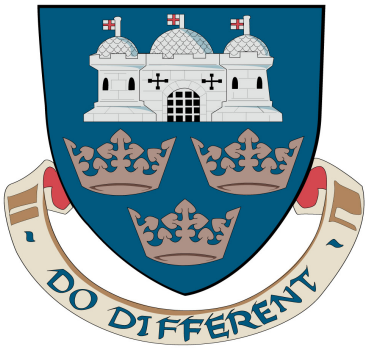
Modelling the Formation of Ordered Acentrosomal Microtubule Arrays



Alexander Ross Mace

School of Computing Sciences

University of East Anglia

A thesis submitted for the degree of

Doctor of Philosophy

April 2015

I would like to dedicate this thesis to my family and friends for their continued support, motivation and advice throughout this thesis

Acknowledgements

I would like to acknowledge my supervisors Dr Wenjia Wang and Dr Scott Grandison for their extended support throughout my PhD. I would like to thank Dr Mette Mogensen, Department of Biological Sciences, University of East Anglia, for her constructive comments and advice on MDCK cells.

Abstract

Acentrosomal microtubules are not bound to a microtubule organising centre yet are still able to form ordered arrays. Two clear examples of this behaviour are the acentrosomal apico-basal (side wall) array in epithelial cells and the parallel organisation of plant cortical microtubules. This research investigates their formation through mathematical modelling and Monte Carlo simulations with the software programs developed ourselves.

In epithelial cells there is a generally accepted 'release and capture' hypothesis for the transfer of centrosomal microtubules onto the side wall array. We use a combination of mathematical and Monte Carlo simulation models to perform the first modelling of this hypothesis. We find that a tubulin concentration dependent dynamic instability is not a good fit to this hypothesis but that a reduced centrosomal nucleation rate in response to an increased number of side wall microtubules makes the hypothesis work in biologically reasonable conditions. We propose that the loss of nucleation rate is a result of ninein being transferred from the centrosome to the side wall. We show OpenCL to be a useful tool in building a simulation program for parameter searches.

We use a Monte Carlo simulation model to investigate how the collision induced catastrophe (CIC) probability affects the formation of the ordered array of cortical plant microtubules. We find that with entrainment an ordered array stops forming once the CIC drops below 0.5. We find that the severing action of katanin is able to restore order at CIC probabilities below 0.5 but the speed at which crossovers must be severed becomes unfeasibly fast as the CIC decreases. This

implies that at very low CICs observed in nature ($\sim$0.1), katanin may be necessary but not sufficient to create the ordered array. We also provide a customisable and intuitive cortical microtubule simulation software to aid in further research.

Contents

Contents **v**

List of Figures **ix**

List of Abbreviations **xv**

1 Introduction **1**

1.1 Acentrosomal microtubules 2
1.2 Epithelial cells 2
1.3 Cortical microtubules 6
1.4 Aims 9
1.4.1 Objectives for Aim 1 9
1.4.2 Objectives for Aim 2 9
1.5 Thesis chapter summary 10

2 Literature Review: Microtubules **12**

2.1 Summary 12
2.2 Microtubule Roles 12
2.3 Microtubules and Dynamic Instability 13
2.4 Nucleation 13
2.5 Microtubule to Microtubule Interactions 15
2.6 Microtubule to Protein Interactions 16
2.7 Models of Dynamic Instability 17
2.7.1 Models of Epithelial cell microtubules 18
2.7.2 Models of Cortical Microtubules 19

3 Research Methodology **22**
3.1 Introduction . . . 22
3.2 Modelling approach . . . 22
3.2.1 MDCK microtubules . . . 23
3.2.2 Cortical microtubules . . . 24
3.3 Collision detection . . . 24
3.3.1 *A priori* . . . 25
3.3.2 *A posteriori* . . . 26
3.3.3 Parameter searches through parallelisation . . . 26
3.3.4 OpenCL versus CUDA . . . 27
3.3.5 How does OpenCL work? . . . 28
3.3.6 Optimising OpenCL . . . 29
3.3.7 OpenCL in practice . . . 30
3.3.8 OpenCL or computing cluster? . . . 31
3.4 Microtubule visualisation . . . 31

4 Software Design and Implementation **33**
4.1 Introduction . . . 33
4.2 Modelling microtubules in MDCK cells . . . 34
4.2.1 Software Requirements . . . 34
4.2.2 Graphical Simulation Model Design . . . 34
4.2.3 Parameter Search Simulation Model . . . 38
4.2.4 Programming Language . . . 38
4.2.5 Implementation and Testing . . . 39
4.2.5.1 IDEs . . . 39
4.2.5.2 Graphical Simulation Model . . . 39
4.2.5.3 Simulation and PDE Models . . . 40
4.3 Modelling microtubules in Cortical Plant cells . . . 40
4.3.1 Requirements . . . 40
4.3.2 Graphical Simulation Model Design . . . 43
4.3.3 Parallelised Simulation Model Design . . . 45
4.3.4 Programming Language . . . 47
4.3.5 Implementation and Testing . . . 47

4.3.5.1 IDEs 47
4.3.6 Graphical Model Evaluation 50

5 Modelling the *Bellett* hypothesis of side wall formation in MDCK cells 52
5.1 Introduction 52
5.2 Cell Geometry 52
5.3 Model Parameters 53
5.4 Simulation 55
5.5 PDE implementation 56
5.5.1 Numerical solution 59
5.5.2 Exploration of parameter space 59
5.6 Results 61
5.6.1 Centrosomal Catastrophe Rate versus Centrosomal Rescue Rate 62
5.6.2 Side wall Catastrophe Rate versus Side wall Rescue Rate . 65
5.6.3 Centrosomal Catastrophe Rate versus Side wall Catastrophe Rate 65
5.6.4 Side wall dependent nucleation rate 67
5.7 Discussion 69
5.7.1 Tubulin dependent dynamic instability 69
5.7.2 Side wall dependent nucleation rate 70
5.7.3 Benefits of OpenCL 71

6 Modelling the role of catastrophe, crossover and katanin in the self organisation of cortical microtubules 72
6.1 Introduction 72
6.2 Methodology 73
6.2.1 Microtubule Collisions 74
6.2.2 Microtubule Order 74
6.2.3 Simulation Environment 75
6.2.4 Software 76
6.3 Results 76

6.3.1 The role of growing to shrinking frequency 76
6.3.2 The role of shrinking to growing frequency 78
6.3.3 The role of collision induced catastrophe (CIC) 78
6.3.4 Katanin and Crossovers 81
6.4 Discussion 84
6.4.1 Catastrophe versus rescue rate 84
6.4.2 CIC versus crossover 85
6.4.3 The effect of katanin at low CIC 86

7 Conclusions **87**
7.1 Summary of Scientific Contributions 87
7.2 Epithelial cell research 88
7.3 Cortical microtubule research 88
7.4 Cortical microtubule software 89
7.5 Future work 89

References **91**

List of Figures

1.1 A graphical representation of microtubules in a polarised epithelial cell. 3

1.2 A diagram of the *Bellett* hypothesis with emphasis on a single microtubule. (1) Microtubules nucleate and grow out in a radial manner from the centrosome and attract +TIP proteins. (2) Microtubule plus ends attach to dynein at the adherens junction and force from the dynein motor triggers the transport of ninein from the centrosome to the adherens junction. (3) The dynein motor then pulls the microtubule away from the centrosome and down towards the base of the cell. (4) Once the minus end is at the adherens junction it is stabilised by the ninein and the plus end of the microtubule continues to grow to the base of the cell. 4

1.3 A diagram showing the results (dotted red arrow) of three different interactions an incident microtubule (solid black arrow) may have when encountering a barrier microtubule. 6

1.4 A diagram showing katanin severing. (a) Katanin approaches a growing microtubule. (b) Katanin severs the microtubule creating two microtubules. (c) The microtubule downstream of the severing site continues in its original growth state whereas the microtubule upstream of the severing site enters a shrinking state. 8

2.1 A figure showing dynamic instability in microtubules. Tubulin is added to the plus end of the microtubule with GTP bound β-tubulin. This GTP is later hydrolysed to GDP whilst part of the microtubule structure and when the microtubule switches to a shrinking state these GDP-bound subunits become exposed at the plus end and are rapidly lost. 14
2.2 A diagram showing the results (dotted red arrow) of three different interactions an incident microtubule (solid black arrow) may have when encountering a barrier microtubule. 15

3.1 A diagram showing the memory and work architecture of OpenCL on a GPU with a CPU as the host device. 28

4.1 A class diagram of the MDCK graphical simulation model. Vector3 is a data type that holds three float values, in this case representing the x, y and z coordinates in 3D space. Mt represents a microtubule, Game1 the simulation, and Program creates and runs the simulation. 36
4.2 A flow chart of microtubule behaviour in the simulation. AZ = attachment zone, and length refers to microtubule length. 37
4.3 A sample 60 minute run of the MDCK graphical simulation model used to validate the simulation with the side wall creating negative feedback on the centrosomal nucleation rate. T = time(min), C = number of centrosomal microtubules, S = number of side wall microtubules. We initially see a rapid expanse of the centrosomal array which then shrinks as the side wall array increases. 41
4.4 Comparison of the MDCK Simulation and PDE models. The data shown for the simulation model is the average of 1000 runs. The simulation and PDE data is very highly correlated with Pearson correlation coefficients greater than 0.998 in all four populations. . 42
4.5 The cortical microtubule model class diagram. 44
4.6 A flowchart of the behaviour of the cortical simulation model. The state transitions indicate spontaneous switches between states that are not induced by interactions with another microtubule. 46

4.7 Testing the cortical microtubule implementation without entrainment or crossovers at 5, 20, 40 and 60 minutes into the simulation. Order ranges from 0 to 1 where 1 is perfectly ordered. The microtubules are not expected to reach an ordered state. 48
4.8 Testing the cortical microtubule implementation with entrainment and no crossovers at 5, 20, 40 and 60 minutes into the simulation. Order ranges from 0 to 1 where 1 is perfectly ordered. The microtubules are expected to reach an ordered state within 60 minutes. 49

5.1 The dimensions and positions of the MDCK cell and its components used in our model. 53
5.2 Parameter search: Centrosomal catastrophe versus centrosomal rescue after 120 minutes. The value of the colours is shown to the right of each map. Increasing C_r has a positive effect on centrosomal total length but becomes negative at high C_r (a). Increasing C_r has a positive effect on the number of centrosomal microtubules (b). In both centrosomal heat maps decreasing C_c has a positive effect at low C_r but this is the reverse at high C_r. Both the side wall total length (b) and the number of microtubules on the side wall (d) decrease with increasing C_c. In both we observe an increase when C_r is increases at high C_c but a decrease followed by an increase when C_c is low. 63
5.3 Parameter Search: Side wall catastrophe versus side wall rescue after 120 minutes. The value of the colours is shown to the right of each map. The total length of the centrosomal array shows a very low sensitivity to S_c and S_r with a less than 1 µm difference observed across the whole heat map (a). The total length of the side wall population (b), and both the number of centrosomal (c) and side wall (d) microtubules increases as S_c decreases and S_r increases. 64

5.4 Parameter Search of Centrosomal catastrophe versus side wall catastrophe after 120 minutes. The value of the colours is shown to the right of each map. In all four heat maps we observe the same trend. Numbers and length decreases as C_c increases. Increasing S_c, however, causes a reduction in all four heat maps but this is only noticeable when C_c is low. 66

5.5 The ratio of the total length of the side wall array to the total length of the centrosomal array in the centrosomal catastrophe rate versus side wall catastrophe rate parameter search. The value of the colours is shown to the right of the map. 67

5.6 Number and total length of the side wall and centrosomal microtubule populations using the estimated parameters from the parameter search assuming free tubulin concentration only dependence. The total length of the side wall population increases in a linear manner whereas the total length of the centrosomal population briefly increases in the first 5 minutes before flattening out (a). The total number of side wall microtubules increases linearly over time whereas the centrosomal population has an initial sharp increase in the first 5 minutes in the number of microtubules before sustaining a much slower rate of increase over the next 115 minutes (b). 68

5.7 Number and total length of the side wall and centrosomal microtubule populations using the estimated parameters from the parameter search using both a microtubule dependent nucleation rate and free tubulin concentration dependence. The total microtubule length of the centrosomal population rapidly increases until microtubules start transferring to the side wall population. At this point the total centrosomal microtubule length decreases towards zero whilst the side wall population increases past 1200 µm (a). This relationship is also observed in the total number of microtubules with the side wall population reaching 133 microtubules and the centrosomal population decreasing past 20 microtubules (b). 69

