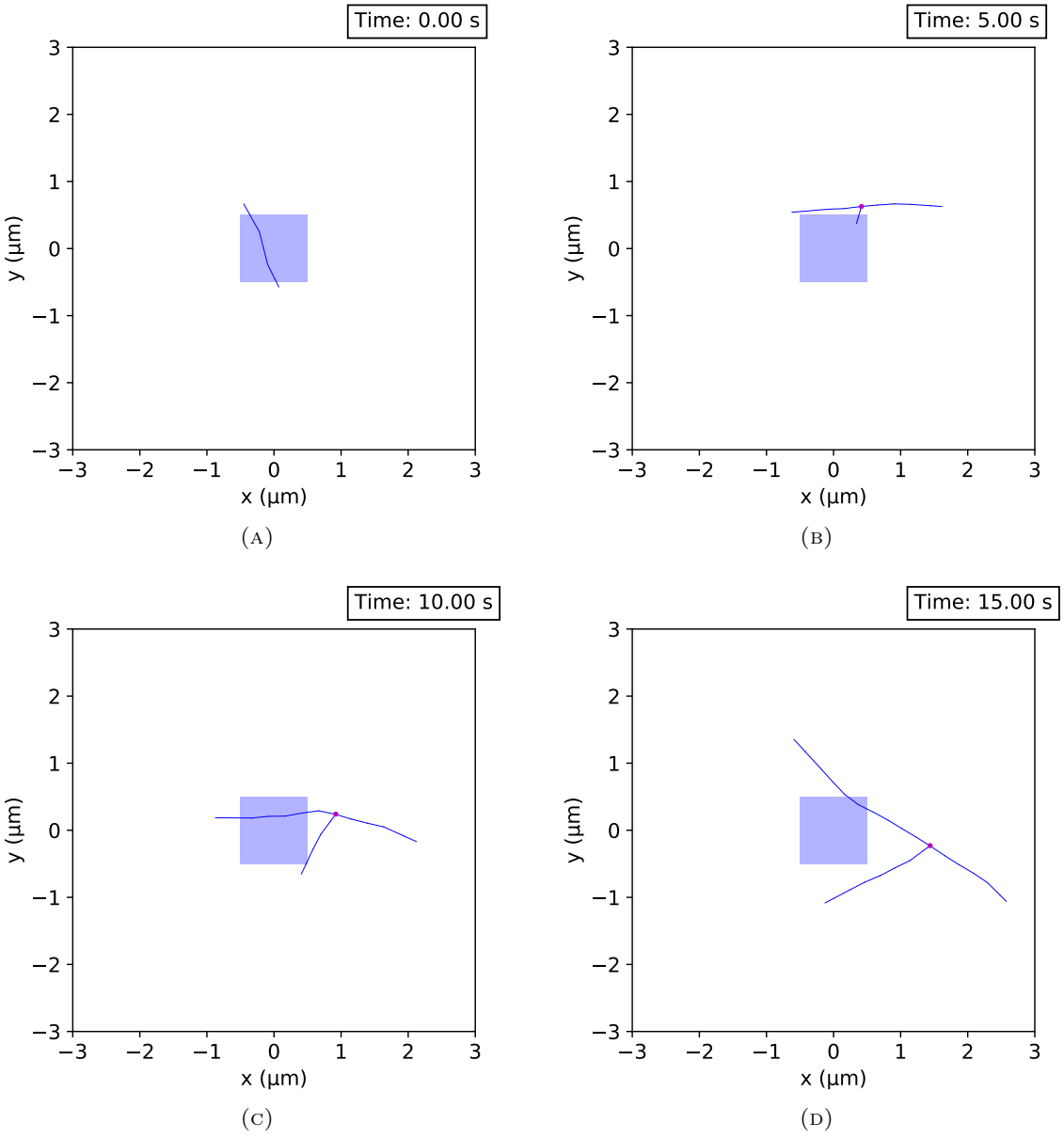
FIGURE 2.10: Diagram showing the discrete sites available for branching on a filament. Each circle represents a site and has a length of l_{mon}. White circles are available sites, branches can form here. Red circles are unavailable sites due to a nearby branch or the pointed end, the number of red circles either side of a branch is equal to the branch spacing length (which in this diagram is $3l_{\text{mon}}$) divided by l_{mon}. Blue circles are unavailable sites due to a branch occupying it.

Branches are capped at their pointed ends (capping will be covered in section 2.3.4.6) therefore they undergo (de)polymerisation at the barbed end only. If a branch with a length of $2l_{\text{mon}}$ is chosen to undergo a barbed end depolymerisation event, the branch is removed from the system. If a monomer in a mother filament with a branch attached to it is removed via depolymerisation then the branch is detached from the mother filament. The branch becomes a linear filament with the cap on its pointed end removed. Whenever branches detach (see section 2.3.4.5 for more) the discrete branching sites that were unavailable due to the branch, have to be made available again. For example, using Figure 2.10 the bottom filament, if the rightmost branch detaches, the seven unavailable sites must be made available again.

Figure 2.11 shows snapshots of a simulation showing branching off of a single filament. The blue square is the nucleation region where the initial filament is placed and initialised to a length of 500 monomers and in a bent configuration (see section 2.7 later on for this). This initial filament (and branch) undergoes polymerisation, depolymerisation and Brownian motion in a fluid of viscosity 0.1Pa s and 300K. The depolymerisation rates used are 1.4s^{-1} and 0.8s^{-1} for barbed and pointed ends respectively. These depolymerisation rates will be used in all future simulations unless stated otherwise. The G-actin concentration is fixed at 5µM and the Arp2/3 complex concentration is fixed at 50nM. The simulation is limited to a maximum of two actin filaments so further filaments do not branch. The small magenta circle shows the branchpoint or Arp2/3 complex.

As with nucleation, after the branch is created it is checked to see if it intersects any other object in the simulation, if it does then the branching event is cancelled and the branch removed.

Since the pointed end of a branch is in the mother filament this is a steric hindrance violation. This is the first example of an 'allowed exception' to steric hindrance. If both the mother and daughter are short rigid steric hindrance is ignored between mother and branch. If at least one of them is long rigid or flexible the branch ignores only the subunit of the mother its pointed end is on, and the two adjacent subs. This is to prevent long branches that loop round being able to cross or polymerise into their mothers (or vice versa).



(A) (B) (C) (D)

FIGURE 2.11: Snapshots of a simulation showing polymerisation and the creation of a single branch off a filament. The branchpoint/Arp2/3 molecule is represented as a magenta circle.

2.3.4.5 Branch detachment

In the previous section a scenario was highlighted where branches could detach from their mothers - when there is depolymerisation of the monomer the branch is attached to. Branch detachment has been observed to happen even when the mother filament is not depolymerising, as branches are 'metastable'. Certain proteins such as cofilin can also promote branch detachment [15, 16]. In the model, branch detachment happens with a flat rate k_{dbr} per branch. The probability of any branch detaching in a given timestep is therefore

$$P_{\text{dbr}} = k_{\text{dbr}}\Delta t. \tag{2.52}$$

Figure 2.12 shows snapshots of a simulation showing a branch detachment event. The simulation is very similar to the previous (Figure 2.11), as the same random number seed is used. All the conditions are the same except branch detachment occurs with a rate 0.1s^{-1}. Therefore Figure 2.12A and B are exactly the same as Figure 2.11 A and B. Branch detachment occurs between five and ten seconds. Once the branch is detached, the two linear filaments can not intersect each other.

Immediately after a branch detachment event there are two filaments that are crossing. The pointed end of the filament that was a branch still sits inside what was the mother filament. This is therefore an 'unallowed exception' to steric hindrance; it breaks the steric hindrance rules we have set. Most often, this will be solved by future Brownian motion or depolymerisation steps.

2.3.4.6 Capping and Uncapping

Capping is the process of blocking both polymerisation and depolymerisation from a particular end. Filaments originally have both ends uncapped (except for branches which are pointed end capped), and when/if an end undergoes capping it is then capped. Capped ends can be uncapped at a later time. Capping occurs with a rate per concentration of capping protein k_{cap}, and differs between barbed end ($k_{\text{cap}}^{\text{barb}}$) and pointed end ($k_{\text{cap}}^{\text{point}}$). These rates are taken from *in vitro* work, barbed end capping is more well studied [114] than pointed end capping [110]. The probability of barbed end capping in any given timestep is therefore

$$P_{\text{cap}}^{\text{barb}} = k_{\text{cap}}^{\text{barb}} C_C \Delta t, \tag{2.53}$$

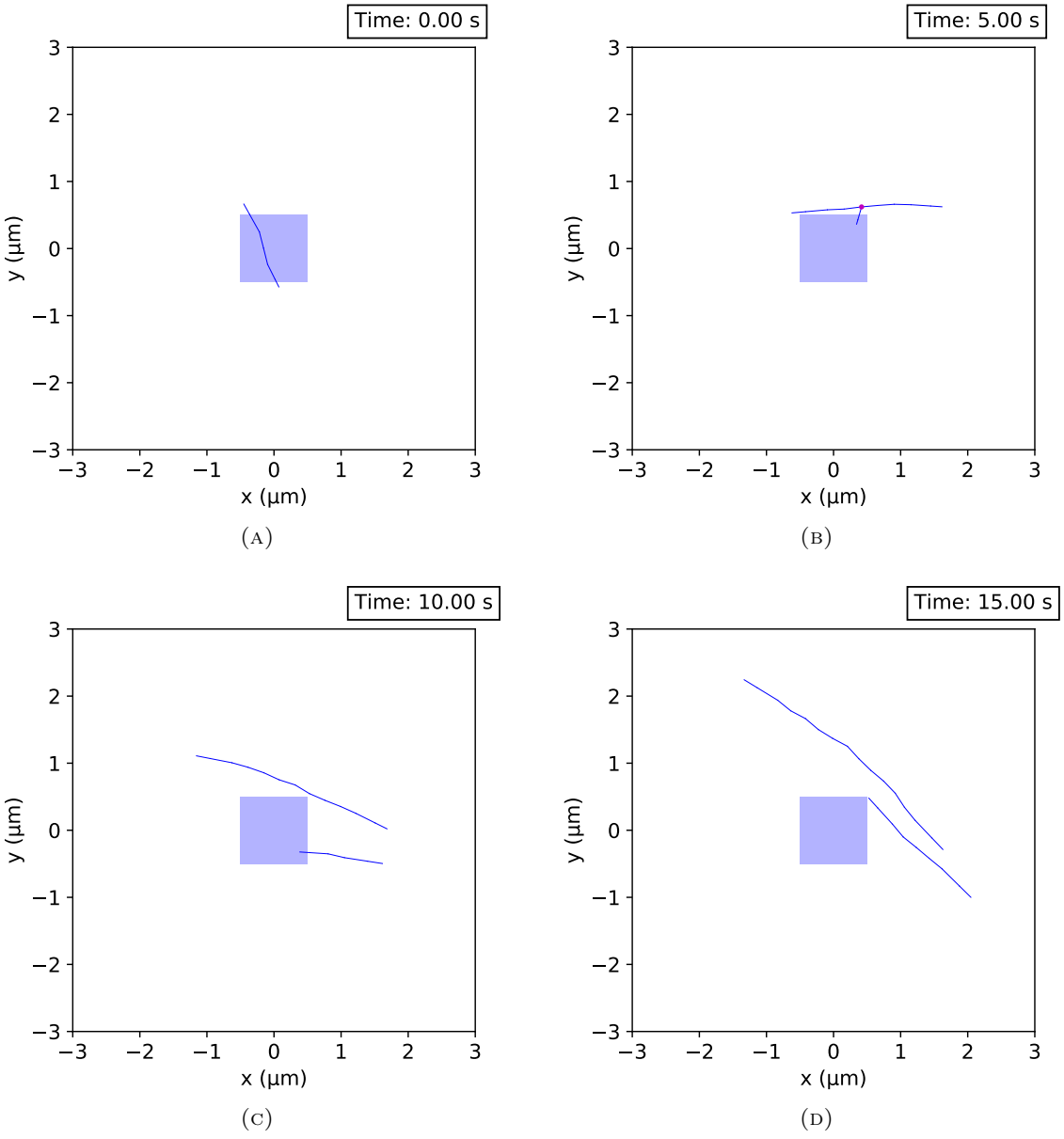
and for pointed end capping it is

$$P_{\text{cap}}^{\text{point}} = k_{\text{cap}}^{\text{point}} C_C \Delta t, \tag{2.54}$$

where C_C is the concentration of capping protein.

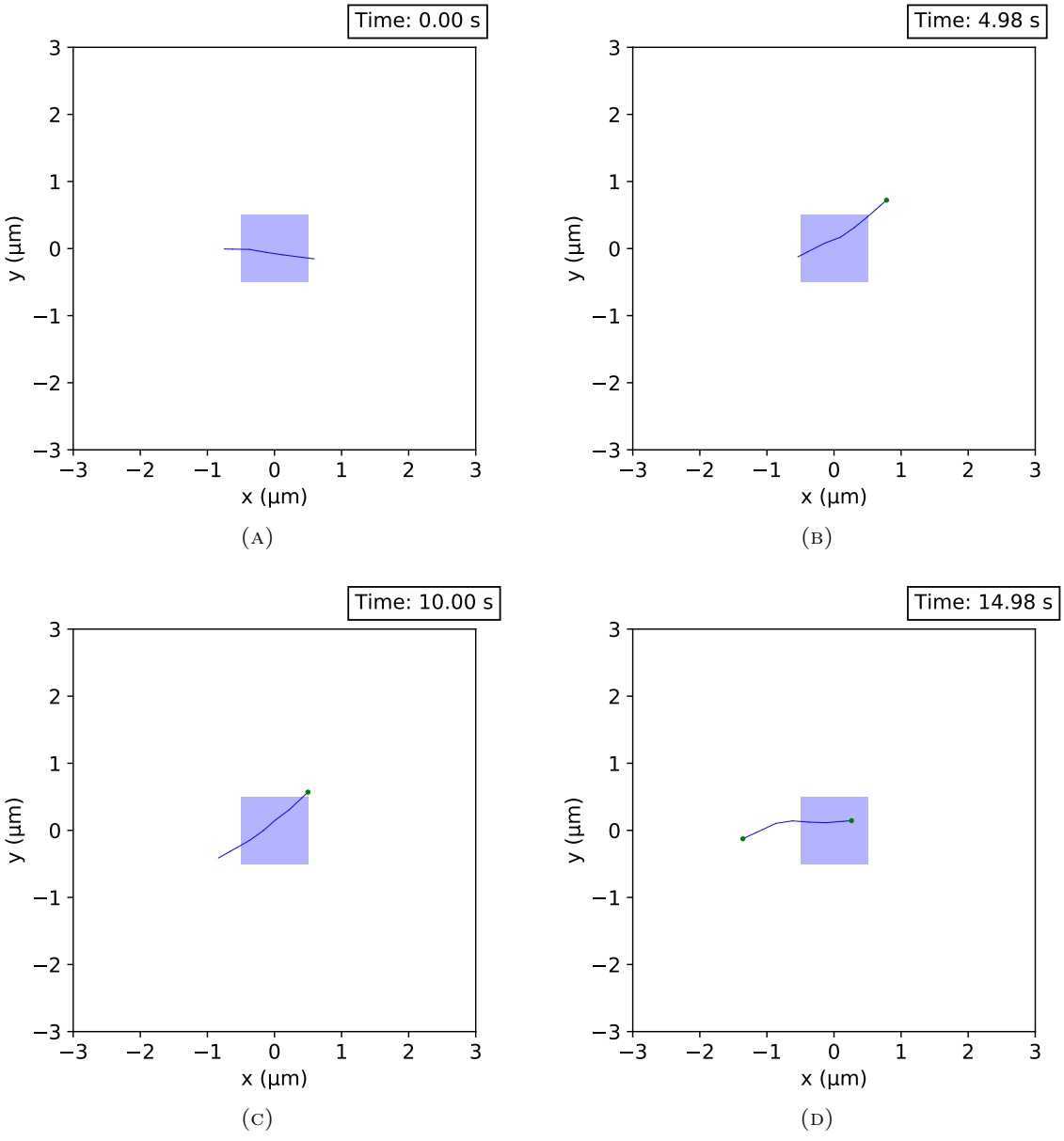
Figure 2.13 shows snapshots of a simulation showing capping of a single filament. The initial filament is placed inside the blue nucleation region as previously shown. The rates are $3.5\mu\text{M}^{-1}\,\text{s}^{-1}$ and $0.8\mu\text{M}^{-1}\,\text{s}^{-1}$ for barbed and pointed end capping respectively [110, 114]. The G-actin concentration is fixed at $5\mu\text{M}$ and the capping protein concentration is fixed at $40\text{n}\text{M}$. All other conditions as previous. The small green circles are the capping proteins associated with the capped ends of the filament.

Once in a capped state, ends can uncap again. Uncapping occurs with a flat rate k_{uncap} for barbed ($k_{\text{uncap}}^{\text{barb}}$) and pointed ends ($k_{\text{uncap}}^{\text{point}}$). Again, barbed end uncapping dynamics are much more well studied and the rate is taken from [114]. The pointed end capping rate is based on the assumption given in [110] that pointed end uncapping would lead to branch detachment, therefore $k_{\text{uncap}}^{\text{point}}$ is set to be equal to k_{dbr}. The probability of uncapping at the barbed end per barbed end capped filament in a given



(A) (B) (C) (D)

FIGURE 2.12: Snapshots of a simulation showing polymerisation, the creation of a single branch off a filament and then the detachment of that branch.



(A) (B) (C) (D)

FIGURE 2.13: Snapshots of a simulation showing polymerisation and capping of a single filament. Capped ends are shown as small green circles.

timestep is therefore

$$P_{\text{uncap}}^{\text{barb}} = k_{\text{uncap}}^{\text{barb}} \Delta t, \tag{2.55}$$

and for pointed end uncapping

$$P_{\text{uncap}}^{\text{point}} = k_{\text{uncap}}^{\text{point}} \Delta t. \tag{2.56}$$

2.3.4.7 Crosslinking and unlinking

Crosslinking is the creation of an attachment, called a crosslink, between two filaments. For a crosslink to form the filaments need to have available sites that are within a certain distance of each other. The available sites on a filament are the same as for branches: a crosslink cannot form within d_{br} of a branch or within d_{cl} of a crosslink. A crosslink can also not form within either d_{br} or d_{cl}, whichever is greater, from the pointed end. Figure 2.14A is a diagram to illustrate the crosslinking availability and is similar to Figure 2.10 for branches. Figure 2.14B shows a scenario where both crosslinking and branching is on, crosslink and branches therefore compete for space.

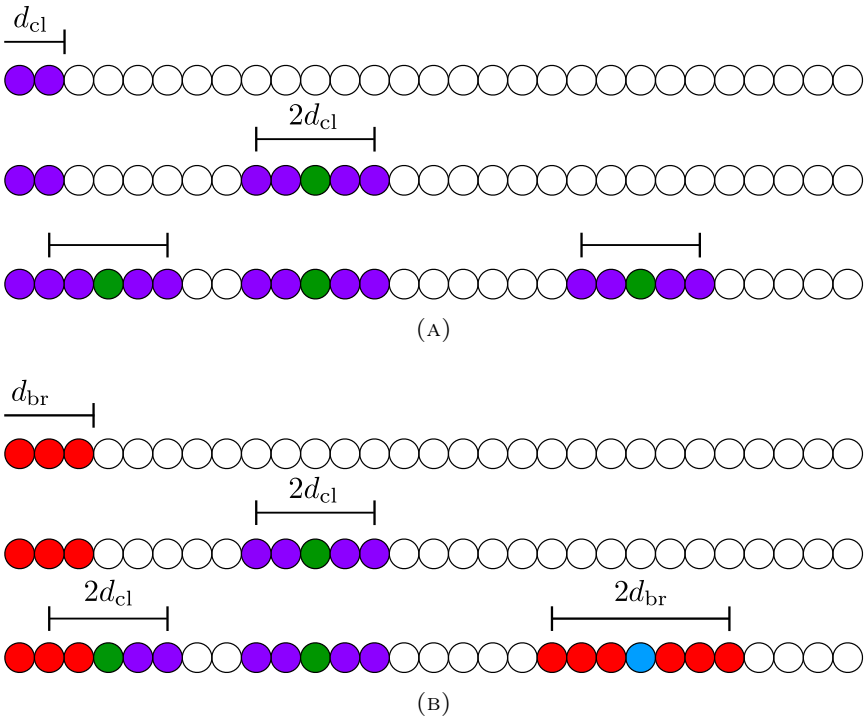


FIGURE 2.14: Diagram showing discrete sites available for crosslinking. Crosslink spacing is $2l_{\text{mon}}$ in this example. Purple circles are unavailable sites due to a nearby crosslink or pointed end. Green circles are unavailable sites due to the presence of a crosslink there. **(A)** Situation where crosslinking is on, but branching is not. **(B)** expanding on Figure 2.10, this figure includes spacing restrictions from both branches and crosslinks.

Crosslinking forms with a rate per possible crosslink (k_{cl}), as crosslink formation has this geometric element. Figure 2.15 is a flow chart illustrating the algorithm used for creating crosslinks. Solid black lines show the progression of the algorithm. Solid green arrows depict the flow from a 'Yes' answer, dashed red arrows depict the flow from a 'No' answer. Dotted blue arrows depict the end of the iteration of the current loop. For example the first dotted blue arrow shows when all the nearby subunits to subunit j have been checked, we repeat for the next subunit, so j increases by one ($j \mathrel{+}= 1$). In general, the operation $a \mathrel{+}= 1$ is shorthand for an increase of the variable

a by one. q is a random number drawn from a uniform distribution between 0 and 1. The algorithm for determining the formation of a crosslink is described below.

In every timestep each filament is evaluated to see if it undergoes crosslinking. First for every available site on the filament the steric grid is used to find any subunits belonging to a different filament that could be within a distance (s_{cl} : the crosslink distance) from it. All available sites on that subunit are then obtained and for each one, a point-to-point distance check is carried out. If the point-to-point distance between the two sites on each filament is less than $s_{\text{cl}} + 2r_{\text{steric}}$ then a crosslink could form there and so the pair of sites is added to a list of possible crosslinks that could form. These steps are repeated until we have a complete list of all the possible crosslinks that could form from that filament in that timestep. Once we have a list of potential crosslinks we use a MC method to determine if one of those will form, this is described below. When moving on to the next filament(s), any potential crosslink that involves a filament already checked is ignored, as this would be double counting.

The probability of forming a crosslink on a given filament is

$$P_{\text{cl}} = k_{\text{cl}} N_{\text{poss-cl}} \Delta t, \tag{2.57}$$

with $N_{\text{poss-cl}}$ being the number of possible crosslinks for that filament in that timestep.

When a filament crosslinks the availability of neighbouring monomers are adjusted, just as in formation a branch, with monomers being within d_{cl} of the crosslinked monomer being designated as unavailable. As discussed in sections 2.3.2.2 and 2.3.2.4 the presence of a crosslink alters the behaviour of motion of the filaments. With short rigid filaments, the crosslink distance is kept the same as the original distance when it formed so less than or equal to s_{cl}. For long rigid and flexible filaments, the crosslink is modelled as a spring with rest length s_{cl}.

Crosslinks are another example of an 'allowed exception' to steric hindrance. Steric hindrance is ignored between the subunit one side of the crosslink is on and the two adjacent subunits, and the subunit the other side of the crosslink is on and the adjacent subunits.

Figure 2.16 shows snapshots of a simulation showing crosslinking between two filaments. Both filaments are placed inside the nucleation region and initialised with a length of 500 monomers. The crosslinking rate per potential crosslink is k_{cl} is 10^{-3}s^{-1}, the crosslink distance s_{cl} is 100nm, the crosslink spacing d_{cl} is 37 monomers ($\sim$100nm). The G-actin concentration is fixed at 5µM. The crosslinks are shown as green rectangles with a circle at their centre.

Unlinking is the reverse of crosslinking where a pre-existing crosslink is removed. Unlinking happens with a rate k_{unlink} per crosslink. The probability of any given crosslink being removed in a given timestep is therefore

$$P_{\text{unlink}} = k_{\text{unlink}} \Delta t. \tag{2.58}$$

After unlinking, the list of available sites is updated.

Like in branch detachment, unlinking is another example of an 'unallowed exception' to steric hindrance. Unlike branch detachment it may not cause a steric hindrance violation, as crosslinks do not always produce filaments that overlap.

Unlinking can also happen automatically as in debranching, when the monomer that is crosslinked is removed due to depolymerisation.

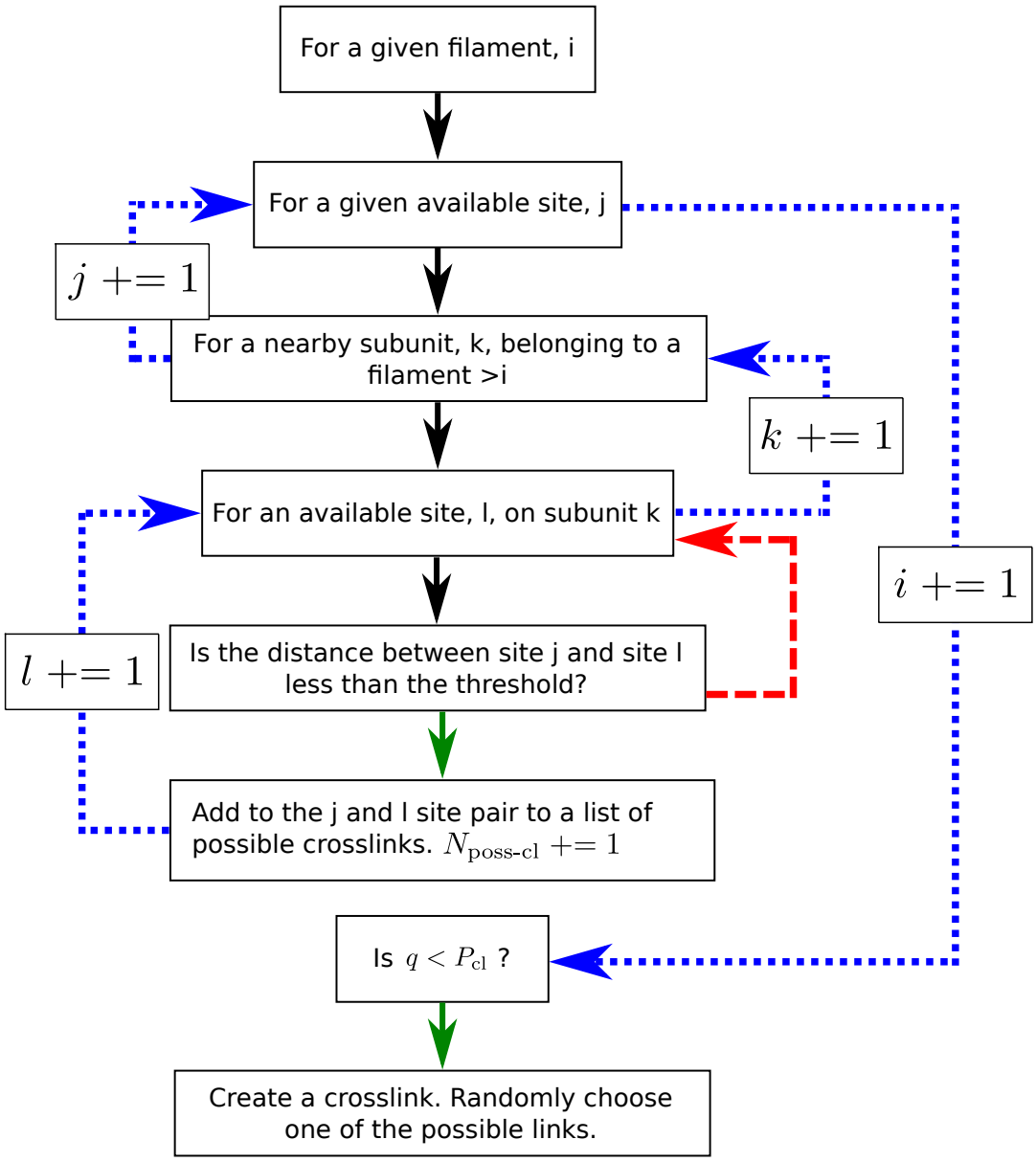
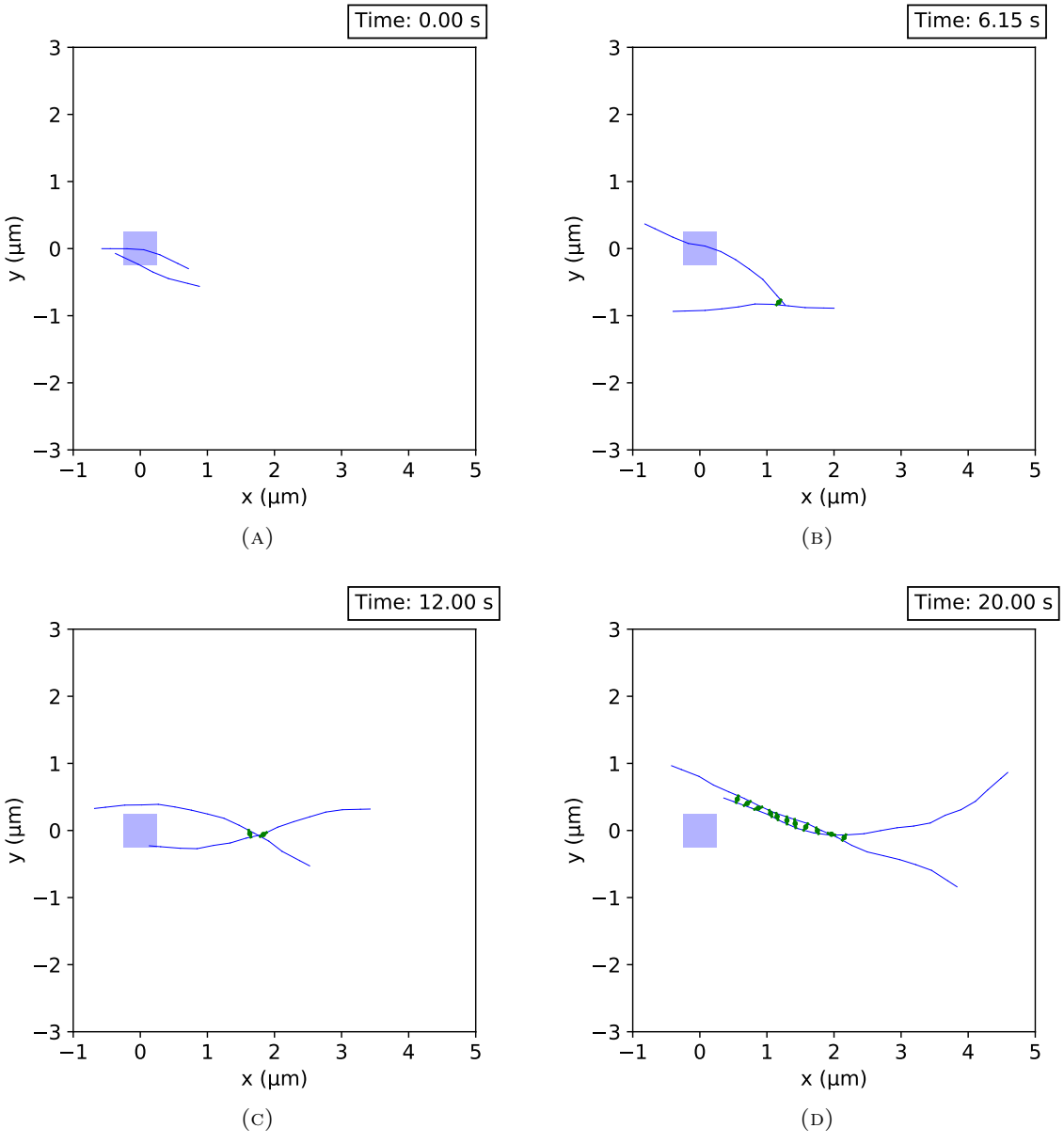


FIGURE 2.15: Flow chart describing the algorithm used for creating crosslinks. Solid black lines show the progression of the algorithm. Solid green arrows depict the flow from a 'Yes' answer from the question, whereas dashed red arrows depict the flow from a 'No' answer. Dotted blue arrows depict the end of the iteration of the current loop. In general, the operation a += 1 is shorthand for an increase of the variable a by one. q is a random number drawn from a uniform distribution between 0 and 1.



(A) (B) (C) (D)

FIGURE 2.16: Snapshots of a simulation showing polymerisation and crosslinking of two filaments. Crosslinks are shown in green as thick lines with a circle at their centre.

2.3.4.8 Severing

Severing is the breaking of a single filament into two filaments. Severing occurs with a rate per filament length k_{sever} based on *in vitro* studies [115, 116]. The probability of severing a filament in a given timestep is therefore

$$P_{\text{sever}} = k_{\text{sever}} L \Delta t. \tag{2.59}$$

If a filament is to sever, first a severing point along the filament must be determined. The severing point must correspond to the end of a monomer to ensure the two resultant filaments have an integer number of monomers. A random integer is chosen from a uniform distribution which gives the sever point monomer. The filament is resized to be the filament on the pointed end side of the sever point and a new filament is created to be the filament on the barbed end side of the sever point. If either of the two resulting filaments is less than $3l_{\text{mon}}$ in length it is dissociated. All attached branches and crosslinks on the barbed end side of the sever point need updating so they 'belong' to the new barbed end filament. If either of the two resulting filaments is less than $3l_{\text{s}}$ then they need to be straightened so that they are rigid filaments. If the new filament(s) is flexible, the ends have to be straightened such that the final two subunits are in line with each other.

Figure 2.17 is a diagram illustrating a severing event. The first, red arrow show the severing event taking place. The black cross in the first top depiction of the filament marks the sever point. Severing produces another 'unallowed exception' to steric hindrance, since immediately after the severing event, the two filaments break steric hindrance[1]. It can be seen that the filament end has straightened when severing has taken place. The second, green arrow shows some time later after Brownian motion has corrected the steric hindrance violation.

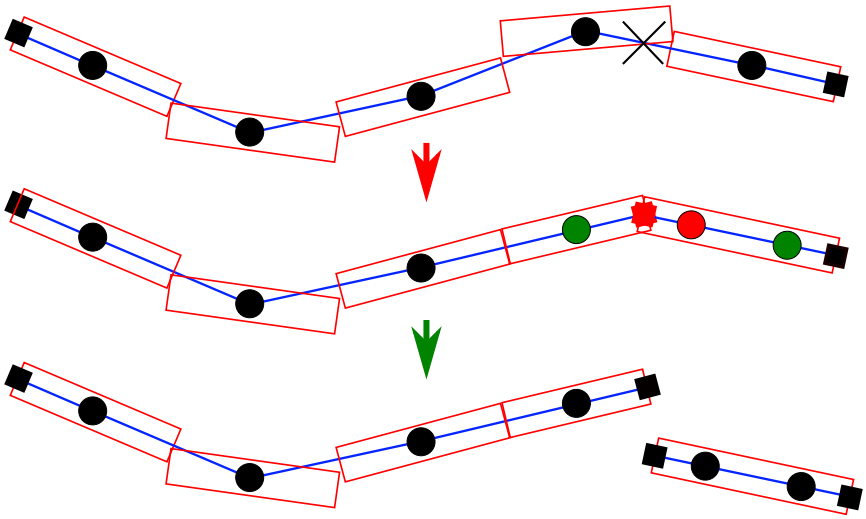


FIGURE 2.17: Diagram representing a severing event where time progresses from top to bottom. The black cross shows the sever point: where the filament will be cut. The middle depiction of the actin filament shows the sever event taken place, this would result in a steric hindrance break. The third one shows some time later when the filament(s) diffuses away.

[1]Steric hindrance takes into account the radius of the filaments (section 2.3.3). Filament subunits are treated as rods with a spherical cap and a radius r_{steric}. Therefore a severing event breaks steric hindrance.

Figure 2.18 shows snapshots of a simulation showing a severing event of a single filament. The severing event takes place between 5.37 (Figure 2.18B) and 5.42 seconds (Figure 2.18C). The severing rate per unit length k_{sever} is $0.02\mu\text{m}^{-1}\,\text{s}^{-1}$.

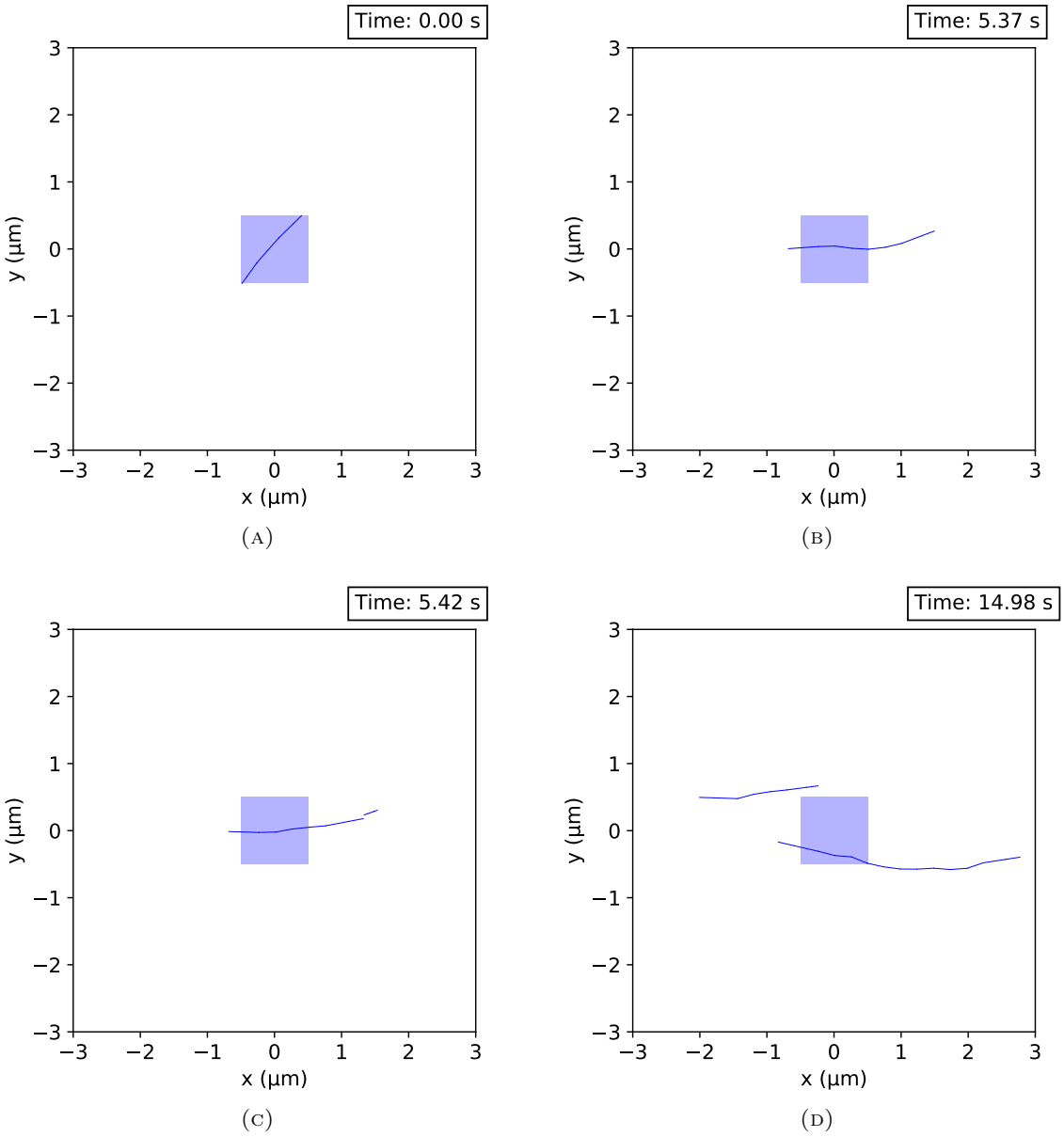


FIGURE 2.18: Snapshots of a simulation showing polymerisation and severing of a single filament, forming two filaments. **(B)** is just before the sever event takes place and **(C)** just after.

2.3.5 Spatial regions

In section 2.3.4.1 we mentioned the use of 2D regions in space in which nucleation of filaments occur. In this section we will discuss the use of regions in the model which expand to more processes than just nucleation.

Regions can confine nucleation, branching, capping, anti-capping (protection from capping) and severing in space. Regions can be rectangular/square, circles or rings (rings just in the case of nucleation). Rectangular regions are given a unit vector to

describe their direction, meaning they do not have to be Cartesian axis aligned. Nucleation regions are necessary in any simulation, since the nucleation rate is strictly a rate density, and is dependent on the area of the region. Regions other than nucleation are optional.

For branching, the branching rate is dependent on the number of available (polymerised) monomers in the filament (see section 2.3.4.4 and Figure 2.10). With a branching region there is an extra requirement on polymerised monomer sites for them to be available; they have to be within a branching region. Any polymerised monomers outside any branching regions are unavailable for branching just as if they were too close to another branch for example.

For capping, only filament ends within a capping region can cap. For anti-capping any filament ends within an anti-capping region cannot cap.

For severing, the severing rate is dependent on the length of the filament (see section 2.3.4.8). With a severing region this length is now the length of the filament inside any severing regions. The severing monomer chosen must be inside a severing region.

Figure 2.19 shows snapshots of a simulation showing the effect of several spatial regions. The blue region (rectangle in the centre) is a nucleation region just as we have previously had. The nucleation rate density is $5\times10^{-4}\mu m^{-2}\,\mu M^{-3}\,s^{-1}$ and the area of the nucleation region is $5\mu m^2$. The magenta region (left rectangle) is a branching region. The Arp2/3 complex concentration is fixed at 20nM. The green region (right rectangle) is a capping region. The capping protein concentration is fixed at 200nM. The red regions (top and bottom rectangles) are severing regions. The severing rate per length is $1\mu m^{-1}\,s^{-1}$. Polymerisation and depolymerisation happens everywhere. The G-actin concentration is fixed at 10µM.

2.4 Membrane

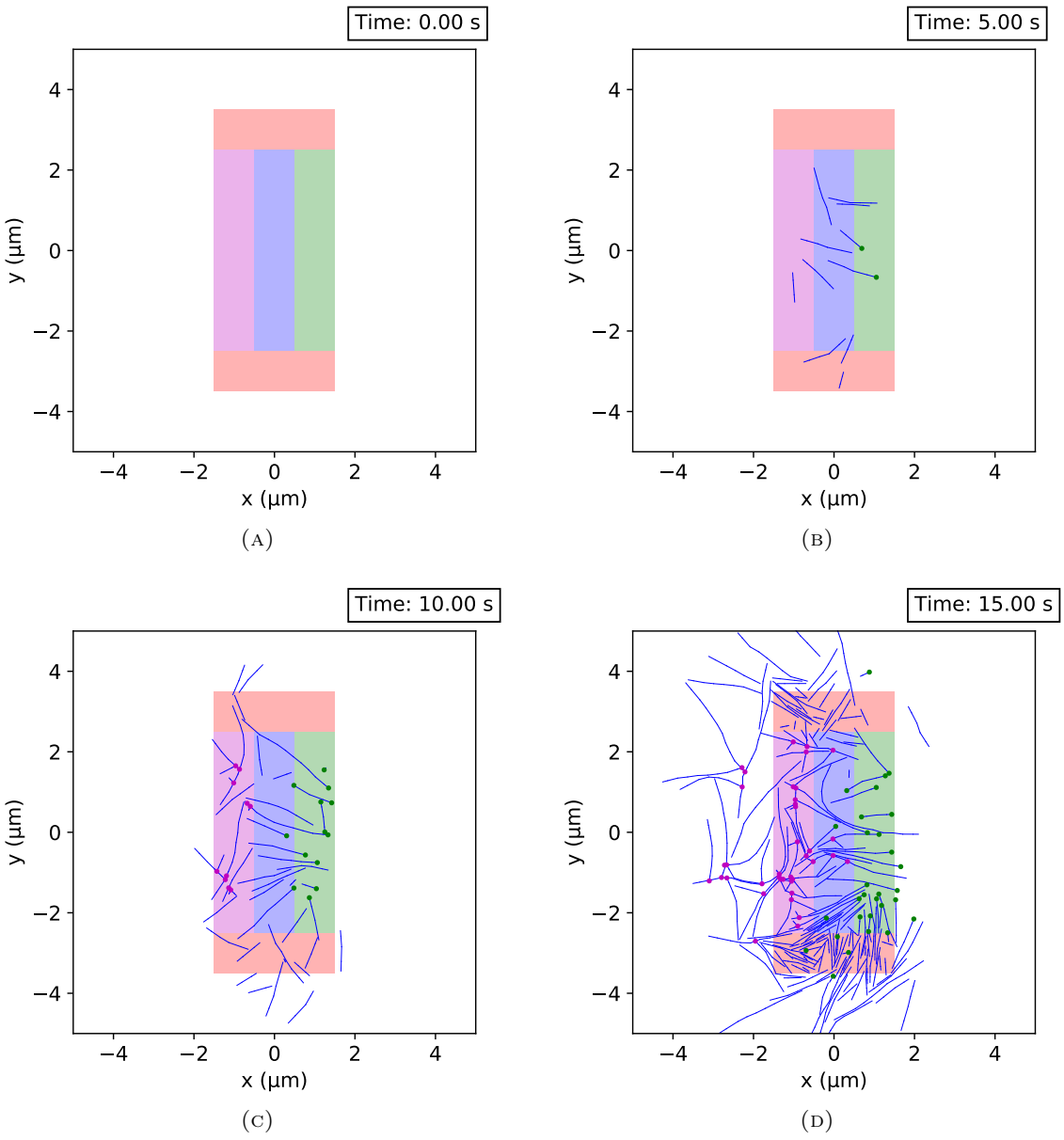
When investigating force generation in phagocytosis by actin filaments we need a membrane for the actin to push against. We started with looking at the simplest case: a one-dimensional wall, and later incorporated a more complex representation of a cell membrane in two dimensions.

2.4.1 Simple wall

The simple wall is a infinitely thin rigid line orientated parallel to the x axis and is characterised by a length, a position in the y direction and an external force which acts in the negative y direction. The wall moves in the positive y direction only by actin filaments pushing it through polymerisation events. Close to the wall the polymerisation is modulated by an Boltzmann factor, the polymerisation probability in a given timestep is therefore

$$P_p = k_p \exp\left(\frac{-f_{\text{wall}} d_{\text{wall}}}{k_B T}\right) C_{\text{G}} \Delta t, \tag{2.60}$$

where f_{wall} is the magnitude of the load force of the wall, d_{wall} is the distance the wall would move given polymerisation of the filament. Therefore $f_{\text{wall}} d_{\text{wall}}$ is the work required to push the wall a distance of d_{wall}. To clarify, f_{wall} is a parameter. Any polymerisation event that would intersect the wall can go ahead, the wall just moves accordingly. The wall always sits on top of the leading actin filament, it can move in the negative y if there is space to do so. There is steric hindrance between the



(A) (B) (C) (D)

FIGURE 2.19: Snapshots of a simulation showing the effect of different regions. The blue region is a nucleation region, the magenta is a branching region, the green is a capping region and the red regions are severing regions. Polymerisation and depolymerisation are allowed to happen everywhere.

wall and actin filaments undergoing Brownian motion. Actin filaments cannot move beyond the wall, any Brownian dynamics move that would result in this is rejected.

Spatial regions (section 2.3.5) can be attached to the wall and move with it. For example, a nucleation region could be attached to the wall which then moves up with the wall meaning actin filaments always nucleate in that region close to the wall.

Figure 2.20 shows snapshots of a simulation showing actin filaments pushing on a simple wall. The wall is shown as a thin red horizontal line. The wall's load force f_{wall} is 50pN. Both a nucleation region and a branching region are present and cover the width of the frame (10µm) and 100nm in height. The regions are attached to the wall and so as the wall moves up, the regions move up too. The nucleation rate density is $5{\times}10^{-3}\mu\text{m}^{-2}\,\mu\text{M}^{-3}\,\text{s}^{-1}$. The Arp2/3 complex concentration is fixed at 100nM.

In Figure 2.20 there is a onset time before the wall begins to move. The filaments must form an underlying network so that new filaments can push off that network and polymerise against the wall. The onset time given these conditions is around thirty seconds.

Similar methods have been used to investigate actin force generation before [66, 68].

2.4.2 Membrane

Our membrane is modelled in a similar way to our actin filaments, in that it is represented as a series of points. However, the start point and the end point are connected by an extra subunit; creating a loop. The membrane is initialised with an overall length and the points are generated along a circular path. Figure 2.21 is a diagram of our membrane model and contains similar concepts to diagrams for actin filaments (e.g. Figure 2.2). In Figure 2.21 there are no 'endpoints' since the membrane is a loop. Points are shown as black circles and move experiencing drag as if they were the rods which are shown as red rectangles. Subunits connecting the points are shown as blue lines.