6.1 The change in order of the microtubule array as a result of change in the growing to shrinking frequency, f_{gs}. (a) the mean order of 200 simulations at 60 minutes when applying a range of multipliers to f_{gs}. Error bars indicate ± standard deviation. The greatest order is observed when f_{gs} is at its WT value. Increasing f_{gs} past the WT value causes a much more rapid decline in order compared to decreasing f_{gs} past the WT value. (b) A snapshot of the model at 60 minutes using ${f_{gs}}^{WT}$. A clear single dominant growth direction is observed. (c) A snapshot of the model at 60 minutes using $0{f_{gs}}^{WT}$. Instead of a single dominant growth direction there appear to be two. (d) A snapshot of the model at 60 minutes using $2{f_{gs}}^{WT}$. The microtubule array is very sparse and does not have a clear dominant growth direction. 77

6.2 The change in order of the microtubule array according to changes in the shrinking to growing frequency, f_{sg}. (a) The mean order of 200 simulations at 60 minutes when applying a range of multipliers to ${f_{sg}}^{WT}$. Error bars indicate ± standard deviation. Increasing f_{sg} past the WT value causes a decline in the order. There is a slight increase in order by decreasing f_{sg} to 0.8 of the WT value but decreasing f_{sg} further causes a sharp decline in the order. (b) A snapshot of the model at 60 minutes using $0.8{f_{sg}}^{WT}$. A single dominant growth direction is observed. (c) A snapshot of the model at 60 minutes using $0{f_{sg}}^{WT}$. The microtubule array is more sparse than the $0.8{f_{sg}}^{WT}$ model and a dominant growth direction less pronounced. (d) A snapshot of the model at 60 minutes using $2{f_{sg}}^{WT}$. The microtubule array is thicker than the $0.8{f_{sg}}^{WT}$ model but a single dominant growth direction is less clear. 79

6.3 (a) The mean order of 200 simulations at 60 minutes when changing the fraction of collisions greater than the entrainment angle that result in a CIC. Error bars indicate ± standard deviation. (b) A snapshot of the model at 60 minutes using 0.8 CIC. A clear single dominant direction is observed. (c) A snapshot of the model at 60 minutes using 0.4 CIC. Whilst the dominant direct appears to be vertical, there are many microtubules growing across it. (d) A snapshot of the model at 60 minutes using 0.1 CIC. There is no clear dominant direction of microtubule growth. 80

6.4 The mean count of crossover expirations of 200 simulations at 60 minutes at CIC values of 0.1, 0.4 and 0.8 using the same parameters as Figure 6.3. Error bars indicate ± standard deviation and the black vertical dotted line shows the mean crossover expiration time. Only the first 49 time intervals are shown on the graphs. 82

6.5 The mean order of 200 simulations at 60 minutes at CIC values of 0.1, 0.4 and 0.8 using the same parameters as Figure 6.3, but also simulating the role of katanin. We define katanin severing time as the time it takes for the katanin protein to localise to and sever a microtubule crossover. Error bars indicate ± standard deviation. . 83

List of Abbreviations

CIC Collision induced catastrophe

MDCK Madin-Darby canine kidney

PDE Partial differential equation

WT Wild type

Chapter 1

Introduction

Microtubules play an integral part in the cell life cycle. They are responsible for a variety of roles including cellular division, intra-cellular transport and arrangement of intra-cellular organelles [Desai and Mitchison, 1997]. A microtubule is a hollow rod composed of α and β-tubulin heterodimers. The α and β subunits are structurally similar showing $\sim$46% amino acid homology [Burns, 1991]. Both subunits have GTP binding sites, however, only the GTP bound to the β subunit can be hydrolysed. The tubulin heterodimers chain together to form protofilaments and each microtubule is composed of 12-17 of these protofilaments stacked as vertical helices, with the exception of a microtubule composed of 13 protofilaments where each protofilament is straight [Chrétien and Wade, 1991]. The diameter of a microtubule is 25 nm but the length of a microtubule can be much longer than this depending on cell type, for example, 40 µm long microtubules have been observed in hippocampal neurons [Yu and Baas, 1994]. Microtubules are polarised with the β-tubulin exposed end considered the plus end and the α-tubulin exposed end the minus end and undergo dynamic instability switching rapidly between growing and shrinking states [Mitchison and Kirschner, 1984].

Arguably, some of the most popular images of microtubules show them as extending across the cell from a fixed point in a radial manner. The point from which they extend is known as the centrosome and its components are highly conserved across a variety of organisms [Doxsey et al., 1994; Stearns et al., 1991]. As such both the centrosome and the microtubules that nucleate from it are well studied [Andersen, 1999]. As microtubules grow there is the recruitment of

microtubule binding proteins which enable the microtubules to interact with other cellular components not only affecting the localisation of the cell components but also the structure of the microtubule array [Akhmanova and Hoogenraad, 2005]. However, a group of microtubules whose behaviour is less understood are those that are not bound to the centrosome - *acentrosomal* microtubules.

1.1 Acentrosomal microtubules

Acentrosomal microtubules, whilst sharing the same make up, are less intuitive than their centrosomal counterparts, posing many questions as to their location, structure and interactions. Acentrosomal microtubules may have their own nucleation complexes *in situ* [Chan et al., 2003; Murata et al., 2005; Zimmerman et al., 2004] or may be transported there via motor proteins [He et al., 2005; Sharp et al., 1997]. Acentrosomal microtubules may be anchored at one end of the microtubule [Moss et al., 2007] or both ends of the microtubule may be free allowing the microtubule to migrate across its environment [Shaw et al., 2003]. These microtubules may be less dynamic than in the centrosomal array with many acentrosomal microtubules being stabilised [Bulinski and Gundersen, 1991]. However, even in the absence of a microtubule organising centre, acentrosomal microtubules are able to form highly organised arrays with two prominent examples of this being epithelial cells and plant cell cortical microtubules.

1.2 Epithelial cells

As epithelial cells polarise there is a rapid shift in the organisation of microtubules. The centrosome moves towards the apex of the cell and the radial array of microtubules shrinks significantly and a new population of microtubules is observed on the side walls of the cell running from the apex of the cell to the base [Bacallao et al., 1989; Bré et al., 1987; Buendia et al., 1990]. A graphical representation of this is shown in Figure 1.1. This behaviour has also been observed in MDCK cells following reassembly of the microtubule array after depolymerisation via nocodazole [Meads and Schroer, 1995]. The establishment of this side

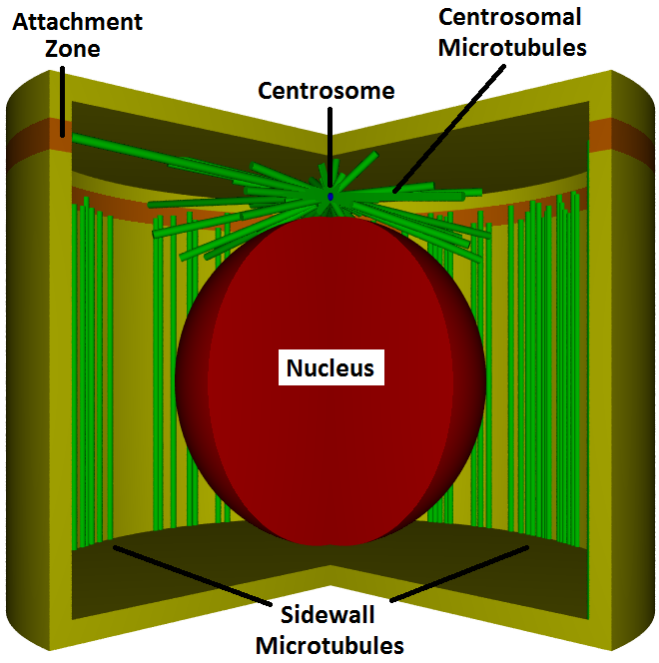


Figure 1.1: A graphical representation of microtubules in a polarised epithelial cell.

wall array is very important for the rapid transport of proteins to the cell surfaces [Mays et al., 1994]. It is believed that the microtubules on the side walls do not nucleate there but instead are transferred there from the centrosomal array through a 'release and capture' model [Mogensen et al., 1997, 2000]. As part of the polarisation process ninein, a centrosomal protein, relocates to the side wall [Mogensen et al., 2000] having being transported there by the centrosomal microtubules [Moss et al., 2007].

In a recent study, Bellett et al. [2009] show that the plus ends of the centrosomal microtubules bind to the side wall at the adherens junction. Also located at these binding sites are β-catenin and cadherin. Based from these observations, Bellett et al. [2009] made the following 'release and capture' hypothesis as to how the side wall array forms (henceforth referred to in this thesis as the *Bellett* hypothesis) (Figure 1.2). Microtubules grow out from the centrosome in a radial manner and attract the +TIP binding proteins EB1 and CLIP-170. The plus ends that encounter the adherens junction (also referred to in this thesis as the attachment zone) are captured by dynein complexes. Ninein is then transported

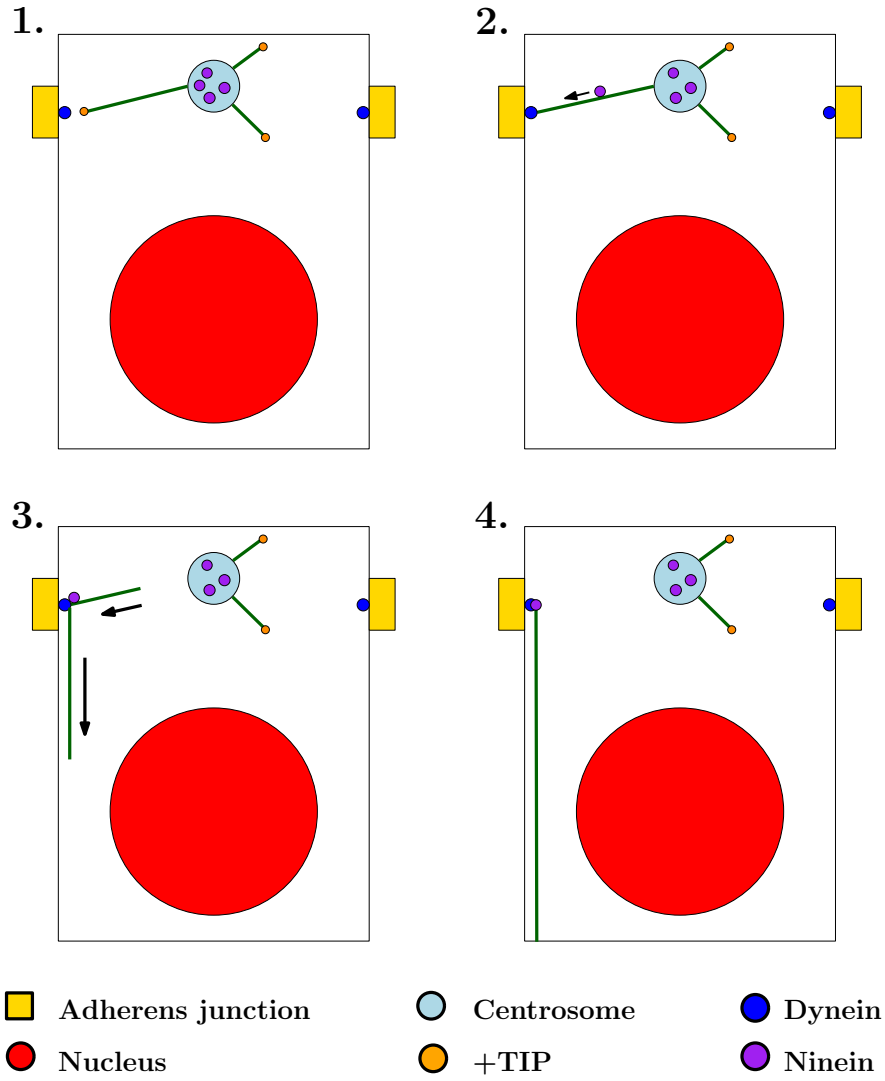


Figure 1.2: A diagram of the *Bellett* hypothesis with emphasis on a single microtubule. (1) Microtubules nucleate and grow out in a radial manner from the centrosome and attract +TIP proteins. (2) Microtubule plus ends attach to dynein at the adherens junction and force from the dynein motor triggers the transport of ninein from the centrosome to the adherens junction. (3) The dynein motor then pulls the microtubule away from the centrosome and down towards the base of the cell. (4) Once the minus end is at the adherens junction it is stabilised by the ninein and the plus end of the microtubule continues to grow to the base of the cell.

via motor proteins along the microtubule from the centrosome to the adherens site. Then force generated by the dynein at the adherens junction pulls the microtubule from the centrosome and down towards the base of the side wall. Once the minus end reaches the adherens junction it is stabilised and bound by the ninein and the microtubule continues to grow towards the cell base. The *Bellett* hypothesis induces two important research questions. Firstly does the hypothesis hold true under biologically reasonable conditions and what assumptions, if any, need to be made. Secondly, whilst the *Bellett* hypothesis implies that the centrosomal array shrinks due to microtubules being pulled from it onto the side wall, this does not fully explain the shrinking/disappearance of the centrosomal array - why do new centrosomal microtubules stop growing? Is there negative feedback from the increased side wall array? We propose two possible reasons for the shrinking on the centrosomal array. The amount of free tubulin (the building block of microtubules) does impact microtubule stability [Janulevicius et al., 2006] so as the side wall increases it would reduce the amount of tubulin and therefore could reduce the stability of the centrosomal array. A second reason is that as ninein is proposed to stabilise the minus ends of the side wall array, it could be that its loss from the centrosome reduces its own ability to stabilise its microtubules.

A suitable candidate cell for computational modelling of epithelial cells is the MDCK cell [Gaush et al., 1966]. It is a model cell type for epithelial cell studies and is widely studied in the academic world with more than 6200 related papers found since 1966 in PubMed [PubMed, 2015]. MDCK cells have several characteristics which make them well suited for epithelial cell studies such as fast growth, clear junctions between cells and a strong apico-basal polarisation that occurs whether it is grown as a mono cellular layer or a full three dimensional culture [Dukes et al., 2011]. Given their popularity in the scientific world they are a strong candidate for modelling techniques. Also as MDCK cells (in addition to cochlear Kolliker epithelial cells) were used in the study where Bellett et al. [2009] derived their hypothesis, it is highly appropriate that they be used in our computational modelling. To the best of our knowledge the *Bellett* hypothesis has not been modelled before.

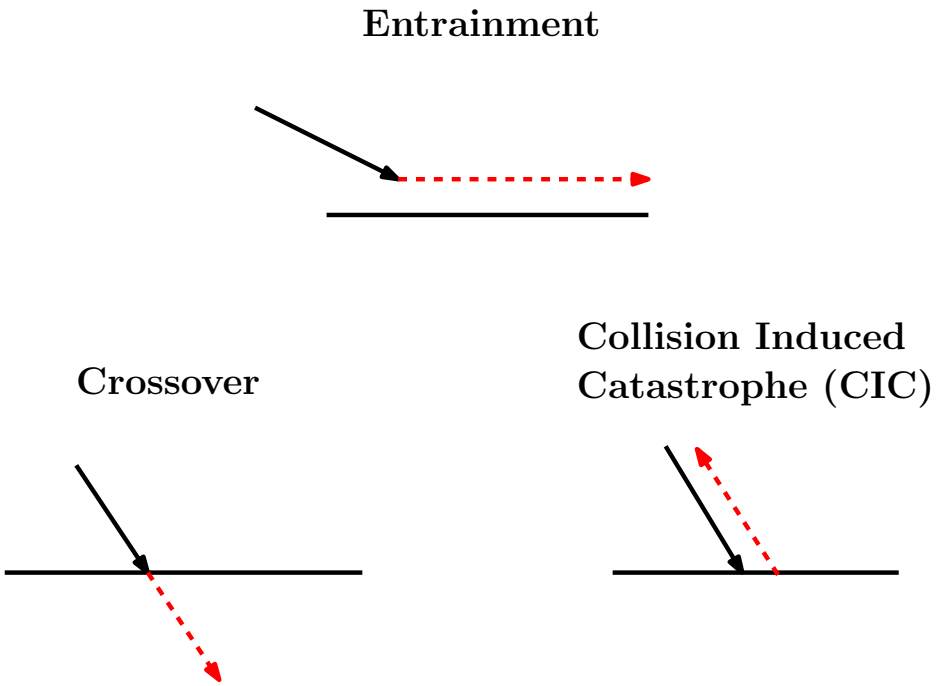


Figure 1.3: A diagram showing the results (dotted red arrow) of three different interactions an incident microtubule (solid black arrow) may have when encountering a barrier microtubule.

1.3 Cortical microtubules

Acentrosomal cortical microtubules in plant cells and the arrays they form are different to those in epithelial cells. Cortical microtubules nucleate on the cell cortex, not from a centrosome, and form highly organised parallel arrays [Wasteneys and Ambrose, 2009]. Cortical microtubules are attached to the plasma membrane [Hardham and Gunning, 1978] creating an almost two dimensional environment. The cortical arrays are believed to provide a structure for factors driving cell elongation though not necessarily through the deposition of cellulose [Himmelspach et al., 2003] but the correct formation of these arrays is very important as they help to determine the growth axis of the cell [Szymanski and Cosgrove, 2009]. The cortical microtubules are neither anchored nor stable at their minus end and are therefore free to grow across the cortex through 'treadmilling' (grows faster at its plus end than it shrinks at its minus end) [Shaw et al., 2003].

Due to the microtubules ability to move across the cortex, and the two di-

mensional environment, microtubules do interact with each other. Possible outcomes of a microtubule to microtubule collision include collision induced catastrophe (CIC), entrainment (microtubule grows parallel to barrier microtubule) and crossover (microtubule grows straight over the barrier microtubule)(Figure 1.3). For more detail on these events see Section 2.5. Entrainment has been shown to be a very strong factor in formation of a parallel array [Allard et al., 2010; Dixit and Cyr, 2004; Eren et al., 2010] but what is less studied is the proportion of non-entrainment events that crossover or undergo CIC. Collisions that do not entrain result in CIC depending on cell type for example a very low 9% in *Arabidopsis thaliana* petiole epidermal cells [Wightman and Turner, 2007] to a moderate 60% in tobacco BY2 cells [Dixit and Cyr, 2004]. This begs the question how does an ordered array form when there is a high probability of crossover (low interaction) between microtubules? Allard et al. [2010] propose that when entrainment alone is allowed that cortical microtubules form an ordered array even when the CIC probability is zero. This, however, seems strange as there does not appear to be any inhibitory factor in their results that would prevent two ordered arrays crossing over each other with one running perpendicular to the other. Therefore it would be important to investigate what is the critical probability of CIC that at which the cortical array is no longer able to form through entrainment alone. Following on from this would be to investigate how plant cells form an ordered array when their observed CIC is less than this critical value. In such circumstances the protein katanin is a good candidate for investigation.

Katanin is a microtubule severing protein [McNally and Vale, 1993] that cuts a microtubule into two segments. The microtubule segment upstream of the cut (newly exposed plus end) enters the catastrophe state (although it can still undergo rescue) whereas the microtubule segment downstream (now minus end exposed at crossover site) continues in its previous state thus increasing the number of microtubules in the cell [Roll-Mecak and Vale, 2006](Figure 1.3). In cortical microtubule arrays the microtubules are preferentially severed at crossover sites [Wightman and Turner, 2007] and 97% of severances target the overlying microtubule (the microtubule that was incident on the collision) [Zhang et al., 2013]. The severing action of katanin has been shown to be essential to the formation of an ordered array [Burk et al., 2001; Wightman and Turner, 2007; Zhang

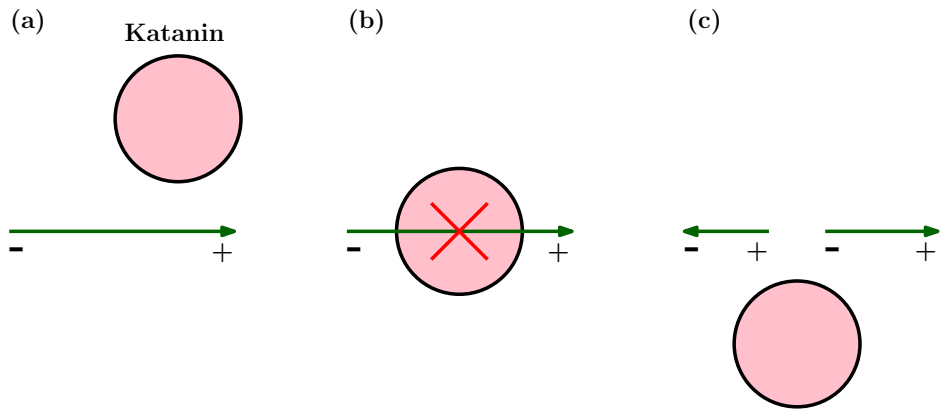


Figure 1.4: A diagram showing katanin severing. (a) Katanin approaches a growing microtubule. (b) Katanin severs the microtubule creating two microtubules. (c) The microtubule downstream of the severing site continues in its original growth state whereas the microtubule upstream of the severing site enters a shrinking state.

et al., 2013]. Therefore a second research question would be at what probabilities of CIC would the action of katanin be sufficient to cause an ordered array? It is also of note that whilst there have been several computational modelling studies on cortical microtubules (for details see Chapter 2), at the time of research, there did not appear to exist modelling software for them. Development of flexible software designed to be used by others would greatly facilitate future modelling studies.

A good candidate cell to be used in computational modelling of cortical microtubules would be one from *Arabidopsis thaliana*. This is an appropriate choice because *Arabidopsis* is not only widely used as a model organism for plant studies but is also used for recent cortical microtubule biological and computational studies [Allard et al., 2010; Eren et al., 2010; Kawamura and Wasteneys, 2008]. This means that there are existing established parameter sets which would be a great aid in developing a model and also other models to compare against.

1.4 Aims

Based on the research questions identified for the formation of acentrosomal microtubule arrays for epithelial cell microtubules and plant cortical microtubules we proposed the following research aims.

1. To model the *Bellett* hypothesis of the formation of the side wall acentrosomal array in a MDCK cell to observe whether it can adequately explain the formation of the side wall microtubule array and explore potential mechanisms for the shrinking of the centrosomal array.

2. To model the role of catastrophe both spontaneous and induced in the formation of the cortical acentrosomal array.

To achieve these aims we define the following objectives:

1.4.1 Objectives for Aim 1

1. Develop a computational model of the *Bellett* 'release and capture' hypothesis.

2. Investigate whether tubulin concentration dependent dynamic instability explains the shrinking centrosomal array.

3. Investigate whether loss of ninein from the centrosome to the adherens junction could explain the shrinking centrosomal array.

1.4.2 Objectives for Aim 2

1. Develop a parallelisable computational model that is representative of cortical microtubule behaviour with a graphical output that is intuitive to use across a range of platforms.

2. Use the computational model to validate the dynamic instability parameters through parameter searches on spontaneous catastrophe and rescue.

3. Use the computational model to observe how the order of the cortical array changes as the probability of CIC decreases from 1 to 0.

4. At CIC probabilities where order fails to occur, model how order changes compared to the response/severing time of katanin.

1.5 Thesis chapter summary

Here we present brief summaries of the other chapters included in this thesis.

- **Chapter 2:** Here we provide a literature review on microtubule dynamic instability, interactions between microtubules and proteins and some of its functions in different cell types. We also include a description of previous computational models of microtubules including general dynamic instability, and models specific to cortical and epithelial microtubules.

- **Chapter 3:** In this chapter we present our research methodology. Based on the aims and objectives presented in Chapter 1 we discuss existing methodologies and their advantages/disadvantages. In particular we consider the appropriateness of trying to develop a mathematical model versus a Monte Carlo simulation model and the use of OpenCL as an alternative parallelisation technology for our parameter searches.

- **Chapter 4:** In the software design chapter we implement the methods discussed in the research methodology chapter to create computational microtubule models for both the MDCK and the cortical microtubules. We discuss the requirements of the software and how they relate to the choice of programming language, both for the standalone graphical models and for the approach we use in parallelisation in order to perform the parameter searches. We present the behavioural logic used in our simulations through flowcharts and class diagrams of our models.

- **Chapter 5:** In this results chapter we present the results of our research into modelling the *Bellett* hypothesis as to the formation of the side wall microtubule array in MDCK cells. We present a large scale parameter search of the relationship between the catastrophe and rescue rates on the centrosomal and side wall arrays as influenced by free tubulin concentration. We do not, however, observe a realistic shrinking of the centrosomal array

in response to an increased side wall array. However, we do observe such a relationship when simulating a loss of nucleating stability at the centrosome in response to ninein being relocated to the side wall.