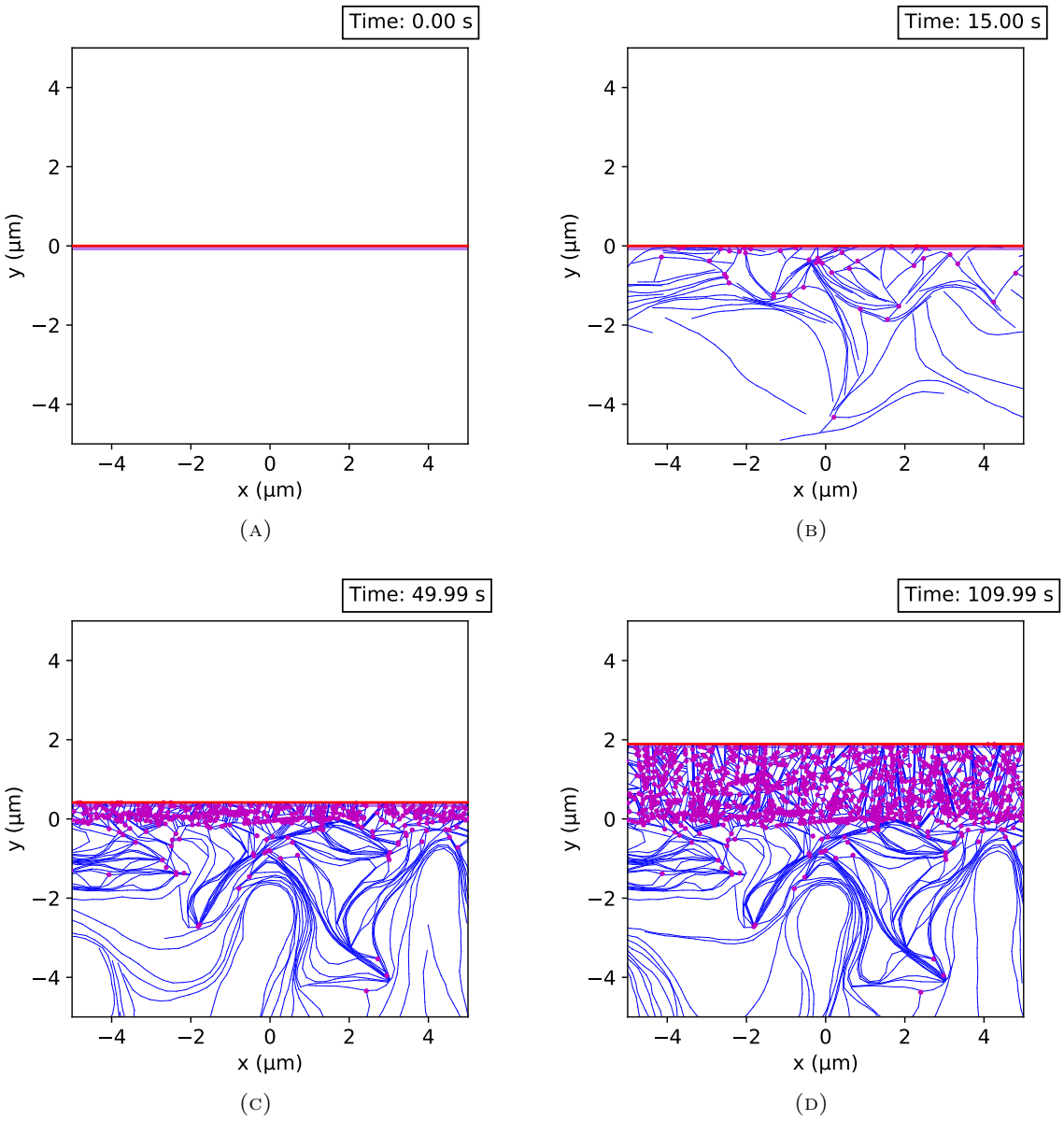
The same Brownian dynamics algorithm described in section 2.3.2.3 is used but the membrane has one additional constraint connecting the final point and the first point. So the matrix $\mathbf{B}_t$ for a membrane has N rows (for each constraint) and $2N$ columns (for x and y components of each point), where N is the number of points in the membrane. All points in the membrane are used in the Brownian dynamics algorithm unlike with actin where the two endpoints are ignored. The thickness of the membrane, s_{mem}, is 5nm [1]. s_{mem} is used for both Brownian dynamics and steric hindrance.

We want our membrane to be able to bend with a physical bending rigidity, κ_b. However since we are in 2D our membrane is a 1D object, as are our actin filaments. Therefore we require an effective persistence length. We make the assumption of spherical symmetry and choose a persistence length such that the bending energy of our circle representing a cross section, is the same as the bending energy of a sphere with the same radius.

We therefore need a length which we call w, such that

$$l_P = \frac{\kappa_b w}{k_B T}, \tag{2.61}$$

where κ_b is the bending rigidity of a membrane.



(A) (B) (C) (D)

FIGURE 2.20: Snapshots of a simulation show an actin network overcoming a load force to push a wall. The wall (thin horizontal red line) has a load force of 50pN in the negative y direction. The filaments nucleate in a thin (100nm thick) region below the wall. Branches also form in this region. These regions move up with the wall. **(A)** The initial condition, no filaments are present. **(B)** 15 seconds in. Filaments have nucleated and they begin to polymerise, branch and interact sterically with one another. **(C)** 50 seconds. The filaments have formed a network and have begun to push the wall up. Pushing is achieved through polymerisation of filaments into the wall. **(D)** The final frame (110 seconds). The wall has continue to move upwards slowly.

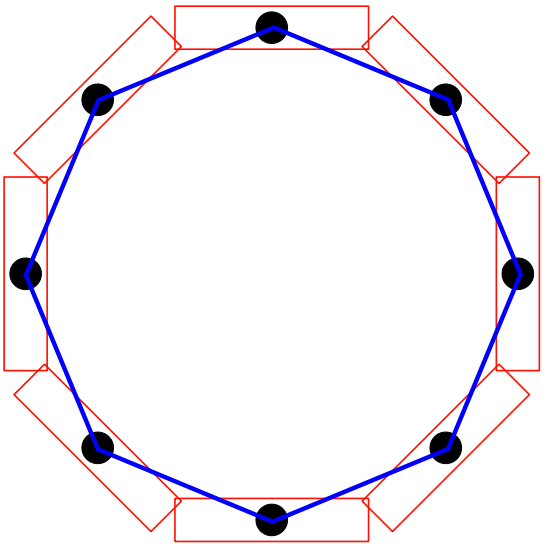


FIGURE 2.21: Diagram showing a representation of our membrane model. This uses the same diagrammatic forms that Figure 2.2 did to depict actin filaments. More explicitly this is its use of points (black circles), rods (red rectangles) and subunits (blue lines). Membranes in simulations will typically have many more points than what is shown here.

To find w let's consider the bending energy of a two-dimensional membrane in three-dimensional space.

$$E_{\mathrm{b}}^{\mathrm{2D}} = \int_A \frac{\kappa_b}{2} \left(\frac{1}{R_1} + \frac{1}{R_2} \right)^2 dA, \tag{2.62}$$

where R_1 and R_2 are the radii of the principal curvatures defined as the maximum and minimum curvatures [74, 75]. A is the surface area of the membrane we are integrating over. We neglect the Gaussian curvature and spontaneous curvature since R_1 and R_2 will be large compared to the bending fluctuations we are interested in. For the Gaussian curvature as $1/R_1$ and $1/R_2$ are small then $1/R_1R_2$ is very small and can be neglected. For spontaneous curvature as the membrane is locally flat we can neglect spontaneous curvature. Through our assumption of spherical symmetry $R_1 = R_2 = R$, we have

$$E_{\mathrm{b}}^{\mathrm{2D}} = 2\kappa_b \int_A \frac{1}{R^2}\, dA. \tag{2.63}$$

The area element dA can be written in spherical polar coordinates in terms of angle elements $d\theta$ and $d\phi$. The angle in the (xy) plane of the simulations is ϕ and the angle out of the plane is θ.

$$dA = R^2 \sin(\theta)\, d\theta\, d\phi, \tag{2.64}$$

and we can consider an arc length increment ds where $ds = R\, d\phi$, therefore

$$dA = R \sin(\theta) d\theta\, ds. \tag{2.65}$$

Now for a sphere we can rewrite equation 2.63 as

$$E_{\mathrm{b}}^{\mathrm{2D}} = 2\kappa_{\mathrm{b}} \int_0^{2\pi R} \int_0^{\pi} \frac{1}{R} \sin(\theta)\, d\theta\, ds, \tag{2.66}$$

and as R is constant for a sphere this integral becomes

$$E_{\mathrm{b}}^{\mathrm{2D}} = \frac{2\kappa_{\mathrm{b}}}{R} [s]_0^{2\pi R} [-\cos(\theta)]_0^{\pi} = 8\pi\kappa_b, \tag{2.67}$$

$8\pi\kappa_b$ is the bending energy of spherical vesicle, and is independent of its radius [74, 117].

Now with a one-dimensional membrane we just integrate over the length s, therefore replacing the area integral as follows

$$\int_A dA \equiv w \int_L ds, \tag{2.68}$$

which assumes w is constant in s. So using equation 2.65, we have

$$\int_0^{2\pi R} \int_0^{\pi} R \sin(\theta)\, d\theta\, ds \equiv w \int_0^{2\pi R} ds, \tag{2.69}$$

therefore

$$w = \int_0^{\pi} R \sin(\theta)\, d\theta = 2R \tag{2.70}$$

assuming R is constant in θ.

If we now look at the bending energy of our circle we have

$$E_{\mathrm{b}}^{\mathrm{1D}} = 2\kappa_{\mathrm{b}} w \int_0^{2\pi R} \frac{1}{R^2}\, ds, \tag{2.71}$$

where again R is constant as it is a circle.

$$E_{\mathrm{b}}^{\mathrm{1D}} = \frac{2\kappa_{\mathrm{b}} w}{R^2} [s]_0^{2\pi R} = 4\pi\kappa_b \frac{w}{R} = 8\pi\kappa_b, \tag{2.72}$$

since $w = 2R$. This is the same result as for a sphere, and so equation 2.61 becomes

$$l_{\mathrm{P}} = \frac{2R\kappa_{\mathrm{b}}}{k_{\mathrm{B}}T}, \tag{2.73}$$

which is what we use for our effective persistence length. We used the letter w for 'width' as for a sphere it is the diameter. A similar method was used for a one-dimensional membrane in [67], where t_{lam} is the thickness of the lamellipodia. However in [67], the curvature in θ was neglected entirely. This is assuming the membrane is flat with a constant thickness, which is suitable for a lamellipodia. Our case is different and we cannot neglect the curvature in θ.

Our membrane is inextensible, which while this is a valid assumption for actin filaments, it is not completely true for membranes. However real membranes can only stretch a small amount before breaking, corresponding to a few percent in area

increase [118, 119]. We therefore argue that having the membrane inextensible is a good control case. Membranes also have reservoirs in form of folds/wrinkles which could be tens of percent in area [120] or even more [121]. There is also 'membrane growth' via a process called exocytosis, which will be covered in chapter 4.

As an alternative to our inextensible membrane we can allow for limited extensibility. We can replace the constraints with springs, and therefore have a 'bead-spring' model, in this case equation 2.33 is replaced with equation 2.11, with the spring forces or tension force being included in $\boldsymbol{F}_t$ as a list of forces on each point $\boldsymbol{F}_i^{\mathbf{spr}}$ with

$$\boldsymbol{F}_i^{\mathbf{spr}} = k_{\text{spr}} \left[(l_{i-1} - d_{i-1})\boldsymbol{u}_{i-1} - (l_i - d_i)\boldsymbol{u}_i \right], \tag{2.74}$$

where l_i is the rest length of the i subunit and d_i is the actual length of the subunit. This shows there is two contributions to each point, one from each spring either side of it. Since the membrane is a loop the force on the first point uses the last subunit as the $i-1$ subunit. k_{spr} is the stiffness of the springs and is equal to $\sigma_{\text{mem}}N$ where σ_{mem} is the membrane tension, this is because our springs are in series. To be clear, this is an alternative option, we therefore have two different models for a membrane - one inextensible, and one extensible.

The membrane exists on the steric grid, just like the actin filaments. Checks are made between membrane and actin filaments to ensure steric hindrance is obeyed. However steric checks are approved/rejected on a per point level. This approve/reject process is described here and Figure 2.22 gives an example showing a section of a membrane. After determining a new configuration from the Brownian dynamics algorithm, each point in turn is updated and the two neighboring subunits are checked for steric hindrance violations. If either (or both) of the subunits violate steric hindrance the new point is rejected and reset back to its previous location. In Figure 2.22 in the top part (before the first arrow) the old configuration is shown as black points connected with blue subunits, the new configuration (that has not been tested sterically yet) is shown as green points connected by red subunits. There are filaments causing steric hindrance violations that are not shown. Each point is tested in turn. The resultant configuration may be a combination of the old configuration and new configuration, as in 2.22. Carrying out steric hindrance checks in this way is necessary to get some shape changes in the membrane and will be important later on for investigating phagocytosis (chapter 4). Steric checks with other subunits within the membrane are unnecessary in our case, and so we neglect them to save computational cost.

2.5 Exclusion zone

The last type of physical object present in the model are exclusion zones. These are either circles or rectangles in space that neither actin filaments or membranes can enter. These are useful in investigating actin networks in the presence of obstacles like in previous work [65]. Exclusion zones will also be used in chapter 4 when investigating phagocytosis, as they can be used as a simple target.

A circular exclusion zone is defined simply as a coordinate point which is the centre point of the circle, and a radius. A rectangular exclusion zone is defined as a two coordinate points, which are the bottom left point and the top right of the rectangle, they therefore are axis aligned rectangles.

After a membrane moves or an actin filament undergoes a process that could cause a steric hindrance violation (e.g. motion or polymerisation), distance checks are carried out against any exclusion zones in the simulation. The distance queries are carried out using Geometric tools [85]. With a circular exclusion zone this is

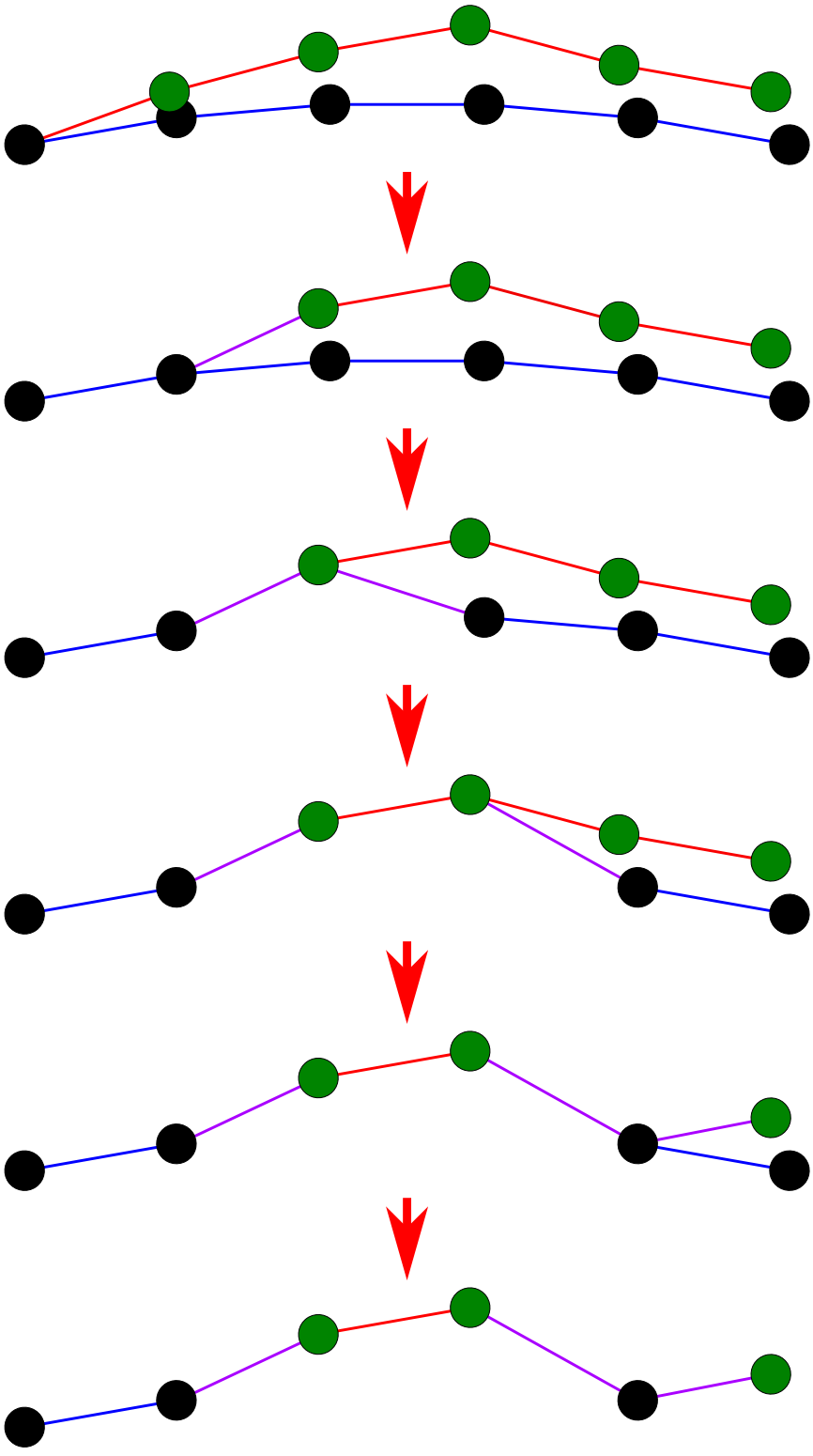


FIGURE 2.22: Diagram showing steric point-wise approve/reject process, showing a section of the entire membrane loop. Starting at the top the old configuration is shown as black points with blue subunits. The new configuration determined from the Brownian dynamics is shown as green points with red subunits. Each point (ignoring the left-most point in this example) is evaluated in turn. Purple subunits are subunits that are created by joining two points where one point is from a rejection and one from an acceptance. This example shows: reject, approve, approve, reject, approve. The resultant configuration is shown at the bottom.

a segment-to-point distance query. With a rectangular exclusion zone this is four segment-segment queries (one for each side), if these pass it either means the segment (actin or membrane subunit) is either completely outside the rectangle or completely inside, therefore an additional simple check is required which checks if a point on the segment is inside the rectangle.

Figure 2.23 shows snapshots of a simulation demonstrating the effects of exclusion zones. The nucleation region is shown as a ring surrounding a circular exclusion zone in green. Two rectangular exclusion zones are above and below the circular exclusion zone. Polymerisation, depolymerisation and branching happen everywhere. Filaments undergo Brownian motion and grow, avoiding the exclusion zones.

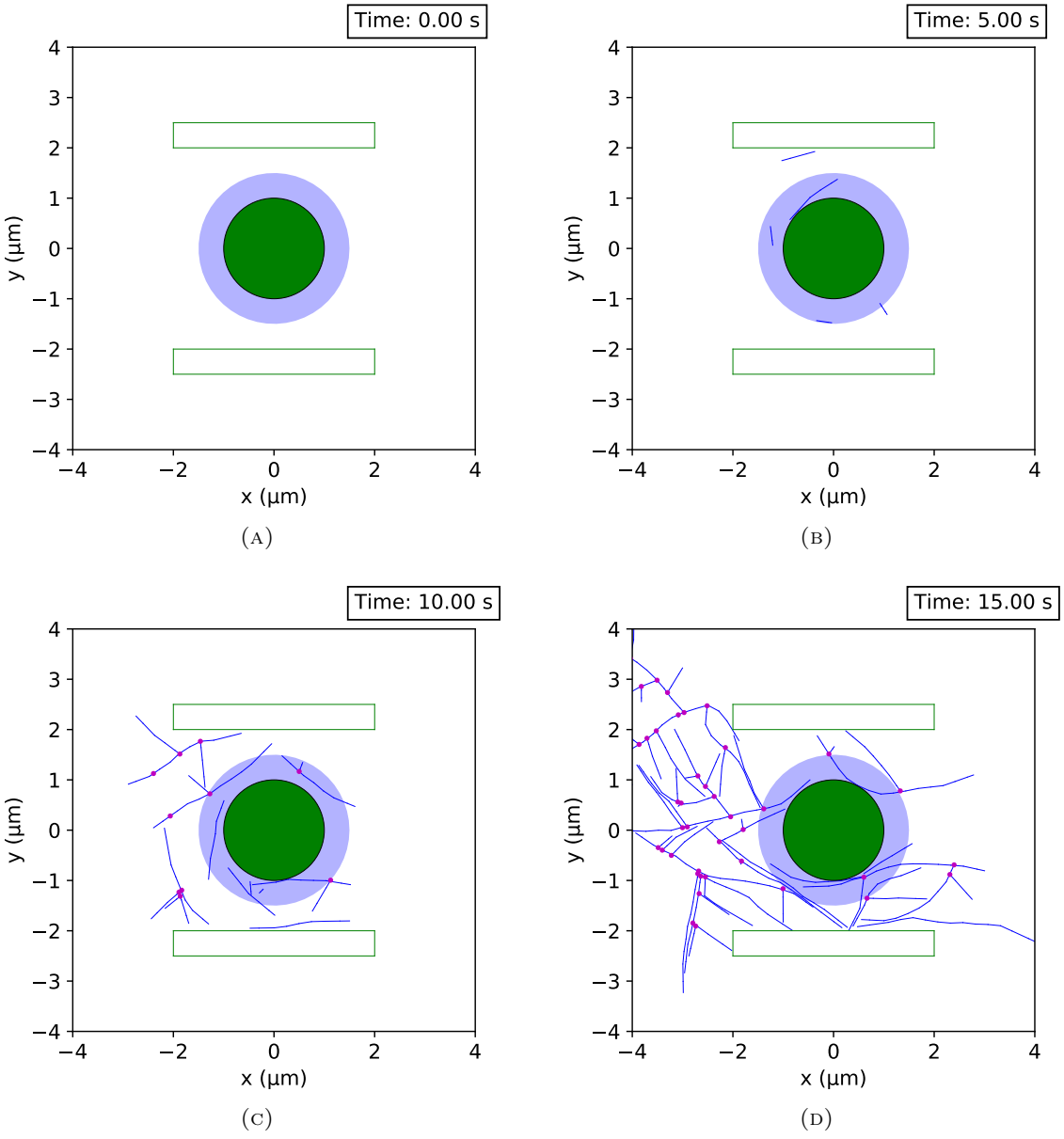


FIGURE 2.23: Snapshots of a simulation showing the effect of exclusion zones. In the centre is a circular exclusion zone shown as a solid green circle, around it is a ringed nucleation region. Above and below the circle are rectangular exclusion zones, shown as empty green rectangles.

Polymerisation, depolymerisation and branching are allowed to happen everywhere.

2.6 Timestep restrictions and stability

An essential consideration of any simulation technique is the numerical stability of the method. Of course we want to be able to use the maximum timestep possible to save wall time when running simulations. In this model we have two methods to consider: the Brownian dynamics for motion of the actin and membrane, and the Monte Carlo method for the different biochemical events actin can undergo. We will cover both methods in turn.

2.6.1 Stability of Brownian dynamics method

Explicit Brownian dynamics methods can become unstable if the forces on the points are evaluated over too large a timestep that they result in large movements which create a positive feedback loop, leading to instability. For more information on numerical stability the reader is directed to [58, 59].

For the general Brownian dynamics method we use the same two timestep restrictions given in [94] which are

$$\Delta t_{\mathrm{Bd}} = \min\left\{\frac{\epsilon\pi\eta a l_s^3}{l_{\mathrm{P}} k_{\mathrm{B}} T}, \frac{(\epsilon l_s)^2}{2D_{\mathrm{S}}}\right\}, \tag{2.75}$$

where ϵ is a fractional error and must be positive and less than one, it is taken to be 0.06 in [94]. We experimented with values of ϵ and found we could increase it to 0.2 and still get very stable simulations. The first term comes from considering the relaxation time of angular rotation in a discrete WLC and as in [94] it is taken from [122]. The second term comes from the requirement of the diffusion distance of a point being less than the subunit length. In [94] D_{S} is the diffusion constant of a single 'bead'. We instead are describing our points in the filament/membrane as rods so we take D_{S} to be the diffusion constant of a single rod. We use the parallel mobility as this is the larger than the perpendicular mobility. Therefore we have

$$D_{\mathrm{S}} = k_{\mathrm{B}} T \left(\frac{\ln\left(l_s/2r\right) + \gamma_{\parallel}}{2\pi l_s \eta}\right), \tag{2.76}$$

where the term in the brackets is very similar to the parallel mobility given in equation 2.13, but with a small change of using the subunit length in the logarithmic term. a is the radius of a hard sphere with a diffusion constant of D_{S} (sometimes named the Stokes radius) this is

$$a = \frac{k_{\mathrm{B}} T}{6\pi\eta D_{\mathrm{S}}} = \frac{l_s}{3\left(\ln\left(l_s/2r\right) + \gamma_{\parallel}\right)}. \tag{2.77}$$

We can now rewrite equation 2.75 as

$$\Delta t_{\mathrm{Bd}} = \left(\frac{\epsilon\pi\eta}{\ln\left(l_s/2r\right) + \gamma_{\parallel}}\right) \min\left\{\frac{l s^4}{3 l_{\mathrm{P}} k_{\mathrm{B}} T}, \frac{\epsilon l_s^3}{k_{\mathrm{B}} T}\right\}, \tag{2.78}$$

the overwhelming majority of the time $\frac{l_s}{3l_{\mathrm{P}}}$ will be much less than ϵ, so in these cases it is the angular rotation term that sets the maximum timestep required for the Brownian dynamics method. It is worth noting the strong scaling with subunit length, which is one limitation of this method. Ideally subunit lengths would be as short as a single monomer (l_{mon}), but this would require very short timesteps.

The other thing to consider for the Brownian dynamics algorithm is the stiffness of any springs in the system. These could be for branches, crosslinks or tether points

(covered later in chapter 3). In general the timestep must be less than the drag coefficient (the inverse of the mobility) divided by the spring's stiffness [58, 59]. Therefore we have a third restriction

$$\Delta t_{\mathrm{Bd}}^{(3)} = \frac{\epsilon k_{\mathrm{B}} T}{k_{\mathrm{spr}} D_{\mathrm{S}}} = \frac{2\epsilon\pi\eta l_s}{k_{\mathrm{spr}}\left(\ln\left(l_s/2r\right) + \gamma_{\parallel}\right)}, \tag{2.79}$$

where k_{spr} is the stiffness of the spring. So we can add this to equation 2.78 giving

$$\Delta t_{\mathrm{Bd}} = \left(\frac{\epsilon\pi\eta}{\ln\left(l_s/2r\right) + \gamma_{\parallel}}\right) \min\left\{\frac{ls^4}{3l_{\mathrm{P}}k_{\mathrm{B}}T}, \frac{\epsilon l_s^3}{k_{\mathrm{B}}T}, \frac{2l_s}{k_{\mathrm{spr}}}\right\}, \tag{2.80}$$

2.6.2 Timestep restrictions of Monte Carlo method

Since the Monte Carlo (MC) method we are using simply converts given rates into probabilities by multiplying with the timestep, this creates another restriction. Obviously if the probability of an event occurring in a given timestep is calculated to be greater than one, then our timestep is too large. The MC events are limited to happening once per timestep (e.g. a filament can't undergo barbed end polymerisation more than once in a single timestep). We therefore want the probability of an event occurring to be much less than one and we therefore determine our timestep from what is likely to be the four most common events: polymerisation and depolymerisation at both ends. We rearrange the equations for the probabilities of these events (equations 2.47, 2.48, 2.49 and 2.50) setting the probability value to be 0.1 and solving for Δt. We therefore have

$$\Delta t_{\mathrm{MC}} = 0.1 \min\left\{\frac{1}{k_p^{\mathrm{barb}} C_{\mathrm{G}}}, \frac{1}{k_p^{\mathrm{point}} C_{\mathrm{G}}}, \frac{1}{k_d^{\mathrm{point}}}, \frac{1}{k_d^{\mathrm{point}}}\right\}. \tag{2.81}$$

Most of the time, barbed end polymerisation is the most likely event and so the first term often sets the maximum timestep.

Of course, other events can happen in a single timestep (e.g. nucleation, branching) and even though they are neglected here in this timestep determination, assertions are placed in the code to ensure that the probability of a single event is always less than 0.1. This is done because some of these events have dynamic probabilities, for example branching is dependent on the number of available monomers that can form a branch on the filament.

2.6.3 Determining the timestep

The timestep used has a hard upper limit of 0.01 seconds, no matter what the conditions in sections 2.6.2 and 2.6.1, it can never exceed this. We chose this value as simulations with a timestep this size run very quickly, there is nothing to be gained by making the timestep larger. Therefore we have our final determination of the timestep as

$$\Delta t = \min\left\{\Delta t_{\mathrm{Bd}}, \Delta t_{\mathrm{MC}}, 0.01\mathrm{s}\right\}. \tag{2.82}$$

2.7 Flow of the simulation method

In this section a description of the flow of the simulation method is given.

2.7.1 Initialisation

A simulation is run by supplying a configuration file with various options and parameter values given as inputs. Inputs of interest are given in section 2.9 in Tables 2.1 and 2.2.

Inputs are first parsed, then any regions are created followed by the creation of any initial actin filaments and, if chosen, the membrane. The timestep is determined at runtime as described in section 2.6. Nucleation regions have their corresponding probability of nucleation (in a single timestep) checked to see if it is less than 0.1. If any are above 0.1, the region is split into two separate regions and those two regions are then checked again and so on. This process is repeated until all nucleation regions have a probability of nucleation of less than 0.1.

By default, initial actin filaments are trimers, but they can be initialised with a longer length and if their length is greater than $3l_s$ they can be set in a bent configuration. This is done by randomly choosing angles between subunits from a normal distribution as

$$\Delta\theta_i \sim N\left(0, \frac{d_i}{l_\mathrm{P}}\right), \tag{2.83}$$

where $\Delta\theta_i$ is $\theta_i - \theta_{i-1}$. This comes from the equipartition theorem [5, 57].

Finally the details of the initial frame is outputted and the simulation begins to progress in time.

2.7.2 Progressing in time

Figure 2.24 is a flow chart provided to help describe the flow of the simulation method. The first step is initialisation described above in section 2.7.1. After initialisation the current time is incremented by the timestep length. The membrane (if there is one present) then fluctuates via the Brownian dynamics method described in section 2.4.2. Steps 4-17 in Figure 2.24 are the optional Monte Carlo biochemical steps that actin filaments can undergo and were described in section 2.3.4. All of these steps, with the exception of nucleation, loop over all actin filaments in the order that they are stored in the computer's memory. This is the order in which they were created (via nucleation, branching or severing) in the simulation. The nucleation step (step 8 in Figure 2.24) loops over all nucleation regions. If other regions are present (branching, capping, severing) then the relevant biochemical process proceeds by looping over the regions then the contents (filaments) of those regions. Step 18 is the motion of the actin filaments and structures, described in section 2.3.2. Just as steps 4-17, step 18 loops over all actin filaments; they are moved and then the new configuration is approved or rejected depending on the result of steric hindrance checks. This is done one by one, filament by filament. Step 19 is allowing for the simple membrane wall described in section 2.4.1 to move downwards if there is space to do so. This is done to account for the external force acting on the wall, pushing it downwards.

Step 20 is writing data to output files. Various outputs, including positions for filaments, regions and membranes are outputted to file every 'frame'. The number of frames is given as an input to the simulation. It is easiest to describe frame print outs using an example. If a simulation is run for say ten seconds, and the user wants a print out of ten frames a second, they must specify they want 101 frames (the extra one comes from the initial frame). The number of timesteps between frames ($N_{\Delta t-\mathrm{frames}}$) is calculated as

$$N_{\Delta t-\mathrm{frames}} = \max\left\{\left\lfloor \frac{t_\mathrm{run}}{\Delta t(N_\mathrm{frames}-1)}\right\rfloor, 1\right\}. \tag{2.84}$$

$N_{\Delta t-\text{frames}}$ is then used to determine if a timestep is a 'frame'. t_{run} is the runtime, N_{frames} is the number of frames specified by the configuration file. Step 20 only runs when the current timestep is a 'frame', as shown in Figure 2.24.

Finally, if the current time is less than the runtime, the system loops back to step 2. Once the current time exceeds (or equals) the runtime, the simulation ends. A Python script can be set to run after the simulation has finished, to automatically print a visualisation of the simulation as images. Video software (FFmpeg [123]) can also be used to then stack the images into a video.

2.7.3 Comparison to standard methods

In section 2.1.2 three relevant standard methods were introduced. In this section we will compare our method with these standard methods.

Our method is somewhat of an amalgamation of the three standard methods. We use a Brownian dynamics method to move filaments and our membrane. This is coupled with an approve-reject system based on steric hindrance which is a similar to methods using the Metropolis-Hastings algorithm. We mentioned earlier in section 2.2 that we initially experimented with a Gillespie method for the biochemical processes. We found this to be much slower computationally as we had short timesteps in which the Brownian dynamics algorithm must also use. We therefore chose an alternative, simpler method with a larger fixed timestep in which many processes can happen. Such a method is described in [82].

With this method of using a fixed timestep rather than Gillespie, some issues can arise. Using this method assumes that events happening in the same time window are independent. In a given time window many events may happen, in reality these events would happen in a particular order and this may impact what happens next. In our simulation method these events always happen in the same order described in section 2.7.2. This could be addressed in the future by, in every timestep, 'shuffling' the order in which biochemical events happen as well as 'shuffling' the order in which actin filaments are chosen in each process. We would still however have potential problems of time dependence within a timestep, but typically the timestep length is such that only a few events happen in a single step.

The approve-reject steric hindrance method could also present problems. Under certain conditions filaments could tangle and 'lock-up', unable to free themselves. Similar methods coupling approve-reject with Brownian dynamics for filaments have been used before [68, 124]. However in both these cases an additional component of interaction between filaments exist. In [124] there is an additional Metropolis-Hastings approve-reject procedure based on an interaction potential. In [68] there is an additional force between filaments that are closer than 1nm. The magnitude of this force increases linearly with decreasing distance. This force was calibrated so that more than 99.9% of the total steps were approved, as the force prevented filament crossings from happening most of the time. It would be worth in the future investigating the effects of including an additional interaction like the ones used in these examples. Both of these examples are also using 3D simulations whereas our method is only 2D. In 2D steric hindrance will play a larger role since the whole system is confined to a plane. The potential move to three dimensions will be discussed later on in the thesis.

2.8 Validation - Correction C

Correction C to go here.

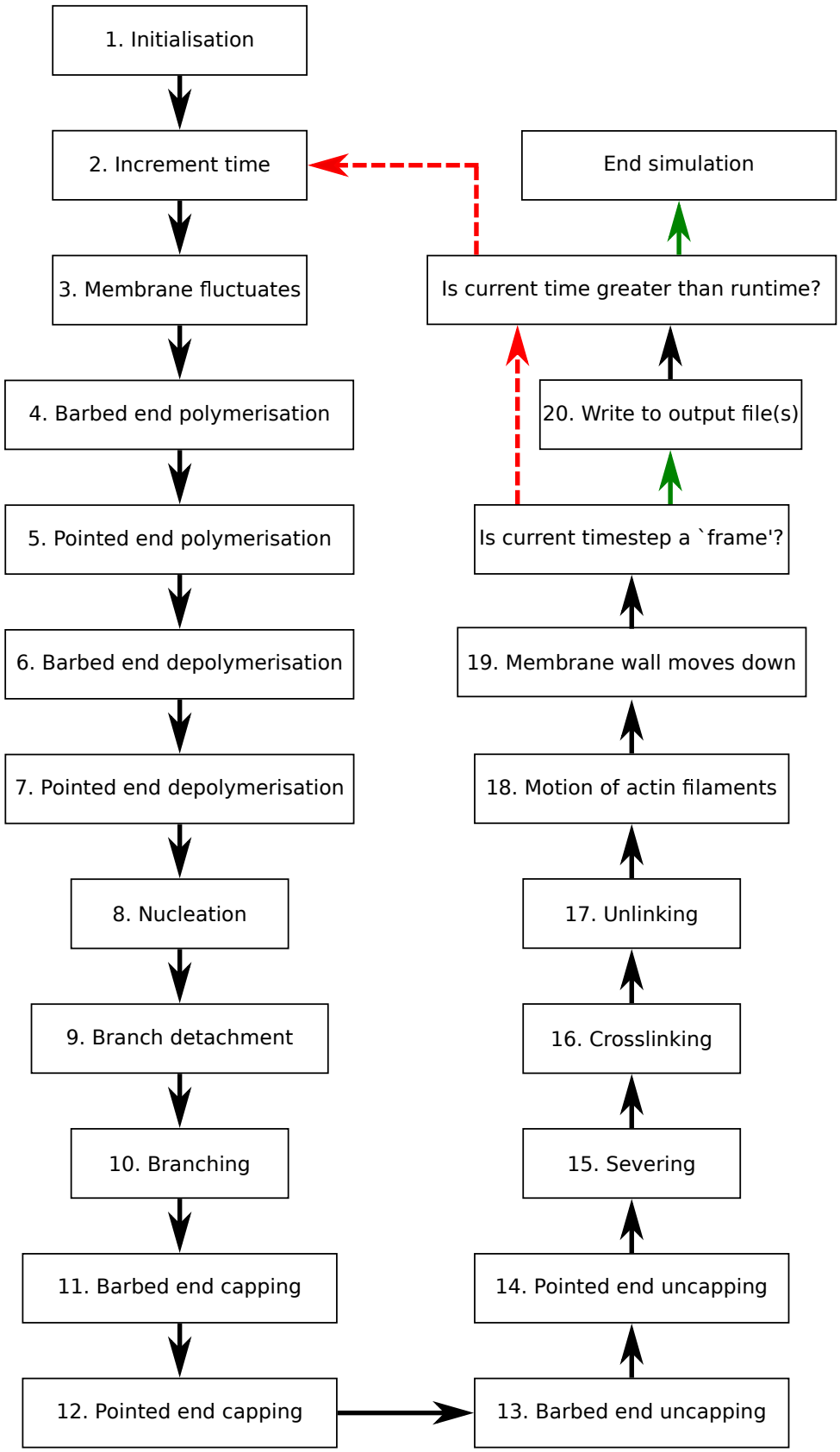


FIGURE 2.24: Flow chart describing the simulation method. Start at the top left. Solid green arrows depict the flow from a 'Yes' answer from the question, whereas dashed red arrows depict the flow from a 'No' answer.

Table 2.1: Constants used in the model.

Symbol	Description	Values	Refs.
r	Radius of actin filament	3.5nm	[1, 21]
r_{steric}	Steric radius of actin filament	7.5nm	[104]
l_P	Persistence length of actin	17µm	[3]
l_{mon}	Actin monomer length	2.7nm	[55]
k_{br}	Branch spring constant	$5\times10^{-5}\text{N m}^{-1}$	here
θ_{br}	Branch angle	1.2 rad	[10]
$k_{\theta_{\text{br}}}$	Branch torsion spring stiffness	$5\times10^{-18}\text{N m rad}^{-1}$	here
κ_{cl}	Crosslink spring constant	5×10^{-5} N m^{-1}	here
s_{mem}	Membrane thickness	5 nm	[1]
l_{mem}	Subunit length of membrane	250 nm	N/a
ϵ	Error coefficient in B.D. algo	0.2	here

2.9 Summary of simulation inputs and constants

Table 2.1 contains the simulation constants values used. Table 2.2 gives the simulation variables used.

2.10 Summary

We have presented the simulation tool developed during the research project. We are able to simulate many actin filaments undergoing Brownian motion and many different biochemical processes. We also have a simple model for a membrane in 2D. In later chapters we will present work that uses this model and in some cases builds upon it. However the tool has been designed to be fairly general and so could be used to model many more interesting systems and scenarios.

TABLE 2.2: Parameters used in the model.

Symbol	Description	Typical value(s)	Refs.
N_0	Initial number of actin filaments	0	N/a
	Maximum number of filaments	10^5	N/a
l_s	Subunit length of filament	500 nm	N/a
L_{init}	Initial length of filaments	2-3l_{mon}	[1, 11, 110]
k_{nuc}	Nucleation rate density	Varies	N/a
k_p^{barb}	Barbed end polymerisation rate	12 $\mu\text{M}^{-1}\,\text{s}^{-1}$	[6]
k_p^{point}	Pointed end polymerisation rate	1.3 $\mu\text{M}^{-1}\,\text{s}^{-1}$	[6]
k_d^{barbed}	Barbed end depolymerisation rate	1.4 s^{-1}	[6]
k_d^{point}	Pointed end depolymerisation rate	0.8 s^{-1}	[6]
C_G	Free G-actin concentration	Varies	N/a
k_{br}	Branching rate per site	5.3×10^{-4} $\mu\text{M}^{-3}\,\text{s}^{-1}$	[110]
C_{Arp}	Arp2/3 conc	1-10µM	[125]
d_{br}	Min distance between branches	13l_{mon}	[10, 23]
k_{dbr}	Debranching rate	1.8×10^{-3} s^{-1}	[110, 126]
$k_{\text{cap}}^{\text{barb}}$	Barbed end capping rate	3.5 $\mu\text{M}^{-1}\,\text{s}^{-1}$	[114]
$k_{\text{cap}}^{\text{point}}$	Pointed end capping rate	0.8 $\mu\text{M}^{-1}\,\text{s}^{-1}$	[110]
C_{cap}	Concentration of capping protein	0.2 - 2 µM	[42, 114, 125, 127]
$k_{\text{uncap}}^{\text{barb}}$	Barbed end uncapping rate	4.0×10^{-4} s^{-1}	[114]
$k_{\text{uncap}}^{\text{point}}$	Pointed end uncapping rate	1.8×10^{-3} s^{-1}	[110]
k_{cl}	Crosslinking rate per possible link	Varies	N/a
d_{cl}	Min distance between crosslinks	Varies	N/a
s_{cl}	Crosslink distance	Varies	N/a
k_{unlink}	Unlinking rate	Varies	N/a
k_{sever}	Severing rate	5×10^{-3} - 0.01 $\mu\text{m}^{-1}\,\text{s}^{-1}$	[115, 116]
η	Viscosity	0.05 Pa s	[128, 129]
T	Temperature	300 K	approx. lab/body
L_{mem}	Length of membrane	Varies	N/a
κ_{mem}	Bending modulus of membrane	$6k_BT$ - $50k_BT$	[69, 130, 131]
Δt	Timestep (upper limit)	0.01 s	section 2.6

Chapter 3

Investigating the effects of actin binding proteins on actin nucleation and polymerisation

This chapter presents work done in investigating *in vitro* actin polymerisation systems. This includes studying the effects of the protein Las17, on actin. This research project involved a large amount of cross-disciplinary collaboration with members of the laboratory of Prof. Kathryn Ayscough; in particular Dr. Ellen Allwood and Dr. John Palmer. We worked closely with Ellen and John to help analyse their experimental data in novel ways. Through this, we managed to improve on outdated analysis techniques and assumptions.

3.1 Introduction

Las17 is the yeast homologue of Wiskott Aldritch Syndrome Protein (WASP) [132], a well known nucleation promoting factor (NPF) of actin. WASP and its homologue Las17, are two examples of many proteins that activate Arp2/3 complex, to produce branched actin nucleation [133]. However, Kathryn Ayscough's group has previously showed that unlike WASP, Las17 can nucleate linear filaments on its own [134]. The presence of Las17 is also thought to increase the elongation rate of actin filaments [134, 135]. Las17 is important in yeast endocytosis, where it activates Arp2/3 complex which is required for branched actin nucleation. However it also nucleates the initial 'mother' filaments that are required to branch off [132].

Our collaborators are interested in Las17, and have identified different regions of the Las17 protein responsible for these different effects [135]. Working with them, we aimed to be able to identify these effects by analysing experiment data from two independent experiments, as well as using simulations and analytical calculations. We wanted to improve analysis techniques used in their lab by considering more mathematically robust methods. This would help us to better understand the relative contributions of polymerisation, depolymerisation and nucleation to actin dynamics.

In this work we study three different biochemical systems, these are 'Actin only', 'Actin and Las17' and 'Actin, Las17 and Arp2/3'. As the names suggest these contain actin, actin in the presence of Las17 and actin in the presence of Las17 and Arp2/3 complex, respectively.

3.2 Introduction to experimental systems

We used two independent experimental systems in this project: pyrene assays and Total Internal Reflectance Fluorescence (TIRF) microscopy (TIRFM).

3.2.1 Pyrene assays

Pyrene assays are used to measure, in bulk, the total amount of polymerised actin over time. Actin is labelled with pyrene, a fluorescent marker. Pyrene-actin fluoresces and pyrene F-actin monomers (actin monomers polymerised within a filament) fluoresce more than pyrene G-actin monomers (individual, free monomers). This increase in intensity is reported to be around 10× [136]. This phenomenon therefore allows pyrene assays to be used to measure actin polymerisation over time.

In general, the method involves mixing G-actin with pyrene and a buffer solution in a cuvette (a small tube like container). To produce samples of varying concentrations of actin, more buffer can be added to dilute the mixture down. The experiment then begins when a salt buffer (like KMEI: which contains potassium and magnesium salts) is added, inducing polymerisation. The cuvettes are placed inside a fluorimeter which is used to measure the intensity of the fluorescence produce by the samples during the experiment.

Data taken is fluorescence intensity over time and an example pyrene actin plot is given in Figure 3.1.

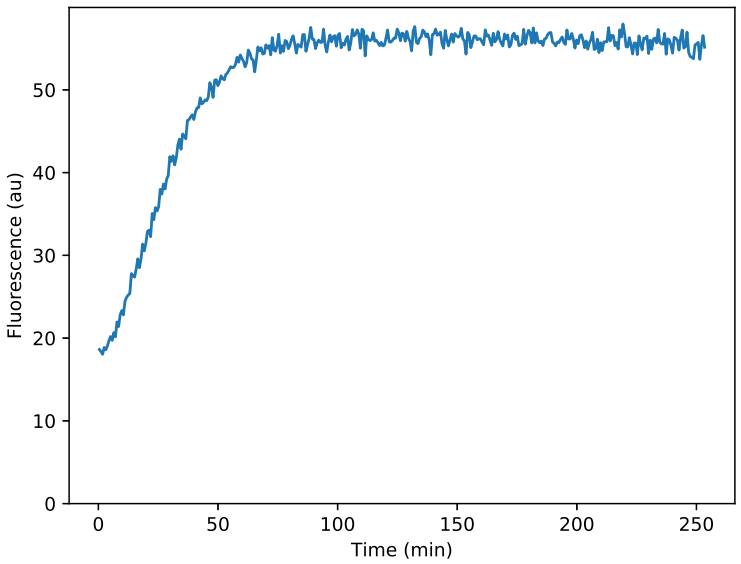


FIGURE 3.1: An example pyrene curve. Fluorescence intensity in arbitrary units is along the y axis, time in minutes along the x. Raw data from 3.10µM actin. The curve shows an increase in fluorescence of actin over time, until it reaches a plateau. Data taken by Ellen Allwood.

Figure 3.1 shows the characteristic three stages in the polymerisation of actin [1]. The initial 'lag phase', the 'growth phase' and the 'plateau' phase when the system has reached steady state.

There are some drawbacks and complications to the pyrene assay which I will address here.

- We measure 'amount of F-actin' in this arbitrary fluorescence unit, which as mentioned also contains a contribution from G-actin.

- It can be seen in Figure 3.1 that the data does not start at zero fluorescence, so what is the source of this background fluorescence? and are we able to remove it?

- There is an inevitable delay between starting polymerisation by adding the salt, and the first measurement on the fluorimeter. This leads to the often interesting very initial part of the polymerisation process being missed.

- Pyrene-actin polymerisation is complicated by different ATP hydrolysis states of the actin (ATP, ADP+P_i intermediate, ADP actin). These have been previously shown to have different levels of fluorescence [111].

- This is a bulk measurement of actin, and so it is a measure of a combination of nucleation, (including branching if Arp2/3 complex is present), polymerisation and depolymerisation. How do we tease these individual processes out? We also have no information about the number of filaments, or any structural/geometrical information.

- There are many experimental variables that are difficult to quantify. Different fluorescence levels of different batches of pyrene-actin, viability of the actin and associated proteins of interest are two issues that can produce variability between experiments.

Some of these issues we will address in our analysis (section 3.6.2) but some are inherent to the experimental system. Therefore we wanted to complement the pyrene assay system with another; TIRFM.

3.2.2 TIRF microscopy

TIRF stands for Total Internal Reflectance Fluorescence and is used in TIRF microscopy (TIRFM). A TIRF microscope is designed to use total internal reflectance to excite only a thin region of the sample close to the surface [137]. This is beneficial because this results in only a small number of fluorophores emitting light, reducing background noise from those which would be out of focus anyway. It is a popular technique for visualising actin filaments which are tethered to the coverslip [60, 138]. The thickness of the observed sample is only around 200nm [60, 139] and so we can treat the image as almost a two-dimensional plane.

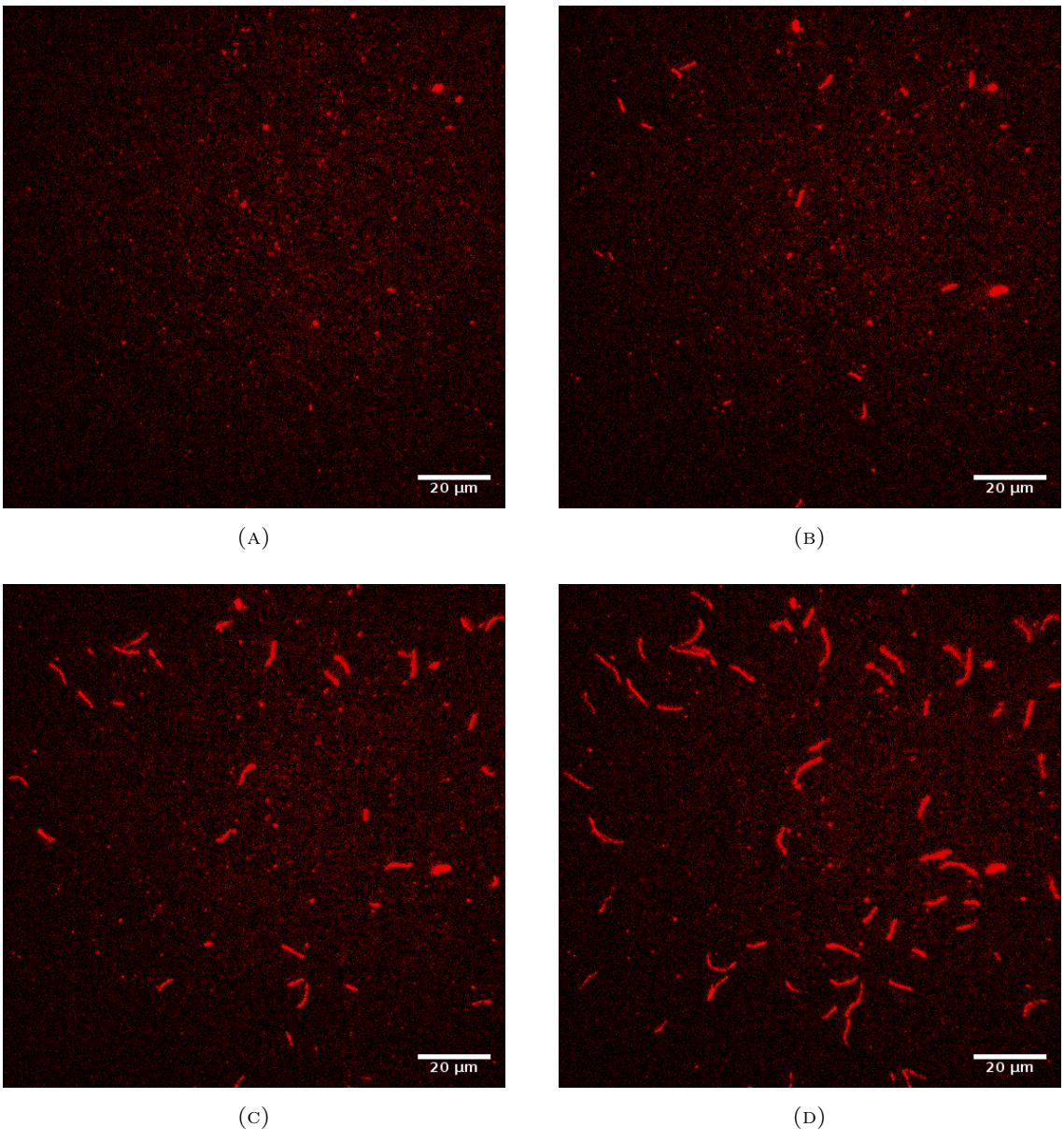
Unlike pyrene assays we therefore can get information on individual actin filaments. TIRFM can be used to study the lengths of filaments, or the geometry and structures that actin can make under certain conditions.