- **Chapter 6:** In this chapter we present our results of our cortical microtubule research. We show that the catastrophe and rescue rates observed in live studies are a good fit for the model as deviation from those values causes a drop in the order of the array. We observe a sigmoid-like relationship between decreasing the CIC rate and the order of the cortical array with the order of the array rapidly falling as the CIC probability drops below 0.5. AT CIC below 0.5 we find that action by katanin is able to bring about an ordered array but at 0.1 CIC the rate at which katanin would have to act appears unrealistically fast which implies that there are additional mechanisms in controlling the order of the cortical array in cells where the CIC probability is very low.

- **Chapter 7:** Finally in Chapter 7 we provide our conclusions that we have formed as a result of our research. We highlight the scientific contributions of our research and discuss future work that may be undertaken.

Chapter 2

Literature Review: Microtubules

2.1 Summary

In this chapter we present a brief background on microtubules. We highlight the roles of microtubules before delving into their structure and how they are added to cell. We discuss some of the interactions microtubules have between themselves and also with other proteins that enable them to perform their role within the cell. Finally we present a selection of previous computational models that have been used to explore microtubule dynamics and order.

2.2 Microtubule Roles

Microtubules are involved in a variety of important cellular processes including:

1. Cellular Transport - Microtubules provide a framework through the cell that allows motor proteins such as dynein and kinesin to quickly transport vesicles and organelles throughout the cell [Gross, 2004].

2. Cellular Replication - Microtubules play an essential role in cellular division as they are responsible for the formation of the mitotic spindle [Carminati and Stearns, 1997]. Due to this role microtubules can be a target for anti-cancer drug therapy [Zhou and Giannakakou, 2005].

3. Cell Motility - Microtubules form part of the structure of *cilia* and *flagella* which contribute to individual cell movement and the movement of substances outside the cell [Haimo and Rosenbaum, 1981].

2.3 Microtubules and Dynamic Instability

Dynamic instability refers to the microtubules switching between growing and shrinking states. Changing from a growing to a shrinking state is known as catastrophe and the reverse as rescue with much greater growth/shrinkage happening at the plus end. A cause for the switching between growth and shrinking states is the establishment of a GTP cap at the plus end of the microtubule [Mitchison and Kirschner, 1984]. Tubulin is added to a growing microtubule with GTP bound to the beta-subunit stabilising the end of the microtubule but over time this is hydrolysed to a GDP destabilising the tubulin(Figure 2.3). Should the protective GTP cap become too small (for example the rate at which GTP-tubulin is added becomes slower than the rate at which the GTP-tubulin is hydrolysed within the microtubule) then the end destabilises and rapidly shrinks. It is proposed that at least 14 GTP-tubulin dimers are required to maintain the stability of the growing microtubule [Caplow and Shanks, 1996; Drechsel and Kirschner, 1994]. The two state model can be described by four parameters being the growth, shrinkage, catastrophe and rescue rates. A third 'paused' state has been observed where the microtubule is neither growing nor shrinking [Brittle and Ohkura, 2005]. The paused state can also be divided into two states, those entering paused from a growing state and those entering paused from a shrinking state thus creating a four state model [Keller et al., 2008].

2.4 Nucleation

There is an additional tubulin subunit, γ-tubulin, that is required for the nucleation of microtubules [Oakley et al., 1990; Oakley and Oakley, 1989]. In animal cells the γ-tubulin is contained in γ-tubulin ring complexes (γ-TuRC) within the centrosome and acts as a building point for the microtubule and stabilises/anchors the minus end [Zheng et al., 1995]. Whilst the majority of microtubule nucleation

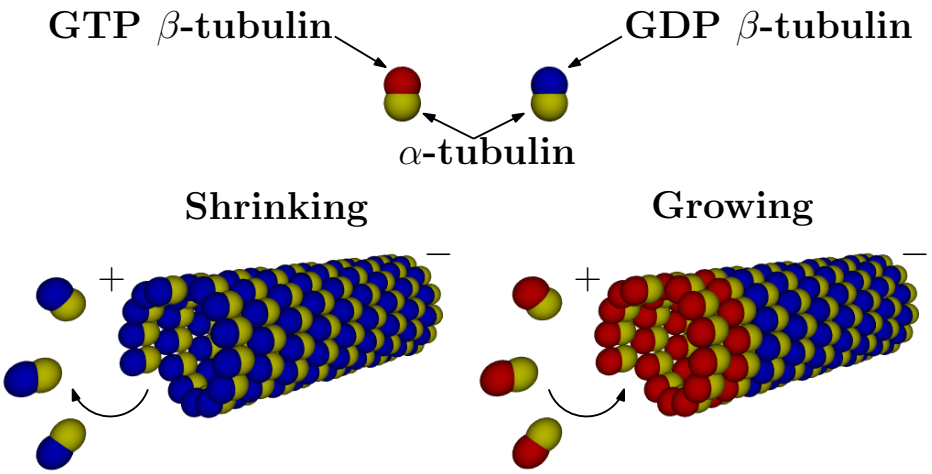


Figure 2.1: A figure showing dynamic instability in microtubules. Tubulin is added to the plus end of the microtubule with GTP bound β-tubulin. This GTP is later hydrolysed to GDP whilst part of the microtubule structure and when the microtubule switches to a shrinking state these GDP-bound subunits become exposed at the plus end and are rapidly lost.

happens at the centrosome in animal cells, nucleation can occur acentrosomally, for example at the golgi body [Efimov et al., 2007]. In plant cells microtubules can also nucleate without the need of γ-TuRCs and a centrosome and potentially have multiple nucleation sites throughout the cell [Erhardt et al., 2002]. In addition, microtubules can also act as nucleation sites for other microtubules through the recruitment of γ-tubulin and termed microtubule dependent nucleation [Murata et al., 2005]. The angle at which the nascent microtubules grow out from the existing microtubule is a range centred around 40°[O. Wasteneys and E. Williamson, 1989]. In the case of branched nucleation there is a bias for the nascent microtubule to grow towards the existing microtubule's plus end although there does not appear to be a bias as to which side of the microtubule that the nascent microtubule branches from [Chan et al., 2009]. In addition to branched nucleation, there are cases where the nascent microtubule grows parallel to the existing microtubule [Chan et al., 2009; Nakamura et al., 2010; Wasteneys and Ambrose, 2009].

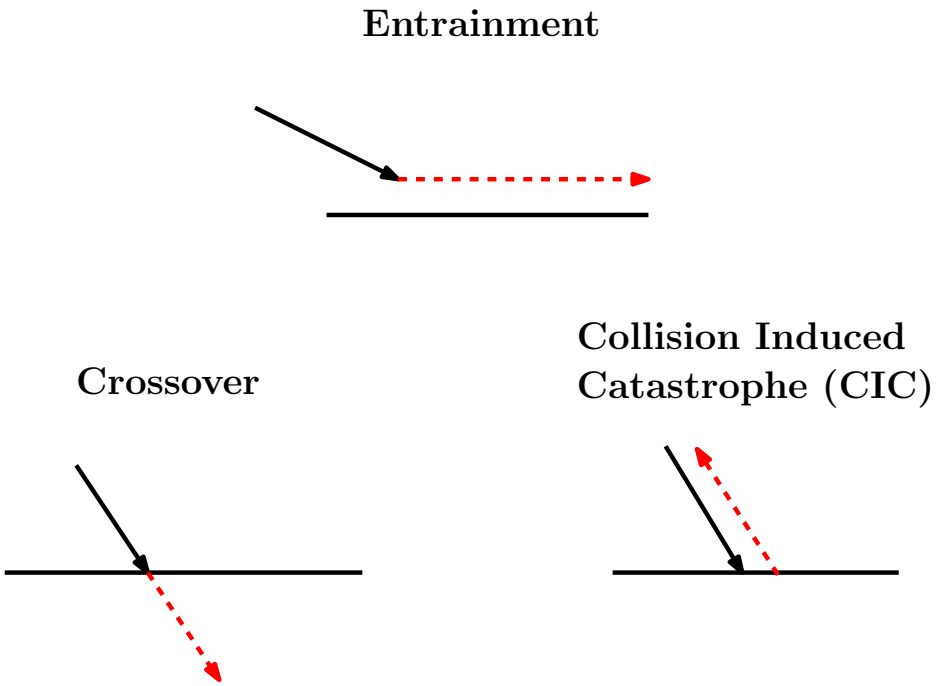


Figure 2.2: A diagram showing the results (dotted red arrow) of three different interactions an incident microtubule (solid black arrow) may have when encountering a barrier microtubule.

2.5 Microtubule to Microtubule Interactions

Whilst microtubule interactions may be quite rare when considering centrosomal nucleation they are quite commonplace in cortical plant cells as microtubules grow attached to the plasma membrane essentially creating a two dimensional environment [Hardham and Gunning, 1978]. This, coupled with multiple sites of nucleation, increases the chances of a growing microtubule encountering a barrier microtubule. When two microtubules collide there are several possible outcomes depending on cell type and the nature of the collision. Examples of possible interaction results are shown in Figure 2.5. The incident microtubule may undergo collision induced catastrophe (CIC) due the force exerted when encountering the barrier microtubule [Janson et al., 2003] and the chance for catastrophe increases as the angle of collision increases [Dixit and Cyr, 2004].

If the angle of collision is shallow (40° [Dixit and Cyr, 2004], 45° [Wightman and Turner, 2007]) the incident microtubule may bend and grow parallel to the

barrier microtubule and is said to have entrained (or undergone 'zippering'), forming bundles of microtubules each separated by a 25 nm gap due to microtubule binding proteins [Chan et al., 1999]. This outcome does not appear to be influenced by whether the growth directions of the incident and barrier microtubules are parallel or anti-parallel to each other [Dixit and Cyr, 2004].

2.6 Microtubule to Protein Interactions

There are many proteins that act to affect microtubule stability and order. Two classes of these proteins are MAPs (microtubule associated proteins) and +TIPs (those that act on the plus end of the microtubule). Some examples of these types of protein are:

1. **MAP-65:** A member of the MAP-65 group promotes both order and stability within microtubules by cross linking parallel microtubules into bundles, creating a 25nm spacing between them [Chan et al., 1999].

2. **CLASP:** CLASPs are found on the cell cortex and assist in the stabilisation of microtubules through rescue [Mimori-Kiyosue et al., 2005]. They are also thought to control the orientation of cortical arrays in plants cells by selectively facilitating the growth of microtubules around specific sharp cell edges [Ambrose et al., 2011].

3. **Septin-GTPase:** A network of septin filaments provides directionality to microtubule growth and promotes growth along this network but inhibiting catastrophe thus keeping the microtubule in a growing state [Bowen et al., 2011].

4. **EB1:** EB1 and its homologues are recruited to the plus end of the microtubule and have a myriad roles including the recruitment of other stabilising and growth promoting factors and also the stabilisation of chromosomes during cellular division [Tirnauer and Bierer, 2000].

5. **KIF17:** KIF17 acts with EB1 at the plus ends of microtubules and has a role in both the stabilisation and polymerisation of microtubules thought to

be due to the selective localisation of APC (adenomatous polyposis coli - a catastrophe inhibitor) and the capping of the plus end [Jaulin and Kreitzer, 2010].

2.7 Models of Dynamic Instability

Gliksman et al. [1993] propose a two state (growth/shrink) Monte Carlo simulation model of microtubule dynamics in newt lung epithelial cells. In this model microtubules are stable at the minus end emulating the attachment of the minus end to the centrosome. The plus end switches between growth and shrinkage according to catastrophe and rescue frequencies. The growth rate is dependent on the amount of unpolymerised tubulin with the cell whereas the shrinkage, catastrophe and rescue rates are static. Each simulation was run until the microtubules reached a steady-state although it is unclear how many independent simulations were done for each parameter set. This model was used to show how changes in the catastrophe, rescue and microtubule nucleation rates could account for the change in microtubule dynamics observed in the mitotic spindle when the cell switches between interphase and mitosis.