Similar to pyrene assays, an actin and buffer solution is made up prior to the experiment. The solution contains methylcellulose which increases the viscosity above that of water to reduce Brownian motion of the filaments. The KMEI buffer is added which induces actin polymerisation, starting the experiment. There is an inevitable delay between this and the first frame of data collection as the camera needs to be setup and adjusted so the frame is in focus; this can take several minutes.

The TIRFM experiments were carried out by John Palmer. Snapshots of an example of what is seen under TIRFM are shown in Figure 3.2, which contains 'Actin only' at 0.5µM.

Data acquired with TIRFM is not without its challenges and drawbacks.

- The experiment itself is time consuming and acquiring microscope time can be competitive.



(A) (B)

(C) (D)

FIGURE 3.2: Snapshots of an example TIRFM experiment. Background has been subtracted and brightness/contrast corrected using Fiji ImageJ [140]. Frame size is 136.5µm×136.5µm. Fluorescent actin is labelled in red. Actin concentration is 0.5µM. **(A)** First frame of data collection: 0 seconds. **(B)** 400 seconds (6min 40s). **(C)** 800 seconds (13min 20s). **(D)** 1200 seconds (20min). Taken by John Palmer.

- Flurophores can 'fall off' actin and speckle the frame. This makes analysis difficult as it is sometimes impossible to determine what is a short filament and what is loose flurophore.
- Untethered polymerised filaments can fall into the field of view from above.
- Filaments often cross which increases difficulty in analysing elongation rates with automatic techniques.
- In general it can be time consuming to measure quantitative data such as elongation or nucleation of filaments.
- Although the image quality is generally high, TIRFM is still a diffraction limited technique. Tens of actin filaments could be aligned together and still appear as a single filament.
- As mentioned above there is a time delay in taking the first measurement which is unavoidable just as in the pyrene system.

Just as with the pyrene system, some of these issues we will address in our analysis (section 3.5.2). Other issues are dealt with by using the pyrene and TIRFM systems together. TIRFM's main advantage over pyrene assays is the availability of visual information on a individual filament level. For example we can get the number of filaments and the lengths of filaments directly through the images. However, image analysis proves difficult and time consuming.

The pyrene system's main advantages over TIRFM include that the more direct quantitative measurement of fluorescence intensity is much easier to analyse. Several concentrations of actin and actin with other proteins can be run all at the same time on the fluorimeter, making data acquisition much quicker than TIRFM.

3.3 Differential equations

To help understand actin dynamics in our three systems we developed a mathematical model where the actin system is described by a set of coupled differential equations.

3.3.1 'Actin only' and 'Actin and Las17'

We treat the 'Actin only' and the 'Actin and Las17' systems as the same in terms of the equations used to describe them. In both systems there is no nucleation of branched actin, all F-actin filaments should be linear filaments. We assume that all nucleation in the 'Actin and Las17' system is due to Las17 nucleation alone[1]. When comparing the two systems we expect the parameters describing nucleation, polymerisation and depolymerisation to change.

The coupled differential equations are presented below. They describe the change in the number of G-actin monomers, F-actin monomers and F-actin filaments.

$$\dot{c}_1 = k_d c_f - k_p c_1 c_f - \beta k_n c_1^{\beta}, \tag{3.1}$$

$$\dot{c}_p = -k_d c_f + k_p c_1 c_f + \beta k_n c_1^{\beta} = -\dot{c}_1, \tag{3.2}$$

[1]Depending on the actin concentration used, there is 6.5 - 31× as many actin monomers than Las17 molecules (see experimental details for concentrations used in section 3.6.1). However, there obviously will be less number of filaments than actin monomers at any given time. Las17 is therefore assumed to be in excess and nucleation assumed to be dominated by the Las17 mediated pathway, but perhaps more investigation is required (see later in section 3.8).

$$\dot{c}_f = k_n c_1^{\beta}. \tag{3.3}$$

The dot notation is used to indicate derivatives with respect to time, i.e. $\dot{c} = \frac{dc}{dt}$. c_1 is the concentration of G-actin monomers in the system, c_p is the concentration of F-actin monomers in the system, c_f is the concentration of filaments in the system. k_d is the depolymerisation rate per F-actin filament concentration and is the sum of depolymerisation at both barbed and pointed ends. k_p is the polymerisation rate per G-actin concentration per F-actin filament concentration, again for both ends. k_n is the nucleation rate per G-actin concentration to the power of β, which is the size of the smallest actin filament in actin monomers. The smallest actin filament can also be called a nucleation 'seed'. In our simulation model described in chapter 2 and from the literature, this is always assumed to be equal to three. However some previous work, while accepting that the size of the 'seed' is three monomers, suggested through curve fitting that the power to which the nucleation scales with G-actin concentration is two [110]. We assume that the size of the 'seed' in number of monomers should be equal to the scaling of nucleation, as shown in the equations. This is because β G-actin molecules must come together quickly to form a 'seed'. The number of collisions should scale with the number of molecules required to come together to form a 'seed'. Due to this disagreement in the literature between scaling and size, we therefore took the opportunity to investigate the value of β.

We have used the units of concentration to describe the amount of actin in the description above. However we could also use these equations with number of molecules instead by using the appropriate conversion. For example c_1 would then represent the number of G-actin molecules, c_f the number of F-actin filaments, k_p would be the polymerisation rate per number of G-actin per filament. Polymerisation rates are typically given in units of per concentration of G-actin per filament or per concentration of G-actin per filament end. Conversion will therefore be necessary to compare with literature values. This will be covered later in section 3.6.2.6.

In section 3.2.1 we covered how pyrene G-actin does produce some fluorescence, and that it has been reported to be around a tenth of pyrene F-actin. Therefore, to compare our differential equations with what is measured in the pyrene assays experiment we output the following

$$c_m = c_p + ac_1, \tag{3.4}$$

where c_m is the concentration of what is measured in a pyrene assays experiment: the concentration of F-actin monomers plus the concentration of G-actin monomers multiplied by the fraction of G-actin to F-actin fluorescence, a.

Initial conditions are

$$c_1(0) = c_0, \tag{3.5}$$

$$c_p(0) = c_f(0) = 0, \tag{3.6}$$

and therefore

$$c_m(0) = ac_0. \tag{3.7}$$

Here c_0 is the initial concentration of G-actin. So here we are assuming that all the actin in the system initially is in monomeric form, there are no filaments initially.

We assume no dissociation of filaments. To include dissociation of filaments using a model of this form would require effectively infinite equations to monitor the lengths of filaments. However we also argue assuming no dissociation is valid for our systems. These equations will be used to model actin polymerisation in the pyrene system, an example was shown in Figure 3.1. There will be some dissociation of filaments in the

experimental system but if polymerisation is much greater than depolymerisation, this will be small. If there is significant dissociation in the experimental system, the model would simply determine a nucleation rate that is smaller than that in the experiment. Since some of the nucleated filaments would immediately dissociate. We can validate this assumption with the simulation model (which includes dissociation), which will be covered later in section 3.6.3.

We expect the values of k_p, k_d, k_n and possibly β to differ between the two systems.

3.3.2 'Actin, Las17 and Arp2/3'

To model this system we need to also account for branched actin nucleation. We take the equations defined above and modify them to include branching. The modified equations are presented below.

$$\dot{c}_1 = k_d c_f - k_p c_1 c_f - \beta k_n c_1^{\beta} - \gamma k_b c_p c_1^{\gamma}, \tag{3.8}$$

$$\dot{c}_p = -k_d c_f + k_p c_1 c_f + \beta k_n c_1^{\beta} + \gamma k_b c_p c_1^{\gamma} = -\dot{c}_1, \tag{3.9}$$

$$\dot{c}_f = k_n c_1^{\beta} + k_b c_p c_1^{\gamma}. \tag{3.10}$$

k_b is the branching rate per F-actin monomer concentration per G-actin monomer concentration to the power of γ. Using the same logic as with β, γ is therefore the number of G-actin monomers in a branch 'seed'. The general consensus is that Arp2/3 complex binds to a single actin monomer and a mother filament to then nucleate a branch. With the Arp2 and Arp3 subunits of the complex representing conformationally the first two actin monomers, and the single actin monomer then providing the third to form a trimer-like 'seed' [141]. However some previous work has proposed that Arp2/3 complex binds to two actin monomers before binding to the mother filament [11]. This assumption has also been previously used in modelling work [110, 111]. In chapter 2 we also assumed γ to be equal to two based on [110], which found a quadratic scaling of branching rate on G-actin concentration to be the best fit to their data. The branching rate should also depend linearly on the concentration of Arp2/3 complex. Across all experiments the initial concentration of Arp2/3 complex is constant, we hide this dependence in k_b.

This branching model is rather simplistic. It does not account for 'occupied sites' along mother filaments, nor does it account for branch spacing. It simply works under the principle that the number of available sites is always the number of F-actin monomers. Branches are also treated the same as linear filaments with regards to their polymerisation and depolymerisation rates. Therefore the fact that branches are pointed end capped is not taken into consideration. De-branching is not explicitly modelled, although due to the treatment of branches as linear filaments, it is as if branches immediately de-branch upon their formation. The initial Arp2/3 complex concentration is consistent across the experiments but this will change over time as branches form. This is not considered in the equations, Arp2/3 complex is treated as constant in time. To address these issues would require more complexity with more equations and parameters, which would defeat the purpose of having these equations in addition to our simulation model. We can test the validity of these simplifying assumptions by comparing with our simulation model.

3.4 Adapting the simulation

In addition to our analytical model we can use the simulation tool described in Chapter 2. However, we needed to adapt it slightly so we can use it to model both of the experimental systems.

3.4.1 Tethering

In the TIRFM system the filaments are tethered to the glass slide. This can be replicated in the model by using a method similar to crosslinking that was previously described in Chapter 2 Section 2.3.2. We model tethering through pairs of 'tether points'. One point is fixed in space and represents the end of a tether stuck to the glass slide. The other point is situated on the filament, at a discrete site corresponding to the middle of a monomer.

As discussed in Chapter 2, in our simulation model we assume that nucleation seeds are three monomers in length, and are therefore trimers. We assume that trimers are immediately tethered to the glass slide on nucleation. Therefore in the model, when a new filament is nucleated, a tether point pair is created on the middle of the trimer (on the second monomer). Both tether points are positioned at the same spatial coordinate but one will be fixed there for all time and the other can move with the filament's monomer it is associated with. Additional tether points are then added later on as the filament grows. New tethers are created at the ends of a filament, according to the method described below.

Firstly, when a new filament is nucleated, two random distances, each corresponding to each end of the filament, are selected according to

$$d_{\text{tetP}}, d_{\text{tetB}} \sim N(\mu_{\text{tet}}, \sigma^2_{\text{tet}}). \tag{3.11}$$

d_{tetP} and d_{tetB} are the distances for the pointed and barbed end respectively. The filament must then grow such that the distance from the relevant end to the initial tether point is a distance greater than the relevant random distance chosen. When this requirement is met, a new tether is created on the relevant end monomer of the filament and another distance is chosen randomly. For example if the tether is placed at the barbed end, a new value for d_{tetB} is chosen again according to equation 3.11. Figure 3.3A helps explain this process. The green monomers are tethered and therefore the filament must grow to the space where the chosen lengths (d_{tetP} and d_{tetB} for pointed and barbed ends respectively) mark, before more tether points are added. If a monomer associated with a tether point is removed via depolymerisation, the tether point is removed. If a filament has all tether points removed it is completely free to move via Brownian motion. The 'distance to the next tether point' (that is used to determine whether to add a new tether point) is set to be the entire length of the filament in this case.

For short rigid filaments with a single tether, the filament just rotates around the tether, there is no translational motion. In this case both tether points are located on top of each other. If the filament is unbranched, the rotation is done as the method described in section 2.3.2.1, using the mobility constant given in equation 2.4. If the filament is branched, the whole branched structure is rotated around the tether point, using the mobility constant from the fitted ellipse (described in section 2.3.2.2). Short rigid filaments that are branches themselves can not tether. This was chosen to simplify motion of branched structures. If short rigid filaments have more than one tether associated with them, they are fixed in space. Figure 3.3B shows a short rigid filament, it will simply rotate around its single tether shown as a red cross.

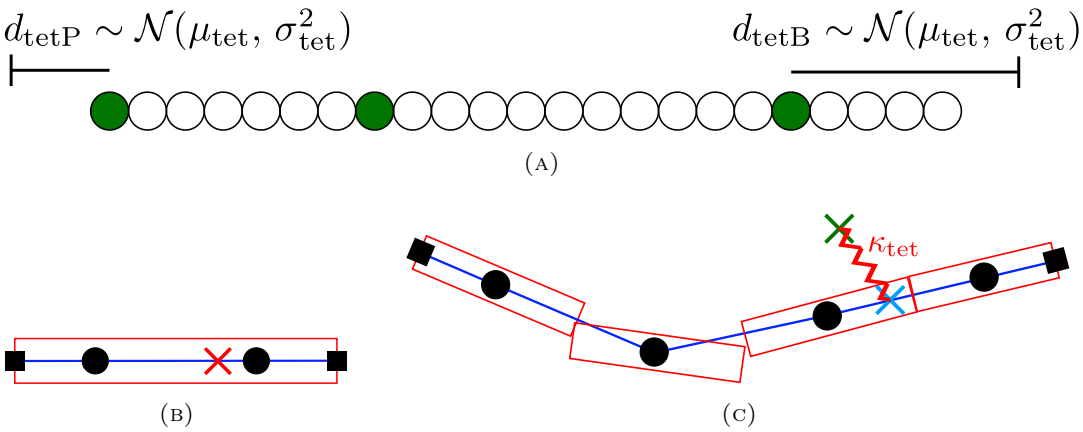


FIGURE 3.3: Tethering. **(A)** Diagram showing monomers along a single actin filament, colour coded to show if a monomer has a tether associated with it (green). Also shown are the distances to the next tether points (d_{tetP} and d_{tetB}). The filament has to grow such that the distance from the relevant end to the next tether point meets or exceeds these distances, at which point a new tether will be added at the relevant end monomer. Figures **(B) and (C)** represent actin filaments and uses the same design as Figure 2.2 for example. **(B)** A short rigid filament with one tether point (marked by the red cross) associated with it, the filament will simply rotate around this point. **(C)** A longer flexible filament with one tether point associated with it. Here the tether acts as a spring, pulling the tethered part of the filament (blue cross) towards the fixed end of the tether (green cross).

For long rigid and flexible filaments, tethers act as springs. We take a similar approach to that taken in previous work, where adhesion complexes in lamellipod protrusion were modelling as springs [68]. The tether spring provides an additional force which is added to the force term $\mathbf{F}_t$ in equation 2.33. The total force acting on the filament from a single tether is

$$\mathbf{F}^{\mathbf{tet}}_{\mathbf{tot}} = \kappa_{\text{tet}} \left(\mathbf{g} - \mathbf{f} \right), \tag{3.12}$$

where $\mathbf{g}$ is fixed tether point (modelling the connection on the glass slide), and $\mathbf{f}$ is the tether point on the filament. This total force is then distributed on the two points that make up the subunit the tether is on (or the end-midpoint if $\mathbf{f}$ is on an end subunit) using the method in equations 2.37 and 2.38.

3.4.2 Limiting G-actin and Arp2/3

For simulating the pyrene system we need to consider a limited resource pool of G-actin and Arp2/3 complex. In the *in vitro* system there is a limited resource of G-actin and Arp2/3 complex whereas in Chapter 2 we assumed an infinite pool. When using simulations to model the pyrene experimental system, we will be simulating over long timescales of several hundreds of minutes. In these timescales the assumption of an infinite pool is not valid and modelling a limited resource is essential.

To achieve an finite pool of G-actin/Arp2/3 complex we define a region in which these molecules exist. This is a three-dimensional rectangular box which can be defined with a particular length, width, height and initial G-actin/Arp2/3 complex concentration. The concentration can be converted into a number of molecules by the following relationship

$$N = C \Delta x \Delta y \Delta z N_A \times 10^{-3}. \tag{3.13}$$

Where N is the number of molecules, C is the concentration of molecules (µM, which is equivalent to $10^{-3}\mathrm{mol\,m^{-3}}$), Δx, Δy and Δz are the width, height and depth of the region in metres respectively, N_A is the Avogadro constant in molecules per mole (mol^{-1}).

The number of free G-actin molecules in the region is decreased by one during a polymerisation event, increased by one during a depolymerisation event, decreased by β during a nucleation event, increased by β during a dissociation event and decreased by γ during a branching event. Polymerisation, nucleation and branching cannot occur outside of the region. If a filament grows to the edge and a depolymerisation event occurs just outside the G-actin region, the G-actin monomer is returned to the region, as if the event had happened within the region. After any of these events the G-actin concentration is recalculated by equation 3.13 (rearranged to find C). The region is not partitioned up and therefore the G-actin is considered well-mixed.

An Arp2/3 region works in the same way. A branching event reduces the number of Arp2/3 molecules in the region by one, and a debranching event increases it by one.

Simulating a region that is comparable in size to the entire pyrene system (370µL) would be time consuming and unnecessary. A common method in simulation to overcome this is to use periodic boundary conditions, to simulate a smaller region that is representative of the larger system. Here we choose to instead design the simulation with a defined geometry of our regions to produce an effect equivalent to using periodic boundary conditions. Figure 3.4 shows the simulation geometry which is described here. We define our nucleation region area, and this is our simulation region. We then define a G-actin region that is larger than this and encompasses the nucleation region, it is made to be as large as it needs to be so that at any point in the simulation no actin filaments can ever grow or treadmill to the edge of this. No actin should ever exist outside of the G-actin region. If we want to model the 'Actin, Las17 and Arp2/3' system we combine the G-actin region with an Arp2/3 region to allow branched actin to form along any part of the F-actin in the system. This removes any artificial edge effects that are not present in the *in vitro* system.

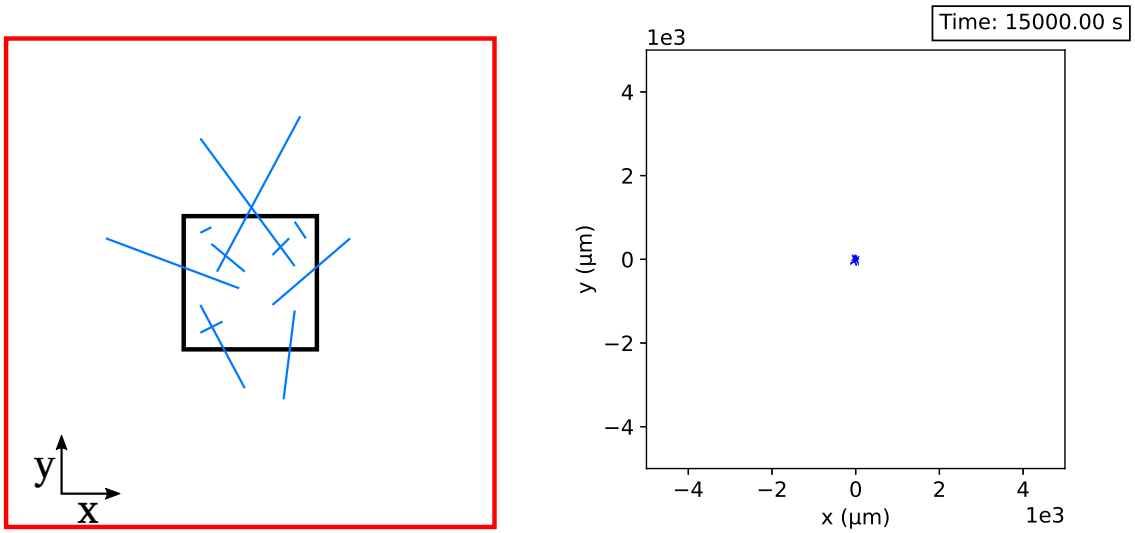
Steric effects and Brownian motion are disabled for simulating the pyrene system as these are unnecessary. The pyrene experimental system is a 3D system where actin filaments are free to undergo Brownian motion. However any steric effects (either between filaments or between filaments and the edge of the glass cuvette) that will affect polymerisation or nucleation will be negligible. We are interested in the bulk fluorescence measurement taken from the pyrene assays and so Brownian motion can also be neglected. A long actin subunit size (l_s) is chosen so all filaments are short rigid. Figure 3.4 shows the simulation geometry.

To get a pyrene style output we simply measure the total length of all filaments in the whole system at every timeframe. We can convert this length to number of F-actin monomers and/or concentration of F-actin monomers to compare with the experimental data. This conversion and comparison is covered later in section 3.6.3.

3.4.3 Latrunculin binding

This section describes a small addition to the simulation tool that has not been currently used but its possible use will be discussed later in section 3.8.

Latrunculin is a toxin which binds to G-actin (in a 1:1 complex) and renders it unavailable for nucleation and polymerisation onto filaments [142]. We model the effect of latrunculin in the simplest way. We start with an initial concentration of



(A) (B)

FIGURE 3.4: Pyrene simulation geometry. **(A)** Diagram of the geometry. The blue lines are actin filaments. The black box is the nucleation region, where new free filaments are formed. The red box is the G-actin region, (can also be combined with an Arp2/3 region) allowing actin to grow beyond the inner region. At the end of the simulation a print out is produced (example shown as **(B)**) which must be checked to ensure no actin has reached the edge of the red box. **(B)** Snapshot of an example pyrene style simulation. The frame encompasses the G-actin region which is 10mm×10mm in area. The much smaller nucleation region which is 100µm×100µm in area is shown in the centre. All the actin filaments in the simulation are in the centre of the frame and at this scale appear as a small blue blob.

latrunculin and use the G-actin region to convert into a number of latrunculin molecules (N_{Lat}), following equation 3.13. Each timestep we simply calculate how many G-actin molecules have been bound by latrunculin, this is

$$\Delta N_G = -N_{\text{Lat}} k_{\text{Lat-on}} C_G \Delta t, \tag{3.14}$$

where $k_{\text{Lat-on}}$ is the rate of latrunculin binding per G-actin concentration ($\mu\text{M}^{-1}\,\text{s}^{-1}$). We then change the number of G-actin molecules by this amount and also record the number of G-actin-latrunculin bound complexes ($N_{\text{Actin-Lat}}$) which is initially zero but increases every timestep by $N_{\text{Lat}} k_{\text{Lat-on}} C_G \Delta t$.

Unbinding is carried out in a similar way with

$$\Delta N_G = N_{\text{Actin-Lat}} k_{\text{Lat-off}} \Delta t, \tag{3.15}$$

where $k_{\text{Lat-off}}$ is the unbinding rate of latrunculin (s^{-1}).

3.5 TIRFM results

3.5.1 Data

All TIRFM experiments were carried out by John Palmer.

TIRFM was used to investigate three different systems: 'Actin only', 'Actin and Las17', and 'Actin, Las17 and Arp2/3'. The actin concentration was 0.5µM in all experiments. This concentration was chosen as it proved difficult to analyse elongation concentrations more than this, as the filaments grow faster and overlap much more. The Las17 concentration was 10nM in all but one experiment where it was

50nM. However there was no difference in elongation or nucleation at the different concentrations. 1nM Arp2/3 complex was used in all experiments.

1500cP (1.5Pa s) (viscosity at 2%, 20°C) methylcellulose was used. This was diluted down such that the viscosity of the medium was 2.5mPa s.

For 'Actin only' and 'Actin and Las17', six separate experiments each were carried out. For 'Actin, Las17 and Arp2/3' four separate experiments were carried out.

All analysis was done using Fiji ImageJ [140]. The details of analysis is provided in the next section. Prior to any analysis the TIRFM images had their brightness and contrast auto corrected. Background was removed using the 'Subtract Background' option in Fiji, with a rolling ball radius of 20px.

3.5.2 Analysis and results

3.5.2.1 Tethering

In the TIRFM the actin filaments are tethered to the glass slide. In order to replicate this in our simulation we needed a measurement of the density of tethers along a filament. Analysis was carried out by Callum Winder, a summer student under the supervision of Ellen Allwood, on data from three TIRFM experiments carried out by John Palmer. Callum used ImageJ to measure distances between tether points along actin filaments. This analysis is presented in Figure 3.5, where the three repeats are shown separately.

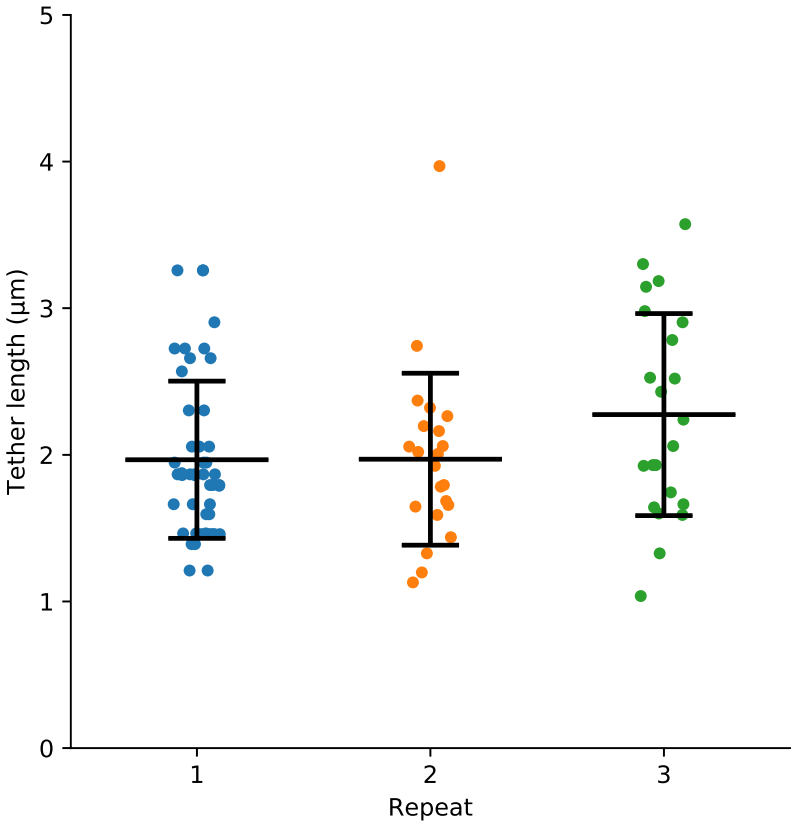


FIGURE 3.5: Measured distances between tethered points along actin filaments. Tether distance (µm) is along the y axis and the three different experiments are shown along the x axis. Data taken by Callum Winder and provided by Ellen Allwood.

Pooling all the data together the overall mean and standard deviation of the tethering lengths is $2.04 \pm 0.60\mu m$.

3.5.2.2 Elongation rate

For analysing the elongation of the actin filaments in the TIRFM data, we used JFilament [143], an ImageJ plugin. In JFilament, tracing a filament and tracking it in time as it grows is possible. The user first chooses a filament they want tracked and simply draws a contour with the mouse. This is then fitted to the filament and can be tracked either forwards or backwards in time. An example image of tracing using JFilament is given in Figure 3.6.

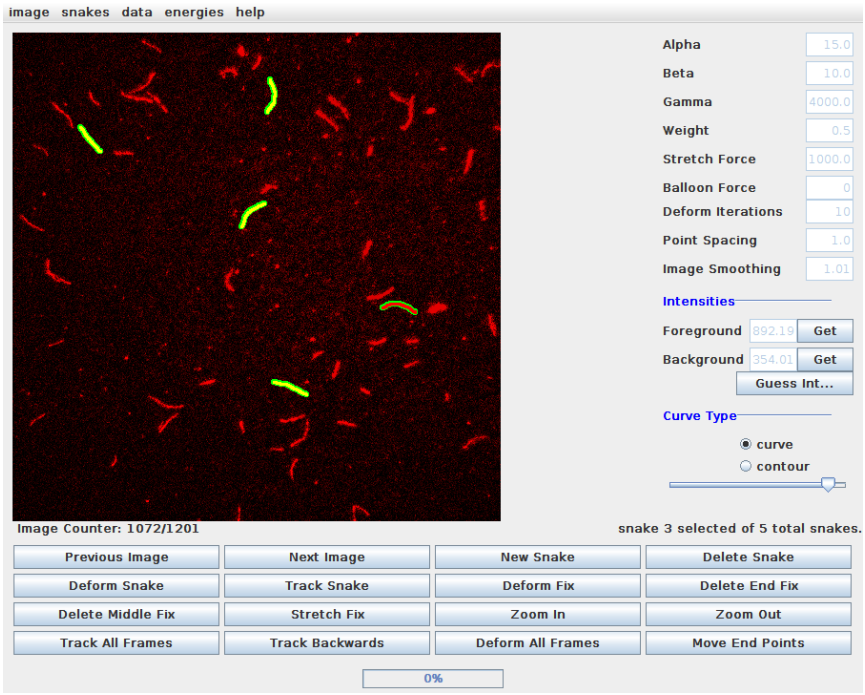


FIGURE 3.6: Screenshot of the JFilament plugin user interface. Traced filaments are shown as yellow contours with a green outline. One traced filament (rightmost) is selected and is shown with a red line running through it. Parameter options are shown on the right.

Once a filament has been tracked through time, its length at each frame can be outputted. Work was required to determine parameter values that work best and overcoming certain problems. For example, to overcome issues with overlapping filaments the filament can be chosen just at the point of overlap and then tracked backwards in time. We found tracking in reverse to be much better than tracking forward in general. Filaments would sometimes fall into the frame from above, but this was rare. We avoided tracing these filaments. We worked closely with John Palmer to use JFilament successfully. Once confident with the plugin, John then used JFilament to analyse the whole dataset. However, in some cases where JFilament failed to work, manual tracing was used instead. Figure 3.7 shows a typical example of elongation data from one experiment in the 'Actin only' system.

John Palmer then fitted the lengths in time data to linear lines using Graphpad Prism [144]. A fit was done for each filament to get the elongation rate for each filament as the gradient of the fit. The elongation rate should be considered the sum

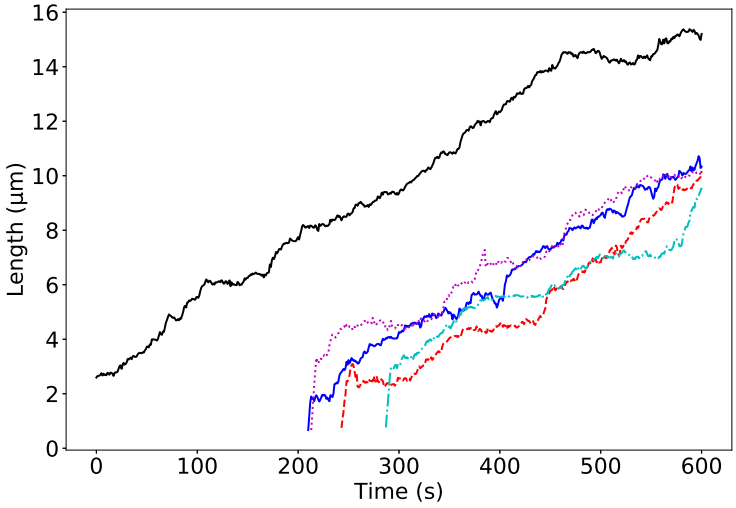


FIGURE 3.7: Typical example of elongation data from the 'Actin only' dataset. Each coloured line represents a single traced actin filament. Length in microns along the y axis against time in seconds along the x axis.

of both barbed and pointed end polymerisation minus the sum of both barbed and pointed end depolymerisation.

Figure 3.8 shows the elongation data measured in all three systems. Each point is the elongation rate of a single filament. Blue points are from the 'Actin only' system, orange for 'Actin and Las17' and green points are from the 'Actin, Las17 and Arp2/3' system. The error bars show the means and standard error of the mean which are given explicitly in Table 3.1.

TABLE 3.1: Measured elongation rates for each system. Uncertainty given is the standard error of the mean.

System	Elongation rate ($nm\,s^{-1}$)
Actin only	15.74 ± 0.80
Actin and Las17	26.5 ± 1.2
Actin, Las17 and Arp2/3	25.8 ± 1.3

We therefore see a ∼65% increase on elongation from actin alone to actin when Las17 is present. The addition of Arp2/3 complex did not significantly change the elongation rate from the 'Actin and Las17' system. The elongation rate of actin alone we measured is slightly higher than previously reported in TIRFM [145].

3.5.2.3 Nucleation rate

For an estimation of the nucleation rate of actin in TIRFM we used the ImageJ plugin, Ridge Detection [146]. Ridge Detection was set up to automatically identify filaments in a given frame. The input parameters were chosen to identify filaments but avoid false positives. This was a difficult balance to achieve as false positives were common due to presence of loose fluorophores. Tweaking parameters to avoid the loose fluorophores then resulted in missing some small filaments. It is also sometimes

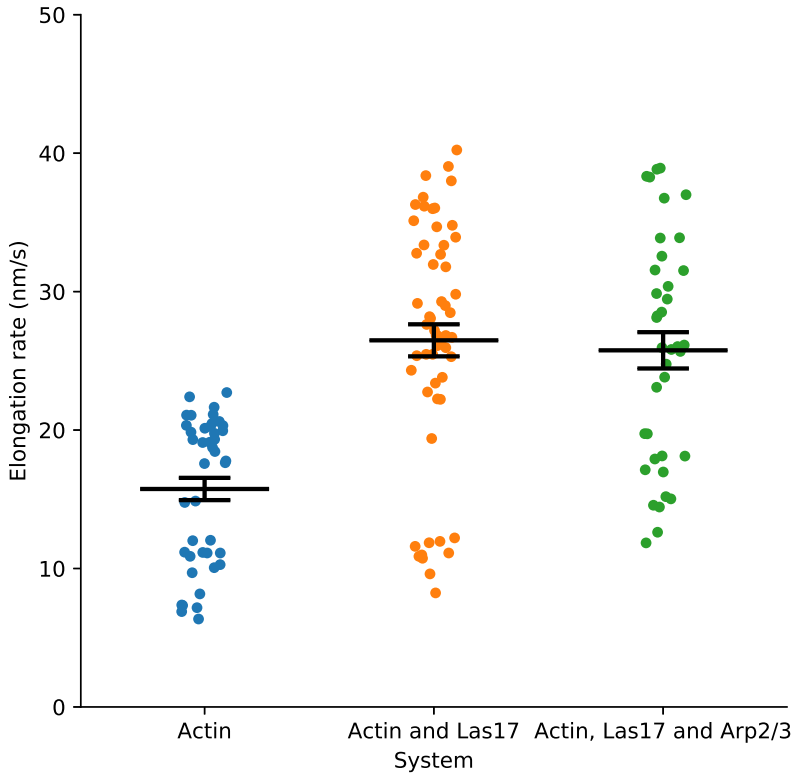


FIGURE 3.8: Elongation rate results. Each point is a elongation rate measured from a fit to elongation data for a single filament. Blue points are from the 'Actin only' system, orange from the 'Actin and Las17' system and green from the 'Actin, Las17 and Arp2/3' system. The mean and standard error of the mean are shown for each system.

difficult, even manually, to ascertain what is a filament and what is fluorescent dye. Once the parameters were calibrated for a typical frame in each experiment, Ridge Detection was run on all the frames. The parameters used were very similar between experiments. Once Ridge Detection has been run we can use the output to simply count the number of filaments in each frame in the videos. A screenshot of Ridge Detection in action is given in Figure 3.9.

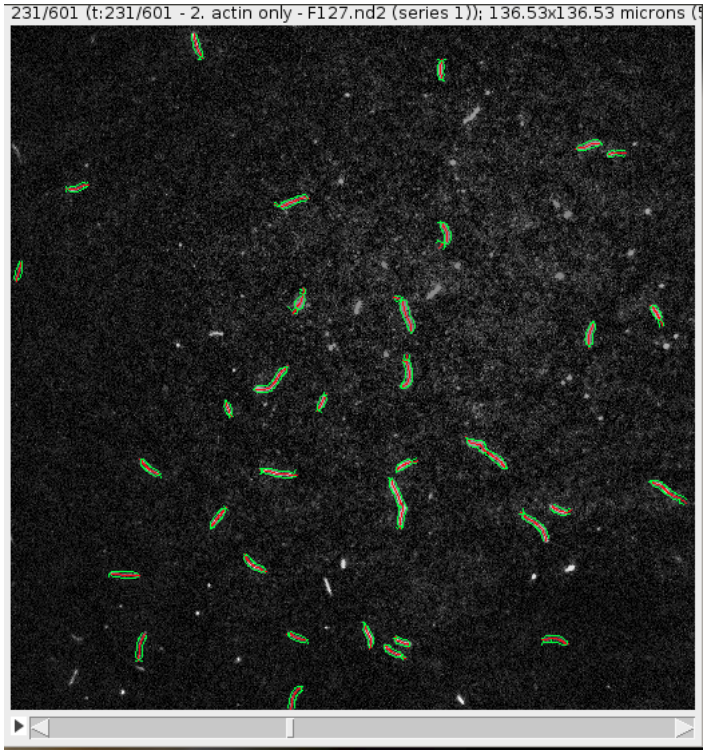


FIGURE 3.9: Screenshot showing result of Ridge Detection plugin on a single frame in the 'Actin and Las17' dataset. Identified filaments are shown as red lines with green outlines.

A typical example from one TIRFM experiment ('Actin only') is shown in Figure 3.10. We then fitted this data for all experiments to linear lines using Origin [147]. The gradient of the linear fit is related to the nucleation rate.

The gradient is the number of filaments nucleated per time. However we need to divide this by the volume that we observe on the TIRFM. We assume a depth of 200nm [60, 139] which we multiply by the area of the frame to get the volume.

This nucleation analysis method was run on the two systems without branching: 'Actin only' and 'Actin and Las17'. We did not analyse the 'Actin, Las17 and Arp2/3' dataset as this method would identify branches as filaments created by nucleation. Nucleation and branching rate analysis could be done in the future (see section 3.8 on future work).

The nucleation rates presented in Figure 3.11 are given in units of $\mu m^{-3}\,s^{-1}$ and are technically nucleation rate volume densities. The nucleation rate will also depend on the concentration of actin. Since in the TIRFM experiments John Palmer always used the same concentration of actin (0.5µM), we do not need to account for this.

We see a ∼57% decrease of nucleation from the 'Actin only' system to the 'Actin and Las17' system. This is the opposite of what is expected and is perhaps a result of limitations of this analysis method. This result will be discussed later in section 3.7.

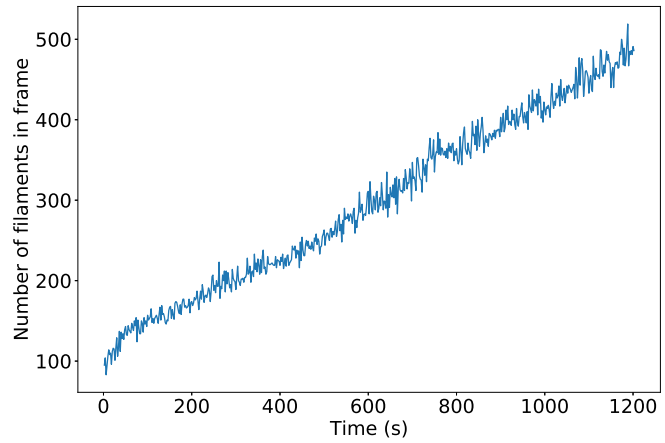


FIGURE 3.10: Typical example of number of filaments in time using Ridge Detection. Taken from 'Actin and Las17' dataset. Number of filaments is shown on the y axis against time in seconds along the x axis.

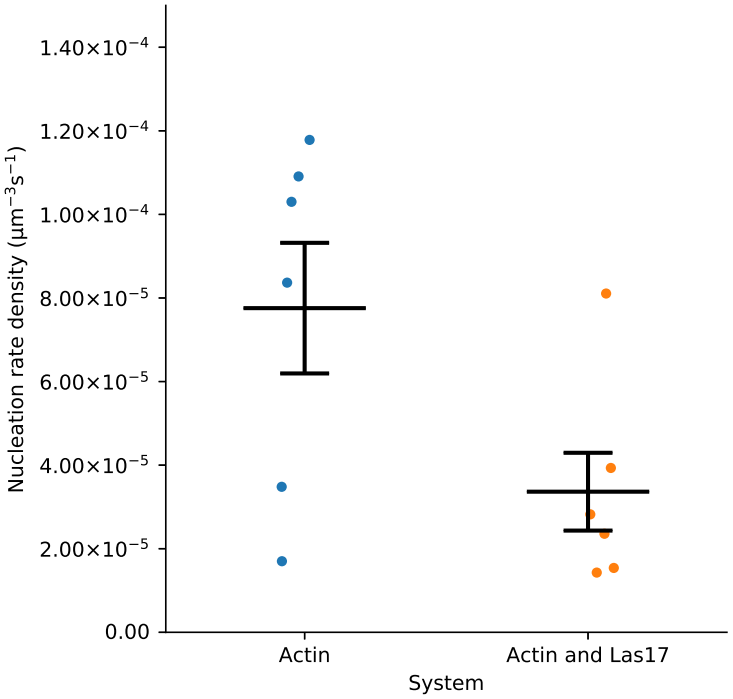


FIGURE 3.11: Nucleation rate results. Each point shows a nucleation rate density from a single experiment. Blue points are from the 'Actin only' system and orange from the 'Actin and Las17' system. The means and standard error on the mean are shown.

TABLE 3.2: Measured nucleation rate densities for each system. Uncertainty given is the standard error of the mean.

System	Nucleation rate density ($\mu m^{-3} s^{-1}$)
Actin only	$7.8 \times 10^{-5} \pm 3.4 \times 10^{-5}$
Actin and Las17	$3.36 \times 10^{-5} \pm 0.93 \times 10^{-5}$

3.5.2.4 Assumption of linear regime

In the two previous sections with both nucleation and elongation, we fitted linear lines to the data. This assumes that the G-actin concentration is not limited. If G-actin were limiting then the amount of nucleation and elongation would slow down over time. We did not observe this in the data and linear fits were good approximations to the data over the timescales of 20-25 minutes. If John Palmer had run the experiments for longer we would expect to see a plateau as the system reaches a steady state. However this was not possible due to photo-bleaching effects.

3.5.3 Simulations

We used our simulation model to reproduce the TIRFM system. This allows us to compare qualitatively simulation to experiment. We can also probe early times that we are unable to observe in experiment.

The TIRFM system is a three-dimensional system however most actin polymerisation and nucleation occurs near the glass surface due to tethering and only around 200nm in height of the sample is observed. Despite tethering, actin filaments can be seen to overlap each other and rarely have their growth or movement hindered by neighbouring filaments. Therefore using our two-dimensional model without any steric hindrance (our simulated filaments can interpenetrate each other) is suitable for simulating the 2D plane at the glass surface. This simplification of the 3D TIRFM system into 2D simulations has been previously made and the 2D simulations were used to investigate TIRFM systems [60].

We ran three different simulations replicating our three actin systems. We used the elongation rates given in Table 3.1 and nucleation rates from Table 3.2. Conversion of these rates was required to input into the simulation. For elongation we assume no depolymerisation and all polymerisation is at the barbed end for simplicity. The conversion for elongation is given in equation by

$$k_p^{\text{barb}} = \frac{k_{\text{e}}}{l_{\text{mon}} C_{\text{G}}}, \tag{3.16}$$

where k_{e} is the elongation rate measured in units of $\text{nm}\,\text{s}^{-1}$, l_{mon} is the length of a monomer in nm (2.7nm), and C_{G} is the concentration of actin in µM (0.5µM).

For nucleation we define an area equal to the size of the frame in the TIRFM (135µm×135µm), and assume again the depth of the TIRFM field to be 200nm. The conversion is given in equation 3.17.

$$k_n = \frac{k_{\text{n}}^{\text{T}} d^{\text{T}}}{C_{\text{G}}^3}, \tag{3.17}$$

where k_{n}^{T} is the nucleation rate volume density measured, d^{T} is the depth of the TIRFM field in µm (0.2µm), and we have a cubic dependence on G-actin concentration which is assumed in the simulation (see Chapter 2).

For the 'Actin, Las17 and Arp2/3' system we assume the same nucleation rate as in the 'Actin and Las17' system. We know the Arp2/3 concentration is 1nM in the experiment. We use the value taken from [110] for the branching rate per available monomer ($k_{\text{branch}} = 5.3 \times 10^{-4}\mu\text{M}^{-3}\,\text{s}^{-1}$) which assumes a linear dependence on Arp2/3 concentration and a quadratic dependence on G-actin concentration. We assume no debranching and a branch spacing of 13 monomers.

In all simulations we assume an infinite pool of G-actin (and Arp2/3 complex for the relevant simulation) as per the discussion in section 3.5.2.4. For tethering we use the findings from section 3.5.2.1 and use a mean μ_{tet} of 2.04µm and standard deviation σ_{tet} of 0.60µm in all simulations. We use a temperature of 290K and a viscosity of 2.5×10^{-3}Pa s from the viscosity of methylcellulose. We use an actin segmentation length l_{s} of 0.5µm. Steric hindrance is turned off so filaments can overlap each other.

We output frames of the same size of the TIRFM frame and do not simulate any boundaries, therefore actin filaments can move out of the field of view like in the experiment. Filaments can polymerise outside of the frame. With the 'Actin, Las17 and Arp2/3' system, filaments can branch anywhere, including outside of the frame. No filaments can be nucleated outside the frame since the nucleation region is the size of the frame.

We ran the simulations for the 'Actin only' and 'Actin and Las17' systems for a runtime of 1500 seconds (25 minutes). The 'Actin, Las17 and Arp2/3' simulation took much longer to run (in CPU wall time) and so was run for a reduced runtime of 1250 seconds (20 minutes and 50 seconds). The CPU time of the 'Actin only', 'Actin and Las17' and 'Actin, Las17 and Arp2/3' simulations were ∼90 hours, ∼120 hours and ∼350 hours respectively.

Figure 3.12 shows four snapshots of the simulation replicating the 'Actin only' system. The snapshots are at (A) 150 seconds, (B) 600 seconds, (C) 1050 seconds and (D) 1500 seconds.

Figure 3.13 shows four snapshots of the simulation replicating the 'Actin and Las17' system. The snapshots are at (A) 150 seconds, (B) 600 seconds, (C) 1050 seconds and (D) 1500 seconds.

Figure 3.14 shows four snapshots of the simulation replicating the 'Actin, Las17 and Arp2/3' system. The snapshots are at A) 150 seconds, B) 600 seconds, C) 1000 seconds and D) 1250 seconds.

These simulations show a good qualitative agreement with the TIRFM experiments. The analysis of individual filament processes (elongation, nucleation, tethering) is therefore validated at a qualitative level.

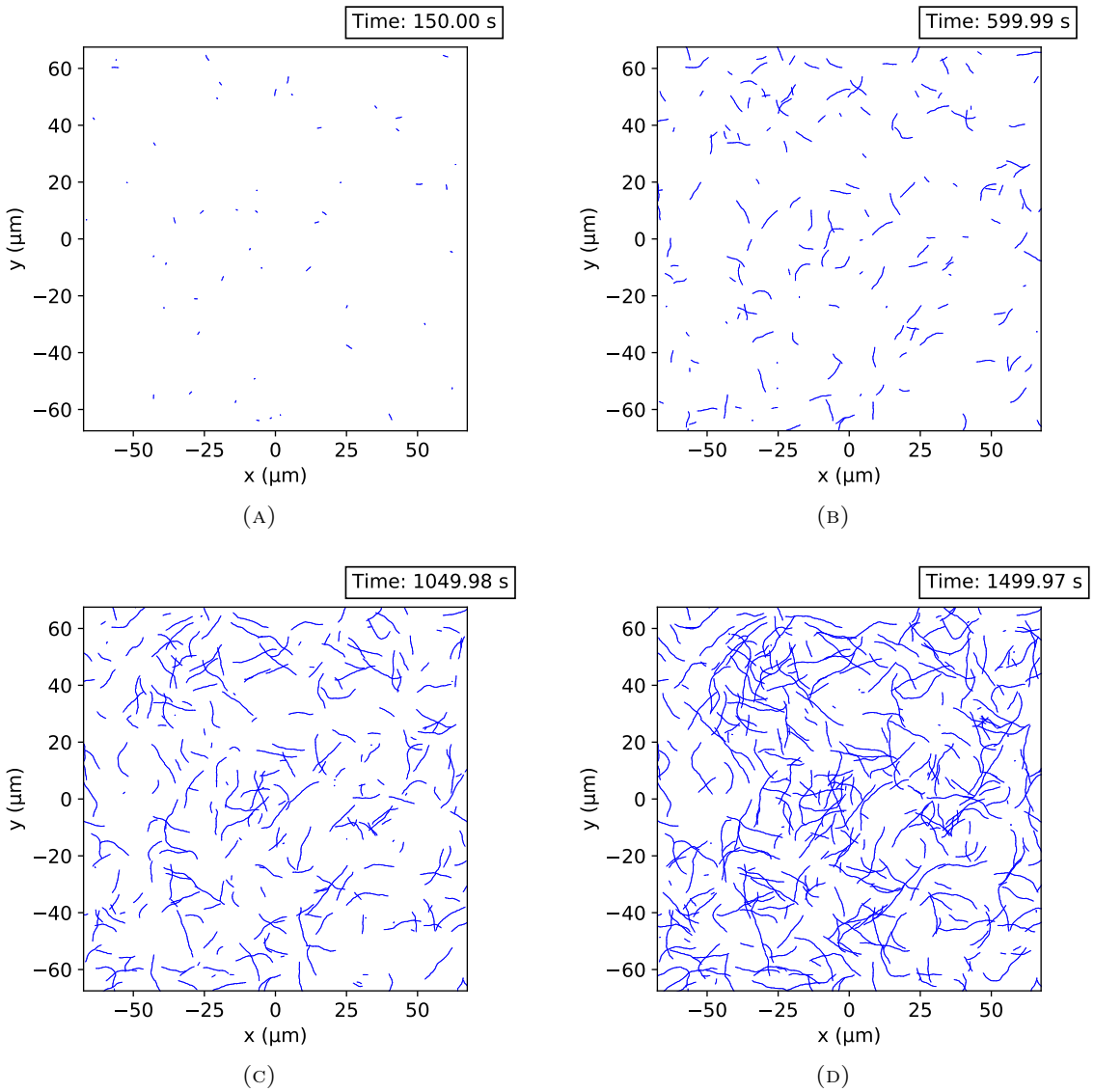
3.6 Pyrene results

3.6.1 Data

All pyrene assay experiments were carried out by Ellen Allwood. Like in the TIRFM system, pyrene assays were used to investigate the three different systems; 'Actin only', 'Actin and Las17' and 'Actin, Las17 and Arp2/3'.