Janulevicius et al. [2006] propose a two state model investigating how the concentration of free tubulin affects microtubule dynamic instability. Similar to [Gliksman et al., 1993] the greater the free tubulin concentration the greater the microtubule growth rate. However, in this model the free tubulin concentration also influences the catastrophe and rescue rates such that the catastrophe rate increases and the rescue rate decreases as the free tubulin concentration decreases. All dynamics occur at the plus end of the microtubule as the minus end is considered to be anchored. Their simulations use a Monte Carlo event based approach whereby each iteration a single event such as tubulin attaching to a microtubule causing it to grow or a growing microtubule undergoing catastrophe occurs. They simulate across a range of cell volumes from 1-100 μm, however, it should be noted that the cell volume is only indicative of the total amount of free tubulin and does not place any physical constraints on the microtubules. They observe that as the cell volume decreases the distribution of microtubule growth and shrinkage times change from exponential to gamma-like distributions. This

change in distribution has been observed in live cells suggesting that a limited amount of free tubulin may have a contributory effect in addition to structural changes in the microtubules over time that may alter their dynamic instability parameters.

Tindemans and Mulder [2010] propose a two state (growing/shrinking) mathematical model representing microtubule dynamic instability and also katanin severing. The system is represented by coupled differential equations; one representing the growing population and the other the shrinking population. Movement between the two populations occurs through spontaneous catastrophe/rescue and severing only. Both populations have the boundary condition such that there are no infinitely long microtubules. The model does not require the minus end to be static and any shrinkage in the minus end can be accounted for by offsetting the plus end growth and shrinkage rates. All rates used within the model are assumed to remain constant throughout including the severing rate which is per unit length. They observe that increasing the severing rate reduces the average length of the microtubules and also tightens the distribution of microtubule lengths. Surprisingly they also show that the total number of microtubules at a steady state is independent of the severing rate which appears to contradict results observed in live cells [Roll-Mecak and Vale, 2006]. However, they explain this as a result of their model parameters staying constant over time which is unlikely to be the case within a live cell.

2.7.1 Models of Epithelial cell microtubules

Reilein et al. [2005] propose a computational model of epithelial cell acentrosomal microtubules but they model the array that forms on the basal-cortex of the cell. They simulate 30 microtubules with dynamic instability only at the plus ends with length dependent catastrophe and rescue. The microtubules grew in a 5 μm by 5 μm two dimensional environment with a random nucleation location and growth direction. A new microtubule is nucleated when a microtubules is lost through complete depolymerisation so that the number of microtubules remains constant. The catastrophe rate was greatly increased if a microtubule passed the environment boundary. If a microtubule interacts with another microtubule

or a randomly placed dot representing APC on the cortex then the rescue rate of the microtubule is increased 30 fold to emulate stabilisation. The model was coded in MatLab. They found that where there were no interactions between the microtubules or APC dots that a steady state length was reached but not a steady pattern. By allowing interactions between microtubules and the APC dots they observed that both a steady state and pattern could be achieved.

2.7.2 Models of Cortical Microtubules

Dixit and Cyr [2004] propose a three state model (growing/paused/shrinking) investigating the role of entrainment in creating an ordered array. In their model microtubule 'shallow' collisions ($< 40°$) can lead to entrainment and 'steep' collisions result in catastrophe or crossover. They performed five independent Monte Carlo simulations of a very small sample of 20 microtubules using a combination of Microsoft excel and CorelDraw. They observed that ordered arrays occurred in each of these five simulations (although the orientation of the array was different in each one) that did not occur when the simulations were performed without entrainment. Whilst both the low number of simulated microtubules and low number of repeats would not be sufficient to draw a statistically significant conclusion it strongly suggests entrainment as a mechanism for cortical microtubule self organisation.

Baulin et al. [2007] propose a simplified two state model (growing/paused) of cortical microtubules. The microtubules are rigid and grow as straight lines. The minus end is constantly shrinking and the plus end is in a growing state but switches to paused when it encounters a barrier microtubule. Once the barrier microtubule no longer blocks the incident microtubule, the incident microtubule returns to a growing state. This is the only interaction between microtubules. The microtubules are nucleated in a random location with a random orientation. Under these conditions order occurs although without a specific orientation. Biasing the growth direction of the microtubules towards a certain angle and/or replacing certain periodic boundaries with hard edges both cause order to occur quicker and also in a certain direction. Their model was also extended to three dimensions where they introduced an additional parameter for the diameter of

the microtubules although it is not stated what value was used for this. Ordered domains were still observed to form in three dimensions (although there was much less distinction between ordered domains due to the additional dimension of movement).

Eren et al. [2010] propose a three dimensional model of an *Arabidopsis* cell presenting it as a cylinder. In their model the ends of the cylinder were considered to be catastrophe inducing edges and therefore there was a strong selective pressure on microtubules to grow into a transverse array. The order of the microtubule array was measured using an entropy parameter. From their model they observe that the order of the cortical array is sensitive to large changes in the entrainment angle (20°, shorter microtubules, higher entropy. 60°, longer microtubules, quicker loss of entropy at the start of the simulation by stabilises with a higher entropy). They observe that allowing entrainment has a greater impact on the order of the cortical array compared to edge induced collision and collision induced catastrophe. They also modelled the effects of microtubule dependent nucleation, whereby 38% of nucleations grew parallel to the existing microtubule and 62% grew branched at angle drawn from a distribution with a mean of 40°. Whilst it was observed that microtubule dependent nucleation had a significant effect on the polarity of the cortical array it did not have a significant effect on the entropy of the array. Whilst they used a Monte Carlo model they only performed 10 independent simulations for each set of parameters. Their model was coded in MatLab.

Allard et al. [2010] investigate the role of entrainment, collision induced catastrophe and branched nucleation using parameters from *Arabidopsis thaliana*. They model the plant cell as a single square face with all four edges as periodic boundaries. They replicate the plus end pause model of [Baulin et al., 2007] and achieve an ordered structure but show that if the microtubules either undergo collision induced catastrophe or crossover instead of a paused state when encountering a barrier microtubule then order is not achieved. They observed that at low rates of collision induced catastrophe the microtubules do not interact enough and at higher rates the lifespan of the microtubules is too short. They propose that entrainment is sufficient to bring about order even the collision induced catastrophe rate is zero. They show that increasing the entrainment angle from

40° to 60° actually reduces the rate at which an ordered array is formed. Similar to [Baulin et al., 2007] they found that changing a set of edges from periodic boundaries to catastrophe inducing biased the orientation of the ordered array to be parallel to those edges. They found that increasing the ratio of branched microtubule dependent nucleation to microtubule independent nucleation caused the microtubules to be clumped in bundles separated by large areas of empty space and hence were not representative of a real cell. For each experiment they performed 10 independent simulations with differing starting conditions.

Ambrose et al. [2011] investigate the role of catastrophe inducing cell edges and the CLASP protein in creating an ordered cortical array. They model the plant cell as a cube with microtubules growing on each of the six faces and use similar growth and entrainment parameters to [Allard et al., 2010]. The 12 edges between faces are divided into 3 groups; longitudinal, transverse and radial with each group having their own catastrophe inducing rate. They simulate the role of the CLASP protein at cell edges by reducing the catastrophe rate at those edges. Using this model they show that by altering the location of CLASP they are able to change the orientation of the microtubule array and replicate microtubule patterns observed in different stages of a live cell.

Chapter 3

Research Methodology

3.1 Introduction

Having defined our aims and objectives in Chapter 1 we present our research methodology that we will use to achieve them. Influenced by previous computational modelling approaches we evaluate the appropriateness of a mathematical or Monte Carlo simulation model for our MDCK and cortical models. We discuss collision detection, parallelisation (required for efficient parameter searches and simulation repeats) and microtubule visualisation techniques.

3.2 Modelling approach

There are two main methods used in the computational modelling of microtubules; a PDE model or a stochastic Monte Carlo simulation model. The PDE model abstracts microtubule behaviour to mathematical equations and is deterministic meaning that given the same set of starting variables the model will always output the same result. This can be very advantageous when performing large scale parameter searches as each set of parameters would only need to be run once. However, whilst it may be possible to develop appropriate mathematical equations to describe the model, the complexity of the system can make it difficult to solve these equations. As such complex microtubule behaviour could increase the run time of the mathematical model potentially offsetting the

advantage of it being deterministic. Therefore current mathematical models of microtubules typically only cover the relatively simple behaviour of dynamic instability, switching between growing and shrinking states and do not include any interactions between microtubules themselves although a recent model included microtubule severing [Tindemans and Mulder, 2010]. This suggests that there would be a large jump in difficulty in modelling more complex behaviour.

Monte Carlo approaches are stochastic meaning that within the model there is an element of randomness which requires that the model be repeated many times in order to observe all events and thus generate a robust result. This has the immediate disadvantage that parameter searches would take longer than a deterministic model but this is greatly offset by the fact that complex microtubule behaviour is easier to simulate and may be more intuitive to develop. It appears that for microtubule simulations that involve some interaction between the microtubules, Monte Carlo methods are the standard approach [Allard et al., 2010; Ambrose et al., 2011; Eren et al., 2010]. As there are substantial differences in microtubule behaviour in the MDCK and cortical plant cells we consider them separately as to whether a mathematical or Monte Carlo approach is more appropriate.

3.2.1 MDCK microtubules

The aim of our MDCK research is to perform large scale parameter searches on the spontaneous catastrophe and rescue rates then it would be of great benefit to produce a mathematical model to reduce the runtime of such parameter searches. The MDCK microtubules are good candidates for mathematical modelling as their behaviour is relatively simple as the microtubules do not interact with each other. The side wall microtubules essentially represent a standard mathematical dynamic instability model with the exception that they stabilise at a certain length defined by the geometry of the cell. The centrosomal population all nucleate from the same point within the cell and if we consider all other components of the cell to be static then the collision events (catastrophe or attachment) can be modelled as a function of the microtubule length. A potential issue is that the centrosomal rescue rate will influence the length dependent

catastrophe rates as any proportion of microtubules that under went collision induce catastrophe at length x and were subsequently rescued must not undergo collision induced catastrophe until it reaches length x once more. Whilst this added complexity can be avoided by setting the centrosomal rescue rate to zero, it would be useful to have a simulation model not only to validate the mathematical model but also to be used when investigating the effects of the centrosomal rescue rate. Also one objective of our MDCK research is to produce a real time graphical output of a MDCK cell to aid in the representation and validation of experimental results. Therefore it is necessary to create a Monte Carlo simulation model for that purpose alone as a mathematical model would not be capable of representing individual microtubules. Even though we had to develop a simulation model we believed that the mathematical model was still worth developing as it was likely to offset its development cost due to increased efficiency in the parameter searches.

3.2.2 Cortical microtubules

Based on our aims we decided that a Monte Carlo simulation method was the appropriate approach to take. Due to the complexity of the microtubule interactions being that much greater than existing microtubule mathematical models, coupled with random nucleation location and growth direction, we determined that it would be very difficult to solve equations which could adequately represent this behaviour. A further requirement of other model is to be able to visualise the patterning formed by the cortical microtubules which would be easier with a Monte Carlo model. Also as the aim is to produce a piece of software that can be used and expanded by other researches, a simulation based model would be much easier to add new and modify existing microtubule interactions.

3.3 Collision detection

An integral part of Monte Carlo simulations where microtubules interact with other bodies through proximity is the detection of collisions between the growing microtubule and other objects (including other microtubules) within the simula-

tion environment. Where microtubules have the potential to collide with other microtubules, a brute force collision detection approach, whereby each microtubule plus end is compared with all other microtubules, would have a complexity of $O(n^2)$. This can add up to greatly increased simulation times where the number of microtubules is large and the number of repeats (necessary in Monte Carlo simulations) is high. Therefore it is necessary to use an appropriate collision detection technique. We can divide these into *a priori* and *a posteriori* techniques.

3.3.1 *A priori*

A priori techniques predict where and when a collision will occur before the microtubules are grown. Such techniques can be highly effective in situations where collisions happen between a growing microtubule and static objects and the microtubule growth behaviour is highly predictable. Such a technique is highly appropriate for our MDCK simulation model. The centrosomal microtubules are all anchored at the centrosome which is in a fixed location in the cell. As the other cellular components are also in static locations then based on the random direction assigned to a microtubule upon nucleation one can calculate which cellular component the microtubule will collide with, the outcome of the collision (attachment or catastrophe) and the length the microtubule at the collision point. As we can calculate the point of impact on the attachment zone we can also infer the length at which the microtubule encounters the cell base once it is on the side wall. Therefore throughout the simulation we only need to monitor the length of the microtubule and when it is greater or equal than its collision length we then apply the appropriate change.

Such a method could have advantages with our cortical microtubule model as it would be possible to calculate at what time step a microtubule could collide with another microtubule (assuming the microtubules remain in a growing state throughout) which would reduce and also potentially simplify the collision checks made. For example we know that microtubule a will collide first with microtubule b in 20 time steps. Therefore no collision detection is necessary until those time steps have elapsed. Then we can test whether b is the expected length (reduced

through periods of pause and shrinkage) or even exists prior to resolving the collision. The disadvantages are that the more complex the microtubule behaviour gets, such as free tubulin dependent growth rates and changes and a non constant growth direction, the harder it is to predict when and where the collisions occur. Also if a change in behaviour was added, the collision code itself would have to be updated limiting its use when developing the simulation software.

3.3.2 *A posteriori*

A posteriori techniques would grow all microtubules in the simulation within a given time step and then calculate whether any collisions have occurred. The point of collision would have to be back calculated and the positions of the microtubules changed to reflect the outcome. This is the approach used in [Allard et al., 2010] and [Ambrose et al., 2011] and would therefore be appropriate to use in our cortical microtubule model. As the collision detection code is independent of the microtubule behaviour (microtubules are either intersecting or they are not) then it would also make the cortical simulation software more robust. A key efficiency issue associated with *a posteriori* collision detection is how often should you check for collisions. The most accurate result would be obtained but having the smallest time step in between collision checks as this minimises collisions being missed. However, the smaller the time step the less likely it is for a collision to occur. As collision detection is one of the most computationally intensive parts of the simulation, it may be necessary to trade off some of the accuracy for a decreased runtime by increasing the time step between collisions (for example Allard et al. [2010] only check for collisions every 0.05 minutes).

3.3.3 Parameter searches through parallelisation

Parameter searches are highly parallelisable and could greatly benefit from heterogeneous computing. Examples vary from the Cell Broadband Engine, which combines a processor core with eight data accelerator cores and is used in the Sony Playstation 3 [Shi et al., 2010] to supercomputers exploiting the linking of thousands of individual processors [Barker et al., 2008] but for many within research the more accessible form of effective heterogeneous is a high performance

cluster. However a perhaps more accessible and potentially more efficient alternative is to use the computer's graphics processing unit (GPU). At the time of research, one of the more powerful CPUs, the Intel® Core™i7-980X is capable of running 16 threads. However, the AMD Radeon™HD 7950, a mid-range GPU, has a total of 1792 stream processors and thus capable of performing over one hundred times more threads than the Intel® CPU. Therefore assuming the code or algorithm in question is parallelisable, it should be possible to decrease the runtime by several orders of magnitude.

3.3.4 OpenCL versus CUDA

To perform operations on GPUs one can use CUDA [Nickolls et al., 2008] which is specific to NVIDIA GPUs, and OpenCL [Aaftab, 2011] which is aimed to be the cross platform standard for GPU computing. The debate as to whether to use CUDA or OpenCL can be argued on two metrics: portability and performance. Portability is the easier metric to measure as CUDA is specific to NVIDIA GPUs and therefore if you are either using an AMD GPU or want to make your application available to the widest audience then OpenCL is the obvious choice. However, fair comparisons to determine the cost of portability on performance between CUDA and OpenCL are difficult due to the differences in architecture between AMD and NVIDIA GPUs and the different coding strategies needed to optimise kernel code (code that is execution in parallel on the GPU) on NVIDIA and AMD GPUs and differences in the compiler used [Fang et al., 2011] i.e. NVIDIA GPUs use a scalar architecture whereas AMD GPUs use a vectorised architecture with a preference for using the *float4* data type [Stone et al., 2010]. The true nature of the portability is shown by [Du et al., 2012] who show that using a kernel that performs matrix multiplication (SGEMM), which is optimised for a NVIDIA Tesla C2050, on a ATI Radeon 5870 causes the performance to drop to approximately 47% at peak compared to using a kernel optimised for the ATI GPU. Likewise, using the ATI optimised SGEMM kernel on the NVIDIA card caused an approximate 90% reduction in performance compared to using the NVIDIA optimised kernel So whilst OpenCL code is likely to give the same output across different GPUs the performance is likely to greatly decrease when

run on a non-optimised architecture. Comparing OpenCL and CUDA performance on a NVIDIA Tesla C1060 using the Parboil benchmark MRI-Q, [Komatsu et al., 2010] observed that, as long as both CUDA and OpenCL code and compilers were optimised to the Tesla architecture, both produced similar sustained performances. This suggests that there is no significant inherent performance advantages to using CUDA over OpenCL on NVIDIA cards (assuming proper optimisation) and therefore, due to OpenCLs portability, OpenCL is used as the heterogeneous computing approach in this thesis.

3.3.5 How does OpenCL work?

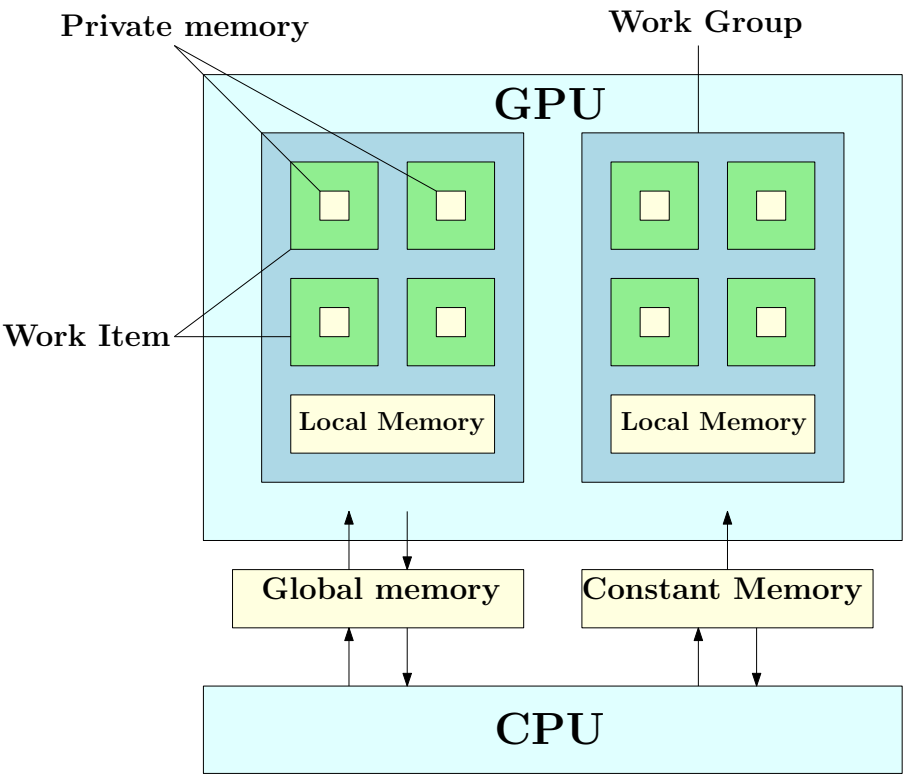


Figure 3.1: A diagram showing the memory and work architecture of OpenCL on a GPU with a CPU as the host device.

Whilst an in depth guide to the inner workings of OpenCL can be found in the OpenCL Specification Guide (at the time that this research was conducted it was version 1.2) [Aaftab, 2011] what follows here is a brief summary of the

runtime of an OpenCL application using a GPU. The code in an OpenCL can be split in two parts - the host code which runs on the CPU and the kernel code which is executed by each of the processing elements on the GPU. Each instance of a kernel (one for each thread) has a unique global ID that ranges from 0 to the total number of threads minus one. Each kernel instance also belongs to a work group and has both a work group ID (ranging from 0 to the number of threads divided by work group size) and a local ID within that work group (ranging from 0 to work group size minus one) meaning that each kernel instance can be uniquely identified by its global ID or a combination of both its work group ID and local ID. There is no limit to the number of threads that can be requested to run but the number than can be run in parallel is limited by the number of processing elements on the GPU. The host code can be written in any language (with popular languages such as C#, C ,C++, Java and Python having OpenCL bindings) and is responsible for specifying the number of threads, the work group size and allocating memory to be used by the CPU and GPU. There are three types of memory that are assigned by the host code: global, constant and local (Figure 3.1). Both global and constant memory are put into a shared cache between the CPU and GPU. Constant memory is read only for the GPU so is used for sending invariant data to be read and processed by the GPU. Global memory is used for data that may be changed over the execution of a kernel and for the transfer of results from the finished kernel to the CPU. Local memory is only available to the GPU but is still assigned by the host code. In addition there is private memory, which is memory that is used by each individual kernel. The time it takes to transfer of data from the CPU to the GPU and back again is an important consideration when deciding whether OpenCL is appropriate to your algorithm as it may offset any gains achieved by parallelisation.

3.3.6 Optimising OpenCL

Although true optimisation would be specific to the exact hardware that is being targeted, we present here some of the more general optimisation strategies [Andrade, 2011].

1. Memory Coalescing: Memory Coalescing is about reducing the number of

read calls the GPU makes to internal memory as each call has associated time cost. For each call a minimum number of elements is loaded into GPU and if each of these elements is used (each kernel access the element corresponding to its global ID) then the access is said to be coalesced. However, if each kernel was accessing the element using a multiple of its global ID then it is probable that elements will be loaded by the GPU which will not be used and is thus an inefficient usage of memory access.

2. Local Memory: The advantage of using local memory over global memory is that read and write access to the local memory is faster than accessing the global memory. Using local memory would therefore be useful if you required multiple accesses to global memory within the same work group and/or as place to store intermediate results.

3. Kernel Calls: Transferring data from the host to the device is computationally expensive and therefore multiple kernel calls should be avoided by keeping as much of the code as possible on the device.

4. Vectorisation: The vector architecture of AMD GPUs causes it to prefer vector calculations with a preferred float width of 4.

5. Loop unrolling: By unrolling loops you reduce the overhead of iterating through the loop although the extent of performance gain would be dependent on how much code is being executed within the loop with the greatest relative gains being observed when the loop contents is minimal.

6. Native Functions: Native functions use native instructions on the device to perform the calculation. For this reason the native functions are generally faster but their accuracy is limited by their implementation on the device.

3.3.7 OpenCL in practice

Khanna and McKennon [2010] use OpenCL to perform numerical modelling of gravitational wave sources. They observe a 25x performance increase using OpenCL on a NVIDIA C1060 Tesla compared to a AMD 2.5 GHz Phenom 9850

quad-core processor. Its also worth noting that they achieved a similar performance gain when using CUDA on the same GPU. Shimobaba et al. [2010] use OpenCL to accelerate the calculation of a computer-generated hologram (CGH), generating a CGH of a resolution of 1920 x 1024 from a 3D object composed of 1024 points in 31 ms using a ATI Radeon HD5850, almost 2000x faster than on the CPU (Intel Core 2 Quad Q6600). They also report an over 10 fold increase in performance from an optimised kernel over an unoptimised kernel (422 ms). Hoffmann et al. [2010] obtain a speed up of approximately 1500 when using an OpenCL implementation of a spiked neural network over a Matlab CPU implementation.

3.3.8 OpenCL or computing cluster?

Whilst OpenCL does appear to have good potential for the parallelisation of microtubule parameter searches there are some drawbacks particularly when compared to using a computing cluster. The first is that there is likely to be a development overhead in translating the model code into a C99 format whereas there is potentially much less overhead in transferring simulation code to the cluster where the cluster supports your chosen programming language. Also OpenCL may not be appropriate to simulate larger and more complex models as it becomes harder to optimise the code to suit the GPU architecture and the limited memory may become an issue. Therefore We plan to use both the cluster and OpenCL in my research with more detailed justifications as to the particular choice given in the Software Design chapter.

3.4 Microtubule visualisation

In previous studies microtubules are typically visualised as lines [Allard et al., 2010; Baulin et al., 2007; Reilein et al., 2005] even in three dimensional studies [Baulin et al., 2007]. In our MDCK graphical model, the microtubules are visualised as cylinders (although they are simulated as lines). This is to aid proper depth perception and clarity as they being rendered in a three dimensional environment and also provide an output closer to those from live cell imaging. A

clear output is important as we will be using the graphical model to validate the microtubule behaviour. In the cortical graphical model the microtubules are visualised as lines (the same way as they are simulated). The reason for this is two fold. Firstly given the two dimensional environment and the relatively small diameter of a microtubule, it would be very difficult to observe any appreciable effect of rendering the microtubules as cylinders. Secondly it is faster to render lines which therefore causes the graphical simulation to progress faster.