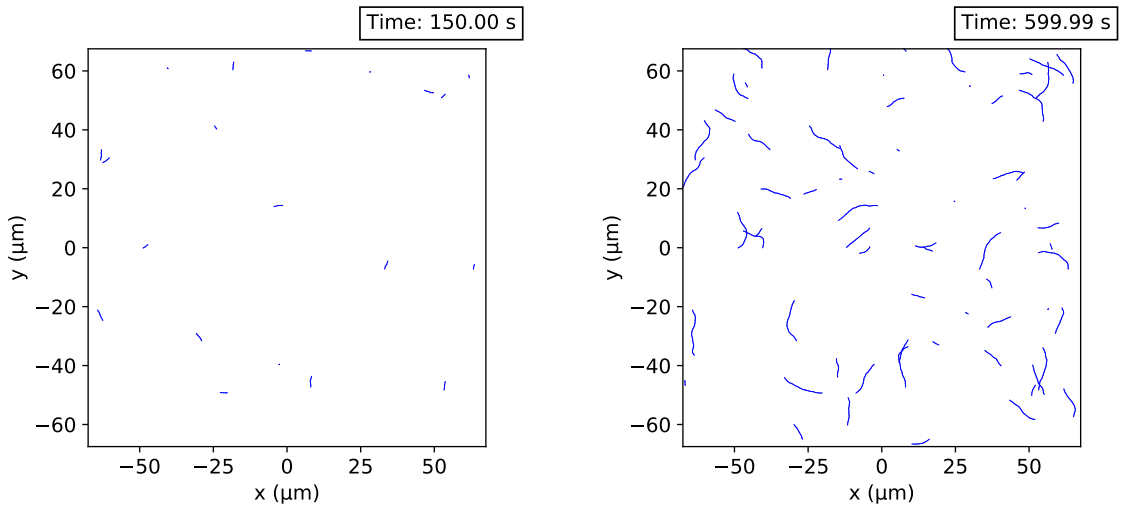
For each system eight different concentrations of actin were used. Ellen starts by making up a 5ml pyrene-actin-buffer solution with an actin concentration of 6.20µM. She then does a dilution series which is taking 4mL of the solution and adding 1mL of G-buffer each time for a total of seven times. Each time the solution is therefore diluted by 20%. Finally, to begin the actin polymerisation process an equal amount of KMEI buffer is added to the actin solutions, therefore diluting each one by further 50%. The final actin concentrations were: 3.10±0.31µM, 2.48±0.25µM, 1.98±0.20µM, 1.59±0.16µM, 1.27±0.13µM, 1.02±0.10µM, 0.812±0.081µM, 0.650±0.065µM. Here we have assumed a 10% error in the concentrations.

The Las17 and Arp2/3 concentrations are kept the same in all experiments and are 100nM and 2nM respectively. Pyrene-actin represented 9.6% of all the actin in the system.



(A) (B) (C) (D)

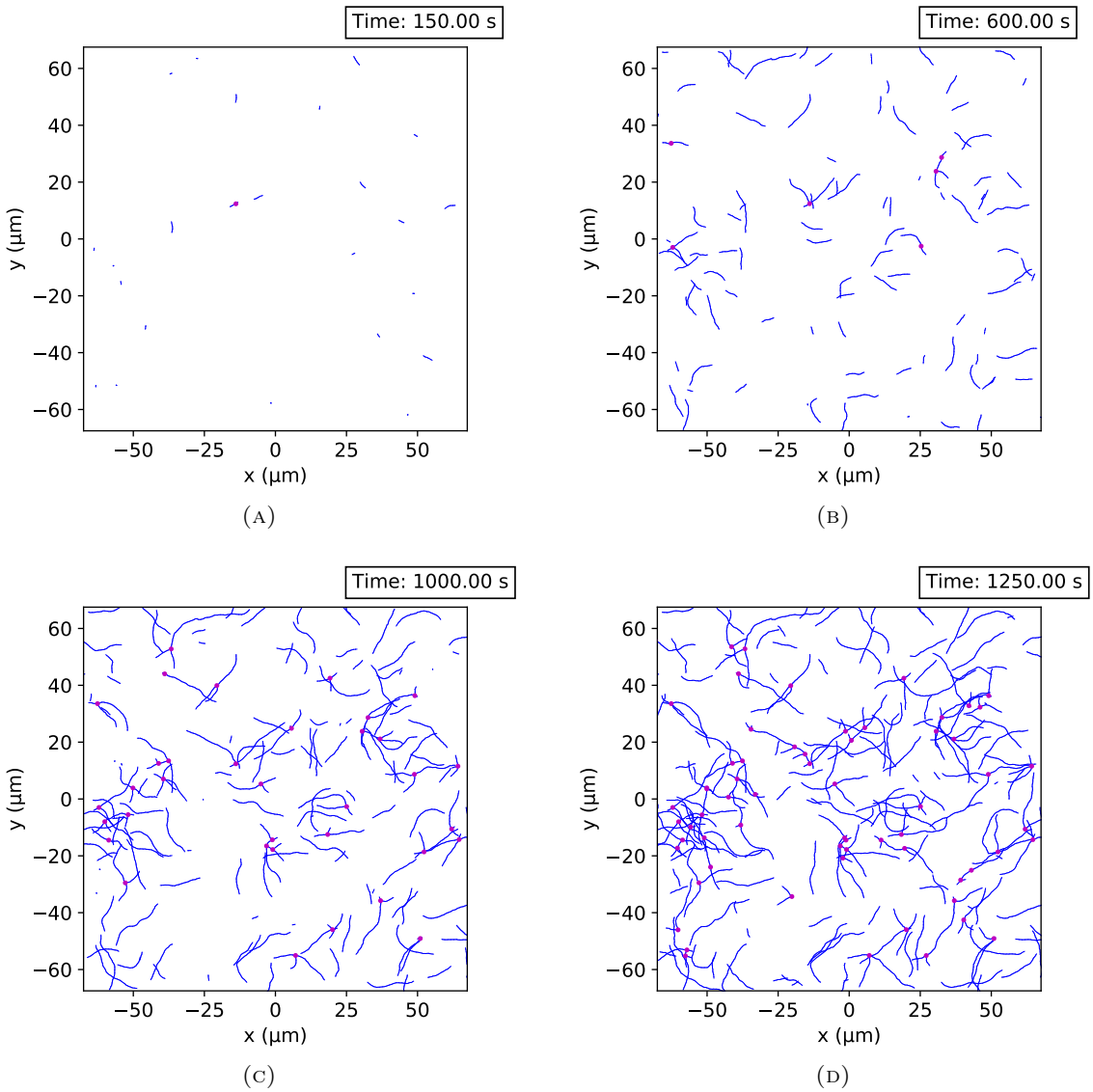
FIGURE 3.12: 'Actin only'



(A) (B)

(C) (D)

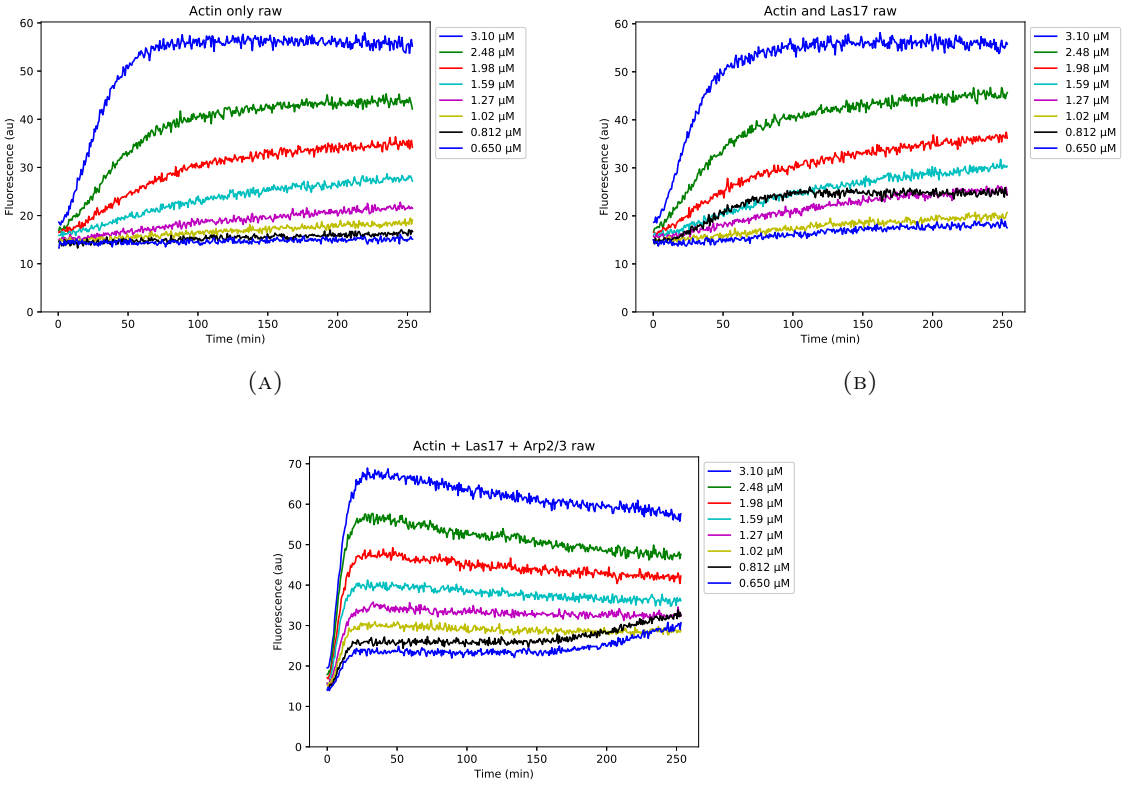
FIGURE 3.13: 'Actin and Las17'



(A) (B) (C) (D)

FIGURE 3.14: 'Actin, Las17 and Arp2/3'

The experiments were run for 250 minutes. The raw data is presented in Figure 3.15.



(A) (B)

(C)

FIGURE 3.15: Raw pyrene data. Each coloured line represents a different concentration of actin. Blue: 3.10µM, green: 2.48µM, red: 1.98µM, sky blue: 1.59µM, magenta: 1.27µM, yellow: 1.02µM, black: 0.812µM and finally blue again: 0.650µM. Fluorescence along the y axis measured in arbitrary units, time along the x axis in minutes. **(A)** 'Actin only' system. **(B)** 'Actin and Las17' system. **(C)** 'Actin, Las17 and Arp2/3' system. Data taken by Ellen Allwood.

One noticeable characteristic of the curves from the 'Actin, Las17 and Arp2/3' system (Figure 3.15C) is the reduction seen after the system has plateaued. This effect is more apparent in the higher actin concentrations. We will ignore it for now but it will be discussed later in section 3.7.

3.6.2 Analysis and results

To understand the three systems in the pyrene experiments we developed an analysis pipeline for this data. In this section the stages of our pipeline are described in order.

3.6.2.1 Adding the time delay

The first step is the addition of the initial time delay. Ellen Allwood begins the experiment by adding the KMEI salt buffer to the actin solution, starting the actin polymerisation process. There then is a delay between that and the first measurement from the fluorimeter, this time is recorded using a stopwatch. We then add this delay to the data which shifts the curves to the right.

3.6.2.2 Removing the background

We assume no filaments are present at the start of the experiment. However, in Figure 3.15 it can be seen that the initial fluorescence is not zero (even taking into account the short time delay). We define a 'baseline' which is the initial fluorescence at the very start of the experiment when the salt buffer is added, before the first measurement. This baseline is composed of a constant background fluorescence term and the fluorescence of monomers. We can write down an equation for the fluorescence at any given time during the experiment as

$$\alpha f(t) = c_{\mathrm{p}}(t) + a c_1(t) + B, \tag{3.18}$$

where α, which will be discussed later, is a conversion factor from fluorescence to concentration, in units of $\mu\mathrm{M}\mathrm{fl}^{-1}$ (where fl is the arbitrary unit of fluorescence). $f(t)$ is the total fluorescence in the system at time t and is on the y axes in Figure 3.15. $c_{\mathrm{p}}(t)$ is the total concentration of actin monomers in F-actin form at time t. $c_1(t)$ is the total concentration of actin monomers in G-actin form at time t. a is the fractional term for fluorescence of G-actin monomers to F-actin monomers. B is the constant background term.

By considering the initial condition assumption that there are no filaments we can write this as

$$\alpha f(0) = a c_0 + B, \tag{3.19}$$

where c_0 is the total concentration of actin and is known.

We can rearrange equation 3.19 as

$$f(0) = \frac{a}{\alpha} c_0 + B_{\mathrm{f}}, \tag{3.20}$$

where $B_{\mathrm{f}} = \frac{B}{\alpha}$ which is the constant background but in fluorescence units. To determine B_{f} we first need to measure $f(0)$. We use Origin software [147] to fit linear lines to the first few minutes of fluorescence data. The example shown in Figure 3.16 is the fit to the 'Actin Only' 1.98µM curve, only the initial 12 minutes have been fitted in this case. All twenty four (eight for each system) curves were fitted in this way.

The fit to the initial data (example is presented in Figure 3.16) gives a gradient and intercept both with uncertainties. The shaded region is simply the region between the upper and lower limit in the uncertainty of $f(t)$. In general terms, for a straight line defined as $y = mx + c$, the uncertainty of y is $\Delta y = \sqrt{((\Delta m)x)^2 + (\Delta c)^2}$, where the delta refers to the associated error (assuming no uncertainty in x). The upper and lower limits of the shaded region are then $y + \Delta y$ and $y - \Delta y$ respectively.

The intercepts of these fits are what we determine to be the initial fluorescence $f(0)$.

We can then fit equation 3.20 for each of our three systems. The example shown in Figure 3.17 is from the 'Actin, Las17 and Arp2/3' system data.

This fit therefore gives B_{f} as the intercept and $\frac{a}{\alpha}$ as the gradient. We can then remove the constant background, B_{f} from the data.

3.6.2.3 Converting to concentration units

To convert to concentration units we consider the plateau region of the curve. The curve plateaus as the system reaches steady state. Steady state is achieved when polymerisation is equal to depolymerisation and the concentration of G-actin is $\frac{k_d}{k_p}$.

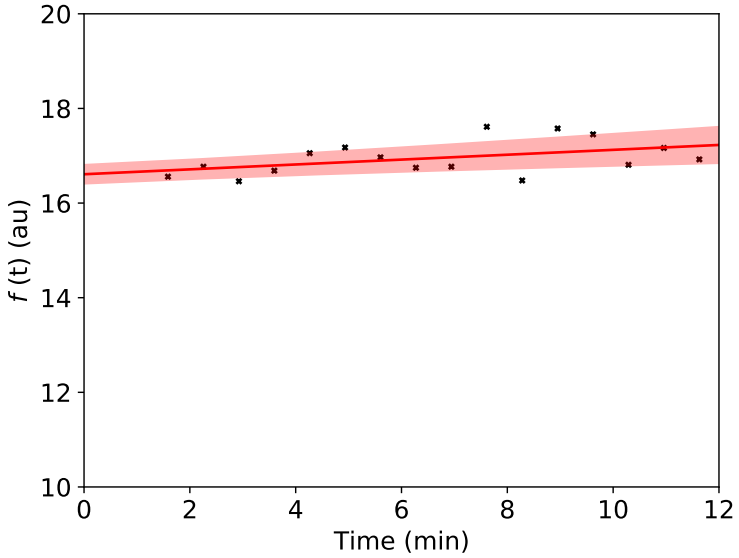


FIGURE 3.16: Example baseline fit 1.98µM 'Actin Only' data. Fluorescence along the y axis in arbitrary units, time in minutes along the x axis. The fit is shown as the red line with associated uncertainty shown as the red shaded region. The initial linear part of the curve is fitted which for this data corresponds to the first 12 minutes. The baseline is determine from the y intercept of the fit.

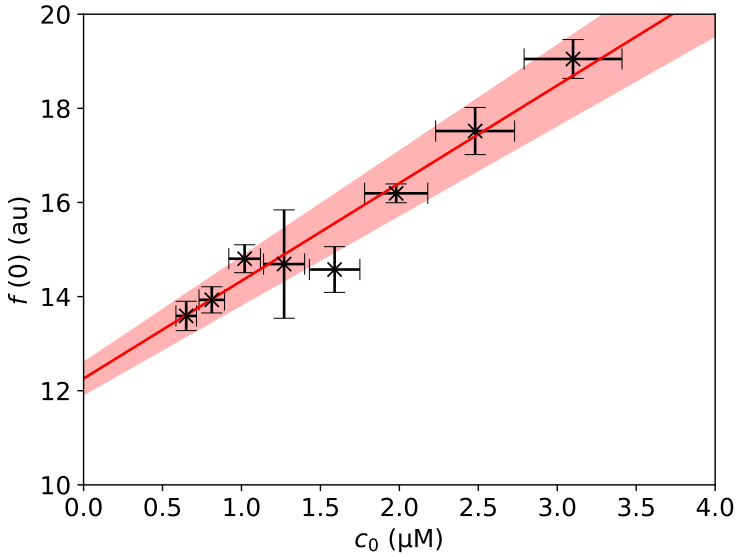


FIGURE 3.17: Example of fit to equation 3.20 'Actin, Las17 and Arp2/3' dataset. Each data point is from a single curve for an actin concentration shown along the x axis. Initial fluorescence is shown along the y axis. Horizontal error bars are the assumed 10% error in initial concentration. Vertical error bars are the uncertainty in the value of $f(0)$ from fitting to first few minutes of fluorescence data previously described. The uncertainty in the fit is the red shaded region. The y intercept corresponds to the constant background B_f. The gradient corresponds to $\frac{a}{\alpha}$.

By using this we can adapt equation 3.18 to

$$c_{\text{tot}}(t = ss) = \alpha f(t = ss) - B = \left[c_0 - \frac{k_d}{k_p}\right] + a\frac{k_d}{k_p}, \tag{3.21}$$

where the left hand side is the background reduced fluorescence converted to concentration units, defined $c_{\text{tot}}(t) = \alpha f(t) - B$. $t = ss$ means the time is such that the system is at steady state and the curve has plateaued. The term in the square brackets is the concentration of F-actin monomers and the last term is the concentration of G-actin monomers multiplied by the fluorescence ratio a. For any two curves we can therefore write

$$\alpha \Delta f(t = ss) = \Delta c_0, \tag{3.22}$$

where $\Delta f(t = ss)$ is the difference between fluorescence steady state values of the given two curves, and Δc_0 is the difference of total actin concentration.

We again turn to Origin and first fit linear lines to the plateaued region of each (time delay adjusted) curve to find a steady state fluorescence value $f(t = ss)$. The example given in Figure 3.18 is a fit to data from the 'Actin, Las17 and Arp2/3' system 1.59µM. The blue data points in Figure 3.18 are points not included in the fit. We choose which points to fit manually each time. The fluorescence value of where the green cross marks is what we measure as the steady state fluorescence value $f(t = ss)$. The time value of where the green cross marks is equal to the time value of the first point in the plateaued region of which we fit. Only the curves that were deemed to have reached steady state were fit in this way. The curves we fitted were: 'Actin Only' 3.10µM, 2.48µM, 1.98µM and 1.59µM, 'Actin and Las17' 3.10µM, 2.48µM and 1.98µM, and finally 'Actin, Las17 and Arp2/3' all eight. In Figure 3.15C the curves for 0.812µM and 0.650µM show a strange increase towards the end of the experiment. This increase is ignored as we suspect this is not real. We also believe the 0.812µM curve for 'Actin and Las17', which was shown in Figure 3.15B is incorrect.

We then fit equation 3.22 to all the different combinations of curves that have reached steady state. This therefore gives us the value of α and by extension a since we already know $\frac{a}{\alpha}$. The example of this fit given in Figure 3.19 is from the 'Actin only' system.

This method assumes that pyrene fluorescence is linearly dependent with pyrene-F-actin concentration. It was previously reported that this was not the case [148], but this is not what we see in our case. We see a strong linear relationship between concentration and fluorescence in all three of our experimental systems, one example of which is shown in Figure 3.19.

Once α is known we can then convert the data to concentration units by multiplying the fluorescence values by α. The final processed curves are shown in Figure 3.20. The curve for 0.812µM actin in the 'Actin and Las17' system has been omitted, we previous mentioned we believe this to be an error.

It is worth noting that these concentrations are actually scaled up. The pyrene-actin represents just 9.6% of the actin in the system, and so what is measured in the fluorimeter is around one tenth of what is actually present. For simplicity in understanding, this fact is ignored in the conversion since it is just a scaling up that is done. These curves in concentration units should be representative of the entire system, but it is the pyrene labelled actin we are actually measuring.

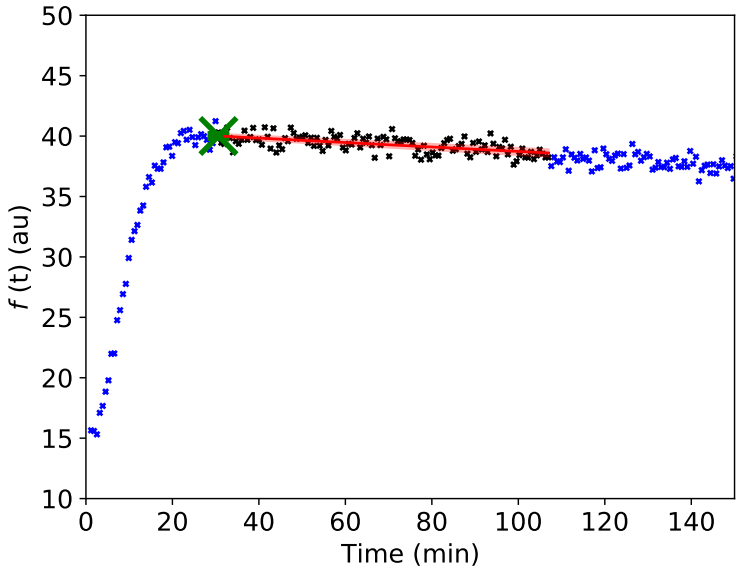


FIGURE 3.18: Typical example of the fit to find the steady state fluorescence value for a given curve. This example is 1.59µM 'Actin, Las17 and Arp2/3' system. Black data points are those chosen to be included in the fit whereas blue data points are not included. The red line is the fit result and associated uncertainty is shown as the red shaded region. The green cross marks where the steady state begins. Fluorescence shown along the y axis and time in minutes shown along the x axis.

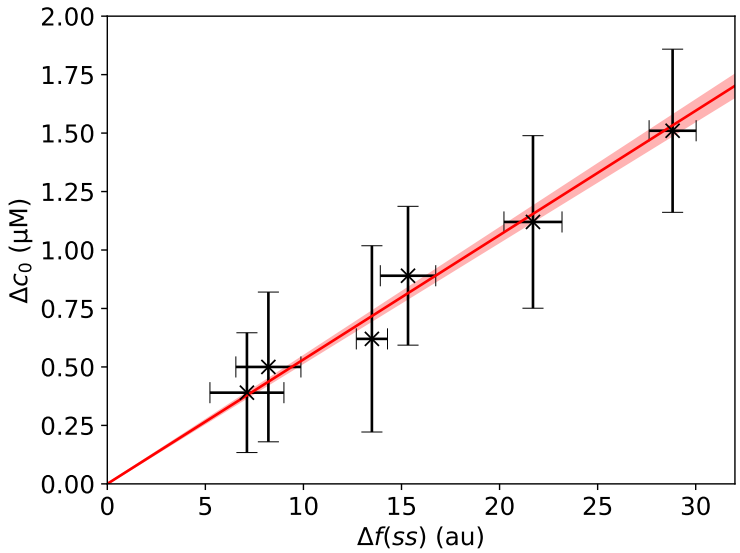
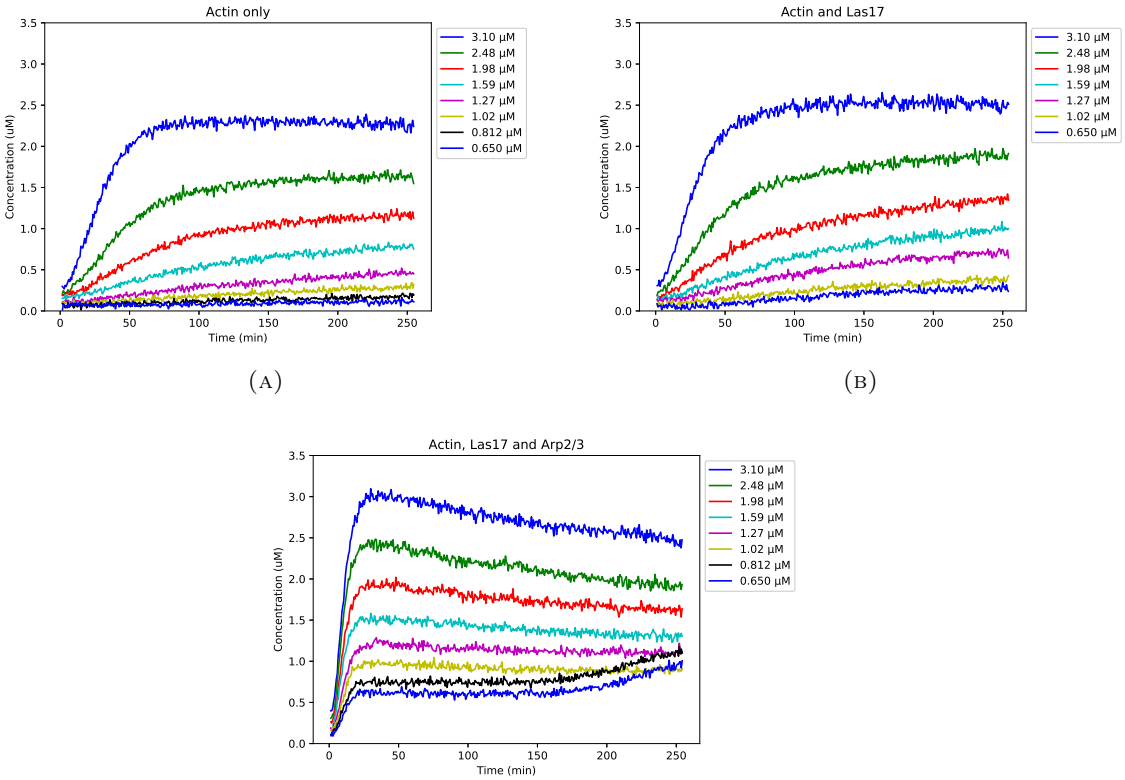


FIGURE 3.19: Typical example of a fit to equation 3.22. This example is the 'Actin only' system. Fit result is shown as a red line with associated uncertainty as the red shaded region. Each data point represents a comparison between two curves. The y axis shows the difference in actin concentration (µM). The x axis shows the difference in steady state fluorescence value. The gradient of this fit is α.



(A) (B)

(C)

FIGURE 3.20: Pyrene assay data adjusted by adding the time delay recorded by Ellen Allwood, removing the constant background B_f and converting into concentration units. Each coloured line represents a different concentration of actin. Blue: 3.10µM, green: 2.48µM, red: 1.98µM, sky blue: 1.59µM, magenta: 1.27µM, yellow: 1.02µM, black: 0.812µM and finally blue again: 0.650µM. Concentration along the y axis in µM, time along the x axis in minutes. **(A)** 'Actin only' system. **(B)** 'Actin and Las17' system. The 0.812µM curve has been omitted. **(C)** 'Actin, Las17 and Arp2/3' system. Data taken by Ellen Allwood.

3.6.2.4 Determining the steady state concentration

We can then find the steady state G-actin concentration, $\frac{k_d}{k_p}$ by first simplifying equation 3.21 to give

$$c_{\text{tot}}(t = ss) = c_0 + (a - 1)\frac{k_d}{k_p}. \tag{3.23}$$

Followed by fitting this equation to the data for the intercept $(a-1)\frac{k_d}{k_p}$. The example of this fit is shown in Figure 3.21 and is from the 'Actin and Las17' system.

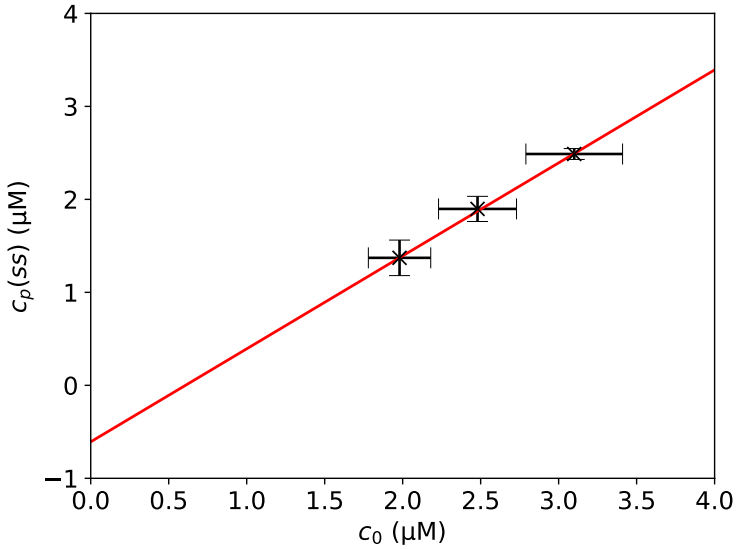


FIGURE 3.21: Typical example of the fit to find the critical concentration using equation 3.23. 'Actin and Las17' system. Y axis is the steady state concentration value µM and x axis is the actin concentration µM. Red line shows fit result and associated uncertainty is the red shaded region. The intercept is $(a-1)\frac{k_d}{k_p}$.

3.6.2.5 Results

The values obtained from this analysis pipeline are given in Table 3.3.

TABLE 3.3: Values of quantities determined from analysis.

System	a	B_{f} (fl)	α $(\mu\text{M}fl^{-1})$	k_d/k_p (µM)
Actin only	0.0911 ± 0.0082	13.01 ± 0.18	0.0532 ± 0.0015	0.902 ± 0.085
Actin and Las17	0.086 ± 0.010	13.50 ± 0.22	0.0594 ± 0.0012	0.665 ± 0.080
Actin, Las17 and Arp2/3	0.113 ± 0.013	12.26 ± 0.34	0.0546 ± 0.0003	0.055 ± 0.010

We measured the G-actin to F-actin fluorescence intensity fraction, a to be approximately 0.1 in all three systems, agreeing with what has been reported previously [136]. The critical concentrations for the 'Actin only' and 'Actin and Las17' systems were higher than expected. Actin alone typically has been reported to have a critical concentration of around 0.2µM [6, 145]. The critical concentration value measured for the 'Actin, Las17 and Arp2/3' system was as expected, since the addition of Las17 should decrease the critical concentration rate below the 0.2µM we expected for actin

alone. The significant difference of over a magnitude in critical concentration between 'Actin and Las17' and 'Actin, Las17 and Arp2/3' was not expected. The critical concentration of branched systems will be discussed later in 3.8.

3.6.2.6 Determining nucleation rate and order

By using the equations outlined in section 3.3.1 we can determine the nucleation rate k_n and the nucleation order β in the two linear systems ('Actin only' and 'Actin and Las17'). The 'Actin, Las17 and Arp2/3 system' has one too many parameters and this system is covered later in the section 3.6.2.7. For fitting to our differential equations we used Wolfram Mathematica [149]. The method is explained below.

First, data that has had the time delay added and most of the steady state plateaued portion removed is imported into Mathematica. We neglect the long plateaued region as we want to concentrate the fit to the growth part of the curve as that is where k_n has an effect. The steady state concentration of any actin system is independent of the nucleation rate. After importing, the data has the background (B_f) removed and is multiplied by α to convert into concentration units. Both values for B_f and α are taken from Table 3.3.

The units we work in is actually number of molecules rather than concentration, so a conversion between the two is needed. Similar to equation 3.13 we must multiply concentrations by the product of the volume in m^3, Avogadro's constant and 10^{-3} to get number of molecules. The volume of the sample in the cuvette used is 370µL ($3.7\times10^{-7}m^3$). However we consider a volume eight orders of magnitude smaller than this ($3.7 \times 10^{-15}m^3$) as fitting to large numbers takes much longer computationally. The data imported is therefore converted into number of molecules against time by multiplying by our conversion factor.

For values in the equations themselves, we use an estimate for k_p of $10\mu M^{-1}\,s^{-1}$, this is an order of magnitude estimate based on previous work [6, 145]. We divide k_p by our conversion so it becomes a polymerisation rate per filament per number of G-actin molecules per second. We then use the values from Table 3.3, the critical concentration (which gives us k_d), and a for monomer fluorescence. The only unknowns in the equations are β and k_n. To determine these we used the function `ParametricNDSolveValue` to numerically fit the equations to the data, solving for k_n for different integer values of β (1,2,3,4). However `ParametricNDSolveValue` failed to work with β values greater than one, so manual fitting by eye was necessary.

Figure 3.22 shows a typical example of a fit result.

The value of k_n that is determined is in units of $N_G^{-\beta}\text{min}^{-1}\times10^{-8}$ (N_G is the number of G-actin molecules). It is converted into $\mu m^{-3}\,\mu M^{-\beta}s^{-1}$.

For each system (Actin only, Actin and Las17) and for each actin concentration we therefore get values for k_n for each of our different values of β. The correct choice of β would produce a constant k_n value across the different actin concentrations. We first normalise the nucleation rates by the nucleation rate for the highest actin concentration (3.10µM). Then, using Origin, fit a linear line to each of the four sets corresponding to the values of β. A plot of the fits are given in Figure 3.23.

The gradients from the fits are presented in table 3.4.

The model which uses the correct value for β should theoretically produce a fit with a gradient value of $0\mu M^{-1}$, since there should be no dependence of the value of k_n on concentration. The analysis of the 'Actin only' system suggests that actin nucleation scales cubicly with G-actin concentration, and therefore F-actin 'seeds' are trimers. The analysis of the 'Actin and Las17' system suggests that nucleation by Las17 also scales cubicly. The weighted means of the cubic nucleation rates were

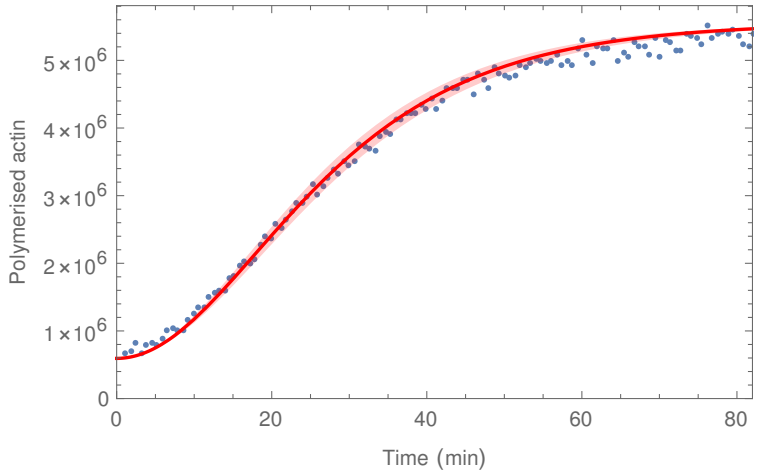
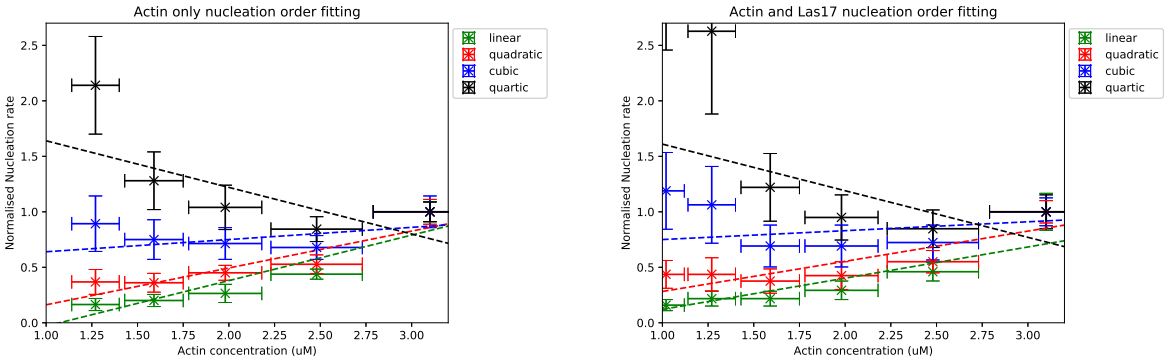


FIGURE 3.22: Typical example of a fit to a pyrene assay curve to determine the nucleation rate. Example is shown is 3.10µM 'Actin and Las17' system assuming a quadratic dependence of nucleation on actin concentration ($\beta = 2$). Y axis shows the quantity c_m defined in the equation 3.4, in units of number of molecules. X axis is simply time in minutes.



(A) (B)

FIGURE 3.23: Nucleation order results. Y axis is the nucleation rate density normalised to the value for 3.10µM. X axis is the actin concentration (µM). Colours refer to different values of β used: green for linear ($\beta = 1$), red for quadratic ($\beta = 2$), blue for cubic ($\beta = 3$) and black for quartic ($\beta = 4$). Dashed lines are the fit results. **(A)** 'Actin only' system. **(B)** 'Actin and Las17' system.

TABLE 3.4: Gradients of nucleation scaling fits for the four different models of nucleation.

System	Linear (μM^{-1})	Quadratic (μM^{-1})	Cubic (μM^{-1})	Quartic (μM^{-1})
Actin only	0.41 ± 0.10	0.332 ± 0.090	0.11 ± 0.12	-0.42 ± 0.29
Actin and Las17	0.281 ± 0.072	0.272 ± 0.081	0.08 ± 0.11	-0.42 ± 0.36

$1.09 \pm 0.09 \times 10^{-6}\mu m^{-3}\,\mu M^{-3}\,s^{-1}$ and $1.35 \pm 0.12 \times 10^{-6}\mu m^{-3}\,\mu M^{-3}\,s^{-1}$ for 'Actin only' and 'Actin and Las17' respectively. The nucleation rate in the 'Actin and Las17' system is therefore ~24% higher than that in the 'Actin only' system.

3.6.2.7 Determining branching rate and order

We used the equations given in section 3.3.2 to fit the data of the 'Actin, Las17 and Arp2/3' system. We assumed the same nucleation order as the 'Actin and Las17' system ($\beta = 3$). We assumed the same nucleation rate from the 'Actin and Las17' system. We make these assumptions because Arp2/3 should not affect linear nucleation. Although interestingly, recent work has shown some Arp2/3 activators can activate Arp2/3 such that it can nucleate linear filaments [138, 150]. Las17 has not been found to do this.

The weighted mean of the cubic nucleation rates of all concentrations in the 'Actin and Las17' system was $1.35 \pm 0.12 \times 10^{-6}\mu m^{-3}\,\mu M^{-3}\,s^{-1}$. The only remaining unknowns are γ and k_b. We fit k_b for three different values of γ, (1,2,3). We used the same method as the nucleation fitting for treating the data, which require values for a, α, B_f and $\frac{k_d}{k_p}$. We take these values from the 'Actin, Las17 and Arp2/3' system in Table 3.3. We again assume a k_p of $10\mu M^{-1}\,s^{-1}$. Required conversions to units of number are done as for the nucleation fitting. Fitting was done manually.

A typical example fit is shown in Figure 3.24, which is clearly poor compared to that of Figure 3.22.

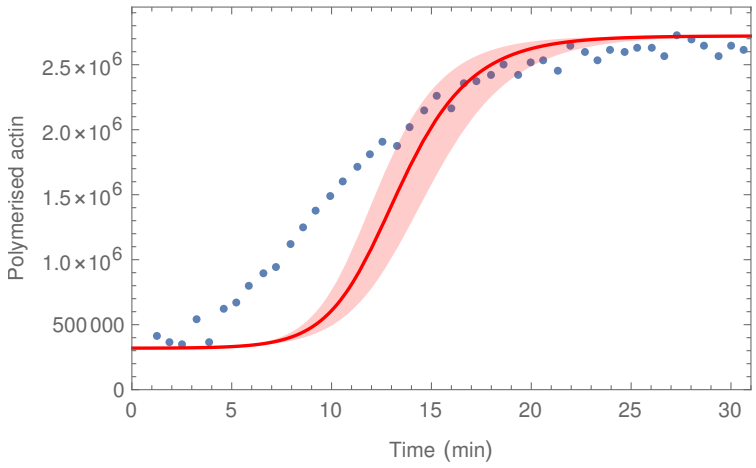


FIGURE 3.24: Typical example of a fit to a pyrene assay curve to determine the branching rate. Example is shown is 1.27µM 'Actin, Las17 and Arp2/3' system assuming a quadratic dependence of branching on actin concentration ($\gamma = 2$). Y axis shows the quantity c_m defined in the equation 3.4, in units of number of molecules. X axis is time in minutes.

Fitting the equations to the individual curves produced poor results of which Figure 3.24 is a typical example. The shape of the fit rarely matched that of the data. This is most likely due to the assumptions made in the equations, or perhaps the assumption of the value of the polymerisation rate. Despite the poor nature of these fits we continued with the same method used in the previous section to plot the branching rate against actin concentration for our three values of γ. As with the nucleation fitting, we normalise k_b to the value of k_b for the highest actin concentration, 3.10µM. We use Origin to fit linear linear lines to each of our three sets of data. A

plot of the fits are given in Figure 3.25. Results of the branch order fits are given in Table 3.5.

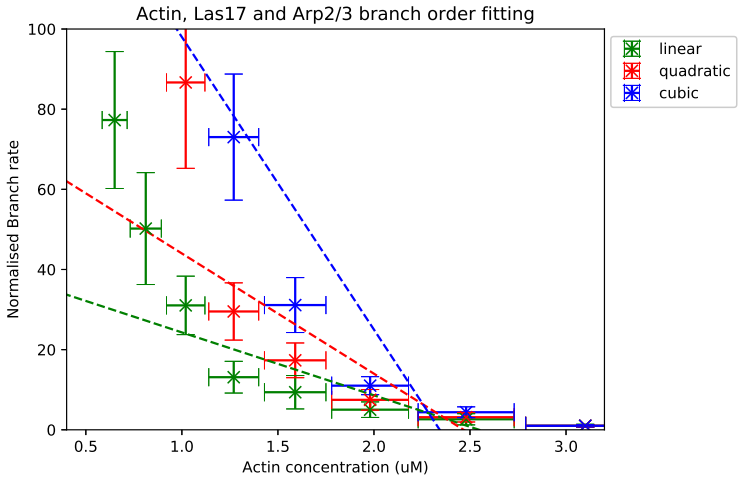


FIGURE 3.25: Branching order results for the 'Actin, Las17 and Arp2/3' system. Y axis is the branching rate per F-actin monomer normalised to the value for 3.10µM. X axis is the actin concentration µM. Colours refer to different values of γ used: green for linear ($\gamma = 1$), red for quadratic ($\gamma = 2$) and blue for cubic ($\gamma = 3$). Dashed lines are the fit results.

TABLE 3.5: Results of branching scaling fits for the three different models of branching.

System	Linear (μM^{-1})	Quadratic (μM^{-1})	Cubic (μM^{-1})
Actin, Las17 and Arp2/3	-15.7 ± 6.6	-30 ± 17	-73 ± 51

The linear fits shown in Figure 3.25 are also poor. All three models show a strong dependence on actin concentration, which increases with order. The data also does not appear to be even linear in actin concentration. From this analysis it is difficult to draw any conclusions. A linear dependence is most likely but more analysis and theory work is required to determine this.

3.6.3 Simulations

We used our simulation model to replicate the pyrene system. This allows us to validate both the analysis of the experimental data and the analytic differential equations that were solved numerically for sections 3.6.2.6 and 3.6.2.7. Just like with the TIRFM simulations in section 3.5.3 we can view early times not possible in the experimental system.

We ran simulations replicating the three systems: 'Actin only', 'Actin and Las17' and 'Actin, Las17 and Arp2/3'. For each system we ran simulations for each of our eight actin concentrations. We setup each simulation's geometry carefully considering what was discussed in section 3.4.2. The nucleation region was a square of side length 100µm, and we assume a depth of 200nm like in the TIRFM. This then gives the nucleation region having an effective volume of 2000µm^3 (2×10^{-15}m^3). This simulation size was chosen as it is large enough to produce sufficiently good statistical

averages but small enough for the simulation to be able to run in a reasonable amount of time (at most a few hours). We therefore need our G-actin (and Arp2/3) grid to have the same volume but be much larger in cross-sectional area. The geometry of the G-actin (and Arp2/3) region is therefore a square of side length 1cm with a depth of 0.02nm, this again gives a volume of 2000µm^3. Again to be clear, the 'depth' of these regions is simply to calculate a volume for conversion of molecule concentration to and from molecule number, the simulation model is still in 2D.

In general, the simulation outputs the total length of all actin filaments, as well as total number of G-actin (and Arp2/3) molecules in the system for each frame. We choose to output a frame every one second in the simulation. To compare with experiment, we need to convert filament lengths and molecule numbers into concentrations.

The total lengths of F-actin are first divided by the monomer size (2.7nm) to get total monomers in the system. We then use equation 3.13 (rearranged for concentration) to convert into concentration. We use the same conversion to convert G-actin molecule number into concentration. We can then plot the F-actin concentration and G-actin concentration in time. An example simulated pyrene assay plot is provided in Figure 3.26.

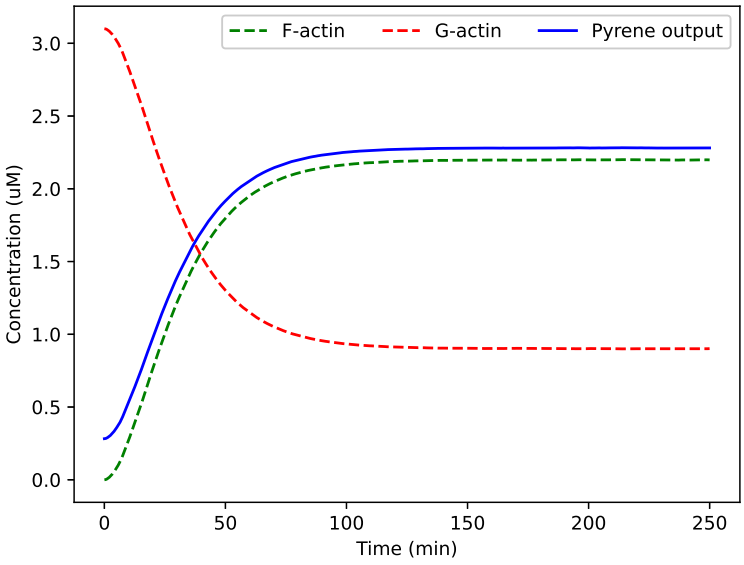


FIGURE 3.26: Example plot of the output from a pyrene style simulation. The concentration (µM) is given along the y axis, and the time in minutes along the x. The green dashed line shows the total F-actin in the system, the red dashed line shows the total G-actin in the system. The solid blue line shows what would be measured in the pyrene experiment (with the constant background removed). This example is with an actin concentration of 3.10µM, exact geometry described in the text.

In Figure 3.26 the simulation is initialised with an actin concentration of 3.10 µM, and polymerisation and nucleation parameters are representative of the 'Actin only' system. The dashed green line shows the F-actin concentration and the dashed red line shows the G-actin concentration. The solid blue curve is replicating what is measured in the pyrene experimental system. The solid blue line is the F-actin concentration plus the G-actin concentration multiplied by a, the the fluorescence ratio which was introduced earlier. The solid blue line is therefore equation 3.18

but with the constant background removed. To compare with experimental data we therefore plot this quantity as our simulated pyrene assay style output.

Simulations were run for 250 minutes. We assumed all polymerisation and depolymerisation is at the barbed end, for the pyrene-style measurement with 'Actin only' and 'Actin and Las17' this has no impact, see section 3.6.3.2 for discussion about this for 'Actin, Las17 and Arp2/3'. The barbed end polymerisation rate was $10\mu M^{-1}\,s^{-1}$ which was the same assumption made in sections 3.6.2.6 and 3.6.2.7.

The barbed end depolymerisation rates were 9.02 s^{-1}, 6.65 s^{-1} and 0.55 s^{-1} for 'Actin only', 'Actin and Las17' and 'Actin, Las17 and Arp2/3' respectively. These rates were chosen to give the critical concentration measured and given in Table 3.3.

Three repeats were carried out for each concentration of actin in each system.

3.6.3.1 'Actin only' and 'Actin and Las17'

The nucleation rates used were $2.17\times10^{-7}\,\mu m^{-2}\,\mu M^{-3}\,s^{-1}$ and $2.70\times10^{-7}\,\mu m^{-2}\,\mu M^{-3}\,s^{-1}$ for for 'Actin only' and 'Actin and Las17' respectively. These values were taken from averages of the cubic nucleation rates found in section 3.6.2.6. The average rates from section 3.6.2.6 were multiplied by 0.2µm to take into account the depth of the nucleation region.

Figure 3.27 shows the simulation results alongside the processed experimental data. The solid lines are the averages across three repeats and the associated shaded regions shows the standard deviation across the repeats. In some cases the shaded regions are difficult to see as they are comparable to the thickness of the solid lines.

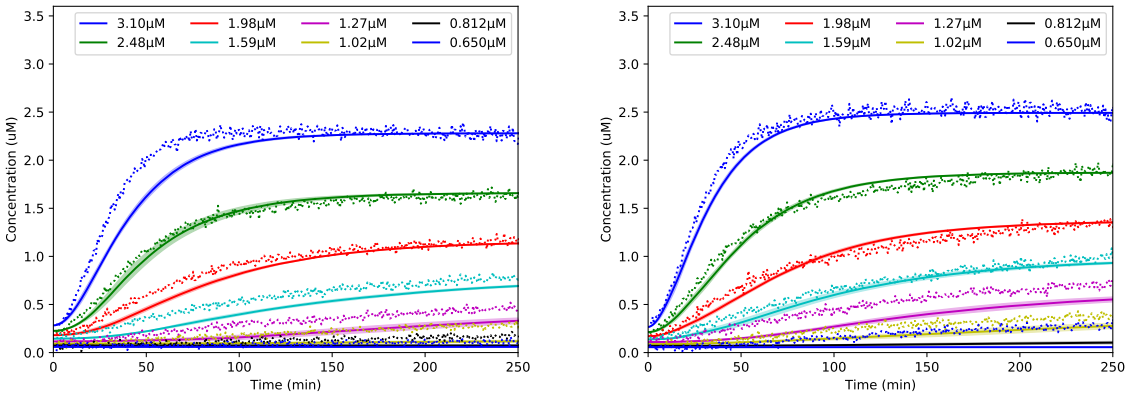


FIGURE 3.27: Results of pyrene-style simulations. The simulated actin concentrations over time are shown as the solid lines (average of three repeats) with associated shaded regions showing the standard deviation. The processed experimental data previously plotted in Figure 3.20 is plotted as points. **(A)** 'Actin only' system. **(B)** 'Actin and Las17' system.