Chapter 4

Software Design and Implementation

4.1 Introduction

In this chapter we discuss our approach to the design and implementation of the software used in our research. At the time of this research there did not, to the best of the authors knowledge, exist specific software for the modelling of microtubules. One of the most probable reasons for this is that computational microtubule models tend to target a certain behaviour or hypothesis of microtubules within a specific cell type which makes it difficult to extend said models to a wider range of cell types. Also as microtubule behaviour is often simplified for simulation models, for example abstracted to straight lines instead of curved rods, there may not be a significant computational overhead in coding that simulation. Therefore in order to perform our research it was necessary to code our own simulation models *de novo*. Due to the large differences in requirements for modelling MDCK cells compared to cortical plant cells we present each separately beginning with the MDCK software as this was the first to be done in our research. For both models we discuss the initial requirements, the implementation and testing of the software. In this chapter we do not cover the parameters used in the software as this varies on the experiment nor the derivation of the PDE models as these are presented in the respective MDCK and cortical results

chapters.

4.2 Modelling microtubules in MDCK cells

4.2.1 Software Requirements

Based on our aims and objectives and our planned research the software requirements are to be able to:

1. Simulate an MDCK cellular environment with microtubule behaviour according to the *Bellett* hypothesis.

2. Produce a graphical output to visualise the microtubule array in aid in presentation and validation of results.

3. Run large scale parameter searches targeting the catastrophe and rescue rates and output meaningful results in a time efficient manner.

A simulation model is necessary to satisfy the second requirement and in doing so also satisfies the first requirement. This simulation model could then be stripped of its graphical output and then parallelised on a cluster for the parameter searches. Whilst this approach would greatly reduce development time given that there would have to be a simulation model it would not lead to efficient parameter searches. For each parameter set the simulation would need be repeated > 100 times to produce a reliable result. Even using a low resolution parameter such as 10x10 would require a minimum of 10000 simulations. In addition, the UEA cluster limited the number of concurrent jobs per user to $\sim$250 which would further compound the overall inefficiency of this approach. Therefore our aim was to develop an additional mathematical model to be used for as many parameter sets as possible.

4.2.2 Graphical Simulation Model Design

To begin, we consider the physical components of the simulation model which do not have any actions themselves. These are listed along with their represented

shape in Table 4.1. Whilst the centrosome is indeed a 3D object we abstract it to a single point in our model as it represents the nucleation point of all microtubules in the model. We simplify the shape of the MDCK cell by modelling it as a cylinder. The free tubulin would not be represented visually as it is not typically visible in cell imaging studies that involve microtubules and therefore it is sufficient to represent it as a concentration only.

Table 4.1: The physical components and their representation in the simulation model.

Component	Representation
Nucleus	Sphere
Centrosome	Point
Cell membrane	Hollow cylinder
Adherens junction	Region of cell membrane
Free tubulin	Numeric (concentration)

The microtubule is the remaining physical component of the model and is the actor in the simulation. In keeping with good object orientated programming practices the microtubule is encapsulated in its own class to improve both re-usability and understandability of the code. As microtubules are hollow rods an accurate representation of them would be as hollow cylinders. However, as the diameter of a microtubule is very small relative to the size of the other physical components in the simulation and the microtubules do not interact with themselves then they can be simplified as 2D lines in a 3D space for the purposes of running an efficient simulation but these 2D lines can be visualized as cylinders for the purposes on the graphical representation.

We show the UML diagram of our epithelial cell graphical model in Figure 4.1. The Mt class represent a microtubule. Its variables and functions are mostly self-explanatory with the exception of the state variable which would take three different values; 0 = shrinking, 1 = growing, 2 = paused. The Game1 class (so called as this model was developed using the Microsoft XNA Game Studio) simulates the cell by iterating through the simulation logic, shown as a flowchart in Figure 4.2, with each iteration being indicative of one time step until the required simulation time is reached. The output of the simulation can be parsed to a file and include the number and total length of the microtubules in each

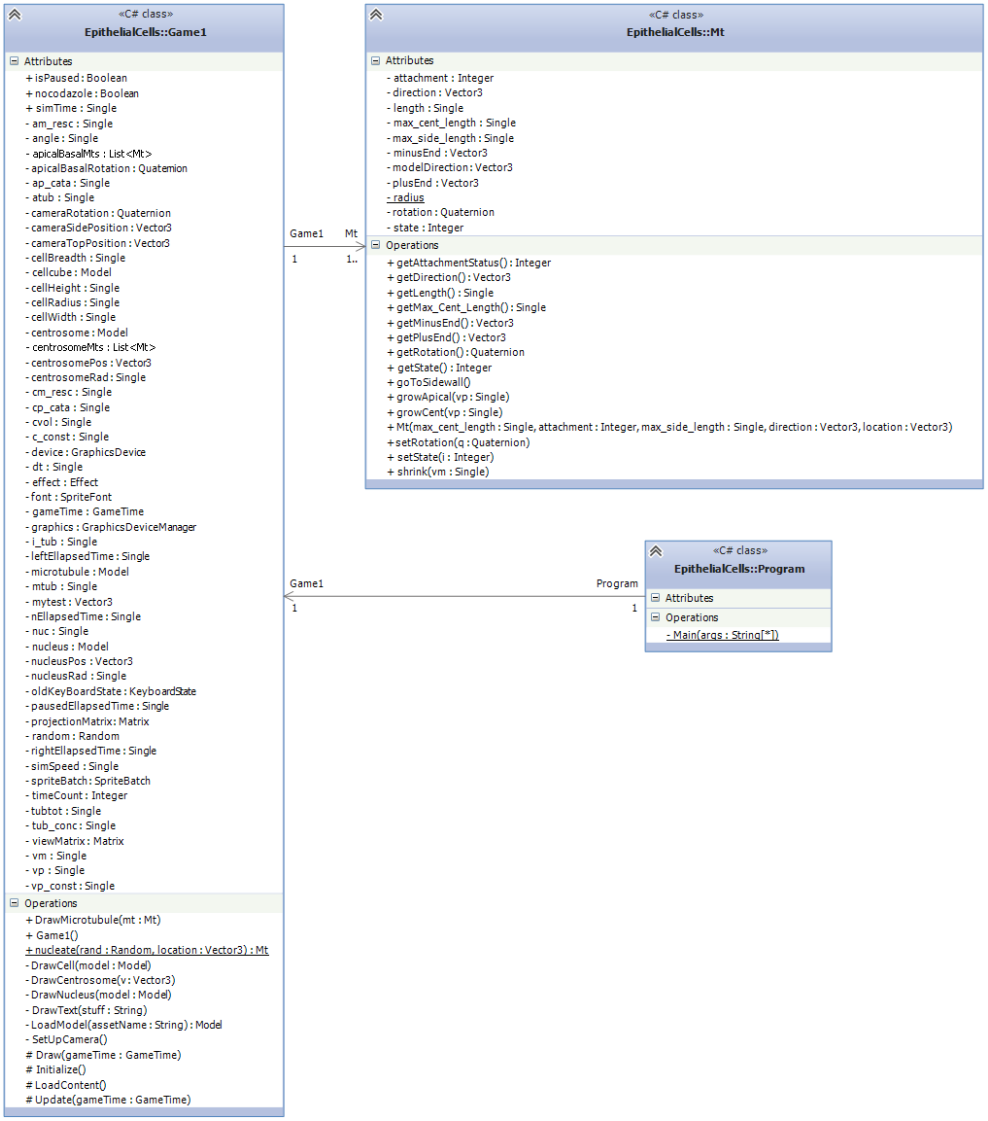


Figure 4.1: A class diagram of the MDCK graphical simulation model. Vector3 is a data type that holds three float values, in this case representing the x, y and z coordinates in 3D space. Mt represents a microtubule, Game1 the simulation, and Program creates and runs the simulation.

population at given times during the simulation. The graphical output can be disabled for faster run speeds

It should be noted that there is no collision detection method shown in the main class. This is because as stated in our research methodology we use an *a*

priori collision detection method and therefore only need to calculate potential collisions upon nucleation.

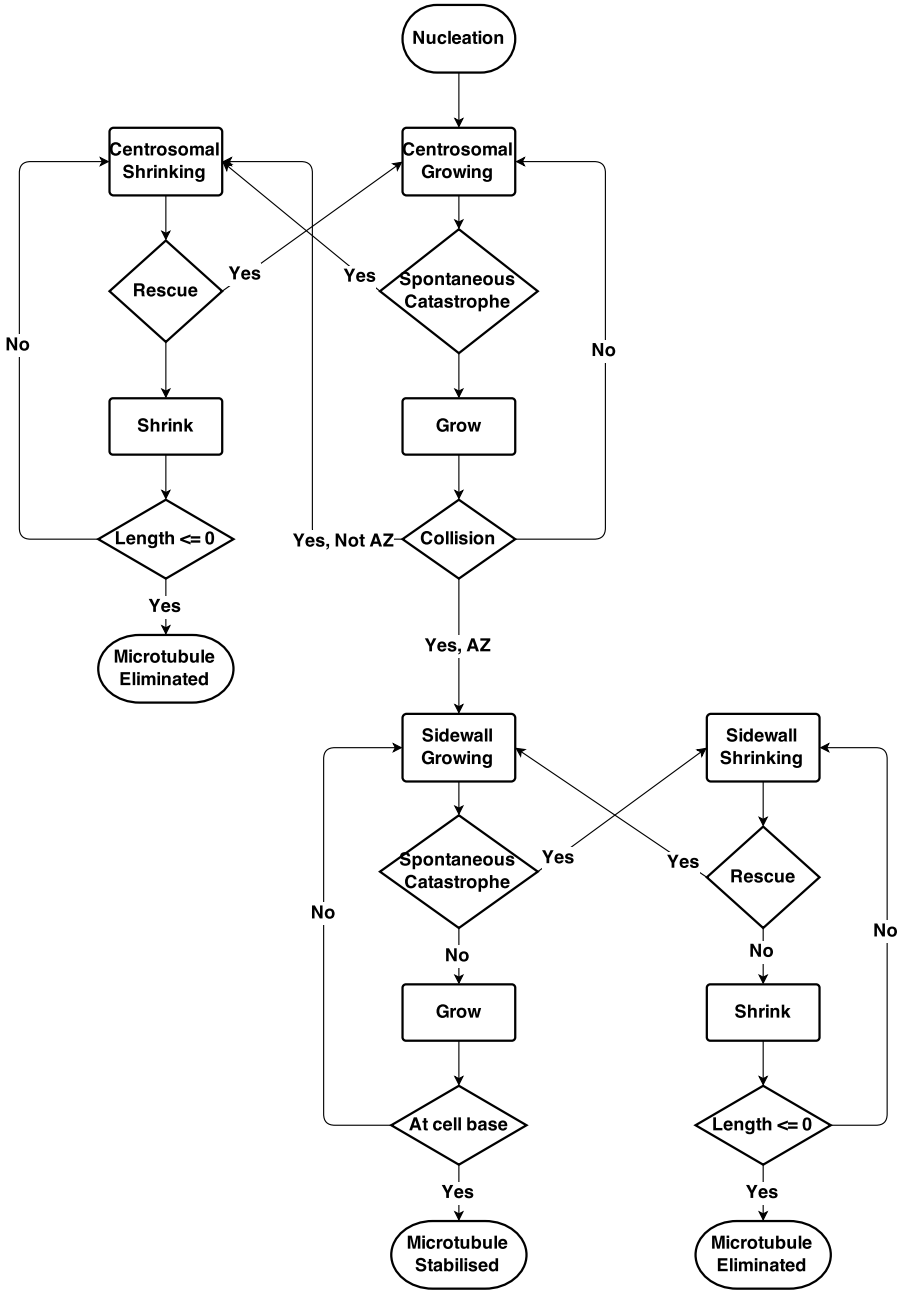


Figure 4.2: A flow chart of microtubule behaviour in the simulation. AZ = attachment zone, and length refers to microtubule length.