Generally the simulations tend to show lower values of concentration for the same time point in the time before reaching steady state than in the experiments. Another way to think about it is that the simulations take longer to reach steady state than in the experiments. The steady state plateau shows good agreement with the data as expected. Simulated steady state values which seem to not agree (for example 1.27µM actin in the 'Actin and Las17' system) are because the simulation has not reached steady state by the end of the 250th minute. We propose the discrepancy is due to the assumption of no dissociation in the differential equations used to find the nucleation rate. As discussed in section 3.3.1, this assumption will result in an underestimate for the nucleation rate. We could run the simulations with dissociation

disabled to validate this (see 3.8 later). Lower concentrations such as 0.812µM and 0.650µM show poor agreement with the experiments. For example 0.650µM 'Actin and Las17' system does not show any polymerisation, since the critical concentration from Table 3.3 is higher than 0.650µM. Whereas in the experiment polymerisation is seen. This suggest the critical concentrations in both 'Actin only' and 'Actin and Las17' determined from section 3.6.2 may be too high.

3.6.3.2 'Actin, Las17 and Arp2/3'

The assumption made in section 3.6.2.7 was that the nucleation rate measured in 'Actin and Las17' would be the same in 'Actin, Las17 and Arp2/3', this assumption is made again and the nucleation rate of 2.70 $\times 10^{-7}$ $\mu m^{-2}\,\mu M^{-3}\,s^{-1}$ is used for 'Actin, Las17 and Arp2/3'.

For the 'Actin, Las17 and Arp2/3' system we ran two sets of simulations. In both simulations we consider a branch spacing of 13 monomers as covered in chapter 2. We assume no debranching.

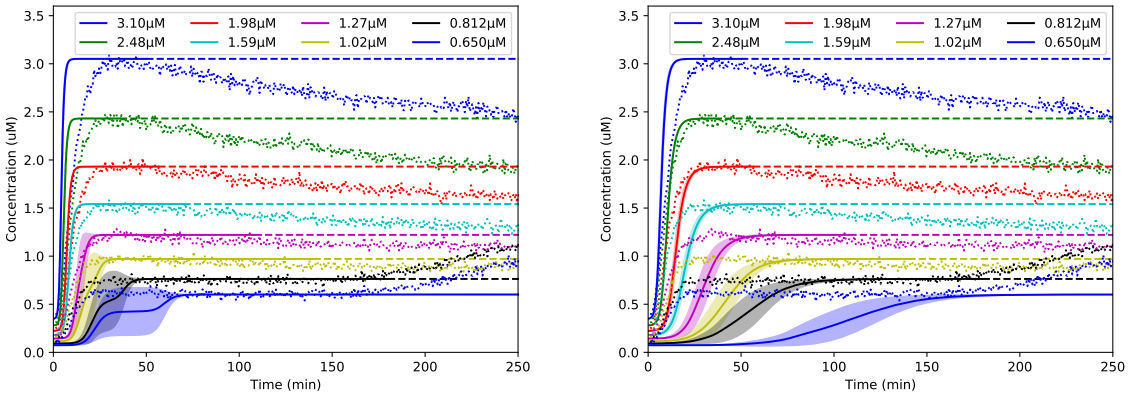
In the first set we use the average branching rate taken from the linear model (the best fitting model of the three evaluated) in section 3.6.2.7. This was 1.14 $\times 10^{-5} N_p^{-1} \mu M^{-1}\,s^{-1}$ (N_p is the number of F-actin molecules), however as was discussed in section 3.6.2.7, this assumes a constant Arp2/3 complex concentration. We then divide this value by the concentration of Arp2/3 complex in the experiments (2nM, resulting in a branching rate of 5.69 $\times 10^{-3} \mu M^{-2}\,s^{-1}$ per branching site. In the differential equations, no branching spacing was assumed and so the number of branching sites was equal to the number of F-actin monomers. We avoid attempting to account for this in this conversion but can investigate the impact of this assumption at a later date (see section 3.8 later on).

We carried out a second set of simulations because the linear model in section 3.6.2.7 was still a poor fit to the data. In the first set we used an average branching rate, but the branching rate had a clear strong dependence on G-actin concentration. With the second set, we used a branching rate of $5.3 \times 10^{-4} \mu M^{-3}\,s^{-1}$ taken from [110], this assumes a quadratic dependence on G-actin concentration. This is the typical model of branching we use in the simulation tool which was covered in chapter 2.

Figure 3.28 shows the simulation results of the two branching models against the experimental data for the 'Actin, Las17 and Arp2/3' system. These simulations took much longer than those presented in Figure 3.27 and so were stopped at predetermined times earlier than 250 minutes. The times were such that the particular simulation had reached steady state. Since the remainder of the simulation would simply be fixed at this steady state, it is unnecessary to continue the simulation. In Figure 3.28 the solid lines are explicitly simulated whereas the dashed lines are not but trace out what the simulation would have produced.

Both branching models show quite a poor agreement with the experiments. Using Carlsson's model (Figure 3.28(B)) gives simulations that are quicker to reach steady state at high actin concentrations (3.10µM, 2.48µM) but slower at lower actin concentrations (1.59µM) and lower. This suggests an issue with scaling, that the quadratic dependence is too high.

Using our branching model with a linear dependence on G-actin (Figure 3.28(A)) is perhaps a better fit to the data. However the scaling issue seen in simulations run with Carlsson's model is still present but to a lesser extent. This is as expected from the differential equation fit results, as a linear dependence was a better fit than quadratic but still showed a strong dependence on G-actin concentration.



(A) (B)

FIGURE 3.28: Results of pyrene-style simulations for the 'Actin, Las17 and Arp2/3' system. Simulated actin concentrations are shown as the solid lines (average of three repeats) with associated shaded regions showing the standard deviation. The processed experimental data previously plotted in Figure 3.20(C) is plotted as points. **(A)** Our linear branching model. **(B)** Carlsson's [110] quadratic branching model.

It should be noted that the curves in Figure 3.28 are averages over three repeats. The curves are not necessarily typical curves for one simulation. This is apparent in Figure 3.28(A) in the lower actin concentrations, where the shape of the growth phase is not typical. There is a large amount of variation at the lower actin concentrations in Figure 3.28, due to the stochastic nature of the simulations. The more filaments there are, the faster the total system grows as there are more ends that are polymerising. Unlike linear nucleation, the amount of branched nucleation at any given time depends on the amount of F-actin in the system, since the branching rate used is a rate per branching site. Therefore branched nucleation leads to positive feedback loops where more branches equal more F-actin and so equal more branches and so on. At these low concentrations, the trajectory a simulation could take has larger variability. A few branches forming early by chance alone could then start the positive feedback loop and ignite significant polymerisation. Another simulation could fail to form branches early, and take much longer to reach steady state. These simulations could be repeated with a larger simulation size to reduce the variation seen between repeats.

More work is required to estimate the errors in the analytical model due to the assumptions made there, which are not made in the simulations. This is discussed in section 3.8 later. In the simulations all polymerisation and depolymerisation happens at the barbed end, while this has no effect on the two linear systems it will have an impact here. By having all (de)polymerisation at the barbed ends the fact that branches are pointed end capped has no effect.

We are unable to account for the reduction of the measured pyrene assay output over time in the experiments. This effect is more apparent in the higher actin concentrations. 'Overshoots' in actin polymerisation have been investigated before and one possible explanation is to do with ATP hydrolysis states [111]. However it is strange that this effect is seen only in the 'Actin, Las17 and Arp2/3' system. This could be an experimental issue we are currently unable to account for. Another possible source could be branch detachment at long times but this idea would need exploring.

Ignoring the reduction, the steady state values in the simulations show good agreement with the experiments.

3.7 Discussion and conclusions

We attempted to further understanding of actin polymerisation and nucleation and the effects of particular actin binding proteins Las17 and Arp2/3 complex. We wanted to understand these systems using mathematical rigor with a quantitative approach to analysis of experimental data. We made use of two complementary experimental methods, analytical theory and simulations to do this.

TIRFM proved somewhat difficult and time consuming to analyse but has huge potential in further study (see next section 3.8). We identified and used semi-automated and automated techniques for measuring polymerisation and nucleation in TIRFM. The elongation rates measured were higher in the 'Actin and Las17' and 'Actin, Las17 and Arp2/3' systems than in the 'Actin only' system, as expected. The elongation rates were not significantly different between 'Actin and Las17' and 'Actin, Las17 and Arp2/3' which is also as expected. The nucleation rates measured were lower in the 'Actin and Las17' than in the 'Actin only' system, which is the opposite of what is expected. However this is probably due to limitations in the automated analysis technique. If necessary, nucleation analysis of TIRFM videos could be done manually.

We identified what was needed to simulate the tethering seen in TIRFM, and this was then measured by Callum Winder and used. We managed to replicate qualitatively the TIRFM system in simulation, which is useful as a validation of measurements from analysis. I will also argue the merits of being able to show others simulation videos that replicate experimental systems. Whether in a talk, at a conference or in outreach, and to people of different backgrounds, demonstrating what is possible in simulation in an easy to understand format such as in video is powerful.

The pyrene system also was not without challenge. We provided one set of experiments here but this was a result of many iterations carried out by Ellen Allwood. To be able to carry out analysis at this quantitative level required new levels of precision in the methodology of the experiments. An example is recording the exact time taken from the start of polymerisation to the first reading. Another example is using a higher pyrene concentration than was previously used in the lab, to increase sensitivity which is important at lower actin concentration.

The analysis pipeline that takes raw data measured in fluorescence intensity and containing a background, and outputs background subtracted data in concentration units is novel. The pipeline we developed is easy to understand, as it is all based on a simple equation for actin polymerisation (equation 3.18). We neglected the consideration of the different ATP hydrolysis states of actin for simplicity. This has previously been proposed to be important in pyrene fluorescence where 'overshoots' are found [111]. Overshoots are not present in the data for 'Actin only' and 'Actin and Las17', but are in higher concentrations in 'Actin, Las17 and Arp2/3'. We are unsure why they appear in these cases. In these curves with an overshoot we simply treat the steady state concentration to be around the maximum point of the curve. If these overshoots are what was seen and proposed in [111], then our determination of steady state concentration would correspond to the steady state of (mostly) ATP-actin, before it hydrolyses to ADP-actin. The pipeline required determination of some properties 'by eye', and this was sometimes difficult. For example, we had to determine which actin concentrations did reach steady state inside the 250 minutes, for example we determined 'Actin only' 1.59µM and 'Actin and Las17' 1.98µM to have reached steady state, but this isn't completely clear. However this was validated by good fits using this data (example Figure 3.18).

The critical concentrations found using our analysis pipeline were higher than expected from the literature [6, 145] in the 'Actin only' and 'Actin and Las17' systems.

However this was drastically different in the 'Actin, Las17 and Arp2/3' system (lower by an order of magnitude), and more in line with what is expected.

In the fitting to differential equations and simulations, we assumed a value for the polymerisation rate k_p, and this was constant in the three systems. This is possibly not the case, since Las17 has been shown to increase elongation which is likely to be increasing polymerisation rate rather than decreasing depolymerisation rate [134, 135]. It would be of interest to repeat the fitting and simulations by assuming a constant depolymerisation rate k_d, to see if any significant differences arise. This is also discussed in the next section on future work (section 3.8).

We manage to get good agreement with fitting of the differential equations to the data of 'Actin only' and 'Actin and Las17'. The absolute value of the nucleation rates measured should be taken lightly, given the assumptions made. However the measured relative increase in nucleation between 'Actin only' and 'Actin and Las17' of $\sim$24% carries more weight, since the same assumptions are present in both measurements.

We were able to find the order of nucleation with G-actin concentration to be cubic for both systems.

Finding the order of branching in the 'Actin, Las17 and Arp2/3' system proved difficult. We found strong dependence on the G-actin concentration for all three models of branching investigated. There are many more assumptions made in this system and we believe that perhaps some of them are not valid. For example, not considering branches to be pointed end capped and neglecting branch spacing. However, the simulations are useful in testing the validity of these assumptions and this is covered in the next section (3.8).

Finally we used simulations to validate our analysis pipeline and differential equation fitting. With the 'Actin only' and 'Actin and Las17' systems we see good agreement with the data but this highlighted a possible underestimation in the nucleation rates measured. We believe this is due to the assumption in the differential equations to neglect dissociation.

The simulations replicating the 'Actin, Las17 and Arp2/3' system highlights the issue with incorrect scaling of the branching rate to G-actin concentration. Both models were unable to agree with the whole dataset. The assumption of only barbed end polymerisation activity may also be having an effect here and is worth exploring. Branched actin systems are much more complicated since the critical concentration for a linear filament would be different than the critical concentration for a pointed end capped filament. For more discussion on this see section 3.8.

The aim of this work was to be able to take existing experimental methods and through novel analysis, mathematical modelling and simulation, understand the actin systems on a deeper quantitative level. Prior to this work, pyrene assays were analysed more qualitatively. For example, drawing conclusions on the polymerisation rates by doing comparisons between gradients of the growth phases. Another example would be assuming that the initial 'slope' in the 'lag phase' is due to nucleation alone. All while still in arbitrary units of fluorescent intensity [135]. We showed that the shape of the pyrene curve up until reaching steady state is dependent on nucleation, polymerisation and depolymerisation. It is impossible, in polymerisation assays alone, to determine any of these independently.

We've provided an analysis pipeline, differential equations and a simulation tool to be able to continue this work and further our understanding of actin systems.

3.8 Future work

This project has opened up many potential avenues for further study.

3.8.1 Repeat experiments

To continue the work carried out here would need repeat experiments of the pyrene assays. Perhaps running the experiments for longer than 250 minutes would be helpful, to ensure more of the concentrations reach steady state. This then provides more data to carry out the analysis pipeline, converting into concentration units and determining the critical concentration of the system. More repeats of the whole pipeline is crucial to validate our results so far.

It would also be interesting to look at different concentrations of Las17 in pyrene assays. In our differential equations we assumed all nucleation in the 'Actin and Las17' system was via the Las17 mediated pathway. It would be useful to test this assumption by increasing the concentration of Las17 to determine if nucleation rate increases. We could determine the concentration at which this assumption becomes valid, whether that is less than the 100nM used here or more.

3.8.2 Further analysis of TIRFM data

The nucleation analysis of the TIRFM data used a automatic technique with possibly large errors that are currently unquantified. This analysis should be repeated with either a more established automatic technique with quantified errors or a purely manual technique. We are reluctant to accept the relative nucleation change between the two linear systems, as this is the opposite of what was found in the pyrene assays and is in disagreement with previous work [134, 135].

Analysis of branching rates could also be done in the TIRFM videos of the 'Actin, Las17 and Arp2/3' system. However this is non-trivial and will probably have to be done manually. Care must be taken to identify what is a branch and what is a linear filament orientated such that it looks like a branch.

3.8.3 Using pyrene style simulations to quantify the validity of assumptions

Simulations could be essential in understanding the 'Actin, Las17 and Arp2/3' system as many complex phenomena are present. We could estimate deficiencies in the analytical model by testing each simplifying assumption one by one.

In the previous section we touched on the differences between the linear systems and the branched system, and there is more work to be done here too. The critical concentration of a linear system is simply

$$C^*_{G,(\mathrm{lin})} = \frac{k_d^{\mathrm{point}} + k_d^{\mathrm{barb}}}{k_p^{\mathrm{point}} + k_p^{\mathrm{barb}}}, \tag{3.24}$$

where now we are considering the polymerisation dynamics at the two different ends (barb for barbed and point for pointed). $C^*_{G,(\mathrm{lin})}$ is the critical concentration of the linear system. Whereas for a branched system we have to take into account the number of branches and linear filaments. The critical concentration becomes

$$C^*_{G,(\mathrm{br})} = \frac{N_{\mathrm{lin}} k_d^{\mathrm{point}} + N_{\mathrm{tot}} k_d^{\mathrm{barb}}}{N_{\mathrm{lin}} k_p^{\mathrm{point}} + N_{\mathrm{tot}} k_p^{\mathrm{barb}}}, \tag{3.25}$$

where N_{tot} is the total number of filaments in the system (branches and linear filaments) and N_{lin} is the total number of linear filaments in the system. This difference is overlooked in our simulations by having all the polymerisation activity at the barbed end. Simulations could be run with some pointed end polymerisation activity, exploring the polymerisation differences between linear filaments and branches. Following this, simulations could be run where branches immediately debranch once nucleated, exploring the assumption made in the differential equations.

Branch spacing is another area worth exploring, a simple investigation could involve running simulations without any branch spacing.

We also want to test the assumption of no dissociation across all three actin systems, as we think this results in an overestimation of the nucleation rate.

3.8.4 Using depolymerisation pyrene assays to determine the depolymerisation rate, k_d, in the pyrene systems

At the end of the pyrene actin polymerisation experiment described in section 3.2.1 Ellen Allwood then begins a depolymerisation pyrene assay. She adds Latrunculin A (LatA) to each sample such that the concentration ratio of LatA to actin is 2:1. For example for 3.10µM actin, Ellen would add LatA to give a concentration of 6.20µM LatA in the sample. The LatA for each actin concentration is diluted according in DMSO, a solvent. The total volume of LatA diluted in DMSO is 7.5µl.

The raw data for just the 'Actin only' system is presented as an example in Figure 3.29. The data for only half the actin concentrations is provided as the other half are used as a control. For the control Ellen Allwood adds DMSO, which is the solvent LatA is dissolved in. Adding 7.5µl of even just pure DMSO promotes some depolymerisation as the volume of the sample is increased and therefore the concentration of G-actin is decreased below steady state. This is why the control is carried out, to account for depolymerisation from this small volume change. The DMSO control curves are omitted from Figure 3.29 for simplicity. The recorded time delay between adding the LatA and starting the fluorimeter read has been added to Figure 3.29.

Figure 3.29 gives the raw data but we could assume that the values of α and B_f are the same as those given in Table 3.3. Therefore converting to concentration and removing the background would be trivial.

Future work is required to use this dataset to determine a depolymerisation rate k_d for each system. Preliminary simulations suggest this is non-trivial, as even though there is a 2× excess of LatA over actin, actin polymerisation still occurs and can not be fully neglected. More work is needed to investigate this and develop a method to extract out k_d. If successful this would eliminate the assumption of k_p which was made and discussed in section 3.7, since determining k_d also determines k_p since the critical concentration (k_d/k_p) is known.

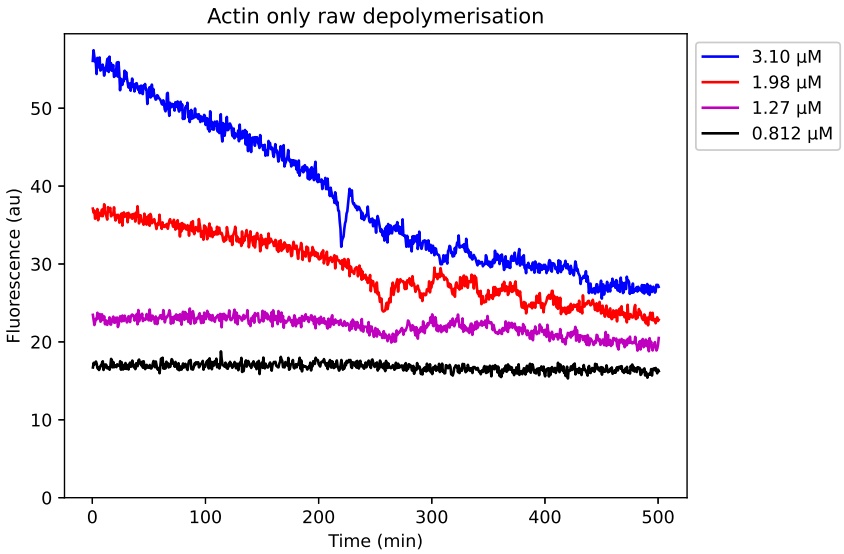


FIGURE 3.29: Raw pyrene depolymerisation assay data, example of 'Actin only' system. The recorded time delay has been added. Fluorescence intensity is shown on the y axis, time since depolymerisation initiated in minutes along the x. Each coloured line represents a different concentration of actin. Blue: 3.10µM, red: 1.98µM, magenta: 1.27µM and black: 0.812µM.

Chapter 4

Investigating Phagocytosis

This chapter covers research into actin force generation in phagocytosis. We first present further development to the simulation tool described in chapter 2 to be able to use it to investigate phagocytosis. We then present our exploration of the target size dependence which relates to some experimental work done in Sheffield. We also present simulation results which look into the effect of changing actin nucleation, branching and polymerisation rates on the wrapping time. Finally we discuss future work, which includes some preliminary results of scenarios worth investigating as well of lots of ideas for further study.

4.1 Introduction

The simulation tool we have developed (see chapter 2) can, with a few adjustments, be used to study actin force generation in phagocytosis. In chapter 1 we highlighted the gap in knowledge surrounding the role of the underlying actin cytoskeleton in phagocytosis. Our research marks the first attempt to simulate phagocytic events from receptor engagement to fusion using a single filament model of actin polymerisation, interacting with a model membrane.

We aimed to explore the relative importance of actin processes, such as nucleation, branching and elongation, by controlling these parameters and measuring how they affect phagocytosis. We also aim to investigate properties of the membrane and target, such as the importance of membrane growth, and target size dependence.

4.2 Adapting the simulation model

In this section we will describe the adaptations to the simulation model which were required for its use in exploring phagocytosis.

4.2.1 Target

The target is what the cell engulfs. In section 2.5 we described the 'exclusion zone' object that we have in our simulation toolbox. We can use a circular exclusion zone as a simple model of a target that the membrane can wrap around.

We model phagocytosis starting at the point of receptor engagement. Therefore we always need to set the geometry of any phagocytosis simulation so that the target sits on the membrane. We also want to model the effect of activated receptors, and we do this through what we call 'activated regions' (defined in section 4.2.1.1).

4.2.1.1 Activated regions

Receptors are proteins that bind to specific extracellular proteins called ligands. Once bound they can initiate a cellular response, for example a biochemical signalling pathway [1]. In phagocytosis, receptors in the membrane engage with ligands on the target, once engaged they become activated - triggering a a complex biochemical response. Many receptor-ligand pairs exist and have been studied in detail [29, 36]. This receptor-ligand binding is the first stage in often complex biochemical signalling pathways that involve many proteins. The end result in these pathways is the stimulation of actin polymerisation. This is done via the activation of proteins such as Arp2/3 complex and formins [28, 29, 36]. We wanted to simplify this complex activation process and concentrate on the mechanics rather than the biochemistry. Therefore our membrane subunits become 'activated' when they move within a threshold distance d_{act} from the surface of the target. This replicates receptor engagement and the threshold distance represents the length of the receptor-ligand complex.

What do we mean for a membrane subunit to be 'activated'? In section 2.3.5 we described the spatial regions we have in our model. 'Activated' membrane subunits have a spatial region created underneath them at the timepoint of activation. This region persists for the remainder of the simulation. As the membrane subunit moves, the associated region(s) move with it. We can choose which combination of spatial regions appear. Typically we have three different regions at once. A nucleation region, modelling the recruitment of G-actin and the activation of nucleation proteins such as formins [29, 36]. A branching region, which models the recruitment and activation of Arp2/3 complex. Arp2/3 complex is activated at the cup and has been shown to be the dominant player in actin polymerisation in phagocytosis [29, 36]. Finally we include an anti-capping region, modelling formins and Ena/Vasp proteins. Ena/Vasp proteins have been shown to be recruited to the cup [44, 45], and these proteins have been suggested to be anti-cappers [41–43]. Formins have also been shown to be effective anti-cappers [42]. We introduced capping in the introduction (section 1.1.3), but only briefly introduced anti-capping. Anti-capping is not uncapping, it is protection from capping. So in our simulations, actin filament ends that are within anti-capping regions cannot cap.

The length of the activated region is equal to the length of the membrane subunit. The width of the activation region defines how far from the membrane, activated proteins such as Arp2/3 complex are localised. Recently, super resolution localisation microscopy was used to identify regions of actin nucleating proteins in yeast undergoing endocytosis [151]. In this work, 90% of the WASP and Myosin-1 (activators of Arp2/3 complex) were found to be localised within 200nm from the membrane. 90% of Arp2/3 complex itself was found within 300nm but the main bulk was inside 150-200nm. We therefore use a width of 200 nm for our branching region. For simplicity we assume the same for all other activated regions such as nucleation and anti-capping.

We also wanted a simple way to model the diffusion of activated proteins laterally underneath the membrane. Receptor-ligand complexes remain fairly immobile on the membrane [34, 152] but the downstream activated proteins are free to diffuse underneath the membrane. To model this we simply have that when any membrane subunit becomes 'activated', the two adjacent neighbour membrane subunits also become activated. We could have added complexity here, for example incorporating a time delay for adjacent subunits to become activated, or continuing the 'diffusion' beyond just the adjacent subunits. We instead took this simple approach but recognise this could be developed in the future. Early simulations showed that without adjacent subunits

becoming 'activated', progression of phagocytosis would stall.

4.2.2 Membrane fusion

The end of phagocytic engulfment occurs when the cell membrane undergoes fusion, creating a phagosome. We are not interested in the later stages of phagocytosis such as phagosome maturation [29]. We therefore wanted to include membrane fusion in the simulation to mark the completion of phagocytosis. Our method for modelling fusion is described in this section.

At each timestep after the membrane has fluctuated (described in section 2.4.2) a check is performed to determine if fusion should happen. This check consists of distance measurements between non-adjacent subunits along the membrane. This is a line segment to line segment distance query carried out using Geometric tools library [85]. If the distance measured is less than the thickness of the membrane (s_{mem}) the two subunits are said to be in contact and the membrane will fuse.

In general, the path drawing out the minimum distance between two line-segments always involves one of the endpoints of one the subunits (unless the segments are colinear or crossing). In our simulations two segments being exactly colinear is extremely unlikely to happen. Crossing of membrane segments could happen, since steric checks within the membrane are not carried out. However, due to small timesteps, the membrane generally will fuse before it has chance to cross over itself. When membrane fusion has been determined to happen, then the three points involved in the distance measurement (two endpoints from one subunit and one from the other subunit) are identified. They are identified by carrying out line segment to point distance queries (again by using Geometric tools). Each of the four possible point-segment pairs are queried and the three points are identified from the point-segment pair that is closest.

Figure 4.1 shows the membrane during a phagocytosis simulation, taken just before and during a fusion event. This helps to illustrate the membrane fusion process which is described here. Figure 4.1A shows the whole membrane just before fusion. Figure 4.1B shows a zoomed in version, highlighting the three fusion points as red circles. Fusion takes place by first deleting these three points. Then, a new membrane object representing the phagosome is created. The phagosome is made up of all the membrane points between the three points. So in Figure 4.1B, these are the blue points below the three red points (most of the points cannot be seen due to the image being cropped). These membrane points are removed from the membrane. Then the three deleted points are replaced with an average point that is added to both the phagosome and the membrane. To be clear this is not the same point, this is two points (one in the membrane, one in the phagosome) with the same coordinate. This is shown in Figure 4.1C which is now showing the timestep of the fusion event, the three red points have replaced with an average point that given to both the membrane (orange) and the phagosome (green). Figure 4.1D shows the overlay of before and during.

Both the new membrane and the phagosome have all their subunit rest lengths (the lengths they should be) set to be equal. The new subunit length is calculated by simply dividing the total length by the number of points. This is to avoid one or two subunits being much larger than the rest, which is actually the case in Figure 4.1C. In subsequent timesteps the Brownian dynamics algorithm ensures the points will relax such that the actual lengths tend towards this new subunit length.

The phagosome's effective persistence length is calculated by the same method described in section 2.4.2. Since the membrane itself has reduced in length it also needs its effective persistence recalculating using equation 2.61 with w recalculated as

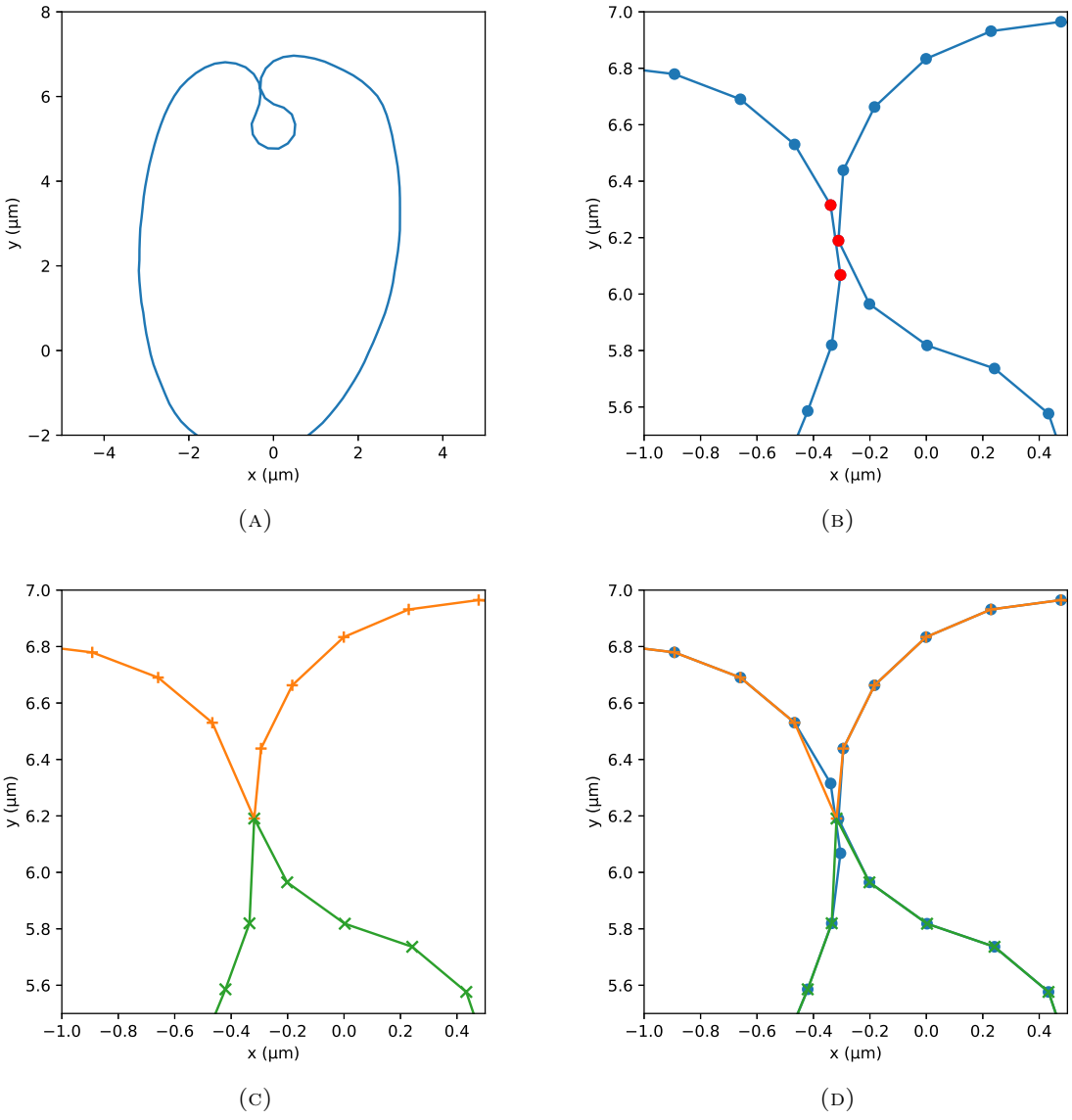


FIGURE 4.1: The fusion process. Taken from a simulation of phagocytosis. **(A)** The timestep immediately before fusion takes place. The whole membrane is shown as a blue line. **(B)** The timestep immediately before fusion takes place. Zoomed in to the fusion area of interest. The membrane is shown as a blue line. Blue points mark the points along the membrane. The three red points show the three points involved in the fusion event. **(C)** The timestep of fusion. The new membrane is shown as orange points connected by straight lines. The phagosome is shown in green. **(D)** Overlay of the timestep immediately before fusion and the timestep of fusion. This is a combination of Figure B and C. Here you can see that the three original points (in blue) are replaced by the one shared point.

L/π (see section 2.4.2). Any activated regions associated around the fusion area are removed.

As a result of fusion some actin filaments might now be breaking steric hindrance. Any filaments that intersect either the membrane or the phagosome, or are completely outside of the membrane or inside the phagosome, are forcibly depolymerised until the steric hindrance violation is no more.

If all possible fusion events have completed (the number of fusion events occurred equals the number of targets in the system) the simulation continues for a further two seconds before finishing. In this two second window, activation of further regions is not allowed.

4.2.3 Membrane growth

In Chapter 2 we explained how our membrane model can be either inextensible like our actin filaments, or allow for some extensibility by connecting the points with springs. In this section we discuss a process that is related but importantly different, the membrane growing. In the cell, this can be done by the process of exocytosis, and this is the model of membrane growth we focus on here. In exocytosis small intracellular membrane vesicles move to and fuse with the membrane, therefore acting as a source of extra membrane material. An exocytosis event therefore increases the surface area of the membrane [1, 153]. Previous studies have shown this to be important in phagocytosis; facilitating engulfment by providing extra membrane for wrapping and relieving membrane tension [120, 153, 154].

We wanted to investigate exocytosis in phagocytosis with our model. We incorporated a simple method that uses the same principles of those described in section 2.3.4. We use an exocytosis rate k_{exo} and, using the Monte Carlo method previously described, determine each timestep whether an exocytosis event will happen. If an event is to happen a membrane subunit is first chosen randomly from a uniform distribution. Then a length increment is chosen,

$$\Delta l_i \sim N\left(\mu_{\text{exo}}, \sigma^2_{\text{exo}}\right), \tag{4.1}$$

where i is the randomly chosen membrane subunit, μ_{exo} is the mean of exocytosis lengths and σ_{exo} is the standard deviation of exocytosis lengths.

The length increment Δl_i is added to the rest length of subunit i, l_i. The total rest length L therefore also increases by Δl_i. The effective persistence length is recalculated using equation 2.61, just like after a fusion event. No points in the membrane are explicitly moved. We simply let the Brownian dynamics algorithm in subsequent time steps take care of that. The membrane subunit will grow to its rest length as either the constraints or the springs are relaxed to the new, longer, rest length.

If at any point a subunit grows to a length greater than or equal to twice the original subunit length, it splits into two. A point is simply added halfway along the subunit, creating two subunits of equal length.

4.3 Successful phagocytosis: an example

To demonstrate our model of phagocytosis we present here a typical example. A simulation was run with a membrane of length 37.75 µm, made up of 151 subunits. This would give a membrane with a radius of just above 6µm, which is typical for a neutrophil [155]. In the simulation the membrane has a bending modulus of $20k_BT$

[69]. The membrane's length is inextensible and remained constant throughout the simulation

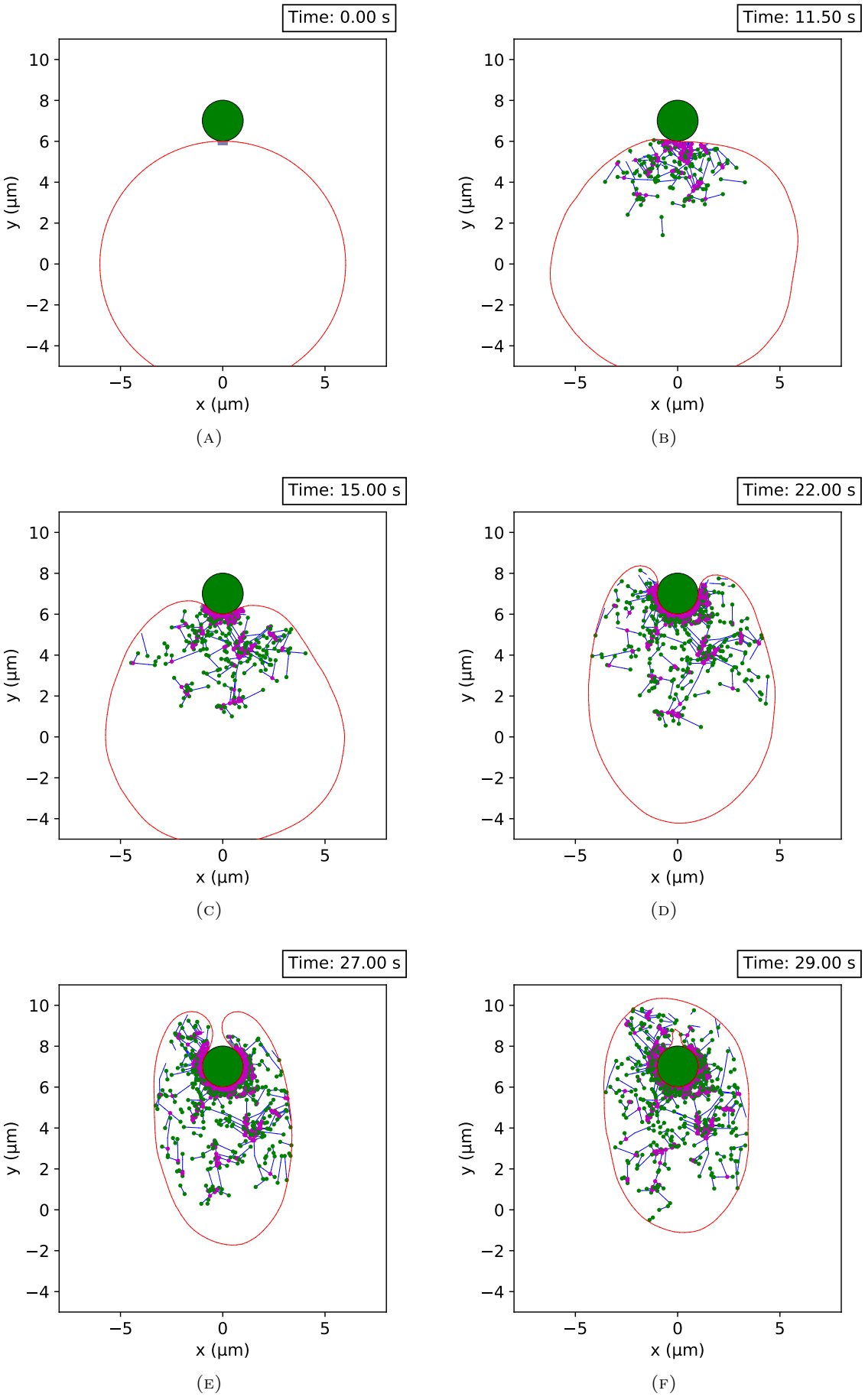
The target is 1µm in radius and is fixed in space, sat on top of the initial membrane. The activation distance, d_{act}, is 12nm, which is the measured length of a receptor-ligand complex [156]. Activated regions are branching, nucleation and anti-capping.

The G-actin concentration C_{G} is 10µM, which is an order of magnitude estimate for in the cell. This considers that most of the G-actin in the cell is in a non-polymerisable pool [157]. We therefore assume G-actin is fixed at this concentration. The nucleation rate density k_{nuc} is 0.02 $\mu\text{m}^{-2}\,\mu\text{M}^{-3}\,\text{s}^{-1}$, an estimate taken from [68][1]. The Arp2/3 complex concentration is fixed at 1µM, an estimate [125]. Barbed end and pointed end capping is turned on with a capping protein concentration, C_{cap} of 0.2 µM, the lower end of the range given in Table 2.2. Uncapping, crosslinking, debranching and severing are all turned off for simplicity. All other parameter values used are taken from Table 2.2. This simulation took ∼17 hours of CPU time to run. Snapshots of the simulation are presented in Figures 4.2 and 4.3.

In this example there are no actin filaments initially (Figures 4.2, 4.3 A). Filaments begin to nucleate in the active regions. In the first few seconds the filaments are unable to push the membrane outwards and form a cup. The filaments are short and move around quickly due to Brownian motion. As more filaments nucleate and grow they experience more drag from the fluid and start to slow down. They also experience steric hindrance from neighbouring filaments which further hinders their motion. Eventually, a base actin network is formed (in this example this takes ∼11 seconds, Figures 4.2, 4.3 B). New filaments can then push the membrane, pushing off this base. By fifteen seconds a phagocytic cup has formed, hugging the bottom edge of the target (Figures 4.2, 4.3 C). This cup is then extended outwards by further filaments and branches, pushing it round the target, progressively (Figures 4.2, 4.3 D). The membrane then makes it around the target (Figures 4.2, 4.3 E), where it will fuse. Fusion occurs at ∼28 seconds. Once fusion has happened we end up with a phagosome surrounding the target and the cell membrane, which has relaxed to circular in shape (Figures 4.2, 4.3 F). As is seen in Figures 4.2, 4.3 F, the phagosome is a funny shape and has some additional membrane on the top that forms what we call a 'hat'. This is because the point of fusion usually is not near the surface of the target, but rather a small amount of distance higher up. This results in the circumference of the phagosome being slightly longer than the circumference of the target. Since the actin which has 'pinned' the membrane to the surface of the target still remains, the phagosome cannot relax. In cells undergoing phagocytosis the underlying actin cortex is thought to undergo rapid depolymerisation in the later stages of engulfment, to make way for internalisation of the target [48, 158]. If we incorporated this into the model we would expect to see the phagosome relax to be circular in shape.

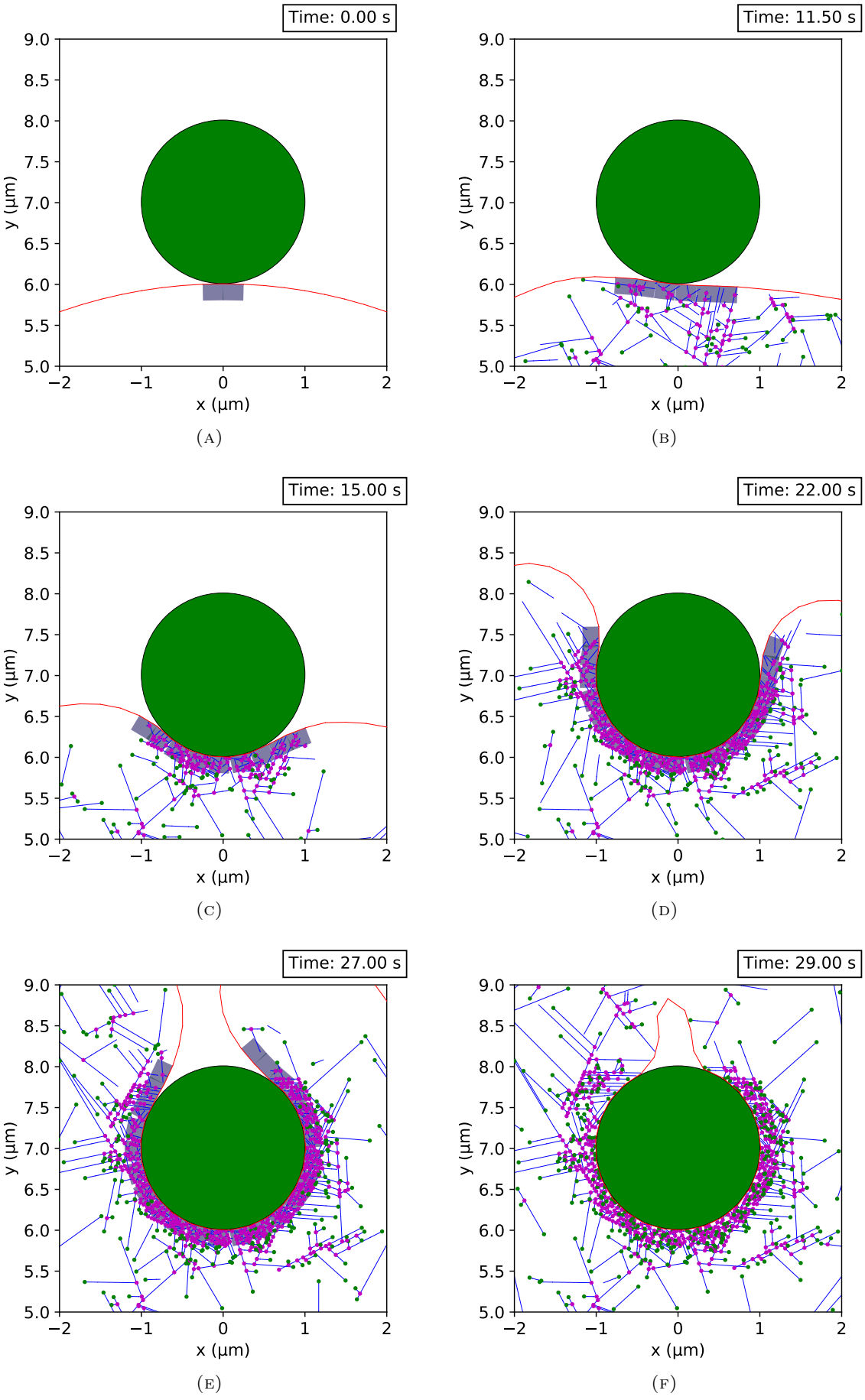
Figure 4.4 shows the progression of engulfment in time for our example. The engulfment at any given time is the length of the cell membrane in contact with the target divided by the target's circumference. Engulfment is calculated on a membrane subunit level. Contact is defined as the distance between the surface of a membrane subunit and the surface of the target being less than 1nm. In Figure 4.4 the time delay of ∼11 seconds can be seen, as there is no engulfment in this early phase. Then progression of engulfment is linear until fusion happens (marked by the vertical orange line). Engulfment does not progress to a value of one, as fusion happens before the top of the target is wrapped (see Figure 4.3 F).

[1]The nucleation rate density used in [68] was $20\mu\text{m}^{-2}\,\text{s}^{-1}$, the factor of 1000 difference comes from multiplying by C_{G}^3.



(A) (B) (C) (D) (E) (F)

FIGURE 4.2: Snapshots of a simulation showing phagocytosis. Time is shown in the top right: **(A)** the initial frame at 0 seconds, **(B)** 11.5 seconds, **(C)** 15 seconds, **(D)** 22 seconds, **(E)** 27 seconds **(F)** 29 seconds. Colour scheme for actin filaments, accessory proteins and exclusion zone (target) the same as in chapter 2. Membrane is shown as a thin red line. Like with actin filaments, the thickness of the membrane is not to scale. This simulation took ∼17 hours of CPU time to run.



(A) (B) (C) (D) (E) (F)

FIGURE 4.3: Snapshots of a simulation showing phagocytosis. Zoomed in. Time is shown in the top right: **(A)** the initial frame at 0 seconds, **(B)** 11.5 seconds, **(C)** 15 seconds, **(D)** 22 seconds, **(E)** 27 seconds **(F)** 29 seconds. This simulation took ∼17 hours of CPU time to run.

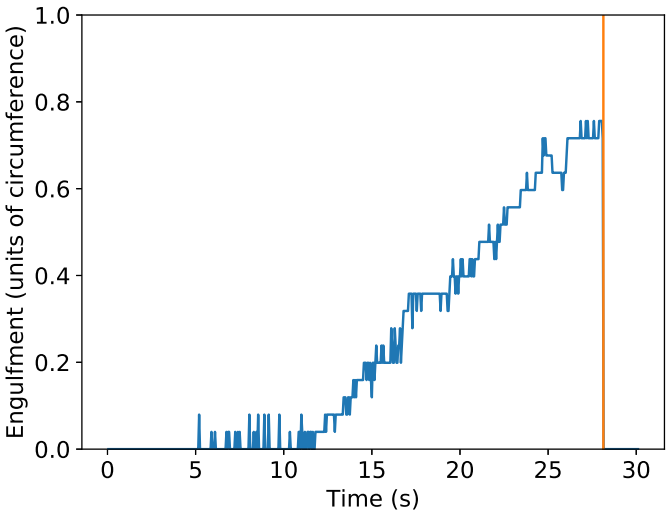


FIGURE 4.4: Progression of engulfment in time for our simple example phagocytosis simulation shown in Figures 4.2 and 4.3. Engulfment, along the y axis, is measured as the length of cell membrane in contact with the target divided by the target's circumference. Time along the x axis in seconds. The blue line shows the progression of engulfment whereas the orange vertical line marks the timepoint at which fusion happens.

4.4 Investigating size dependence

A simple parameter to investigate is the size of the target. In particular we wanted to find the relationship between time taken to reach fusion and the radius of the target. For this we collaborated with Jaime Cañedo, a PhD student in the lab of Dr. Simon Johnston in the Department of Infection, Immunity and Cardiovascular disease at Sheffield. Jaime is interested in the biophysics of phagocytosis and investigated target size experimentally. For this, he incubated J774 murine macrophages together with polystyrene beads, and measured uptake over thirty minutes. Jaime's data is presented in Figure 4.5.

Figure 4.5 shows that in the experiments with the 0.25µm beads and the 1µm beads there was less uptake than in the ones with 0.5µm beads. This data therefore suggests the relationship between target size and 'ease of phagocytosis' is non-monotonic. There instead exists an optimum size of the target and in this case it is 0.5µm radius.

An optimum size has been suggested to exist before in modelling work [69]. To investigate this we ran simulations with the same target radii as those in the experiments: 0.25µm, 0.5µm, 1µm, 1.5µm and 3µm. J774 macrophages are 7.5-9µm in radius [159]. We replicated the macrophage with a membrane length of 52µm (a radius of ∼8.3µm). All other parameters were the same as the example in section 4.3. We enforced a cut off time of sixty seconds.

It is important to highlight the key differences between our simulations and the experiment described above. In the experiment what is measured is the amount of successful phagocytosis in a given time. This essentially tells us something about the probability of success of phagocytosis at different target sizes. The data suggests that this probability is maximum at some optimum target size. In the simulations we measure the time taken to wrap targets, or the speed of phagocytosis. We measure this rather than probability or number of successful phagocytosis events for various

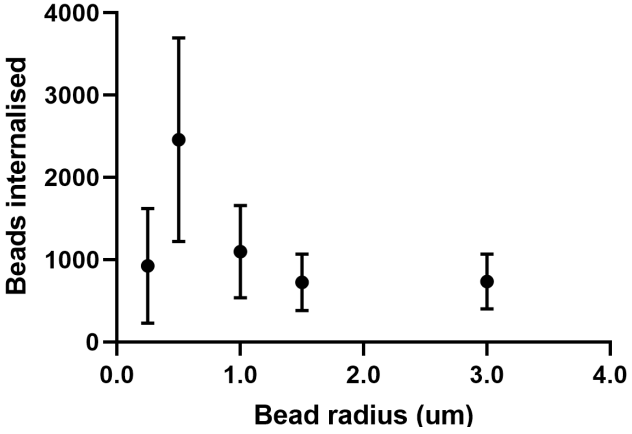


FIGURE 4.5: Uptake of polystyrene beads by J774 murine macrophages in a time window of thirty minutes. The bead radius (µm) is along the x axis and the number of beads internalised by the macrophages is along y axis. The number of repeats for each data point (in order of increasing target size) are 6, 12, 6, 6 and 3. The error bars show the standard deviation across the repeats. Data and Figure courtesy of Jaime Cañedo.

reasons. We are incapable in the simulations to truly replicate the experimental system where many macrophages are in a medium with many targets. In the experiment many more processes than just membrane wrapping are happening, for example the macrophages have to probe their environment to find the beads before receptor engagement. Whereas in our simulations we look at a particular stage of phagocytosis; membrane wrapping of the target. Measuring the time taken is an intuitive and simple measure of 'ease of phagocytosis' in the simulations. Measuring the amount of successful phagocytosis in a given time window is an intuitive and simple measure of 'ease of phagocytosis' in the experimental system. In the experiments it would be difficult to measure time taken between receptor engagement and engulfment/fusion in phagocytosis events due to the necessary time and simultaneously image resolutions required. Therefore we have to be careful when comparing the simulations to the experiments, they are different but to a limited extent they measure the same phenomena; the ease of phagocytosis.

The results of our simulations are presented in Figure 4.6. Each data point is one simulation, we would like to run more repeats in the future. The simulation with the 3µm target did not reach the cut off time or complete fusion. We expect this simulation to either show stalling of phagocytosis, as the target is too large, or to eventually complete engulfment but take much longer than the cut off time of one minute. This would need to be investigated in the near future.

The simulation with the quickest time to achieve fusion was the one with a target radius of 1µm, taking just 27.5 seconds. In the experiment, the optimum bead radius was 0.5µm, whereas in the simulations this took longer to wrap then either the 0.25µm and 1 µm radii targets. The short time to wrap in the 0.25µm radius case is perhaps slightly surprising, but it is important to stress that this is a single simulation. More repeats are necessary to investigate this. We would expect there to be an optimum as small targets are highly curved and so it would require more force generated by the actin in order to bend the membrane around the curved surface. However large objects simply have a larger circumference and it should take longer for the membrane

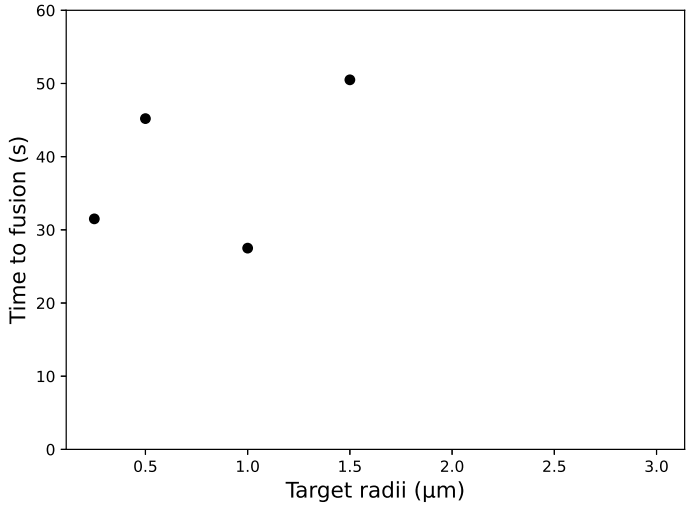


FIGURE 4.6: Target radius along the x axis (µm) against time taken to wrap and fuse along the y axis (seconds). Each data point is taken from a single simulation. CPU times for the simulations are ∼42 hours for the 0.25µm target, ∼65 hours for the 0.5µm target, ∼50 hours for the 1µm target and ∼141 hours for the 1.5µm target.

to be pushed around the entire target. This would therefore be an interplay between membrane tension (which has to be overcome to push around large objects) and membrane bending rigidity (which has to be overcome to push around highly curved objects). This will be discussed later in section 4.7, as an avenue for future study.

4.5 Investigating actin related parameters

What makes our model unique as opposed to previous models of phagocytosis [34, 71, 160] is the inclusion of actin filaments. We have the ability to investigate the effects of actin polymerisation processes on the engulfment of targets.

In this section all simulations were done with a membrane length of 37.75µm and parameter values were the same as in section 4.3 unless otherwise stated. We investigated the dependence of time taken to fuse on the rates of nucleation, branching and barbed end polymerisation. For each parameter set three repeats were run. A cutoff time of one hundred seconds was set.

4.5.1 Nucleation

We first investigated the dependence of the time taken to fuse on nucleation rate density. These simulations always start with a single actin filament as we wanted to look at the extreme condition of a zero nucleation rate. We ran simulations with nucleation rate densities of $0\mu m^{-2}\,\mu M^{-3}\,s^{-1}$, $10^{-4}\mu m^{-2}\,\mu M^{-3}\,s^{-1}$, $10^{-3}\mu m^{-2}\,\mu M^{-3}\,s^{-1}$, $10^{-2}\mu m^{-2}\,\mu M^{-3}\,s^{-1}$, $10^{-1}\mu m^{-2}\,\mu M^{-3}\,s^{-1}$ and $1\mu m^{-2}\,\mu M^{-3}\,s^{-1}$. The results are shown in Figure 4.7, points are the averages and the error bars are the standard deviations across the three repeats.

In Figure 4.7 no data points are plotted for simulations with nucleation rate densities of $0\mu m^{-2}\,\mu M^{-3}\,s^{-1}$ and $10^{-4}\mu m^{-2}\,\mu M^{-3}\,s^{-1}$ because none of the three repeats for each underwent fusion before the 100 second cutoff.

In these simulations we see a dependence of time to fusion on the nucleation rate density, with a monotonic decrease with increasing nucleation rate density.

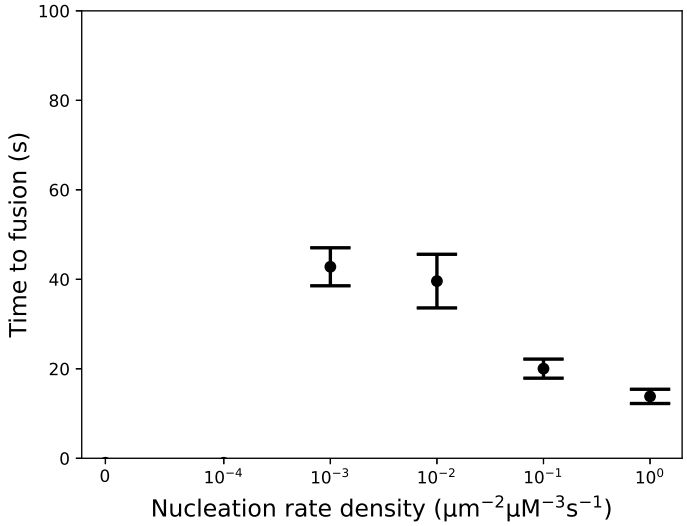


FIGURE 4.7: Nucleation rate density along the x axis ($\mu m^{-2}\,\mu M^{-3}\,s^{-1}$) against time taken to wrap the target and fuse along the y axis (seconds). Each data point is an average of three repeats. Error bars show standard deviations. Data points for $0\mu m^{-2}\,\mu M^{-3}\,s^{-1}$ and $10^{-4}\mu m^{-2}\,\mu M^{-3}\,s^{-1}$ nucleation rate densities are not plotted since no repeat for either managed to achieve fusion before the 100 second cutoff.

4.5.2 Branching

To investigate the effect of branching rate on phagocytosis we ran simulations altering the Arp2/3 complex concentration. We ran simulations with concentrations of 0µM, 10^{-2}µM, 10^{-1}µM, 1µM and 10µM. The results are shown in Figure 4.8, points are the averages and the error bars are the standard deviations across the three repeats. Only two repeats were ran for the 10^{-1}µM Arp2/3 complex concentration value.

In Figure 4.8 no data points are plotted for simulations with Arp2/3 complex concentrations of 0µM and 10^{-2}µM because none of the three repeats for either underwent fusion before the 100 second cutoff.

In these simulations we do see a strong dependence of time to fusion on the Arp2/3 complex concentration and therefore the branching rate. There is a larger difference between the jump in Arp2/3 complex concentration from 0.1µM to 1µM than from 1µM to 10µM. We believe this is because the actin filaments already have many branches and are nearly saturated with branches at 1µM, increasing beyond this is 'diminishing returns'. We would be interested in running more simulations at a higher Arp2/3 complex concentration to verify this. Our model suggests branching is necessary for phagocytosis to occur, since simulations without branching do not show any cup formation or membrane extension around the target. Linear filaments instead simply align parallel with the membrane.

4.5.3 Barbed end polymerisation

To investigate the effect of filament growth on phagocytosis we ran simulations altering the barbed end polymerisation rate. We ran simulations with barbed end polymerisation rates of $1\mu M^{-1}\,s^{-1}$, $5\mu M^{-1}\,s^{-1}$, $10\mu M^{-1}\,s^{-1}$, $20\mu M^{-1}\,s^{-1}$ and $30\mu M^{-1}\,s^{-1}$. The results are shown in Figure 4.9, points are the averages and the error bars are

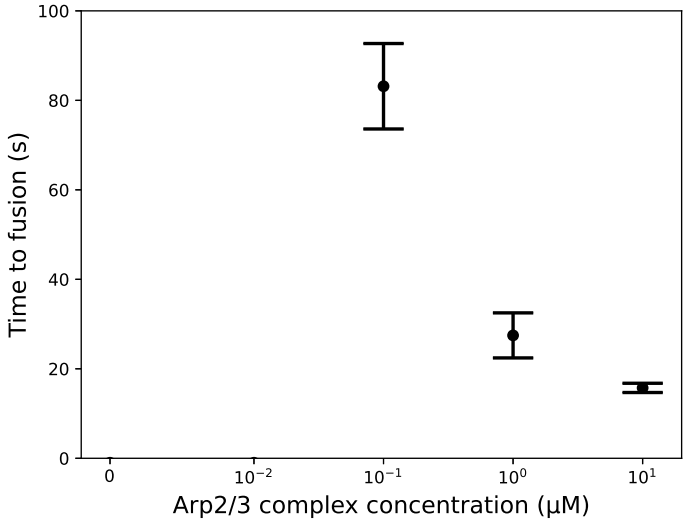


FIGURE 4.8: Arp2/3 complex concentration along the x axis (μM) against time to taken to wrap the target and fuse along the y axis (seconds). Each data point is an average of three repeats, apart for the 0.1μM Arp2/3 complex concentration, which is an average of two repeats. Error bars show standard deviations. Data points for 0μM and 10^{-2}μM Arp2/3 complex concentration are not plotted since no repeat for either managed to achieve fusion before the 100 second cutoff.

the standard deviations across the three repeats. Only two repeats were ran for the $5\mu M^{-1}\,s^{-1}$ polymerisation rate value.

In Figure 4.9 no data point is plotted for simulations with the barbed end polymerisation rate of $1\mu M^{-1}\,s^{-1}$ because none of the three repeats for this underwent fusion before the 100 second cutoff.

In these simulations we see a dependence of time to fusion on the barbed end polymerisation rate. This dependence is most dramatic between $5\mu M^{-1}\,s^{-1}$ and $10\mu M^{-1}\,s^{-1}$.

4.6 Discussions and conclusions

We have built a tool capable of investigating the role of actin in phagocytosis. We can explore the processes of actin polymerisation such as branching, elongation and nucleation in a phagocytosis.

We further developed the simulation tool presented in chapter 2, providing a defined start point (activation of regions) and an endpoint (membrane fusion) to phagocytosis. We also added in the functionality of a growing membrane to explore the importance of exocytosis in wrapping large targets. We have added more functionality some of which we have yet to explore, this will be covered in section 4.7 on future work.

We first wanted to explore target size dependence in our model. In the experimental data from Jaime Cañedo there exists an optimum target size for 'ease of phagocytosis'. We ran simulations and saw a minimum time to fusion at 1μm target radii. We gave a possible explanation of curvature vs size for this optimum and this can be tested (see section 4.7). We cannot draw any conclusions from this data with only four simulations (since each data point in Figure 4.6 is a single simulation). More repeats of these simulations will be necessary to draw conclusions.

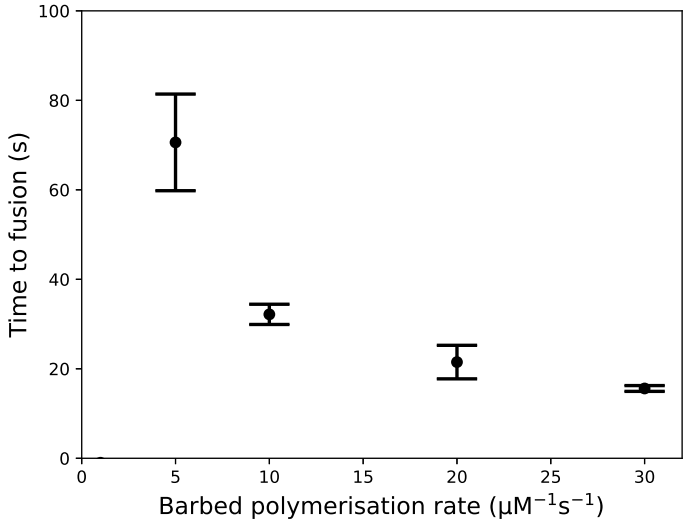


FIGURE 4.9: Barbed end polymerisation rate along the x axis ($\mu M^{-1}\,s^{-1}$) against time taken to wrap the target and fuse along the y axis (seconds). Each data point is an average of three repeats, apart for the $5\mu M^{-1}\,s^{-1}$ polymerisation rate, which is an average of two repeats.. Error bars show standard deviations. The data point for $1\mu M^{-1}\,s^{-1}$ polymerisation rate is not plotted since no repeat for this managed to achieve fusion before the 100 second cutoff.

We then presented simulations investigating the dependence of time to fusion on the rates of three different actin processes; nucleation, branching and barbed end polymerisation. In all three we see a decline in time to fusion with an increase in the rate, suggesting all three help to move the membrane around the target as expected.

Nucleation of linear filaments is important, as a zero nucleation or low nucleation rate simply does not build up a base network for further filaments to push off. Filaments diffuse away from the cup region and cap quickly. Figure 4.7 shows a nucleation rate density of $10^{-3}\mu m^{-2}\,\mu M^{-3}\,s^{-1}$ is enough (along with the other parameter values) to successfully wrap the target. Increasing the nucleation rate density by three orders of magnitude only reduced the time taken to fusion by a factor of around three.

Without branching, phagocytosis was never seen to happen as linear filaments would simply orientate themselves parallel to the membrane and grow along the cell's circumference. Branching is necessary to push the membrane outwards around the target. Figure 2.11 shows a strong dependence on branching with perhaps in between 0.1μM and 1μM Arp2/3 complex concentration typically being necessary to successfully wrap the target in under one hundred seconds. An increase of another order of magnitude from 1μM did still see a further reduction in fusion time. There must be a saturation point where all the sites on F-actin in the regions are branched almost immediately after becoming available. Further investigation would be running simulations at Arp2/3 complex concentrations of above 10μM.

Increasing barbed end polymerisation also had a beneficial impact on phagocytosis. This is hardly surprising since faster growing filaments can exert more force over a given time.

It is worth noting that by increasing barbed end polymerisation and nucleation rate we also indirectly increase the branching rate, since the branching rate is dependent on the amount of F-actin in the system.

We have only begun using this tool we developed to look at phagocytosis from

the perspective of single filament actin processes. This work opens many avenues for further investigation. In the next section we will share ideas and preliminary work into some of those ideas that need expanding.

4.7 Future work

4.7.1 Exploration of target size dependence

We only began to use the simulations to explore the target size dependence on wrapping time. Certainly more repeats of the simulations described in section 4.4 are necessary.

We could also explore this idea of the competition between membrane tension and membrane bending rigidity, by changing the properties of the model membrane. We could simply decrease the bending rigidity of the membrane to see if this lowers the optimum target size as it will become easier for the membrane to bend around the higher curvature. We could also modulate tension by using the extensible membrane model and changing the value of the spring constant, k_{spr} in equation 2.74. By decreasing the tension we would expect to see the optimum target size increase.

4.7.2 Investigating exocytosis

Our model suggests that exocytosis is not necessary for successful phagocytosis, as many simulations reach fusion in tens of seconds. However, with larger targets with radii comparable to the radii of the phagocyte, the cup progression can stall. The inextensible membrane simply is not long enough to wrap the large target. Therefore we do want to explore whether exocytosis helps the phagocytosis of large targets. Comparisons could be made between simulations without exocytosis and simulations with. We expect exocytosis to make a difference with large targets, enabling them to be engulfed. Exocytosis may also help decrease the time taken to phagocytose a target.

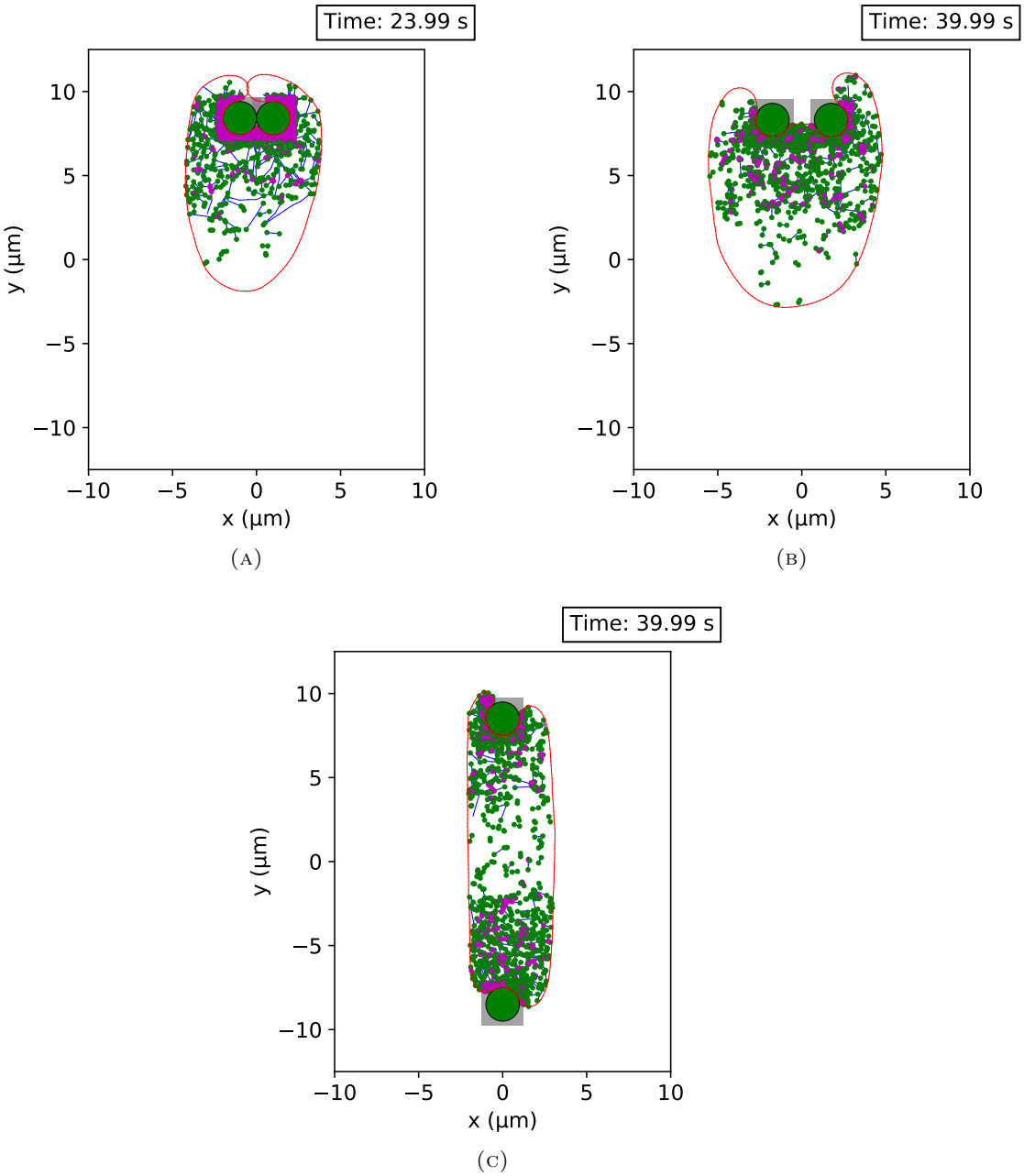
4.7.3 Two targets

We are not just limited to looking at a single target but can investigate the engulfment of multiple targets at once. We were inspired by (currently unpublished) work done in the group of Prof. Holgar Kress at the University of Bayreuth, Germany. Holgar is interested in macrophage phagocytosis of two targets *in vitro*. By placing beads on the surface of macrophages at different distances from each other, his group explores whether there is a single phagocytic cup taking up both beads, or two phagocytic cups.

We want to investigate this in our model by replicating the simple experiment. We have all the ingredients necessary but only did not have the time in this research project. We did run some preliminary simulations with an early version of the simulation model. This version lacked activated regions, and so regions were put in manually around the targets. This also lacked membrane fusion and exocytosis. The membrane had a length of 47.1µm and the targets were 1µm in radius. Arp2/3 complex concentration was 0.5µM. All other parameters were the same as in our simple example from section 4.3.

Snapshots from these early simulations are provided in Figure 4.10 to demonstrate this is possible. In these preliminary simulations we explored three scenarios. The first scenario is where the two targets where so close to each other their surfaces were in contact (Figure 4.10A). The opposite scenario was where the two targets were placed

on opposite sides of the membrane (Figure 4.10C). Finally there is an intermediate scenario (Figure 4.10B).



(A) (B) (C)

FIGURE 4.10: Preliminary simulations into two target phagocytosis using an early version of our simulation tool. The regions are fixed in space. **(A)** Targets are in contact with each other. **(B)** The two targets are at an intermediate distance from each other. **(C)** The two targets are at a maximum distance from each other.

Figure 4.10 shows the timepoint of 'fusion' (the membrane meeting) in the case of Figure 4.10A or the end timepoint, which was a cutoff time of 40 seconds. Figure 4.10C suggests a need for exocytosis, since the membrane cannot engulf both targets. Figure 4.10B could suggest two phagocytic cups at this intermediate distance but will need to be rerun with our more complete model with activated regions.

4.7.4 Actin cortex

The actin cortex is important when considering the mechanical properties of the cell. The mechanical properties, such as the tension, of the cell are determined by contributions from both the membrane and the underlying cortex and it is often difficult to disentangle them [161]. In the membrane-cortex system, it is the properties of the cortex that dominates over the membrane [161, 162]. Our membrane model could in fact represent the membrane-cortex system, with the extra polymerising actin filaments underneath pushing it around the target. However, this is not thought to be what happens. Rather the filaments grow and push off the underlying cortex to form the cup. In the later stages of engulfment, the underlying cortex then undergoes rapid depolymerisation to make way for internalisation of the target [48, 158]. In our simulations we showed no such initial network is required, instead there is a time delay for the simulation to build up a local network to push off. However if we wanted to look at this initial stage of pushing off a pre-existing cortex we added the option to include a separate cortex, underneath our actin filaments. We could simply make a cortex out of filaments, but this would be computationally taxing and simulations would be too slow to run. We can get the same effect by representing the cortex as a single object, like our membrane.

We treat the cortex as inextensible, which is admittedly very crude considering the cortex in cells is active contractile [25]. However as a simple initial model requiring minimal adaptation from the membrane model we have, we treat the cortex as passive and inextensible. We use the same method that was described in section 2.4.2 for our membrane. This means our model of the cortex is a loop of points and has an effective persistence length.

In cells, the actin cortex is tethered to the membrane [25, 163]. We model tethers as springs with a non-zero rest length, connecting our cortex object to our membrane object. The rest length of these tethers is 20nm, which is taken from measurements of the membrane-cortex distance using super resolution microscopy [163].

The development of this cortex model is complete and working within our simulation tool. A single snapshot of a simulation with our cortex model is provided in Figure 4.11. The cortex is shown as a thin blue band underneath the membrane.

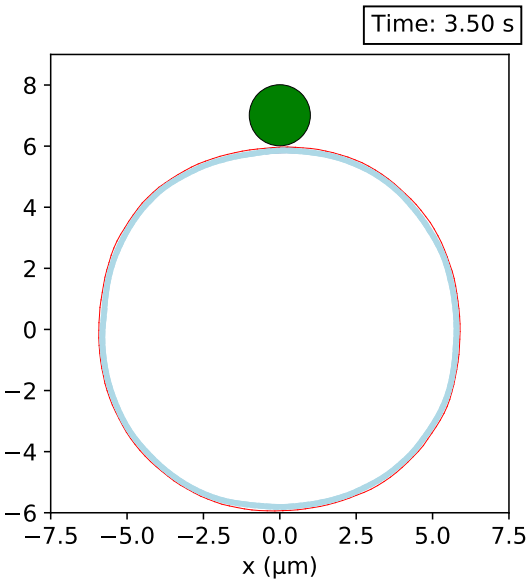


FIGURE 4.11: A snapshot from a simulation with a membrane (red) and cortex (light blue) that is tethered to it.

4.7.5 Target motion

In this research project we looked at targets that were stationary in space. In the initial frame the target is placed so that it sits on our cell, it remains there for the rest of the simulation. We wanted to investigate the effects of target motion on phagocytosis. By modelling the Brownian motion of the target does this prevent phagocytosis by the actin generated force simply pushing the target out of the way rather than wrapping it?

We can have our target undergoing Brownian motion. To include Brownian motion of the spherical target we consider the Stokes-Einstein equation

$$D_{\text{sph}} = \frac{k_{\text{B}}T}{6\pi\eta r_{\text{sph}}}. \tag{4.2}$$

The Stokes-Einstein equation provides the diffusion constant (D_{sph}) of a perfect sphere of radius r_{sph} in a fluid of viscosity η.

The target is moved with a method similar to how short rigid filaments are moved (see section 2.3.2.1). Each timestep small movement increments are chosen

$$dx, dy \sim N\left(0, 2D_{\text{sph}}\Delta t\right). \tag{4.3}$$

The target's coordinates are updated by adding dx and dy to the x and y coordinate respectively. Steric hindrance is then checked, against the membrane and any actin filaments. If steric hindrance is violated the target is moved back.

Preliminary simulations suggest that for small targets (roughly an order of magnitude smaller radius than the phagocyte) the wrapping of the membrane around the target that is seen in Figures 4.2 and 4.3, is not seen. The target is simply pushed away. More investigation is required with larger targets as we expect phagocytosis will still happen. We could extend our model to investigate the importance of adhesion when the target is moving. We could model adhesion of the target to the membrane quite easily using springs connecting the target to the membrane.

4.7.6 Other areas of study

We could easily repeat the type of work done in section 4.5 but with other parameters of actin. We could change the capping rate/capping protein concentration. We could also include severing or branch detachment. However it is not obvious what would be learnt from just investigating dependence on these rates, we need to think carefully before adding more ingredients with more parameters.

We could set up the simulation to include the promotion of depolymerisation and severing at the base. This would replicate what is thought to happen in the *in vivo* case, which allows for the recycling of actin monomers and related proteins, and clearance of the base ready for the target to be internalised. We could couple this with having activated regions that 'turn off' and only propagate at the tips of the cup, either side of the target. Future work could investigate the effect of our alternative extensible membrane model described in section 2.4.2. There is a lot of potential of interesting study here with what we have developed and is already in the simulation tool.

Going further than what we can already do with the tool we have created, we could look at different target shapes, such as rods. We could have a deformable target similar to our membrane.

4.7.7 Modelling phagocytosis in three dimensions

Perhaps the greatest limitation of our model is the fact we are in two dimensions. While we argue that the simplicity of the 2D world can help us understand the 3D world, phagocytosis is a 3D process. We have made some assumptions, mainly in our membrane model, to take into account the fact we are in 2D, but it would be very exciting to repeat what we have done in 3D.

A complete, three-dimensional model of an actin cytoskeleton interacting physically with a membrane sheet to push it around a 3D target would be game changing.

Chapter 5

Modelling infection dynamics of the fungal pathogen *Cryptococcus neoformans*

This chapter covers a short research project that investigated the infection dynamics of a particular pathogen, *Cryptococcus neoformans* (Cn). We collaborated with Kate Pline, a PhD student in the lab of Dr. Simon Johnston at the University of Sheffield's Department of Infection, Immunity and Cardiovascular disease. We first analysed Kate's experimental data using mathematical modelling and probability theory. We then developed a computational model of infection dynamics to help to understand the progression of Cn infections both *in vivo* and *in vitro*.

5.1 Introduction

Cryptococcus neoformans (Cn) is an opportunistic fungal pathogen of humans. Opportunistic meaning Cn mainly causes infections in people with compromised immune systems. Infections usually start with inhalation of Cn cells into the lungs. If untreated, Cn infections can lead to cryptococcal meningitis which is often fatal. Cryptococcal meningitis is responsible for nearly 200,000 deaths worldwide every year and is one of the leading cases of death in those with HIV/AIDS [164].

Cn cells have an outer polysaccharide capsule that help them to evade phagocytosis. The capsule interferes with phagocytic receptor binding [165] and can grow during infection [166], meaning Cn cells can become increasingly more difficult to phagocytose as infection progresses.

Macrophages are known to be the important phagocyte in dealing with Cn infections, but their role is complicated by the fact that Cn can proliferate inside them. Macrophages can kill Cn cells but also can act as a 'trojan horse' helping to spread the infection [165, 167].

Cn can also undergo a process called vomocytosis. Vomocytosis can be thought of as the opposite of phagocytosis, where phagocytosed Cn cells are expelled from the macrophage. The phagosome fuses with the membrane resulting in the Cn cell(s) being released without damaging the Cn cell(s) or the macrophage [165].

The dynamics of Cn infections is therefore very complicated and infections may even start with as few as a single pathogen [168], meaning the dynamics are highly sensitive to stochasticity.

We were approached to help investigate Cn infection dynamics by Kate Pline, a PhD student in the lab of Dr. Simon Johnston from the Department of Infection, Immunity and Cardiovascular disease. We aimed to develop quantitative understanding of Kate's experimental systems to help understand the role of macrophages. We

also wanted to investigate the possibility of predicting infection outcome from early stages in the infection. Finally, the ultimate aim was to use what we learn from Kate's experimental systems to help understand cryptococcal infections in patients, and how current anti-fungal treatments help.

5.2 The *in vivo* zebrafish model

Simon Johnston's lab use zebrafish as a model organism for investigating Cn infections *in vivo*.

5.2.1 Experimental method and results

In this section the experimental method is described and results shown. All work described in this section was done by Kate Pline under the supervision of Simon Johnston.

Zebrafish are injected with Cn cells and incubated for two hours. The Cn strain used is the Green Fluorescent Protein (GFP) expressing H99-GFP strain [169]. H99-GFP Cn cells express GFP in their cytoplasm so they fluoresce green. Once injected, the zebrafish are then imaged under a widefield fluorescence microscope, before being returned to the incubator. The images are analysed to determine the number of Cn in the fish using one of two methods. Either manually counting the number of Cn in the image, which is appropriate for a small number, or using a pixel conversion method to convert green pixels (from the fluorescence in the Cn) to a number. This pixel conversion method has been used before in [166]. Three days later they are again imaged under the microscope and the Cn number is again measured. This then allows for comparison between the initial infection and the infection three days later. This data is presented in Figure 5.1A as a plot of initial Cn number against Cn number at three days post infection (dpi). To be more exact the second measurement took place sixty-four hours after the initial.

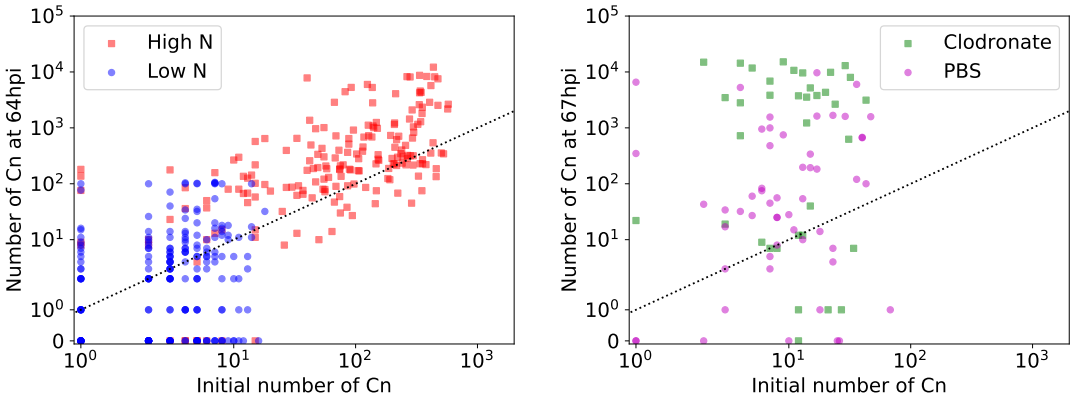
To investigate the role of macrophages the experiment is repeated but with macrophages depleted by using clodronate liposomes. The use of clodronate to deplete macrophages has previously been done before [166]. Liposomes containing clodronate are injected into the zebrafish one day prior to Cn injection. A control experiment is also carried out by injecting with liposomes containing phosphate-buffered saline (PBS) instead of clodronate. The data is presented in Figure 5.1B. To be exact the second measurement took place sixty-seven hours after the initial.

It is worth noting that the measurement technique has the limitation of being unable to measure an infection above 10^4 Cn number, any true value above this would be measured at 10^4. This is due to the fluorescent signal saturating the field of view at around 10^4 Cn.

In Figure 5.1 a line of equality is shown as a dotted black line. This is to highlight if the infection progresses (above the line of equality) or deteriorates (below the line of equality).

5.2.2 Rudimentary analysis

Pathogens such as Cn grow exponentially, the simplest model of growth is according to the equation $N(t) = N_0 \exp[kt]$, where k is a rate constant describing the growth, $N(t)$ is the number of pathogens at time t and N_0 is the number of initial pathogens. We first fitted the data presented in Figure 5.1 to this equation, with t equalling 64 and 67 hours for the original and clodronate/PBS experiment respectively. By fitting



(A) (B)

FIGURE 5.1: Initial number of Cn injected against number measured at three days post infection in zebrafish. Each plotted point is a single experiment. The dotted black line is the line of equality. Data courtesy of Kate Pline. **(A)** Number measured at 64 Hours post infection. Datasets presented here are both the 'low N' (where Cn number is counted manually), as blue circles, and the 'high N' (where fluorescent pixels are converted into Cn number), as red squares. **(B)** Number measured at 67 Hours post infection. Macrophage depleted zebrafish (using clodronate liposomes) plotted as green squares and PBS liposomes control plotted as magenta circles. Both datasets in **(B)** were measured using the pixel conversion method.

a linear line constrained to have a y-axis intercept of zero (which means if no Cn cells are added initially, none will be present three days later). The rate constant is then

$$k = \frac{\ln(m)}{t}, \tag{5.1}$$

where m is the gradient of the line (and so represents $\exp[kt]$) determined from the fit.

A better fit can be achieved by taking the logarithm (base 10) of both sides of the exponential growth equation, and fitting to the logarithm of the data. In this case the gradient is constrained to be one and the rate constant is determined by

$$k = \frac{c}{t \log_{10}(e)}, \tag{5.2}$$

where c is the y-axis intercept (and so represents $kt \log_{10}(e)$) determined from the fit.

For the experiment with macrophages, fitting gave $k = 0.0199 \pm 0.0016$ h^{-1} for the high N dataset (R^2 value of 0.29) and $k = 0.0112 \pm 0.0017$ h^{-1} for the low N (R^2 value of 0.17). For the experiment with depletion of macrophages using clodronate $k = 0.0578 \pm 0.0087$ h^{-1} (R^2 value of -0.079), and for the PBS control $k = 0.0332 \pm 0.0056$ h^{-1} (R^2 value of -0.097). These fits show a higher rate of infection growth in the fish with macrophages depleted, suggesting a role for macrophages to control infection as expected. However, there is large amounts of variability in the data due to high stochasticity, especially in the 'low N', clodronate and PBS datasets. A fit with a negative R^2 value suggests that a horizontal line would be a better fit. Negative R^2 values occur with the clodronate and PBS datasets. In Figure 5.1B there does not appear to be any correlation between initial Cn number and Cn number three days later. This could be due to the low doses used, resulting in a stochasticity dominated system. In the clodronate dataset a large amount of experiments had a three day Cn number of around 10^4, the saturation point of the technique. Therefore in these

experiments the infection could very well be higher than what is shown in Figure 5.1B. We see in Figure 5.1A, the 'high N' dataset shows a clear dependence of three day number on initial dose, whereas again, this is not clear in the 'low N' dataset. We will revisit this in section 5.4.4.

5.3 *In vitro* work

The *in vivo* system is rather complex, with many components interacting with each other. To investigate the role of these key components individually, *in vitro* experiments were carried out and analysed.

5.3.1 Experiment

This section describes the *in vitro* experiments. All experiments were carried out by Kate Pline under the supervision of Simon Johnston. Two different sets of experiments were carried out.

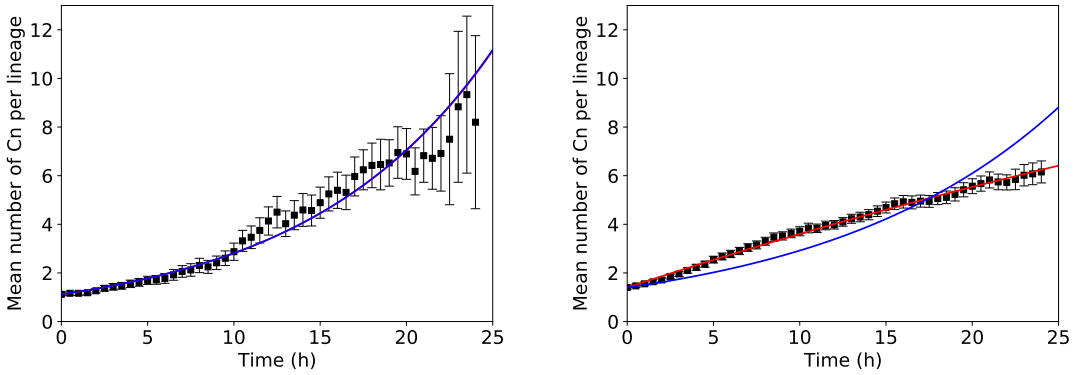
In the first experiment Cn cells are added to a growth medium at a concentration of 10^6ml^{-1}. This is then imaged every thirty minutes under a microscope for twenty-four hours. Populations of Cn cells are referred to as lineages, and initially these are most often only a single cell. Lineages are identified from the initial Cn cell(s), and the number of cells in each lineage is measured at each of these thirty minute intervals.

The second experiment is similar but also includes J774 murine macrophages at a concentration of 10^5ml^{-1}. Many Cn cells are added at a concentration of 10^6ml^{-1}. After two hours all extracellular Cn are washed out leaving only the macrophages, some of which contain intracellular Cn populations, which arise through having been phagocytosed. Each population inside a macrophage is what we call a lineage and is tracked over the twenty-four period, again by simply counting every thirty minutes. Some macrophages will have phagocytosed more than one cell and therefore strictly speaking the population growing inside the macrophage is several lineages (the number of lineages equal to the number of initial cells phagocytosed). Since we are unable to determine the number of cells the macrophage phagocytosed, we will refer to these cases as a single lineage.

Therefore the first experiment aims to investigate Cn growth outside of cells whereas the second experiment aims to investigate Cn intracellular growth. Strictly speaking, the first experiment does not measure extracellular growth since there are no macrophages present, it measures growth in the medium. However I will refer to the first experiment as 'extracellular' and the second as 'intracellular'.

5.3.2 Analysis

For the extracellular experiments forty-nine lineages were tracked. One-hundred and eighteen lineages were tracked for the intracellular experiments. For both extracellular and intracellular datasets, an average number of Cn cells (across the lineages) was calculated for each of the forty-nine timepoints (zero to twenty-four hours with a resolution of thirty minutes). Some populations were not tracked for the whole twenty-four hour period as they exit the field of view at some point. Vomocytosis events are present in the intracellular dataset, for simplicity in our understanding, we neglect these rare events. Any data corresponding to a lineage that undergoes vomocytosis is removed, but only from the timepoint of the first vomocytosis event onwards. This data is presented in Figure 5.2.



(A) (B)

FIGURE 5.2: In vitro data. Average number of Cn per lineage at particular timepoints are shown as the black squares, error bars are standard errors. The blue line is the fit to equation 5.3 and the red line is the fit to equation 5.4. **(A)** Extracellular. **(B)** Intracellular.

In both datasets there is never any loss of fluorescent signal seen, which could mean one of either two things; there is never any death of Cn, or there is death but the Cn continues to fluoresce for the remainder of the experiment, therefore being indistinguishable from a living Cn. The only possible way we could tell if a Cn might be dead is if it does not reproduce - however this is no guarantee, since a living Cn may also not reproduce. To investigate this we looked in the data for any of the initial Cn that never reproduce in the whole experiment. In the extracellular experiment, of the forty-five lineages that begin with just a single Cn, only one fails to replicate, this Cn exits the field of view at the twenty-two hour mark. So we can say that in twenty-two hours only one in forty-five Cn has not replicated. In the intracellular experiment, of the eighty-two lineages that begin with a single Cn, only two do not reproduce and both of these remain in the frame for the entire twenty-four hour experiment. Therefore in twenty-four hours only two in eighty-two have failed to replicate. These findings suggest that if there was death, it would be of a much lower rate than replication, at least in the initial stages anyway.

To investigate the values of the replication rate and death rate we derived some simple analytical expressions, and fitted the data to them. In the simple case where there is no death we have $\frac{dN}{dt} = k_r N(t)$, where N is the number of Cn and k_r is the replication rate. Therefore

$$N_{\text{tot}}(t) = N_{\text{A}}(t) = N_0 \exp[k_r t], \tag{5.3}$$

where $N_{\text{tot}}(t)$ is the number of Cn measured in experiment at time t, $N_{\text{A}}(t)$ is the number of Cn alive at time t.

In the case where there is death but no loss of fluorescent signal, and assuming all injected Cn are alive to begin with, following from the same logic as above we can start with $N_{\text{A}}(t) = N_{\text{A}}(0) \exp[\,(k_r - k_d)\,t]$, where k_d is the death rate of a Cn. Denoting N_{D} as the number of dead Cn we have $\frac{dN_{\text{D}}}{dt} = k_d N_{\text{A}}(t)$, then $\frac{dN_{\text{D}}}{dt} = k_d N_{\text{A}}(0) \exp[\,(k_r - k_d)\,t]$, integrating and solving for the initial condition of $N_{\text{D}}(0) = 0$ gives

$$N_{\text{tot}}(t) = N_{\text{A}}(t) + N_{\text{D}}(t) = \frac{1}{k_r - k_d} N_0 \left(k_r \exp[(k_r - k_d)\,t] - k_d\right). \tag{5.4}$$

By setting $k_d = 0$ h^{-1} equation 5.4 reduces down to equation 5.3.

We then fitted equation 5.3 to both datasets, constraining N_0 to be the value of the first datapoint in the set. For the extracellular dataset this gave $k_r = 0.092 \pm 0.001$ h^{-1} and for the intracellular dataset $k_r = 0.074 \pm 0.002$ h^{-1}. In Figure 5.2 these fits are plotted as blue lines. In Figure 5.2A it can be seen by eye that the extracellular data is nicely exponential and the fit is good (R^2 value of 0.97). However in Figure 5.2B the fit is poor (R^2 value of 0.87). As the intracellular data looks linear, this may suggest there is death.

We then fitted equation 5.4 to both datasets with the same constraint on N_0, but now of course we have a two parameter fit for k_r and k_d. For the extracellular dataset this gave $k_r = 0.092 \pm 0.005$ h^{-1} and $k_d = 0.00 \pm 0.01$ h^{-1} and for the intracellular dataset $k_r = 0.168 \pm 0.004$ h^{-1} and $k_d = 0.181 \pm 0.007$ h^{-1}. Equation 5.4 with these parameter values is plotted as red lines in Figures 5.2A and 5.2B. This extracellular fit (again R^2 value of 0.97) is in agreement with the fit to equation 5.3 (the red line and blue line are on top of one another), suggesting no death is present. However, the intracellular fit (R^2 value of 0.99) suggests there is death, and in fact the death rate is higher than the replication rate.

We can investigate further by considering the rate values from our fits with our knowledge that only a very small proportion never replicate. We can calculate, using a Poisson distribution, the probability of a given Cn failing to replicate. For the case where there is no death this is simple:

$$P\left(\text{reps} = 0\right) = \frac{\left(t_{\text{final}} k_r\right)^0 e^{-t_{\text{final}} k_r}}{1} = e^{-t_{\text{final}} k_r}, \tag{5.5}$$

using $k_r = 0.092 \pm 0.001$ h^{-1} and a t_{final} of 22 hours for our extracellular case gives a probability of 0.132 ± 0.003. This means we expect 13.2% (or around 6) of our initial Cn to never divide in twenty-two hours. As previously discussed, in the experiment only 2.2% (1) do not divide. We can back-calculate k_r given that $P\left(\text{reps} = 0\right) = 0.022 \pm 0.022$ (uncertainty calculated as $\sqrt{\frac{(1/45)(44/45)}{45}}$), this gives $k_r = 0.173 \pm 0.045$ h^{-1}, higher than the fitting suggests.

With death this calculation becomes a little more complicated. In a given time interval a Cn can either replicate, die or do nothing. The probability of an initial Cn failing to replicate is one minus the probability of it replicating.

$$P\left(\text{reps} = 0\right) = 1 - k_r \left(\frac{1 - e^{-t_{\text{final}}(k_r + k_d)}}{k_r + k_d} \right), \tag{5.6}$$

using $k_r = 0.168 \pm 0.004$ h^{-1}, $k_d = 0.181 \pm 0.007$ h^{-1} and a t_{final} of 24 hours as in the intracellular case, we get a probability of 0.519 ± 0.015. This means we expect 51.9% (or around 42-43) of our initial Cn to never divide in twenty-four hours, drastically different from the 2.4% (2) measured. If we assume no death (so equation 5.6 reduces to equation 5.5) and back-calculate k_r given that $P\left(\text{reps} = 0\right) = 0.024 \pm 0.017$, we get $k_r = 0.155 \pm 0.029$ h^{-1}.

So what does this suggest? It is possible that the replication rate is not constant in time but actually decreases, and so most Cn initially divide but then their growth is hindered. This effect is much more dramatic in the intracellular case, it may be that there is no significant death, but the ability of Cn to replicate is hindered. Macrophages are known to limit the nutrients available to pathogens in phagosomes and actively try and kill them, for example by reducing the pH or introducing digestive enzymes into the phagosome [170, 171]. Cn are known to be resistant to a number of

these mechanisms [165] and it is likely that what we have observed is the interaction between the macrophage killing and growth restriction mechanisms with the resistance mechanisms of Cn. In other words, we may have observed the incomplete inhibition of cryptococcal growth that on average results in linear growth of population inside macrophages.

5.4 Stochastic simulation

To better understand the Cn infected zebrafish system we turned to numerical modelling.

5.4.1 Model description

We developed a simple stochastic simulation of infection dynamics, the details of which are presented in this section. We consider replication and death of individuals representing individual Cn which exist in either a single population, or two separate populations - intending to model the intracellular and extracellular populations.

Simulations are carried out using the Gillespie algorithm [81], a Monte Carlo (MC) technique widely used for describing the time evolution of chemical systems. In the description of our implementation below, the simplest case of a single population model is used as an example. For a more detailed explanation and analysis of this technique see [82].

A given simulation is initialised with an initial number of Cn, N_0, as well as (some of) the parameters given in table 5.1. The first step after initialisation is to calculate the sum of all the propensity functions (α) that describe all the events that can take place. The propensity function describing replication is $\alpha_r(t) = N(t)k_r$ and for death, $\alpha_d(t) = N(t)k_d$, where $N(t)$ is the number of Cn at time t, and k_r and k_d are the replication rate and death rate of an individual Cn respectively. The sum is therefore simply

$$\alpha(t) = \alpha_r(t) + \alpha_d(t) = N(t)k_r + N(t)k_d, \tag{5.7}$$

The second step is the calculation of the dynamic timestep τ, this is given by

$$\tau(t) = \frac{1}{\alpha(t)} \ln \frac{1}{r_1}, \tag{5.8}$$

where r_1 is a random number uniformly distributed in (0, 1). For an explanation of where equation 5.8 comes from, see the appendix of [80] or [82].

The third step is the choosing of the event, this is done by drawing a second random number (from the same distribution as r_1), r_2. The possible events that can happen between the time t and $t + \tau(t)$ are a single replication, where N increases by one, or a death, where N decreases by one. The event that takes place is chosen using the following logic,

$$N(t + \tau(t)) = \begin{cases} N(t) + 1, & \text{if } r_2 < \alpha_r(t)/\alpha(t) \\ N(t) - 1, & \text{otherwise.} \end{cases} \tag{5.9}$$

Here the top outcome is a replication event whereas the bottom outcome is a death event.

The time is then advanced by $\tau(t)$ and the algorithm is repeated until the time reaches the runtime of the simulation, t_{end}.

Table 5.1: Parameters used in the model.

Symbol	Description	Values	Refs.
N_0	Initial number of Cn	10^0 - 10^3	As experiment
t_{end}	Runtime of simulation	64 or 67 h	As experiment
k_r	Cn replication rate*	0.092 h^{-1}	Estimated here
k_d	Cn death rate*	0 h^{-1}	Estimated here
P^{intra}	Probability of Cn being intracellular†	0.25	[166]
k_r^{intra}	Cn intracellular replication rate†	0.168 h^{-1}	Estimated here
k_d^{intra}	Cn intracellular death rate†	0.181 h^{-1}	Estimated here
k_r^{extra}	Cn extracellular replication rate†	0.092 h^{-1}	Estimated here
k_d^{extra}	Cn extracellular death rate†	0 h^{-1}	Estimated here

* Used only in the single population case.
† Used only in the two population case.

The derivation of equation 5.8 is given in [82], which finds that, for chemical reactions, the time interval from one reaction to the next is distributed according to the exponential distribution. They assume that each reaction acts independently as a Poisson process. The birth-death processes in our Crypto model are Poisson processes - each event is an individual Poisson process. The time to the next event is independent of the time since the previous event. The birth-death events are independent, for example a Cn is no more likely to replicate after a nearby Cn has died. In our experiments there is no indication of any dependency between events, therefore we can assume that they are independent and random. Given this assumption, the exponential distribution is the appropriate distribution to use here.

5.4.2 Two population version

The *in vitro* experimental analysis suggests different growth dynamics for intracellular and extracellular populations of Cn.

In order to model the *in vivo* system we need to model both intracellular and extracellular populations. We therefore have two separate populations each experiencing their own replication and death rates. The initial population sizes are determined by using a probability P^{intra}, which is the probability of any given Cn being assigned to the intracellular population. This is done using a MC step, where a random number uniformly distributed in the interval (0, 1) is drawn. If the random number is less than P^{intra}, the intracellular population grows by one, if not, the extracellular population grows by one, this process is repeated N_0 times. We make two simplifying assumptions. Firstly, the timescale for phagocytosis is short compared to the timeframe of the experiment [166]. We therefore make the assumption that phagocytosis happens instantly at the beginning of the experiment. Secondly, we neglect vomocytosis, as only around 5-15% of macrophages have reported to undergo vomocytosis in zebrafish [166]. In the simulation these two assumptions are carried out by not allowing transfer of Cn between the two populations.

5.4.3 Parameters used

Table 5.1 gives the numerical values which are either a range or a single value based on our experimental evidence, the references are given in the right hand column.

5.4.4 Results

Simulations were ran using the one population model; modelling the *in vivo* macrophage depleted system from using clodronate filled liposomes. Simulations were also ran using the two population model; modelling the healthy *in vivo* system. We used parameter values suggested from the *in vitro* fitting (section 5.3) for extracellular and intracelluar replication and death. We initialised each simulation with an N_0 taken from a set of thirty numbers distributed equally in logarithmic space from one to one-thousand. For each N_0, one-hundred simulations were ran, giving a total of three-thousand simulations for each of our two cases.

We ran three sets of simulations to compare with three experimental datasets. First we ran two populations simulations for a runtime of 64 hours to compare with the 'high N' dataset presented in Figure 5.1A. Then we ran one population simulations for 67 hours to compare with the clodronate liposome data (Figure 5.1B). Finally we ran two population simulations for 67 hours to compare with the PBS control data (Figure 5.1B). The results are presented in Figure 5.3.

In Figures 5.3A, C and E individual simulation results are presented as in Figure 5.1: one circle corresponds to one simulation. For Figures 5.3 B, D and F, a mean and standard deviation was calculated for each of the thirty N_0's used, these are shown as error bars. Also for Figures 5.3 B, D and F, the experimental data was binned into ranges, of roughly equal size in logarithmic space, and plotted as a bar chart. The height of each bar in the bar chart is the mean of that bin. The error bars shown are the standard deviation. These bins were [1,15), [15,50), [50,100), [100,250) and [250,600) for the 'high N' data in Figure 5.3B. The bins for the clodronate dataset were [1,8), [8,20) and [20,45). The bins for the PBS dataset were [1,6), [6,10), [10,22) and [22,70). The values printed in each bar of the bar chart are the number of experiments to fall into that bin.

Results of the 64 hour two population simulations, shown in Figure 5.3A and B, are clearly not fully in agreement with the experimental data. Using these rates give an infection that is consistently more than an order of magnitude higher than that measured in zebrafish. The general trend of the simulation is similar to that of the data. The relative (relative to the mean) spread in the simulations are in agreement with the relative spread seen in the experiments at low initial numbers of Cn, but at large initial number this is not the case. Care must be taken when evaluating the spread of the data from these plots, as the y axis is logarithmic.

The one population simulation results, shown in Figures 5.3C and D, show perhaps slightly better agreement in terms of the means at low initial doses. However, again the relative spread is not in agreement at higher initial doses of Cn. Figures 5.3E and F tell a similar story. The bar charts in Figures 5.3D and F highlight the absence of dependence on initial doses present in the experimental data (the bar charts are flat). In the simulations there is a strong trend between initial dose and measured dose three days later. In the next section (5.5) we will discuss potential improvements to the simulation that could be investigated in the future.

We ran the one population simulations with no death, since that is what the extracellular *in vitro* experiments suggested. However in the *in vivo* system with macrophages depleted death is still seen (although possibly reduced). This is not surprising as although macrophages are thought to be important, there are still other anti-microbial cells and proteins capable of killing Cn present in the zebrafish.

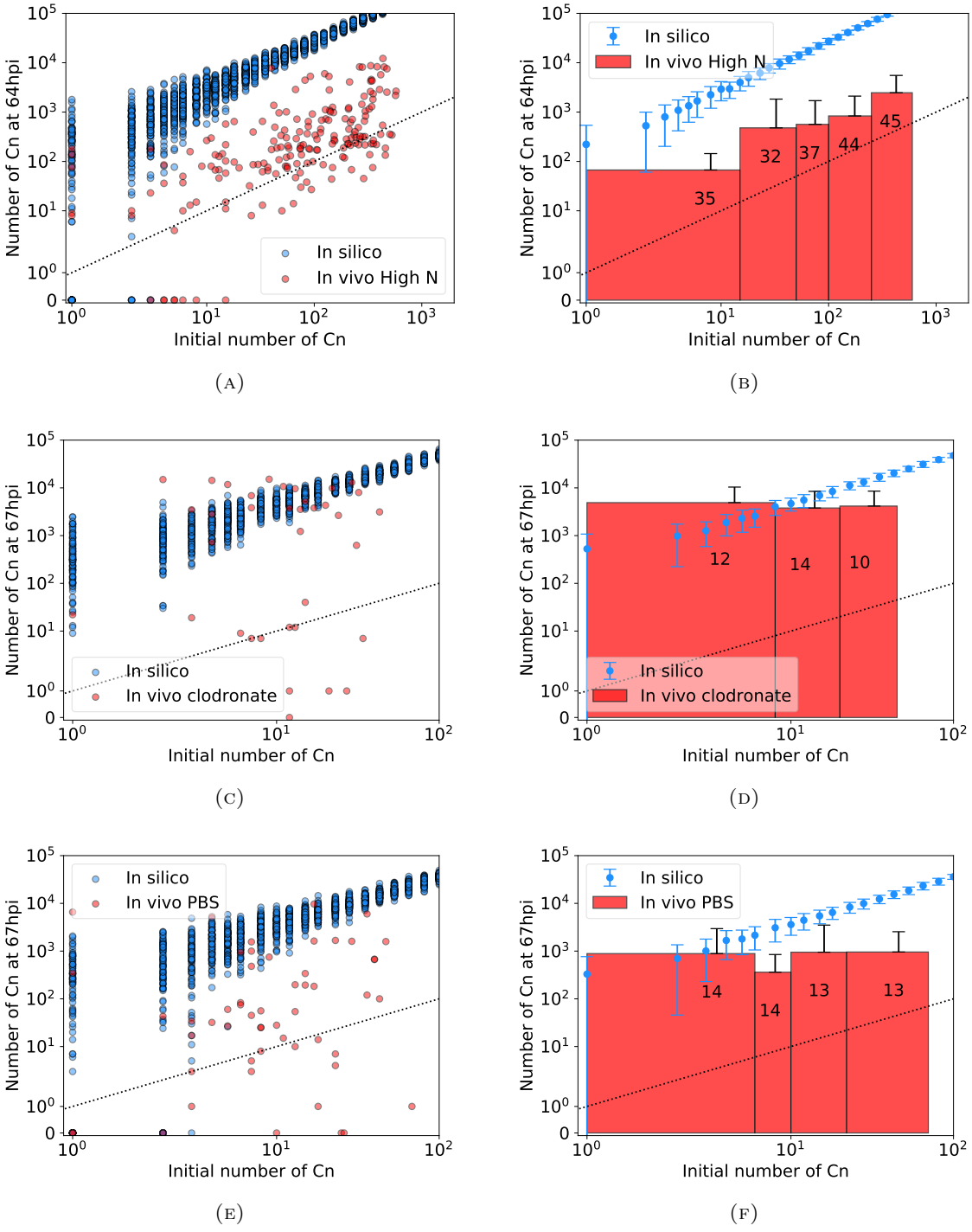


FIGURE 5.3: Simulation results. In **(A)**, **(C)** and **(E)** circles represent results of individual experiments (red) and simulations (blue). In **(B)**, **(D)** and **(F)** each of the thirty set of runs for an initial Cn number has a mean and standard deviation calculated and plotted as a blue circle with error bars. For the experimental data, the data has been binned and plotted as a bar chart. The height of each bar in the bar chart is the mean of that bin and vertical error bars show the standard deviations. The numbers inside each bar are the number of experiments that fall inside that bin. **(A) and (B)**: Results of two population simulations using the rates suggested by the fitting in section 5.3. Results from simulations are plotted as blue circles, while the 'high N' dataset previously plotted in figure 5.1a is plotted as red circles. **(C) and (D)**: Results of one population simulations plotted as blue circles, the clodronate dataset previously plotted in figure 5.1 is plotted as red circles. **(E) and (F)**: Same simulations as plotted in **(A) and (B)** with the PBS dataset.

5.5 Future work

5.5.1 Stochastic simulation: transfer between populations

In section 5.4.2 we discussed the simplifying assumption of not allowing transfer of Cn individuals between populations. This omission could be omitting important aspects in the dynamics of the infection: phagocytosis and vomocytosis. Moving from extracellular to intracellular would represent phagocytosis of Cn. Moving from intracellular to extracellular would represent vomocytosis of Cn. This would be very simple to include in the model and just requires time to investigate the effects.

5.5.2 Fluorescent signal loss *in vivo*

In the *in vitro* experiments covered in section 5.3, no loss of fluorescent signal was ever seen. However in the *in vivo* system, there is loss of signal due to what we assume is Cn death. In Figure 5.1, points that sit below the line of equality represent experiments where the Cn number has decreased overall. The time taken from the death of a Cn to the loss of the fluorescent signal associated with that Cn should not automatically be assumed to be instant, as dead Cn can still fluoresce [172].

We can adapt our numerical model to take this into account. Rather than the number of individuals representing the number of alive Cn, it would represent the number of fluorescent Cn. Alive Cn cells would still die according to a death rate k_d but then dead Cn cells would still fluoresce. Dead Cn cells would then stop fluorescing according to a fluorescent signal loss rate, k_{loss}.

To investigate this fluorescent signal loss *in vivo*, dead Cn were injected into zebrafish and their fluoresence tracked over time. This data was provided by Simon Johnston and is presented in Figure 5.4.

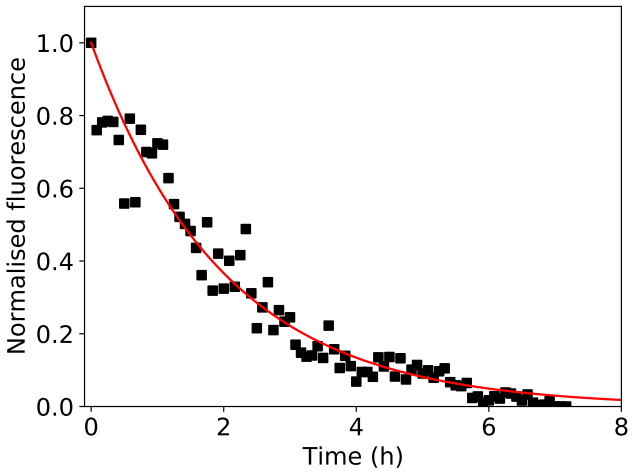


FIGURE 5.4: Fluorescence of dead Cn cells in a region of a zebrafish over time. Normalised to the initial measurement. Red line shows a fit to an exponential decay law. Data courtesy of Simon Johnston.

We fitted the data to a simple exponential decay equation $N(t)/N_0 = \exp[k_{\text{loss}}t]$, where k_{loss} is the decay constant. The fit is shown as the red line in Figure 5.4. This suggests a signal loss rate constant $k_{\text{loss}} = 0.502 \pm 0.013\ \text{h}^{-1}$. This value of k_{loss} could then be used in simulations.

5.5.3 Lineage tracking

In section 5.3.1 the experiments were discussed using the language of lineages. These size of the lineages in Cn number were measured over time but the lineage-tree was not mapped. We do not know which Cn cells divided and which did not. The analysis done in section 5.3.2 using the Poisson statistics was only based on initial Cn from a lineage that starts with a size of one. This is because we know if these Cn replicate or not. Full lineage tracking would help fuel this analysis done in section 5.3.2, as we will know whether all of the Cn cells replicate or not. This may help uncover if the growth and/or death rates change in time.

Preliminary work into lineage tracking was carried out by two undergraduate students under the supervision of Rhoda Hawkins; Joseph Bartram and Sarah Chapman. Joseph and Sarah explored the use of automatic tracking techniques with the *in vitro* dataset covered in section 5.3. They found that the time resolution of the *in vitro* dataset (one image every thirty minutes) was not high enough for accurately keeping track of each cell. However their theoretical work did suggest a higher time resolution of around one image every fifteen minutes would be sufficient. Future work would therefore require repeating the experiments at this higher time resolution to be then able to carry out the lineage tracking.

5.5.4 The clinical system: the role of anti-fungal treatment

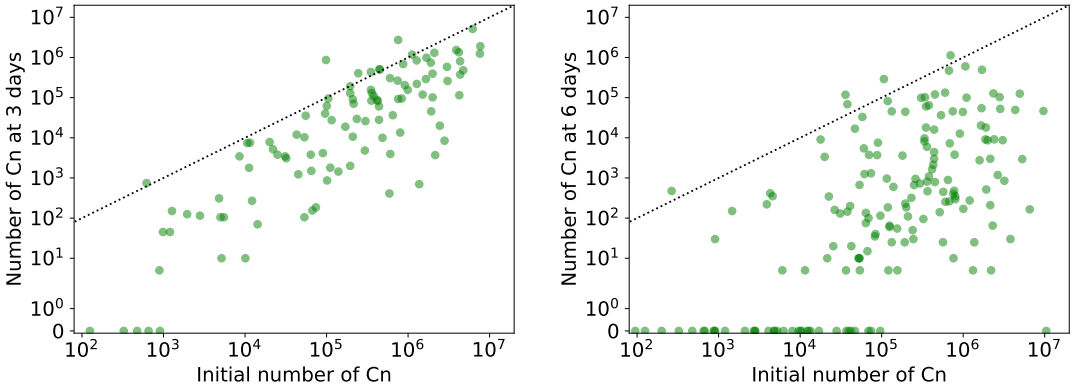
We also want to investigate the role of current anti-fungal treatment given to patients suffering with Cn infection. We did carry out some rudimentary analysis on clinical data, which is presented below, but more investigation is required.

Up until this point we have covered how Cn infections grow from very low initial numbers. However we can apply the same analysis to infections of large numbers that are reduced and cleared through anti-fungal treatment. We used clinical data of Cn counts in cerebral spinal fluid samples taken from patients with cryptococcal meningitis undergoing treatment with anti-fungal drugs. This data was taken from multiple published trials in Southern Africa shared by Prof. Joe Jarvis, London School of Hygiene and Tropical Medicine. Measurements were taken on day zero (prior to the start of treatment) and then various days after, during treatment. In our analysis we focused on day three and day six as these days had a large amount of measurements taken. The data is presented in Figure 5.5.

Fitting equation 5.2 to each dataset gave $k = -0.0324 \pm 0.0027$ h^{-1} (R^2 value of 0.63) and $k = -0.0379 \pm 0.0018$ h^{-1} (R^2 value of 0.63) for the three day and six day datasets respectively. There is a large spread in the data, due to the expected stochasticity. However our fits suggests the anti-fungal treatments are working as expected since there is a general decline in the number of Cn over time.

5.6 Conclusions

We developed a robust quantitative analysis method for the *in vitro* system in section 5.3.2. This suggested complex dynamics in the growth of Cn populations. We uncovered a possible change of dynamics in the intracellular case in time, as most Cn cells replicate but the growth of populations is linear. This may suggest an initial burst of replication followed by a slowing down of replication or possibly death of Cn cells. We provided a possible explanation for this as a competition between the macrophage trying to kill the Cn, verses the Cn trying to resist. This may result in a delay in the macrophage killing the Cn, or the Cn survives but its replication is



(A) (B)

FIGURE 5.5: Progression of Cn infection during anti-fungal treatment. Numbers are measured from samples of cerebral spinal fluid taken from patients. Each green circle represents a patient. The dotted black line is the line of equality. **(A)** Number of Cn measured at 3 days after starting treatment. **(B)** Number of Cn measured at 6 days after starting treatment.

hindered. In the 'extracellular' case this slowing down of replication was also seen, but to a lesser extent.

Simulations help to show the difference in the *in vitro* system to the *in vivo*, as in the *in vitro* system the growth is much higher. The *in vivo* system is obviously much more complex, with many factors contributing to the infection dynamics. To be able to predict infection outcome in this system is non-trivial. Our simulations also show a possible agreement in variability at low initial Cn number but fail to replicate the high variability at high initial Cn number.

Macrophages have a clear impact in the *in vitro* system (compare Figures 5.2 A and B) but it is still unclear *in vivo*. The rudimentary analysis presented in section 5.2.2 suggests higher infection growth without macrophages. However, even with macrophages depleted in the clodronate experiments, some infections were cleared altogether. This suggests macrophages may overall help, but are not necessarily crucial to dealing with Cn infections.

Future work necessary would be lineage analysis of the in vitro data to help shed light on the complex dynamics found. More simulations could be run, sweeping out parameter space to better fit the data, however as always this must be done with care. Adding complexity to the simulation model must also be done with care, to not over complicate and dilute down what could be learnt. Another interesting avenue would be to understanding the clinical system and how anti-fungal treatments affect dynamics of the two-populations.

This short project really only scratches the surface of this fascinating system. We contributed mathematical modelling, probability theory and simulations to aid our collaborators in understanding Cn infections and the interesting interaction Cn has with macrophages.

Chapter 6

Conclusions

In this final chapter we will summarise the research that has been presented. We will then discuss future areas of study including some extensions that could be made to our simulation tool.

6.1 Summary of research presented

In this PhD project we focused on actin, actin's role in phagocytosis and the dynamics of infection. We aimed to shed light on these topics by using computational and mathematical methods, including those we developed ourselves.

In chapter 2 we presented an extensive computational model which captures the fundamentals of actin polymerisation, and the physics of actin biopolymers. In chapters 3 and 4 we described some additional components to the base model, allowing it to be used to investigate the two systems in these chapters.

In chapter 3 we discuss a research project that aimed to uncover more about actin based systems in two particular *in vitro* experiments. We developed novel analysis techniques that challenged pre-existing assumptions surrounding the experimental data. We provided an analysis pipeline for use with the pyrene-actin experimental system. This provided a method for the experiment output of arbitrary fluorescence intensity to be converted into interpretable units of concentration. We also contributed mathematical modelling, simplifying the systems down to a few coupled differential equations. By using our mathematical model alongside our analysis pipeline we were able to find the nucleation order of actin to be three, which therefore suggests that the trimer is the size of the actin seed. We found the same for the system with the actin nucleator Las17. However more repeat experiments are necessary as stated in chapter 3 to verify the results. The branched system provided more challenges but the work we started will hopefully lead to some exciting developments here. There seems to be some disagreement in the existing literature surrounding whether the order of branched nucleation on G-actin concentration should be linear or quadratic [11, 110, 111, 141]. In order to help resolve this disagreement we would need to extend our differential equations perhaps by taking into account branch spacing, pointed end capping or branch detachment at long times. Simulations would also be key in this endeavour too, as we could use them to test the validity of assumptions made in the differential equations.

Chapter 4 concerned our model of phagocytosis. We were inspired by previous models of actin filaments generating forces for example in lamellipod protrusion [67, 68]. We extended our model of actin polymerisation dynamics to investigate force generation by adding in a model membrane and simple target to engulf. This approach allowed us to investigate the effects of actin processes such as elongation and branching, on the 'ease of phagocytosis', determining which processes are of importance. We

see an optimum target size for phagocytosis, which is seen in our collaborator's experimental data as well as in previous modelling work [69]. We found branching of actin to be essential to push the membrane outwards around the target. We also found some nucleation to be essential and increasing nucleation to be beneficial. There are many possible directions to take this work in the future, which was discussed at the end of chapter 4. We highlight the importance of continuing our work here, running more simulations to help investigate size dependence and the importance of membrane exocytosis. Exciting avenues of research would be furthering our preliminary investigations into phagocytosis of two targets. Which looks at whether one phagocytic cup forms or two based on the distance between the two targets. Adding some complexity to the model of the target would also be interesting. This could include looking at different target shapes and orientation with respect to the membrane, and having targets that are able to deform.

Finally in chapter 5 we presented a project which again aimed to uncover more detail from experimental data. Here we wanted to explore the effects of phagocytosis on infections at a larger scale. We looked at infection dynamics, and the interactions between phagocytes and a fungal pathogen. We investigated unexpected growth dynamics of phagocytosed pathogens by providing some simple mathematical models and by considering probability theory. We also provided a computational model, which aims to capture the basics of the progression of an infection over time. We found interestingly that despite most intracellular pathogens replicating at least once, their growth as a population was linear. We explained this as a potential competition effect between the phagocyte trying to kill the ingested pathogens verses the pathogens resisting. Our simulations help to show that modelling infections *in vivo* is non-trivial, with many complex factors contributing towards infection outcome. However the simulation model could easily be extended and enables future investigation of more complex factors.

6.2 Future work

All aspects of this PhD project provide many avenues for further work. We highlighted specific areas for exciting future study in the related chapters. Rather than simply repeat what was suggested there, we instead make some suggestions for extending and using our simulation tool based on our experience using it in chapters 3 and 4.

6.2.1 Extensions to the simulation tool

We did not fully explore everything we developed in the model. In our introduction chapter (section 1.1.3) we mentioned the similarity between bundling and crosslinking, and that the conditions for forming a bundle over a crosslink are not fully understood. We never really investigated this, but it could easily be explored using the features of the model we already have. In chapter 2 we highlighted the modular nature of the source code of our simulation tool. This allows for extensions to be added such as including other actin processes for example the inclusion of motor proteins such as myosin.

The current version of the tool has some limitations. One such limitation is the long wall time required to run certain simulations. Simulations that include a membrane can take days to run, meaning getting enough repeats for good statistics can take time. The main reason for this is the length of the timesteps required to simulate the membrane are very small, roughly 10^{-5} seconds. This comes from the timestep restrictions outlined in section 2.6.1, coupled with the fact that the effective persistence

length for the membrane scales with membrane size (see section 2.4.2). One potential solution to this would be to switch the Brownian dynamics method to an 'implicit scheme'. The current scheme outlined in section 2.3.2.3 is explicit. Simply put, an explicit scheme calculates the state of the system at the later time step from the state of the system at the current time step. Implicit schemes use information about the current state and the later state to determine a solution. An example of an implicit scheme is the backward Euler method. The advantage of implicit schemes are that the size of the time steps required to produce stable solutions can be orders of magnitude greater than those required in explicit schemes. The disadvantages are that they can be difficult to implement and by using larger timesteps they can produce less accurate results [173]. Switching our Brownian dynamics method from the explicit one described to an implicit one would allow much larger timestep sizes allowing us to probe timescales currently unavailable. A smaller subunit size l_s would also be possible, as implicit schemes have been used to simulate actin filaments with a subunit size of just two monomer lengths [61]. This would remove this slight disparity between our scheme for moving 'short rigid' filaments and structures, and the Brownian dynamics method. Another method for decreasing wall time of simulations would be to increase parallelisation of the code. The model already runs some of its algorithms across multiple cores of the computer's CPU but more of the code could take advantage of this. An example of this is polymerisation. Currently polymerisation of actin filaments happens by first determining if the filament should polymerise (by the MC method), if it should it is then polymerised. The steric grid is then checked for any conflicts due to this polymerisation. Then the algorithm progresses to the next filament. In a parallel method, all filaments undergo their MC method check and if they pass they polymerise, all in parallel. Then the steric grid can be checked just once, and any conflicts resolved. Parallelisation could be taken further by rewriting parts of the code using the CUDA programming language so it can run on GPUs [174].

Another limitation of our tool is that it is a 2D model, this was covered in section 4.7.7. Modelling some actin based systems including phagocytosis may be better suited in three dimensions. Limitations of using a 2D model include exaggerated effects of steric hindrance in 2D over 3D. Since in 3D filaments are less likely to be hindered by other filaments as they have an extra dimension to move and grow in. Some scenarios may require 3D when symmetry assumptions break down. There may be something interesting happening in the other dimension that we cannot account for. Modelling in 3D is related to the previous limitation regarding long simulation wall time. Running these simulations in 3D would be more taxing computationally, further slowing down simulations. The technical implementation of moving up a dimension would present some challenges but not be too difficult to achieve. The actin filaments could easily be represented in a 3D coordinate system. The processes such as branching and crosslinking would also be straightforward to scale up to 3D. The membrane could be modelled as a 2D sheet, perhaps represented as a mesh of points, although much thought would be needed to carry this out.

6.2.2 Other systems that we can model

We built our simulation tool to look at actin and phagocytosis specifically, however there are many related cellular processes that this could be used for.

Other processes that involve actin polymerisation could be investigated using our model. These could include cell migration, and the formation of podosomes and filopodia.

In chapter 4 we provided a simple model for an actin cortex which is tethered to our membrane. If some of speedup suggestions from the previous subsection are made then this could be replaced with a more complete cortex made up of many crosslinked individual filaments.

In chapter 3 we developed the model further to be able to further understanding of two experimental systems, and pull more information out from the data. There is no reason why this could not be also achieved for other experimental techniques such as Fluorescence recovery after photobleaching (FRAP) [26] for example. It is helpful to, when possible, couple our simulations with analysis of data and mathematical modelling, as we did in chapters 3 and 5.

6.3 Concluding remarks

One of the advantages of being a physicist working in the field of biophysics is the cross-disciplinary nature of the work. The research presented in chapters 3 and 5 were heavily cross-disciplinary and as a result very rewarding. We were able to provide a more quantitative insight into these experimental systems than usual, and by working together achieve more than what would have been done working alone. Lots of cross-disciplinary discussion surrounded the work presented in chapters 2 and 4 too. Working with collaborators in Sheffield helped in the development of our models, allowing us to focus on the important biology. This is also this case from interactions with others at conferences. I had the pleasure of presenting this work at many different conferences, and the feedback received was valuable.

In this thesis we presented our investigations into actin based systems including phagocytosis, and infection. These fascinating areas of research provided many challenges along the way. We developed novel techniques to help analyse experimental data, as well as simulations to help uncover the underlying physics involved. Our work here is hopefully just the start of many exciting projects to come.

Appendix A

The derivation of torsional spring forces in branched actin

In chapter 2 we presented Equation 2.39, which is

$$\boldsymbol{F}_i^{\textbf{ang-br}} = -\boldsymbol{F}_{i+1}^{\textbf{ang-br}} = k_{\theta_{\mathrm{br}}} \Delta\theta \frac{\hat{\boldsymbol{n}}_i}{d_i}. \tag{A.1}$$

Equation A.1 is the force calculated to maintain a branch at the characteristic rest angle from its mother. It is of the same form as the discrete bending forces we stated in the thesis, these were equations 2.18-2.23. Therefore it can be derived from the bending energy of a polymer, equation 1.2, which is

$$U^b = \frac{\kappa_b}{2} \int_0^L \left(\frac{\partial^2 \boldsymbol{r}(s)}{\partial s^2} \right)^2 ds. \tag{A.2}$$

We first must discretise equation A.2. To help do this, we will write equation A.2 in terms of the tangent which is

$$\boldsymbol{t}(s) = \frac{\partial \boldsymbol{r}(s)}{\partial s}, \tag{A.3}$$

giving us

$$U^b = \frac{\kappa_b}{2} \int_0^L \left(\frac{\partial \boldsymbol{t}(s)}{\partial s} \right)^2 ds. \tag{A.4}$$

For small angles $\tan(\theta) \approx \sin(\theta) \approx \theta$. So we can replace the tangent with any angle relative to a fixed axis [5]. Therefore we have

$$U^b = \frac{\kappa_b}{2} \int_0^L \left(\frac{\partial \theta(s)}{\partial s} \right)^2 ds. \tag{A.5}$$

Then we discretise the integral into

$$U^b = \frac{\kappa_b}{2} \sum_{i=1}^{N-1} \left(\frac{\Delta\theta_i}{\Delta s} \right)^2 \Delta s = \frac{\kappa_b}{2} \sum_{i=1}^{N-1} \frac{(\Delta\theta_i)^2}{\Delta s}. \tag{A.6}$$

Where $\Delta\theta_i = \theta_{i+1} - \theta_i$, so the angle change from one subunit to the next. Now we need to differentiate this (or find the gradient) as

$$\boldsymbol{F^b} = -\nabla U^b. \tag{A.7}$$

Considering a single point or subunit i in the summation, equation A.6 is

$$U_i^b = \frac{\kappa_b}{2}\frac{(\Delta\theta_i)^2}{\Delta s}. \tag{A.8}$$

Which we then differentiate in polar coordinates giving

$$\boldsymbol{F^b}_i = -\nabla U_i^b = -\frac{1}{r}\frac{\partial U_i^b}{\partial \theta}\hat{\boldsymbol{\theta}} = -\frac{\kappa_b}{r}\frac{\Delta\theta_i}{\Delta s}\hat{\boldsymbol{\theta}}. \tag{A.9}$$

r is the radial distance from the fixed axis we defined θ relative to. In the bending forces equations (2.18-2.23) the desired subunit length l_s is used for r. With the branches we do not necessarily want the stiffness of the branch-mother link to be the same as the bending rigidity of an actin filament. The branches are held at their characteristic 70 degree angle by the Arp2/3 molecule. It is the angular strength of the Arp2/3 molecule that determines the stiffness of the branch-mother link. So for our branches we replace $\frac{\kappa_b}{r}$ with the branching torsional stiffness $k_{\theta_{\mathrm{br}}}$. This is not necessarily equivalent numerically, and in the main text we said we experimented with the values for $k_{\theta_{\mathrm{br}}}$. Equation A.9 therefore becomes

$$\boldsymbol{F}_{i+1}^{\textbf{ang-br}} = -k_{\theta_{\mathrm{br}}}\frac{\Delta\theta}{d_i}\hat{\boldsymbol{\theta}}_i. \tag{A.10}$$

Where we have replaced Δs with the length of subunit i. $\Delta\theta$ is now the angle change from the branch rest angle, rather than the angle change from one subunit to the next along a single filament. Since the oscillations about the rest angle will be small, the small angle approximation we used earlier still holds. $\hat{\boldsymbol{\theta}}_i$ is equivalent to what we defined $\hat{\boldsymbol{n}}_i$ to be in the main text.

Appendix B

Verifying the forces in the simulation

B.1 Calculating expected movements of objects analytically

Let's assume a 1D system with a bead of radius r_b and an actin filament of length L and radius r_a. The viscosity of the medium they are in is η, the temperature is T. Let's assume that the filament is aligned with the 1D axis and therefore its movement is characterised by the parallel mobility (inverse of its drag coefficient) given in equation 2.2 but repeated here

$$\mu_{\parallel} = \frac{\ln\left(L/2r_a\right) + \gamma_{\parallel}}{2\pi\eta L}. \tag{B.1}$$

The bead's movement is characterised by its mobility which is

$$\mu_b = \frac{1}{6\pi\eta r_b}, \tag{B.2}$$

from Stokes' Law.

The filament is also polymerising at the barbed end (the end in contact with the surface of the bead).

If we define the distance between the pointed end of the filament and the centre of the bead as the 'separation distance', we can look at the 'speed of separation' which is just how this increases in time as the filament polymerises. We look at this in three different scenarios: the filament undergoes Brownian motion but the bead is fixed, the bead undergoes Brownian motion but the filament is fixed and both objects undergo Brownian motion.

In section 1.3.2 we covered the velocity of a barrier or obstacle that has an external force acting against the polymerising filament, this was given in equation 1.4 and is repeated here

$$v_{\text{bar}} = l_{\text{mon}}\left(k_p C_G e^{\frac{-f_{\text{bar}} l_{\text{mon}}}{k_B T}} - k_d\right). \tag{B.3}$$

We can predict the 'speed of separation' by realising that the actin polymerisation must overcome the viscous drag force of the obstacle. The viscous drag force is

$$f_{\text{drag}} = \frac{v}{\mu}, \tag{B.4}$$

where v is the velocity of the object moving through the medium and μ is the object's mobility. So we have

$$v_{\text{sep}} = l_{\text{mon}} \left(k_p C_G e^{\frac{-v_{\text{sep}} l_{\text{mon}}}{\mu k_B T}} - k_d \right). \tag{B.5}$$

We can use the Taylor expansion on the exponential term giving

$$e^{\frac{-v_{\text{sep}} l_{\text{mon}}}{\mu k_B T}} \approx 1 - \frac{v_{\text{sep}} l_{\text{mon}}}{\mu k_B T}. \tag{B.6}$$

This assumes that $\frac{v_{\text{sep}} l_{\text{mon}}}{\mu k_B T}$ is small, allowing us to neglect higher order terms. Using values typical for our simulations gives this to be of order 10^{-3} to 10^{-1}. Rearranging for velocity gives

$$v_{\text{sep}} \approx \frac{l_{\text{mon}} \left(k_p C_G - k_d \right)}{1 + \frac{l_{\text{mon}}^2 k_p C_G}{\mu k_B T}}. \tag{B.7}$$

Now for the first scenario where the filament can move but the bead is immobile, μ in equation B.4 is $\mu_{\parallel}$. One key thing to note here is that because $\mu_{\parallel}$ depends on the filament's length, and the filament is growing, v_{sep} is not constant in time. For the second scenario μ is substituted with μ_b. For the final scenario μ is substituted with $\mu_{\parallel} + \mu_b$. We simply add the mobilities which increases the separation speed, as both are now undergoing fluctuations.

B.2 Simulating the system and comparing simulation to analytical calculation

We ran simple simulations to validate our model against equation B.7. In the simulations we have a single filament of initial length 999 nm (370 monomers) and a bead. The barbed end of the filament is in contact with the surface of the bead. An initial frame showing the setup is shown in Figure B.1.

We reduce the system to one dimension by only allowing motion in the vertical dimension. The subunit length of the filament is chosen to be long so that it is a rigid filament. The filament can polymerise and depolymerise at its barbed end. Simulations were ran for a runtime of sixty seconds and a given simulation takes only a few seconds of wall time to complete.

From a given simulation we can plot the separation distance against time and compare with the analytical prediction made in the previous section. A simulation was run for each of our three scenarios with a barbed end polymerisation rate of $10\mu\text{M}^{-1}\,\text{s}^{-1}$, G-actin concentration of $1\mu\text{M}$, a barbed end depolymerisation rate of 1s^{-1}, a viscosity of 0.1Pa s and a temperature of 300K. The separation distance against time from the simulation is plotted in Figure B.2 as a blue line. The analytical prediction from equation B.7 is shown as the orange line. Both have been 'zeroed' at the initial time point by subtracting the initial separation distance. The value of L used in the analytical prediction was the filament's starting length and this remained constant in time.

Figure B.2 appears to show that the analytical predictions are underestimating the separation distance in scenarios where the filament is able to move. However this discrepancy is due to large Brownian fluctuations due to the high mobility of the filament. The filament cannot move upwards by a large amount - since it would intersect the bead, but it can move downwards. To investigate this we ran the same

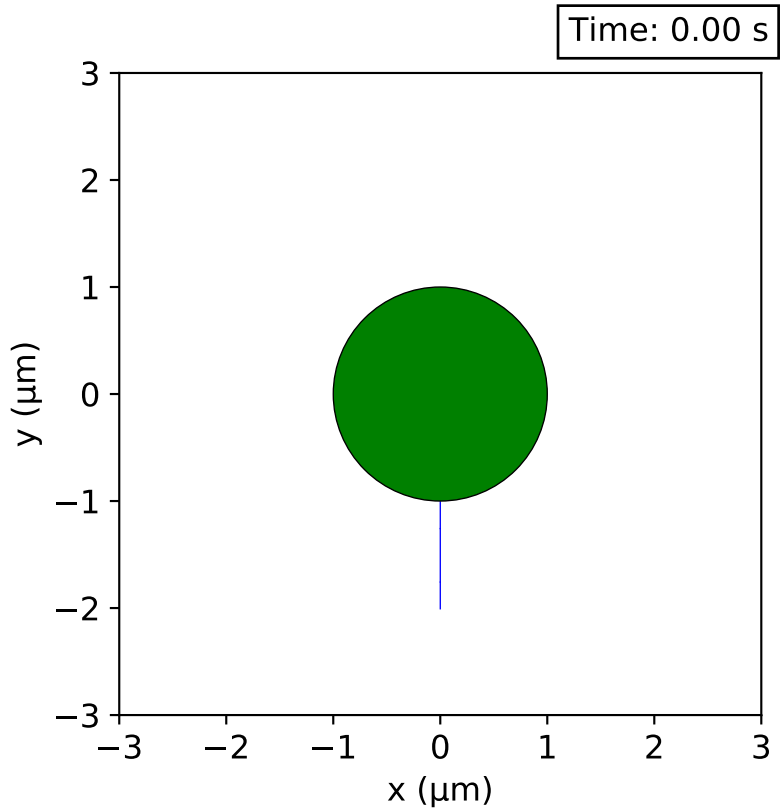


FIGURE B.1: Initial frame of an example of our simulation setup. The filament's barbed end is in contact with the surface of the bead.

simulations but with a viscosity value ten times higher. The results of this is shown in Figure B.3.

In Figure B.3 we see the fluctuations away from the prediction to be much smaller than in Figure B.2.

In both Figure B.2 and Figure B.3 the velocity predictions (orange lines) are very similar across the three different scenarios. The difference between each scenario in equation B.7 is in the value of μ. The second term in the denominator is small and changing μ only has a small effect. The analytical predicted velocity values in Figure B.3 are $24.2\text{nm}\,\text{s}^{-1}$, $23.5\text{nm}\,\text{s}^{-1}$ and $24.2\text{nm}\,\text{s}^{-1}$ for scenarios A, B and C respectively. The raw polymerisation velocity ($l_{\text{mon}}\,(k_p C_G - k_d)$) is $24.3\text{nm}\,\text{s}^{-1}$. For the higher viscosity cases shown in Figure B.3, the velocity predictions are $23.8\text{nm}\,\text{s}^{-1}$, $18.2\text{nm}\,\text{s}^{-1}$ and $23.8\text{nm}\,\text{s}^{-1}$ for scenarios A, B and C respectively.

We then looked across multiple simulations, plotting the average separation velocity against a parameter we vary. The average separation velocity is the separation distance at the end of the simulation minus the separation distance at the start, divided by the runtime of the simulation. We have two sets of simulations which each vary a single parameter. The two parameters we vary are the radius of the bead and the polymerisation rate. For each parameter value we ran three different simulations and plotted the mean and standard deviation as error bars. With the simulations varying bead radius, we used five different values: 500nm, 1µm, 2µm, 5µm and 10 µm. All other parameter values were the same as in the simulations shown in Figure B.2. The results of the simulations and the analytical predictions from equation B.7 are given in Figure B.4.

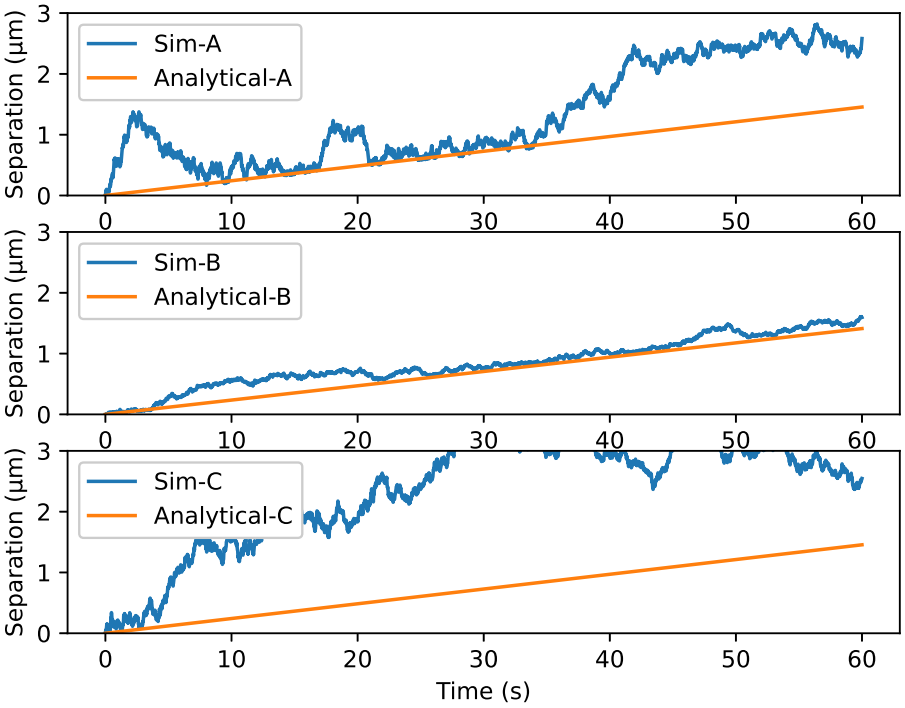


FIGURE B.2: Separation distance between the pointed end of the filament and the centre of the bead against time. The separation distance has been 'zeroed' at time zero by subtracting the initial separation distance. Three simulations are shown for each of the three different scenarios. Top is A: The filament is free to move but the bead is fixed. Middle is B: The bead is free to move but the filament is fixed. Bottom is C: Both bead and filament are free to move. Simulation results are shown as the blue lines, our analytical predictions are shown as the orange lines.

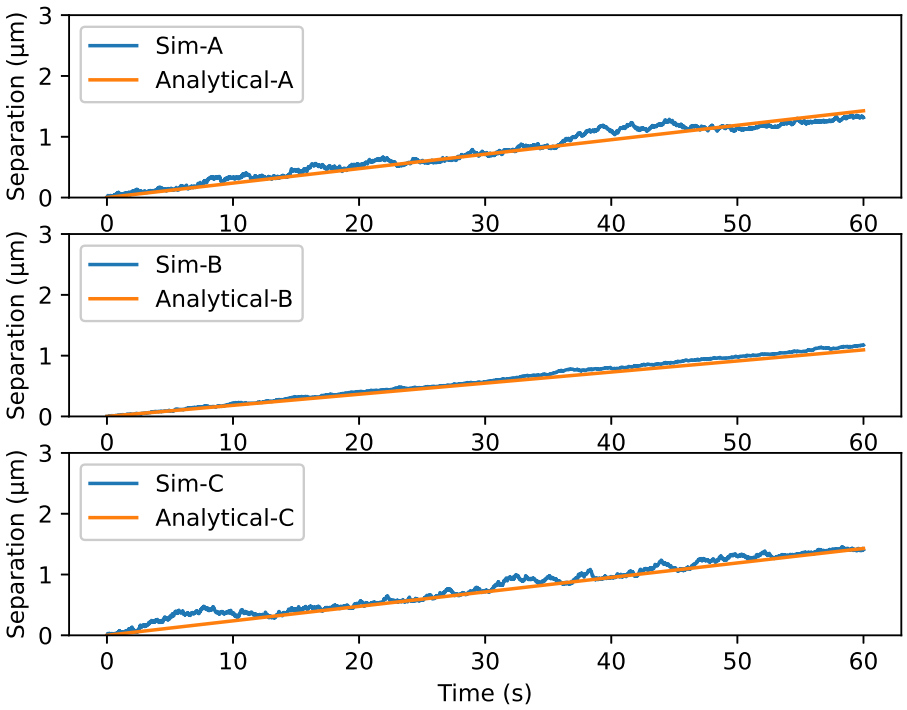
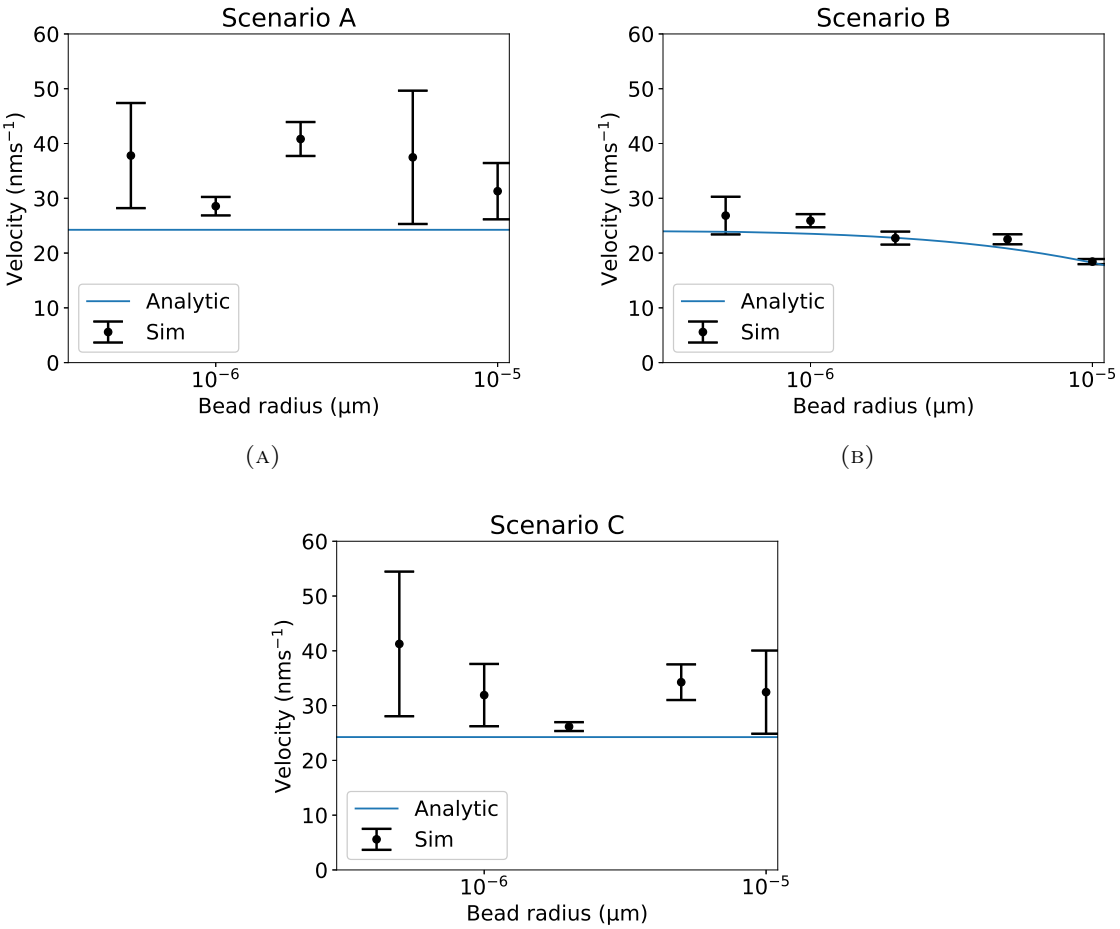


FIGURE B.3: Separation distance between the pointed end of the filament and the centre of the bead against time. The separation distance has been 'zeroed' at time zero by subtracting the initial separation distance. Three simulations are shown for each of the three different scenarios. Top is A: The filament is free to move but the bead is fixed. Middle is B: The bead is free to move but the filament is fixed. Bottom is C: Both bead and filament are free to move. Simulation results are shown as the blue lines, our analytical predictions are shown as the orange lines. Different from Figure B.2 only in that the viscosity is ten times higher here.



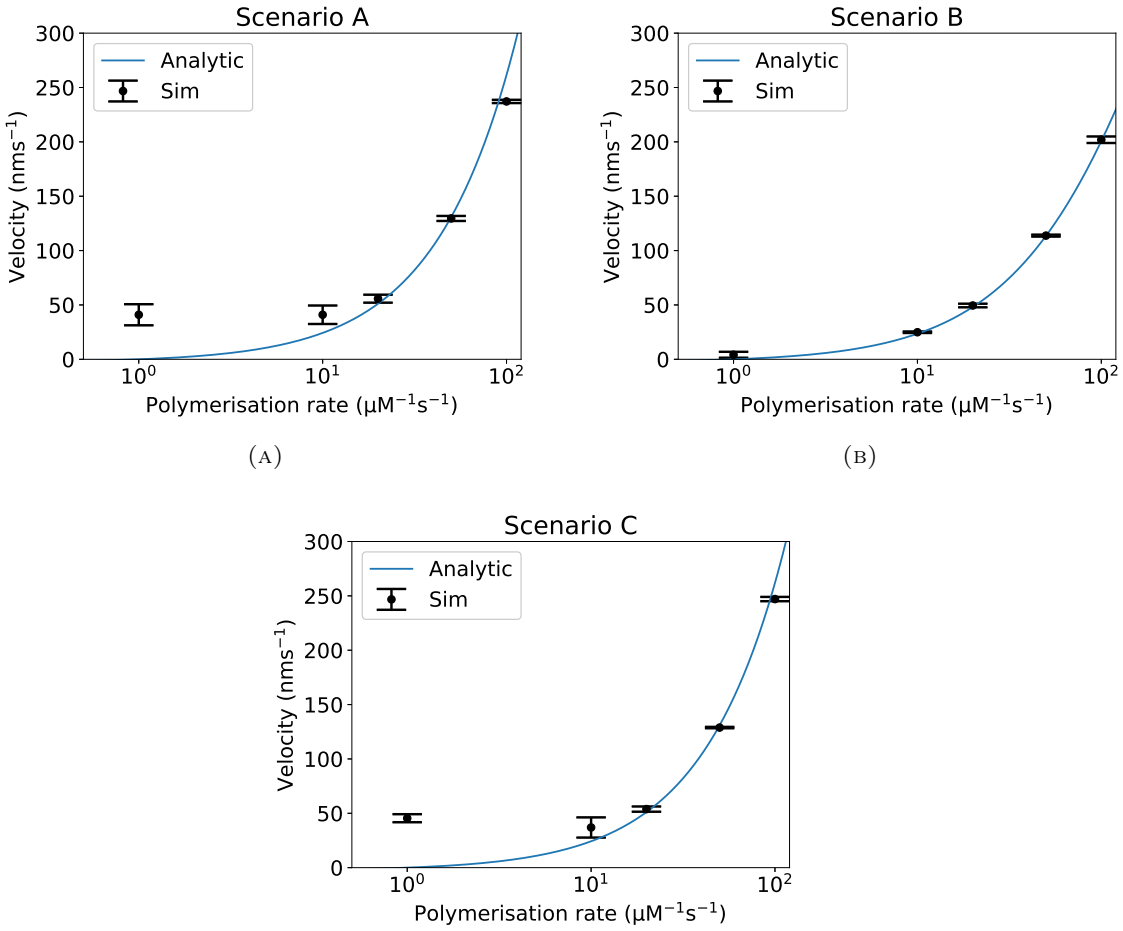
(A) (B)

(C)

FIGURE B.4: Bead radius along the x axis (µm) against average separation velocity along the y axis (nm s^{-1}). Each data point shows the average across three simulations, the error bars show the standard deviation. The blue line is our analytical prediction, equation B.7. **(A)** Scenario A. **(B)** Scenario B. **(C)** Scenario C.

Scenario B shows good agreement between simulations and the analytical prediction. Scenarios A and B agree in that there seems to be no dependence in the simulations of the separation velocity and the bead size. However the simulations show velocities higher than the analytical predictions, which we again attribute to large Brownian fluctuations due to the high mobility of the filament.

For polymerisation rate, we used five different values: $1\mu M^{-1}\,s^{-1}$, $10\mu M^{-1}\,s^{-1}$, $20\mu M^{-1}\,s^{-1}$, $50\mu M^{-1}\,s^{-1}$ and $100\mu M^{-1}\,s^{-1}$. All other parameter values were the same as in the simulations shown in Figure B.2. The results of the simulations and the analytical predictions from equation B.7 are given in Figure B.5.



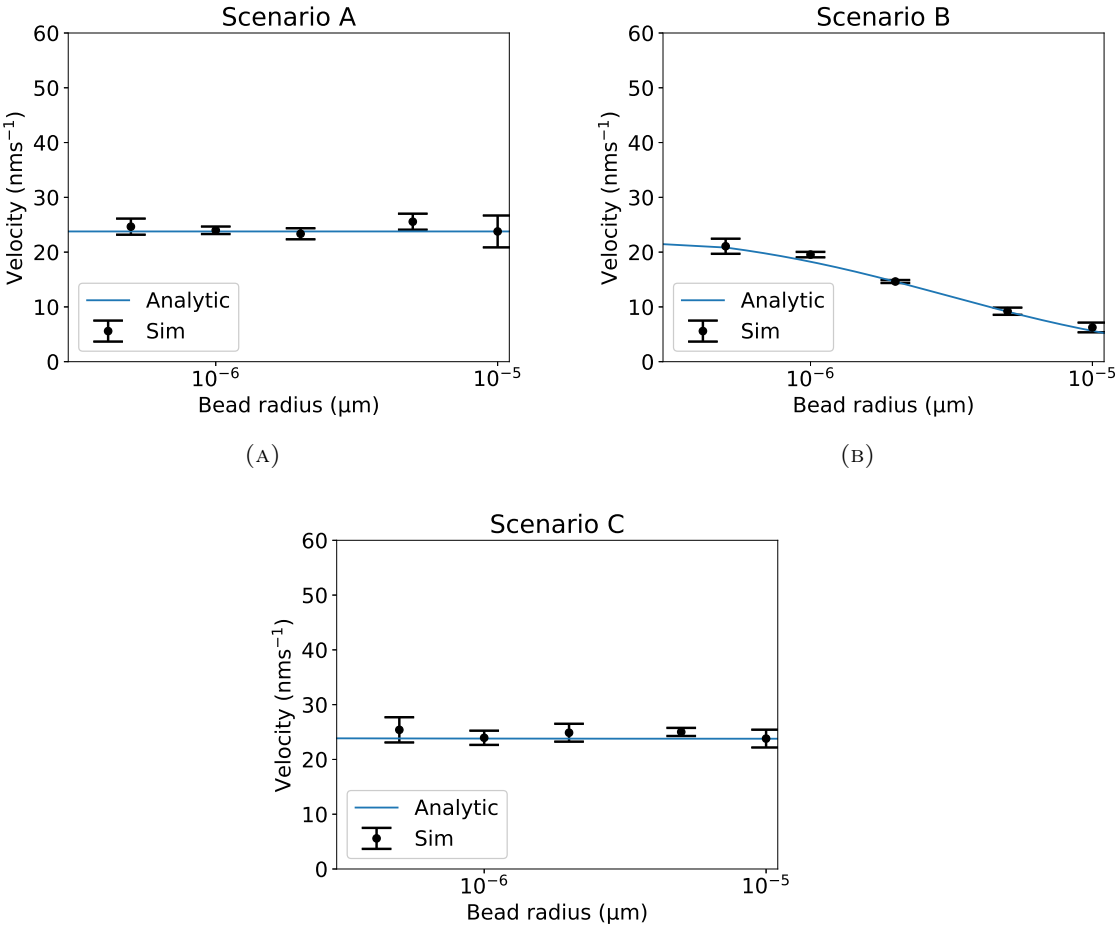
(A) (B)

(C)

FIGURE B.5: Barbed end polymerisation rate x axis ($\mu M^{-1}\,s^{-1}$) against average separation velocity along the y axis ($nm\,s^{-1}$). Each data point shows the average across three simulations, the error bars show the standard deviation. The blue line is our analytical prediction, equation B.7. **(A)** Scenario A. **(B)** Scenario B. **(C)** Scenario C.

Again, we see good agreement in scenario B between simulations and prediction. In scenarios A and C we see good agreement at the higher three polymerisation rates. The simulations using the lower two polymerisation rates again show velocities of separation higher than predicted by equation B.7.

To again check the higher velocities in the simulations than the predictions, we repeated both sets of simulations at the higher viscosity of 1Pa s. Results of the higher viscosity simulations together with the analytical predictions are given in Figures B.6 and B.7.

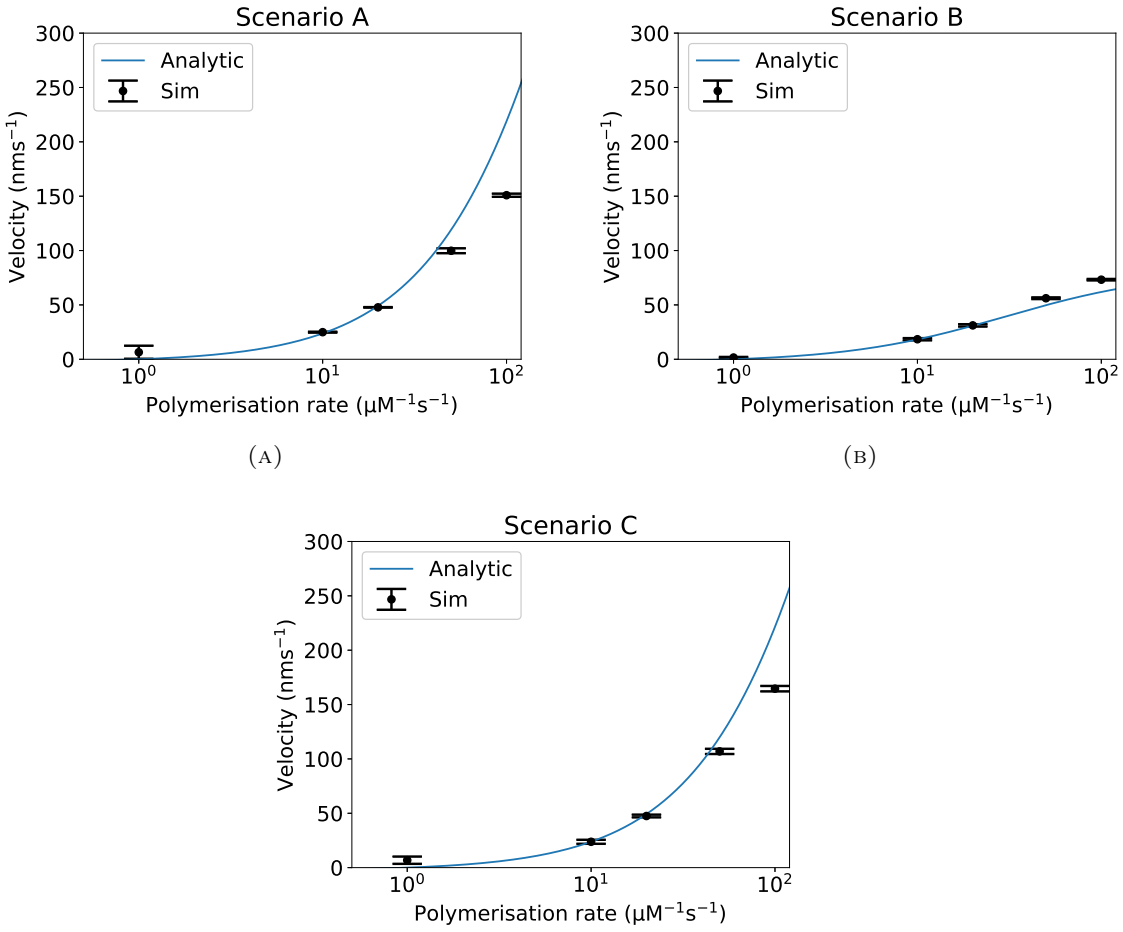


(A) (B)

(C)

FIGURE B.6: Bead radius along the x axis (µm) against average separation velocity along the y axis (nm s^{-1}). Each data point shows the average across three simulations, the error bars show the standard deviation. The blue line is our analytical prediction, equation B.7. **(A)** Scenario A. **(B)** Scenario B. **(C)** Scenario C. Viscosity is ten times higher than results shown in Figure B.4

In Figure B.6, we see a better agreement between the analytical prediction and simulations for scenarios A and C than in Figure B.4.



(A) (B) (C)

FIGURE B.7: Barbed end polymerisation rate x axis ($\mu M^{-1}\,s^{-1}$) against average separation velocity along the y axis ($nm\,s^{-1}$). Each data point shows the average across three simulations, the error bars show the standard deviation. The blue line is our analytical prediction, equation B.7. **(A)** Scenario A. **(B)** Scenario B. **(C)** Scenario C. Viscosity is ten times higher than results shown in Figure B.5

In Figure B.7, we see a better agreement between the analytical prediction and simulations for the lowest two polymerisation rates for scenarios A and C than in Figure B.5. However, we see the simulations and analytical predictions showing poorer agreement with the highest polymerisation rate. This could be due to the analytical prediction using a constant mobility calculated from the starting length of 999nm. This therefore does not account for the growth due to polymerisation reducing the mobility of the filament. This effect would be greater with higher polymerisation rates, meaning the analytical prediction would overestimate the separation velocity. We do perhaps see this as well in Figure B.5 scenarios A and C.

B.3 Buckling

In section 1.3.2 we showed that the stall force of an actin filament is

$$f_{\text{stall}} = \frac{k_B T}{l_{\text{mon}}} \ln\left(\frac{k_p C_G}{k_d}\right). \tag{B.8}$$

If we consider a macroscopic beam of length L under compression there exists a transition from when it is simply compressed as an elastic material to when it buckles. This transition happens at a critical compressive force. If the force is above the transition the beam will bend or buckle due to the energy cost to bend being less than the energy cost to compress. This critical threshold force is

$$f_{\text{buck}} = \frac{\pi^2 \kappa_b}{L^2} = \frac{\pi^2 l_P k_B T}{L^2}. \tag{B.9}$$

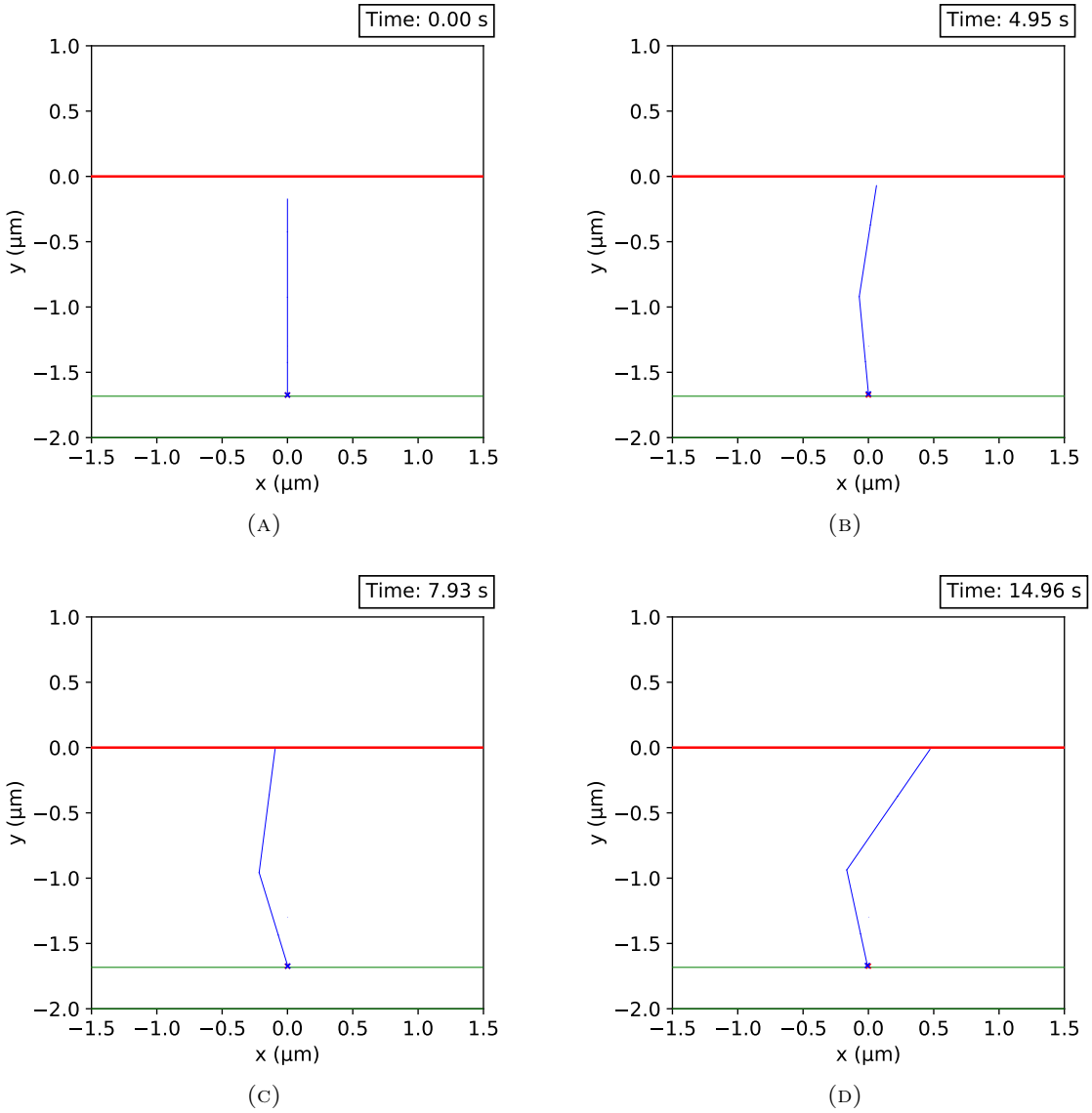
Which is sometimes known as Euler's critical load [21]. We can look at this from the perspective of a critical length rather than a critical force. When considering actin, a filament polymerising against an obstacle will buckle when it is longer than a critical length. We can find this length by setting $f_{\text{stall}} = f_{\text{buck}}$, and rearranging for the filament's length. This gives

$$L_{\text{buck}} = \sqrt{\frac{\pi^2 l_P l_{\text{mon}}}{\ln\left(\frac{k_p C_G}{k_d}\right)}}. \tag{B.10}$$

Using reasonable numbers for actin, L_{buck} tends to be around 300-500nm. Therefore we should expect filaments with lengths greater than 500nm to buckle when they are polymerising against an obstacle.

In our simulations we typically use values for the segment size of 500nm. We argue that this determines the buckling length threshold in the simulations. A single segment is rigid and therefore the filament cannot buckle on a length scale smaller than the segment length. The filaments are also incompressible, meaning any compressive force acting down along the long axis of the filament will result in buckling. There is no transition from compression to buckling, as the filaments cannot compress. We do see buckling as filaments reach an obstacle, but only if the filament is flexible. To demonstrate this, we setup a simulation with a single filament growing against a wall. The filament is tethered at its pointed end, to a fixed rectangular exclusion zone. The wall has an external force of 3.53pN, which is the stall force of a single filament (given the (de)polymerisation conditions in the simulation). The filament is initialised with a length of 1501.2nm (556 monomers) and its segment size is 500nm - meaning it is flexible and can bend. Snapshots of the simulation are shown in Figure B.8.

As the filament reaches the wall it does appear to buckle. The filament cannot compress and continues to polymerise. All the polymerisation force is therefore going into bending the filament, since it is too weak to push the wall upwards.



(A) (B)

(C) (D)

FIGURE B.8: Snapshots of a simulation of a single filament growing against a wall. Time is shown in the top right: **(A)** the initial frame at 0 seconds. **(B)** The filament approaches the wall at 5 seconds. **(C)** The filament 'buckles' or bends at 8 seconds, as it cannot produce the force required to push the wall up but continues to elongate. **(D)** 15 seconds.

References

[1] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts and J. Watson. *Molecular Biology of the Cell*, 4th ed. (Garland Publishing Inc., New York/London, 2002).

[2] J. Howard et al. 'Mechanics of motor proteins and the cytoskeleton', (2001).

[3] A. Ott, M. Magnasco, A. Simon and A. Libchaber. 'Measurement of the persistence length of polymerized actin using fluorescence microscopy', Physical Review E **48**, R1642 (1993).

[4] R. A. Jones. *Soft condensed matter*, Vol. 6 (Oxford University Press, 2002).

[5] C. P. Broedersz and F. C. MacKintosh. 'Modeling semiflexible polymer networks', Reviews of Modern Physics **86**, 995 (2014).

[6] T. D. Pollard. 'Rate constants for the reactions of ATP-and ADP-actin with the ends of actin filaments.', The Journal of cell biology **103**, 2747 (1986).

[7] T. D. Pollard. 'Actin and actin-binding proteins', Cold Spring Harbor perspectives in biology **8**, a018226 (2016).

[8] R. Dominguez and K. C. Holmes. 'Actin structure and function', (2011).

[9] S. J. Winder and K. R. Ayscough. 'Actin-binding proteins', Journal of cell science **118**, 651 (2005).

[10] R. D. Mullins, J. A. Heuser and T. D. Pollard. 'The interaction of Arp2/3 complex with actin: nucleation, high affinity pointed end capping, and formation of branching networks of filaments', Proceedings of the National Academy of Sciences **95**, 6181 (1998).

[11] S. B. Padrick, L. K. Doolittle, C. A. Brautigam, D. S. King and M. K. Rosen. 'Arp2/3 complex is bound and activated by two WASP proteins', Proceedings of the National Academy of Sciences **108**, E472 (2011).

[12] B. A. Smith, K. Daugherty-Clarke, B. L. Goode and J. Gelles. 'Pathway of actin filament branch formation by Arp2/3 complex revealed by single-molecule imaging', Proceedings of the National Academy of Sciences **110**, 1285 (2013).

[13] K. J. Amann and T. D. Pollard. 'The Arp2/3 complex nucleates actin filament branches from the sides of pre-existing filaments', Nature cell biology **3**, 306 (2001).

[14] N. Volkmann, K. J. Amann, S. Stoilova-McPhie, C. Egile, D. C. Winter, L. Hazelwood, J. E. Heuser, R. Li, T. D. Pollard and D. Hanein. 'Structure of Arp2/3 complex in its activated state and in actin filament branch junctions', science **293**, 2456 (2001).

[15] L. Blanchoin, T. D. Pollard and R. D. Mullins. 'Interactions of ADF/cofilin, Arp2/3 complex, capping protein and profilin in remodeling of branched actin filament networks', Current Biology **10**, 1273 (2000).

[16] C. Chan, C. C. Beltzner and T. D. Pollard. 'Cofilin dissociates Arp2/3 complex and branches from actin filaments', Current Biology **19**, 537 (2009).

[17] A.-C. Reymann, C. Suarez, C. Guérin, J.-L. Martiel, C. J. Staiger, L. Blanchoin and R. Boujemaa-Paterski. 'Turnover of branched actin filament networks by stochastic fragmentation with ADF/cofilin', Molecular biology of the cell **22**, 2541 (2011).

[18] B. R. McCullough, L. Blanchoin, J.-L. Martiel and M. Enrique. 'Cofilin increases the bending flexibility of actin filaments: implications for severing and cell mechanics', Journal of molecular biology **381**, 550 (2008).

[19] C. Suarez, J. Roland, R. Boujemaa-Paterski, H. Kang, B. R. McCullough, A.-C. Reymann, C. Guérin, J.-L. Martiel, M. Enrique and L. Blanchoin. 'Cofilin tunes the nucleotide state of actin filaments and severs at bare and decorated segment boundaries', Current Biology **21**, 862 (2011).

[20] J. R. Bartles. 'Parallel actin bundles and their multiple actin-bundling proteins', Current opinion in cell biology **12**, 72 (2000).

[21] L. Blanchoin, R. Boujemaa-Paterski, C. Sykes and J. Plastino. 'Actin dynamics, architecture, and mechanics in cell motility', Physiological reviews **94**, 235 (2014).

[22] C. Yang and T. Svitkina. 'Visualizing branched actin filaments in lamellipodia by electron tomography', Nature cell biology **13**, 1012 (2011).

[23] M. Vinzenz, M. Nemethova, F. Schur, J. Mueller, A. Narita, E. Urban, C. Winkler, C. Schmeiser, S. A. Koestler, K. Rottner et al. 'Actin branching in the initiation and maintenance of lamellipodia', J Cell Sci, jcs (2012).

[24] A. Mogilner and B. Rubinstein. 'The physics of filopodial protrusion', Biophysical journal **89**, 782 (2005).

[25] P. Chugh and E. K. Paluch. 'The actin cortex at a glance', Journal of cell science **131**, jcs186254 (2018).

[26] M. Fritzsche, A. Lewalle, T. Duke, K. Kruse and G. Charras. 'Analysis of turnover dynamics of the submembranous actin cortex', Molecular biology of the cell **24**, 757 (2013).

[27] C. A. Janeway Jr. *Immunobiology* (2020).

[28] R. C. May and L. M. Machesky. 'Phagocytosis and the actin cytoskeleton', Journal of cell science **114**, 1061 (2001).

[29] R. S. Flannagan, V. Jaumouillé and S. Grinstein. 'The cell biology of phagocytosis', Annual Review of Pathology: Mechanisms of Disease **7**, 61 (2012).

[30] S. A. Freeman and S. Grinstein. 'Phagocytosis: receptors, signal integration, and the cytoskeleton', Immunological reviews **262**, 193 (2014).

[31] R. J. Botelho and S. Grinstein. 'Phagocytosis', Current Biology **21**, R533 (2011).

[32] G. Kaplan. 'Differences in the mode of phagocytosis with Fc and C3 receptors in macrophages', Scandinavian journal of immunology **6**, 797 (1977).

[33] S. Tsuboi and J. Meerloo. 'Wiskott-Aldrich syndrome protein is a key regulator of the phagocytic cup formation in macrophages', Journal of Biological Chemistry **282**, 34194 (2007).

[34] D. M. Richards and R. G. Endres. 'Target shape dependence in a simple model of receptor-mediated endocytosis and phagocytosis', Proceedings of the National Academy of Sciences **113**, 6113 (2016).

[35] A. Aderem and D. M. Underhill. 'Mechanisms of phagocytosis in macrophages', Annual review of immunology **17**, 593 (1999).

[36] C. Rosales and E. Uribe-Querol. 'Phagocytosis: a fundamental process in immunity', BioMed research international **2017**.

[37] P. Rougerie, V. Miskolci and D. Cox. 'Generation of membrane structures during phagocytosis and chemotaxis of macrophages: role and regulation of the actin cytoskeleton', Immunological reviews **256**, 222 (2013).

[38] A. D. Hoppe and J. A. Swanson. 'Cdc42, Rac1, and Rac2 display distinct patterns of activation during phagocytosis', Molecular biology of the cell **15**, 3509 (2004).

[39] R. C. May, E. Caron, A. Hall and L. M. Machesky. 'Involvement of the Arp2/3 complex in phagocytosis mediated by FcγR or CR3', Nature cell biology **2**, 246 (2000).

[40] V. Jaumouillé, A. X. Cartagena-Rivera and C. M. Waterman. 'Coupling of β 2 integrins to actin by a mechanosensitive molecular clutch drives complement receptor-mediated phagocytosis', Nature Cell Biology **21**, 1357 (2019).

[41] M. Krause, E. W. Dent, J. E. Bear, J. J. Loureiro and F. B. Gertler. 'Ena/VASP proteins: regulators of the actin cytoskeleton and cell migration', Annual review of cell and developmental biology **19**, 541 (2003).

[42] J. A. Cooper and D. Sept. 'New insights into mechanism and regulation of actin capping protein', International review of cell and molecular biology **267**, 183 (2008).

[43] J. E. Bear and F. B. Gertler. 'Ena/VASP: towards resolving a pointed controversy at the barbed end', Journal of cell science **122**, 1947 (2009).

[44] F. Castellano, C. Le Clainche, D. Patin, M.-F. Carlier and P. Chavrier. 'A WASp–VASP complex regulates actin polymerization at the plasma membrane', The EMBO journal **20**, 5603 (2001).

[45] M. G. Coppolino, M. Krause, P. Hagendorff, D. A. Monner, W. Trimble, S. Grinstein, J. Wehland and A. S. Sechi. 'Evidence for a molecular complex consisting of Fyb/SLAP, SLP-76, Nck, VASP and WASP that links the actin cytoskeleton to Fcγ receptor signalling during phagocytosis', Journal of cell science **114**, 4307 (2001).

[46] D. T. Brandt, S. Marion, G. Griffiths, T. Watanabe, K. Kaibuchi and R. Grosse. 'Dia1 and IQGAP1 interact in cell migration and phagocytic cup formation', The Journal of cell biology **178**, 193 (2007).

[47] E. Colucci-Guyon, F. Niedergang, B. J. Wallar, J. Peng, A. S. Alberts and P. Chavrier. 'A role for mammalian diaphanous-related formins in complement receptor (CR3)-mediated phagocytosis in macrophages', Current biology **15**, 2007 (2005).

[48] V. Jaumouillé and C. M. Waterman. 'Physical Constraints and Forces Involved in Phagocytosis', Frontiers in Immunology **11**, 1097 (2020).

[49] R. J. Botelho, M. Teruel, R. Dierckman, R. Anderson, A. Wells, J. D. York, T. Meyer and S. Grinstein. 'Localized biphasic changes in phosphatidylinositol-4, 5-bisphosphate at sites of phagocytosis', Journal of Cell Biology **151**, 1353 (2000).

[50] O. Kratky and G. Porod. 'Röntgenuntersuchung gelöster fadenmoleküle', Recueil des Travaux Chimiques des Pays-Bas **68**, 1106 (1949).

[51] R. Erban. 'From molecular dynamics to Brownian dynamics', Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **470**, 20140036 (2014).

[52] E. M. Purcell. 'Life at low Reynolds number', American journal of physics **45**, 3 (1977).

[53] A. Mogilner and G. Oster. 'Cell motility driven by actin polymerization', Biophysical journal **71**, 3030 (1996).

[54] A. Mogilner and G. Oster. 'Force generation by actin polymerization II: the elastic ratchet and tethered filaments', Biophysical journal **84**, 1591 (2003).

[55] C. S. Peskin, G. M. Odell and G. F. Oster. 'Cellular motions and thermal fluctuations: the Brownian ratchet', Biophysical journal **65**, 316 (1993).

[56] S. Dmitrieff and F. Nédélec. 'Amplification of actin polymerization forces', The Journal of cell biology **212**, 763 (2016).

[57] S. L. Freedman, S. Banerjee, G. M. Hocky and A. R. Dinner. 'A versatile framework for simulating the dynamic mechanical structure of cytoskeletal networks', Biophysical journal **113**, 448 (2017).

[58] S. S. Andrews. 'Methods for modeling cytoskeletal and DNA filaments', Physical biology **11**, 011001 (2014).

[59] F. Nedelec and D. Foethke. 'Collective Langevin dynamics of flexible cytoskeletal fibers', New Journal of Physics **9**, 427 (2007).

[60] G. Letort, A. Z. Politi, H. Ennomani, M. Théry, F. Nedelec and L. Blanchoin. 'Geometrical and mechanical properties control actin filament organization', PLoS computational biology **11**, e1004245 (2015).

[61] M. Mund, J. A. van der Beek, J. Deschamps, S. Dmitrieff, J. L. Monster, A. Picco, F. Nédélec, M. Kaksonen and J. Ries. 'Systematic Nanoscale Analysis of Endocytosis Links Efficient Vesicle Formation to Patterned Actin Nucleation', (2018).

[62] J. M. Belmonte, M. Leptin and F. Nédélec. 'A theory that predicts behaviors of disordered cytoskeletal networks', Molecular Systems Biology **13**, 941 (2017).

[63] P. Bun, S. Dmitrieff, J. M. Belmonte, F. J. Nédélec and P. Lénárt. 'A disassembly-driven mechanism explains F-actin-mediated chromosome transport in starfish oocytes', Elife **7**, e31469 (2018).

[64] N. Burroughs and D. Marenduzzo. 'Three-dimensional dynamic Monte Carlo simulations of elastic actin-like ratchets', The Journal of chemical physics **123**, 174908 (2005).

[65] A. E. Carlsson. 'Growth of branched actin networks against obstacles', Biophysical journal **81**, 1907 (2001).

[66] D. K. Hansda, S. Sen and R. Padinhateeri. 'Branching influences force-velocity curves and length fluctuations in actin networks', Physical Review E **90**, 062718 (2014).

[67] T. E. Schaus, E. W. Taylor and G. G. Borisy. 'Self-organization of actin filament orientation in the dendritic-nucleation/array-treadmilling model', Proceedings of the National Academy of Sciences **104**, 7086 (2007).

[68] C. H. Schreiber, M. Stewart and T. Duke. 'Simulation of cell motility that reproduces the force–velocity relationship', Proceedings of the National Academy of Sciences **107**, 9141 (2010).

[69] H. Gao, W. Shi and L. B. Freund. 'Mechanics of receptor-mediated endocytosis', Proceedings of the National Academy of Sciences **102**, 9469 (2005).

[70] D. M. Richards and R. G. Endres. 'The mechanism of phagocytosis: two stages of engulfment', Biophysical journal **107**, 1542 (2014).

[71] J. S. Van Zon, G. Tzircotis, E. Caron and M. Howard. 'A mechanical bottleneck explains the variation in cup growth during FcγR phagocytosis', Molecular systems biology **5**, 298 (2009).

[72] M. Herant, V. Heinrich and M. Dembo. 'Mechanics of neutrophil phagocytosis: experiments and quantitative models', Journal of Cell Science **119**, 1903 (2006).

[73] M. Herant, C.-Y. Lee, M. Dembo and V. Heinrich. 'Protrusive push versus enveloping embrace: Computational model of phagocytosis predicts key regulatory role of cytoskeletal membrane anchors', PLoS Comput Biol **7**, e1001068 (2011).

[74] W. Helfrich. 'Elastic properties of lipid bilayers: theory and possible experiments', Zeitschrift für Naturforschung C **28**, 693 (1973).

[75] M. Deserno. 'Fluid lipid membranes–a primer',

[76] S. Tollis, A. E. Dart, G. Tzircotis and R. G. Endres. 'The zipper mechanism in phagocytosis: energetic requirements and variability in phagocytic cup shape', BMC systems biology **4**, 1 (2010).

[77] P. A. Gagniuc. *Markov chains: from theory to implementation and experimentation* (John Wiley & Sons, 2017).

[78] D. Frenkel and B. Smit. *Understanding molecular simulation: from algorithms to applications*, Vol. 1 (Elsevier, 2001).

[79] K. A. Fichthorn and W. H. Weinberg. 'Theoretical foundations of dynamical Monte Carlo simulations', The Journal of chemical physics **95**, 1090 (1991).

[80] D. T. Gillespie. 'A general method for numerically simulating the stochastic time evolution of coupled chemical reactions', Journal of computational physics **22**, 403 (1976).

[81] D. T. Gillespie. 'Exact stochastic simulation of coupled chemical reactions', The journal of physical chemistry **81**, 2340 (1977).

[82] R. Erban and S. J. Chapman. *Stochastic modelling of reaction–diffusion processes*, Vol. 60 (Cambridge University Press, 2019).

[83] J. L. Klepeis, K. Lindorff-Larsen, R. O. Dror and D. E. Shaw. 'Long-timescale molecular dynamics simulations of protein structure and function', Current opinion in structural biology **19**, 120 (2009).

[84] *Boost C++ Libraries*, `https://www.boost.org/`.

[85] *Geometric Tools*, `https://www.geometrictools.com/`.

[86] G. Guennebaud, B. Jacob et al. *Eigen v3*, http://eigen.tuxfamily.org, 2010.

[87] M. Matsumoto and T. Nishimura. 'Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator', ACM Transactions on Modeling and Computer Simulation (TOMACS) **8**, 3 (1998).

[88] M. Doi and S. F. Edwards. *The theory of polymer dynamics*, Vol. 73 (Oxford University Press, 1988).

[89] K. Yang, C. Lu, X. Zhao and R. Kawamura. 'From bead to rod: Comparison of theories by measuring translational drag coefficients of micron-sized magnetic bead-chains in Stokes flow', PloS one **12**, e0188015 (2017).

[90] H. Yamakawa and G. Tanaka. 'Translational diffusion coefficients of rodlike polymers: application of the modified Oseen tensor', The Journal of Chemical Physics **57**, 1537 (1972).

[91] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*, 6th ed. (Pearson, 2007), pp. 149–167.

[92] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery. *Numerical recipes in C: the art of scientific computing*, 2nd ed. (Cambridge University Press, 1992), pp. 692–699.

[93] F. Perrin. 'Mouvement brownien d'un ellipsoide-I. Dispersion diélectrique pour des molécules ellipsoidales', J. Phys. Radium **5**, 497 (1934).

[94] J. Wang and H. Gao. 'A generalized bead-rod model for Brownian dynamics simulations of wormlike chains under strong confinement', The Journal of chemical physics **123**, 084906 (2005).

[95] I. M. Neelov, D. B. Adolf, A. V. Lyulin and G. R. Davies. 'Brownian dynamics simulation of linear polymers under elongational flow: Bead–rod model with hydrodynamic interactions', The Journal of chemical physics **117**, 4030 (2002).

[96] A. Montesi, D. C. Morse and M. Pasquali. 'Brownian dynamics algorithm for bead-rod semiflexible chain with anisotropic friction', The Journal of chemical physics **122**, 084903 (2005).

[97] M. Somasi, B. Khomami, N. J. Woo, J. S. Hur and E. S. Shaqfeh. 'Brownian dynamics simulations of bead-rod and bead-spring chains: numerical algorithms and coarse-graining issues', Journal of non-newtonian fluid mechanics **108**, 227 (2002).

[98] T. B. Liverpool. 'Dynamics of inextensible semiflexible filaments', Physical Review E **72**, 021805 (2005).

[99] D. L. Ermak and J. A. McCammon. 'Brownian dynamics with hydrodynamic interactions', The Journal of chemical physics **69**, 1352 (1978).

[100] S. A. Allison. 'Brownian dynamics simulation of wormlike chains. Fluorescence depolarization and depolarized light scattering', Macromolecules **19**, 118 (1986).

[101] B. Hess, H. Bekker, H. J. Berendsen and J. G. Fraaije. 'LINCS: a linear constraint solver for molecular simulations', Journal of computational chemistry **18**, 1463 (1997).

[102] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark and H. J. Berendsen. 'GROMACS: fast, flexible, and free', Journal of computational chemistry **26**, 1701 (2005).

[103] J. X. Tang and P. A. Janmey. 'The polyelectrolyte nature of F-actin and the mechanism of actin bundle formation', Journal of Biological Chemistry **271**, 8556 (1996).

[104] T. Oda, K. Makino, I. Yamashita, K. Namba and Y. Maéda. 'Effect of the length and effective diameter of F-actin on the filament orientation in liquid crystalline sols measured by X-ray fiber diffraction', Biophysical journal **75**, 2672 (1998).

[105] J. E. Bresenham. 'Algorithm for computer control of a digital plotter', IBM Systems journal **4**, 25 (1965).

[106] X. Wu. 'An efficient antialiasing technique', Acm Siggraph Computer Graphics **25**, 143 (1991).

[107] E. Dedu. *Bresenham-based supercover line algorithm*, (2001) `https://web.archive.org/web/20200728183442/http://eugen.dedu.free.fr/projects/bresenham/` (visited on 28/07/2020).

[108] A. B. Verkhovsky, O. Y. Chaga, S. Schaub, T. M. Svitkina, J.-J. Meister and G. G. Borisy. 'Orientational order of the lamellipodial actin network as demonstrated in living motile cells', Molecular biology of the cell **14**, 4667 (2003).

[109] H. Tang, D. Laporte and D. Vavylonis. 'Actin cable distribution and dynamics arising from cross-linking, motor pulling, and filament turnover', Molecular biology of the cell **25**, 3006 (2014).

[110] A. Carlsson, M. Wear and J. Cooper. 'End versus side branching by Arp2/3 complex', Biophysical journal **86**, 1074 (2004).

[111] F. Brooks and A. Carlsson. 'Actin polymerization overshoots and ATP hydrolysis as assayed by pyrene fluorescence', Biophysical journal **95**, 1050 (2008).

[112] R. D. Mullins, W. F. Stafford and T. D. Pollard. 'Structure, subunit topology, and actin-binding activity of the Arp2/3 complex from Acanthamoeba', The Journal of cell biology **136**, 331 (1997).

[113] V. I. Risca, E. B. Wang, O. Chaudhuri, J. J. Chia, P. L. Geissler and D. A. Fletcher. 'Actin filament curvature biases branching direction', Proceedings of the National Academy of Sciences **109**, 2913 (2012).

[114] D. A. Schafer, P. B. Jennings and J. A. Cooper. 'Dynamics of capping protein and actin assembly in vitro: uncapping barbed ends by polyphosphoinositides', Journal of Cell Biology **135**, 169 (1996).

[115] Q. Chen and T. D. Pollard. 'Actin filament severing by cofilin is more important for assembly than constriction of the cytokinetic contractile ring', J Cell Biol **195**, 485 (2011).

[116] E. Andrianantoandro and T. D. Pollard. 'Mechanism of actin filament turnover by severing and nucleation at different concentrations of ADF/cofilin', Molecular cell **24**, 13 (2006).

[117] G. V. Bossa, T. P. de Souza and S. May. 'Adhesion of like-charged lipid vesicles induced by rod-like counterions', Soft matter **14**, 3935 (2018).

[118] A. Colom, E. Derivery, S. Soleimanpour, C. Tomba, M. Dal Molin, N. Sakai, M. González-Gaitán, S. Matile and A. Roux. 'A fluorescent membrane tension probe', Nature chemistry **10**, 1118 (2018).

[119] W. Rawicz, K. C. Olbrich, T. McIntosh, D. Needham and E. Evans. 'Effect of chain length and unsaturation on elasticity of lipid bilayers', Biophysical journal **79**, 328 (2000).

[120] T. A. Masters, B. Pontes, V. Viasnoff, Y. Li and N. C. Gauthier. 'Plasma membrane tension orchestrates membrane trafficking, cytoskeletal remodeling, and biochemical signaling during phagocytosis', Proceedings of the National Academy of Sciences **110**, 11875 (2013).

[121] L. Figard and A. M. Sokac. 'A membrane reservoir at the cell surface: unfolding the plasma membrane to fuel cell shape change', Bioarchitecture **4**, 39 (2014).

[122] Z. Farkas, I. Derényi and T. Vicsek. 'DNA uptake into nuclei: Numerical and analytical results', Journal of Physics: Condensed Matter **15**, S1767 (2003).

[123] FFmpeg team. *FFmpeg, Version 3.4.8*, Champaign, IL, 2020, `https://ffmpeg.org/`.

[124] S. Ramanathan and D. C. Morse. 'Brownian dynamics algorithm for entangled wormlike threads', The Journal of chemical physics **126**, 094906 (2007).

[125] T. D. Pollard, L. Blanchoin and R. D. Mullins. 'Molecular mechanisms controlling actin filament dynamics in nonmuscle cells', Annual review of biophysics and biomolecular structure **29**, 545 (2000).

[126] A. M. Weaver, A. V. Karginov, A. W. Kinley, S. A. Weed, Y. Li, J. T. Parsons and J. A. Cooper. 'Cortactin promotes and stabilizes Arp2/3-induced actin filament network formation', Current Biology **11**, 370 (2001).

[127] M. Huang, C. Yang, D. A. Schafer, J. A. Cooper, H. N. Higgs and S. H. Zigmond. 'Cdc42-induced actin filaments are protected from capping protein', Current biology **9**, 979 (1999).

[128] D. Arcizet, B. Meier, E. Sackmann, J. O. Rädler and D. Heinrich. 'Temporal analysis of active and passive transport in living cells', Physical review letters **101**, 248103 (2008).

[129] T. Kalwarczyk, N. Ziebacz, A. Bielejewska, E. Zaboklicka, K. Koynov, J. Szymanski, A. Wilk, A. Patkowski, J. Gapinski, H.-J. Butt et al. 'Comparative analysis of viscosity of complex liquids and cytoplasm of mammalian cells at the nanoscale', Nano letters **11**, 2157 (2011).

[130] R. Dimova. 'Recent developments in the field of bending rigidity measurements on membranes', Advances in colloid and interface science **208**, 225 (2014).

[131] B. Pontes, Y. Ayala, A. C. C. Fonseca, L. F. Romão, R. F. Amaral, L. T. Salgado, F. R. Lima, M. Farina, N. B. Viana, V. Moura-Neto et al. 'Membrane elastic properties and cell function', PLoS One **8**, e67708 (2013).

[132] J. J. Tyler, E. G. Allwood and K. R. Ayscough. 'WASP family proteins, more than Arp2/3 activators', Biochemical Society Transactions **44**, 1339 (2016).

[133] D. Winter, T. Lechler and R. Li. 'Activation of the yeast Arp2/3 complex by Bee1p, a WASP-family protein', Current biology **9**, 501 (1999).

[134] A. N. Urbanek, A. P. Smith, E. G. Allwood, W. I. Booth and K. R. Ayscough. 'A novel actin-binding motif in Las17/WASP nucleates actin filaments independently of Arp2/3', Current Biology **23**, 196 (2013).

[135] E. G. Allwood, J. J. Tyler, A. N. Urbanek, I. I. Smaczynska-de Rooij and K. R. Ayscough. 'Elucidating Key Motifs Required for Arp2/3-Dependent and Independent Actin Nucleation by Las17/WASP', Plos one **11**, e0163177 (2016).

[136] J. A. Cooper, S. B. Walker and T. D. Pollard. 'Pyrene actin: documentation of the validity of a sensitive assay for actin polymerization', Journal of Muscle Research & Cell Motility **4**, 253 (1983).

[137] D. Axelrod. 'Cell-substrate contacts illuminated by total internal reflection fluorescence.', Journal of Cell Biology **89**, 141 (1981).

[138] C. J. Balzer, A. R. Wagner, L. A. Helgeson and B. J. Nolen. 'Dip1 Co-opts features of branching nucleation to create linear actin filaments that activate WASP-Bound Arp2/3 complex', Current Biology **28**, 3886 (2018).

[139] K. N. Fish. 'Total internal reflection fluorescence (TIRF) microscopy', Current protocols in cytometry **50**, 12 (2009).

[140] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid et al. 'Fiji: an open-source platform for biological-image analysis', Nature methods **9**, 676 (2012).

[141] J. D. Rotty, C. Wu and J. E. Bear. 'New insights into the regulation and cellular functions of the ARP2/3 complex', Nature reviews Molecular cell biology **14**, 7 (2013).

[142] E. G. Yarmola, T. Somasundaram, T. A. Boring, I. Spector and M. R. Bubb. 'Actin-latrunculin A structure and function differential modulation of actin-binding protein function by latrunculin A', Journal of Biological Chemistry **275**, 28120 (2000).

[143] M. B. Smith, H. Li, T. Shen, X. Huang, E. Yusuf and D. Vavylonis. 'Segmentation and tracking of cytoskeletal filaments using open active contours', Cytoskeleton **67**, 693 (2010).

[144] GraphPad Software. *GraphPad Prism, Version 8 for Windows*, San Diego, California USA, 2020, `https://www.graphpad.com`.

[145] J. R. Kuhn and T. D. Pollard. 'Real-time measurements of actin filament polymerization by total internal reflection fluorescence microscopy', Biophysical journal **88**, 1387 (2005).

[146] C. Steger. 'An unbiased detector of curvilinear structures', IEEE Transactions on pattern analysis and machine intelligence **20**, 113 (1998).

[147] OriginLab Corporation. *OriginPro, Version 2019*, Northampton, MA, USA, `https://www.originlab.com/`.

[148] E. Grazi. 'Polymerization of N-(1-pyrenyl) iodoacetamide-labelled actin: the fluorescence signal is not directly proportional to the incorporation of the monomer into the polymer', Biochemical and biophysical research communications **128**, 1058 (1985).

[149] Wolfram Research, Inc. *Mathematica, Version 12.1*, Champaign, IL, 2020, `https://www.wolfram.com/mathematica`.

[150] A. R. Wagner, Q. Luan, S.-L. Liu and B. J. Nolen. 'Dip1 defines a class of Arp2/3 complex activators that function without preformed actin filaments', Current Biology **23**, 1990 (2013).

[151] R. Arasada, W. A. Sayyad, J. Berro and T. D. Pollard. 'High-speed superresolution imaging of the proteins in fission yeast clathrin-mediated endocytic actin patches', Molecular biology of the cell **29**, 295 (2018).

[152] S. A. Freeman, J. Goyette, W. Furuya, E. C. Woods, C. R. Bertozzi, W. Bergmeier, B. Hinz, P. A. Van Der Merwe, R. Das and S. Grinstein. 'Integrins form an expanding diffusional barrier that coordinates phagocytosis', Cell **164**, 128 (2016).

[153] K. K. Huynh, J. G. Kay, J. L. Stow and S. Grinstein. 'Fusion, fission, and secretion during phagocytosis', Physiology **22**, 366 (2007).

[154] M. Samie, X. Wang, X. Zhang, A. Goschka, X. Li, X. Cheng, E. Gregg, M. Azar, Y. Zhuo, A. G. Garrity et al. 'A TRP channel in the lysosome regulates large particle phagocytosis via focal exocytosis', Developmental cell **26**, 511 (2013).

[155] J. Prinyakupt and C. Pluempitiwiriyawej. 'Segmentation of white blood cells and comparison of cell morphology by linear and naive Bayes classifiers', Biomedical engineering online **14**, 63 (2015).

[156] M. H. Bakalar, A. M. Joffe, E. M. Schmid, S. Son, M. Podolski and D. A. Fletcher. 'Size-dependent segregation controls macrophage phagocytosis of antibody-opsonized targets', Cell **174**, 131 (2018).

[157] D. R.-B. Aroush, N. Ofer, E. Abu-Shah, J. Allard, O. Krichevsky, A. Mogilner and K. Keren. 'Actin turnover in lamellipodial fragments', Current Biology **27**, 2963 (2017).

[158] D. Vorselen, R. L. D. Labitigan and J. A. Theriot. 'A mechanical perspective on phagocytic cup formation', Current Opinion in Cell Biology **66**, 112 (2020).

[159] J. Lam, M. Herant, M. Dembo and V. Heinrich. 'Baseline mechanical characterization of J774 macrophages', Biophysical journal **96**, 248 (2009).

[160] D. M. Richards and R. G. Endres. 'How cells engulf: a review of theoretical approaches to phagocytosis', Reports on Progress in Physics **80**, 126601 (2017).

[161] P. Sens and J. Plastino. 'Membrane tension and cytoskeleton organization in cell motility', Journal of Physics: Condensed Matter **27**, 273103 (2015).

[162] K. J. Chalut and E. K. Paluch. 'The actin cortex: a bridge between cell shape and function', Developmental cell **38**, 571 (2016).

[163] M. P. Clausen, H. Colin-York, F. Schneider, C. Eggeling and M. Fritzsche. 'Dissecting the actin cortex density and membrane-cortex distance in living cells by super-resolution microscopy', Journal of physics D: Applied physics **50**, 064002 (2017).

[164] R. Rajasingham, R. M. Smith, B. J. Park, J. N. Jarvis, N. P. Govender, T. M. Chiller, D. W. Denning, A. Loyse and D. R. Boulware. 'Global burden of disease of HIV-associated cryptococcal meningitis: an updated analysis', The Lancet infectious diseases **17**, 873 (2017).

[165] S. A. Johnston and R. C. May. 'Cryptococcus interactions with macrophages: evasion and manipulation of the phagosome by a fungal pathogen', Cellular microbiology **15**, 403 (2013).

[166] A. Bojarczuk, K. A. Miller, R. Hotham, A. Lewis, N. V. Ogryzko, A. A. Kamuyango, H. Frost, R. H. Gibson, E. Stillman, R. C. May et al. 'Cryptococcus neoformans intracellular proliferation and capsule size determines early macrophage control of infection', Scientific reports **6**, 21489 (2016).

[167] J. Rudman, R. J. Evans and S. A. Johnston. 'Are macrophages the heroes or villains during cryptococcosis?', Fungal Genetics and Biology **132**, 103261 (2019).

[168] J. M. Davis, M. Huang, M. R. Botts, C. M. Hull and A. Huttenlocher. 'A zebrafish model of cryptococcal infection reveals roles for macrophages, endothelial cells, and neutrophils in the establishment and control of sustained fungemia', Infection and immunity **84**, 3047 (2016).

[169] K. Voelz, S. A. Johnston, J. C. Rutherford and R. C. May. 'Automated analysis of cryptococcal macrophage parasitism using GFP-tagged cryptococci', PloS one **5**, e15968 (2010).

[170] P. Escoll and C. Buchrieser. 'Metabolic reprogramming: an innate cellular defence mechanism against intracellular bacteria?', Current Opinion in Immunology **60**, 117 (2019).

[171] S. Kissing, P. Saftig and A. Haas. 'Vacuolar ATPase in phago (lyso) some biology', International Journal of Medical Microbiology **308**, 58 (2018).

[172] S. A. Johnston, K. Voelz and R. C. May. 'Cryptococcus neoformans thermotolerance to avian body temperature is sufficient for extracellular growth but not intracellular survival in macrophages', Scientific reports **6**, 20977 (2016).

[173] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery. *Numerical recipes in C*, 1988.

[174] S. Cook. *CUDA programming: a developer's guide to parallel computing with GPUs* (Newnes, 2012).