4.2.3 Parameter Search Simulation Model

Whilst the aim is to use a PDE model for the parameter searches this may not prove possible for all parameter sets. As we want to avoid serial processing, one option would be to disable the graphical output on the graphical simulation model and add an additional loop to the code such that the simulation ran for 1000 times and output a mean set of results for a given parameter set. However, due to cluster throttling this would not be an efficient way of performing a high resolution parameter search. An alternative would be to parallelise the model on the graphics card through OpenCL, again repeating each parameter set 1000 times to produce a mean output. This does however have two key disadvantages. At the time of production the C99 language used on the kernel did not support classes and the code would have to be rewritten to reflect that. Also it would be difficult to optimise the code for the GPU and the same code would likely run faster on the CPUs used in the cluster. However, as the simulation is reasonably lightweight, converting the code to C99 compatible format would not be complex. The main advantage for using OpenCL would be that a high rate of concurrency could be achieved as there were more than double the cores on our GPU than the server throttle limit plus it would not be liable to cluster downtime or excessive loads caused by factors/persons outside of our control. Therefore we decided to parallelise the simulation model through OpenCL.

4.2.4 Programming Language

The language we used to develop this software was C# with the following justifications. Importantly the authors were already familiar with it. As an efficient object orientated programming language with a good quality IDEs it would allow for rapid development and testing and be suitable for simulations in which thousands of iterations are required. Use of an object orientated language would also permit relatively straight forward addition of added functionality that may be required as research is being undertaken. The XNA framework allows a 3D simulated real time graphical representation of the cell model. There are also freely available libraries such as Cloo.dll and OpenClTemplate.dll which assist in the integration of OpenCL.

There were a couple of disadvantages in using C#. Firstly the UEA (University of East Anglia) cluster did not support C# which would mean that any parameter sets that we could not represent as partial differential equations would have to be run in series. However as this was only one parameter set and preliminary testing showed that the simulation model resolves very quickly it was decided that this would not have a large impact on the time to do the research. Secondly C# is a Windows platform specific programming language and would therefore limit the portability of our software. However, as this research was investigating a specific hypothesis within a specific cell type it was decided that the software was likely to be for personal use only and therefore the lack of portability would not be an issue.

To produce the three dimensional meshes used in our graphical simulation we used the Blender software.

4.2.5 Implementation and Testing

4.2.5.1 IDEs

The code for all projects, including those developed in C# and C, was implemented using the Visual Studio IDE using the XNA framework for graphical rendering.

Whilst the kernel code was written in Visual Studio as it was easier to debug within that environment, we also used the OpenCLCodeChecker utility to ensure that the code would compile correctly before sending it to the graphics card . The OpenCLCodeChecker is a small utility either outputs a green light stating that the code would successfully run on the graphics card or a red light complete with an error log to show where the errors are in the code.

4.2.5.2 Graphical Simulation Model

The first model to be built was the graphical simulation model and this was done in stages. First the microtubule class was constructed and a test harness was used to ensure the microtubules were behaving as expected. The second stage was to implement the graphical representation of the cell and microtubules. Finally the

simulation logic was added and therefore knowing that both the microtubules and the graphical representation worked, we could use not only the numerical outputs of the simulation but also observe in real-time whether the simulation was running correctly.

Figure 4.3 shows a sample run of the graphical simulation model used to ensure that the microtubules are behaving as expected. We observe that at the beginning of the simulation, Figure 4.3b the vast majority of the microtubules are present on the centrosomal array. As the simulation progresses we observe microtubules encountering the attachment zone and being pulled onto the side wall. At 60 minutes the side wall array is well established (Figure 4.3f) whereas the centrosomal array has almost disappeared.

4.2.5.3 Simulation and PDE Models

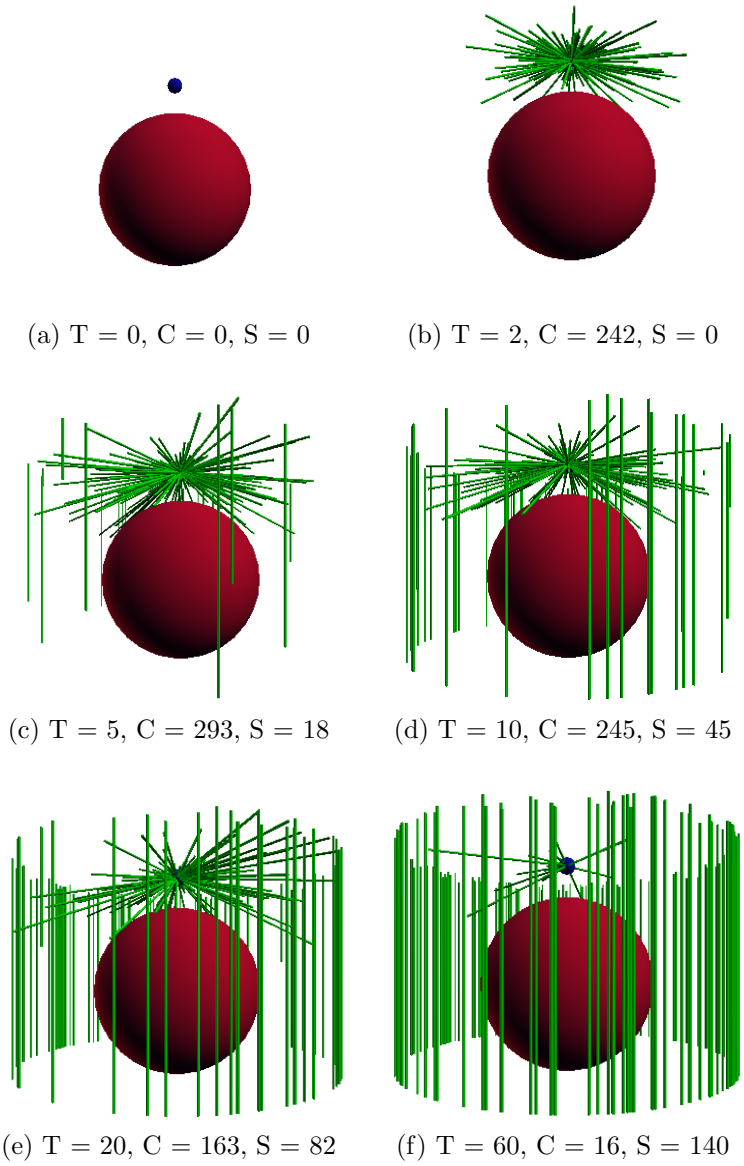
Once we had validated the simulation model we implemented the PDE model and validated it against an average of 1000 runs performed on the simulation model with the results shown in Figure 4.4. We observe very high correlation in both the growing and shrinking populations in both the centrosomal and side wall microtubule arrays showing that the PDE model is correctly implemented. Both the simulation and PDE code we then ported into kernels for use in OpenCL, using an intermediary step of coding them in C to assist in the debugging. The kernel code was then validated using the OpenCLCodeChecker utility before the parameter searches were performed.

4.3 Modelling microtubules in Cortical Plant cells

4.3.1 Requirements

Based on our aims and objectives the software requirements are to be able to:

1. Simulate cortical microtubules and calculate their order.
2. Produce a graphical output to visualise the microtubule array in aid in presentation and validation of results.



(a) T = 0, C = 0, S = 0

(b) T = 2, C = 242, S = 0

(c) T = 5, C = 293, S = 18

(d) T = 10, C = 245, S = 45

(e) T = 20, C = 163, S = 82

(f) T = 60, C = 16, S = 140

Figure 4.3: A sample 60 minute run of the MDCK graphical simulation model used to validate the simulation with the side wall creating negative feedback on the centrosomal nucleation rate. T = time(min), C = number of centrosomal microtubules, S = number of side wall microtubules. We initially see a rapid expanse of the centrosomal array which then shrinks as the side wall array increases.

3. Perform parameter searches targeting spontaneous catastrophe and rescue rates, collision induced catastrophe, and katanin response time and observe

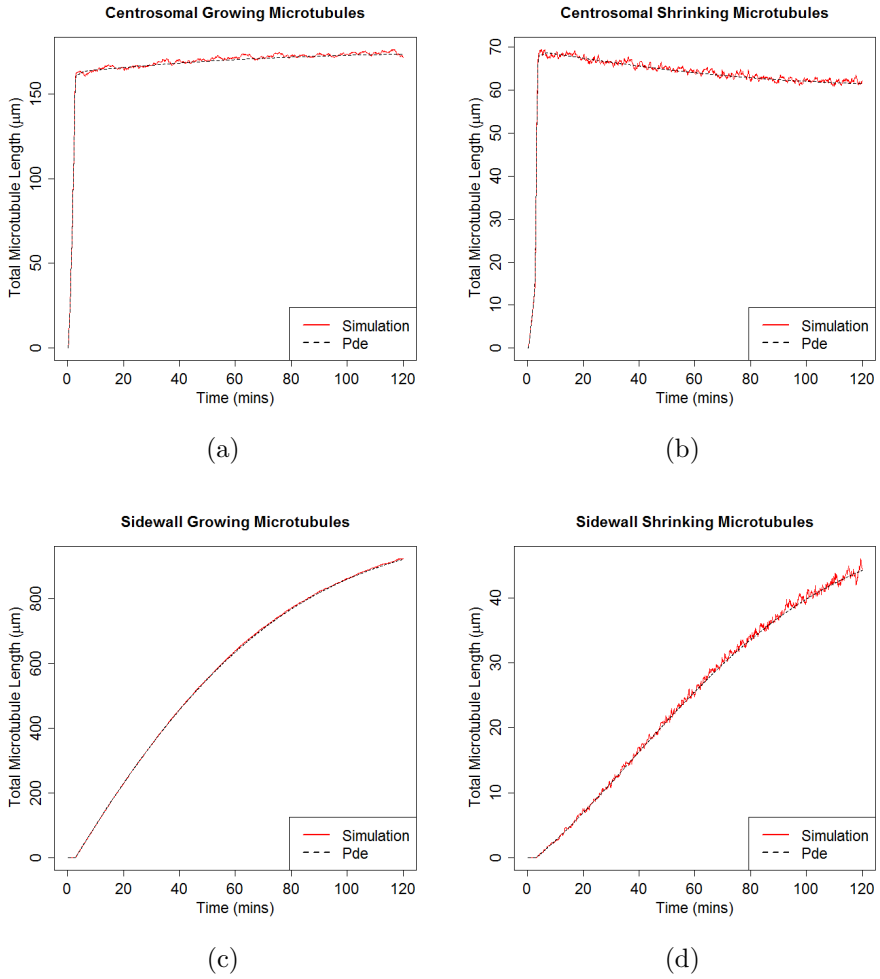


Figure 4.4: Comparison of the MDCK Simulation and PDE models. The data shown for the simulation model is the average of 1000 runs. The simulation and PDE data is very highly correlated with Pearson correlation coefficients greater than 0.998 in all four populations.

their effect on order.

4. Be used by other microtubule researchers.

As we determined in our methodology that a mathematical model would not be suitable for this research we can use a simulation model to satisfy all these requirements.

4.3.2 Graphical Simulation Model Design

In this model there are only two physical components; the cell membrane and the microtubules. As the cortical microtubules grow along the cell membrane the simulation environment can be simplified to two dimensions. As such the cell membrane can be represented as a box with each of the edges being a periodic boundary simulating that a cortical microtubule is able to grow all the way around the cell membrane and back to itself. Although in a 2D environment a microtubule might be best represented as a rectangle (flattened cylinder) we decided to simulate them as a line as the diameter of the microtubule is very small relative to the size of the cell and would increase the speed at which collisions could be resolved.

The class diagram of our cortical model is shown in Figure 4.5. Whilst our intention was to re-use as much code from the MDCK model as possible, it was mostly insufficient due to the added complexity of microtubule to microtubule interactions present in the cortical model. In the cortical model we define a microtubule as being a collection of microtubule subunits (Mts). This is because the addition of entrainment can cause microtubules to have sections with different directions and the periodic boundaries mean that microtubules would not be able to presented as a continuous line. Another class is for the storing of crossover events and tracking them throughout the simulation so that they can later be acted upon by katanin. It stores the two microtubules involved in the crossover, the respective microtubule subunits and the point of crossover and implements a method for simulating the effect of katanin severing said crossover.

There are two output classes; GraphicalVersion and LineCol. Both of these classes are largely identical and accept parameters as command line arguments but differ in their output. The GraphicalVersion, as its name suggests, outputs a real time graphical model of a single simulation along with the microtubule order, number and length. The purpose of this class is to get a visual output of how combinations of parameters can effect ordering whilst providing an insight into what patterns may emerge as additional information to the order parameter. LineCol, on the other hand, does not have any of the graphical parameters and instead is designed to repeat a simulation for a given set of parameters multiple

times and then output the means of the microtubule order, number and length at time intervals in order to provide a more robust result.

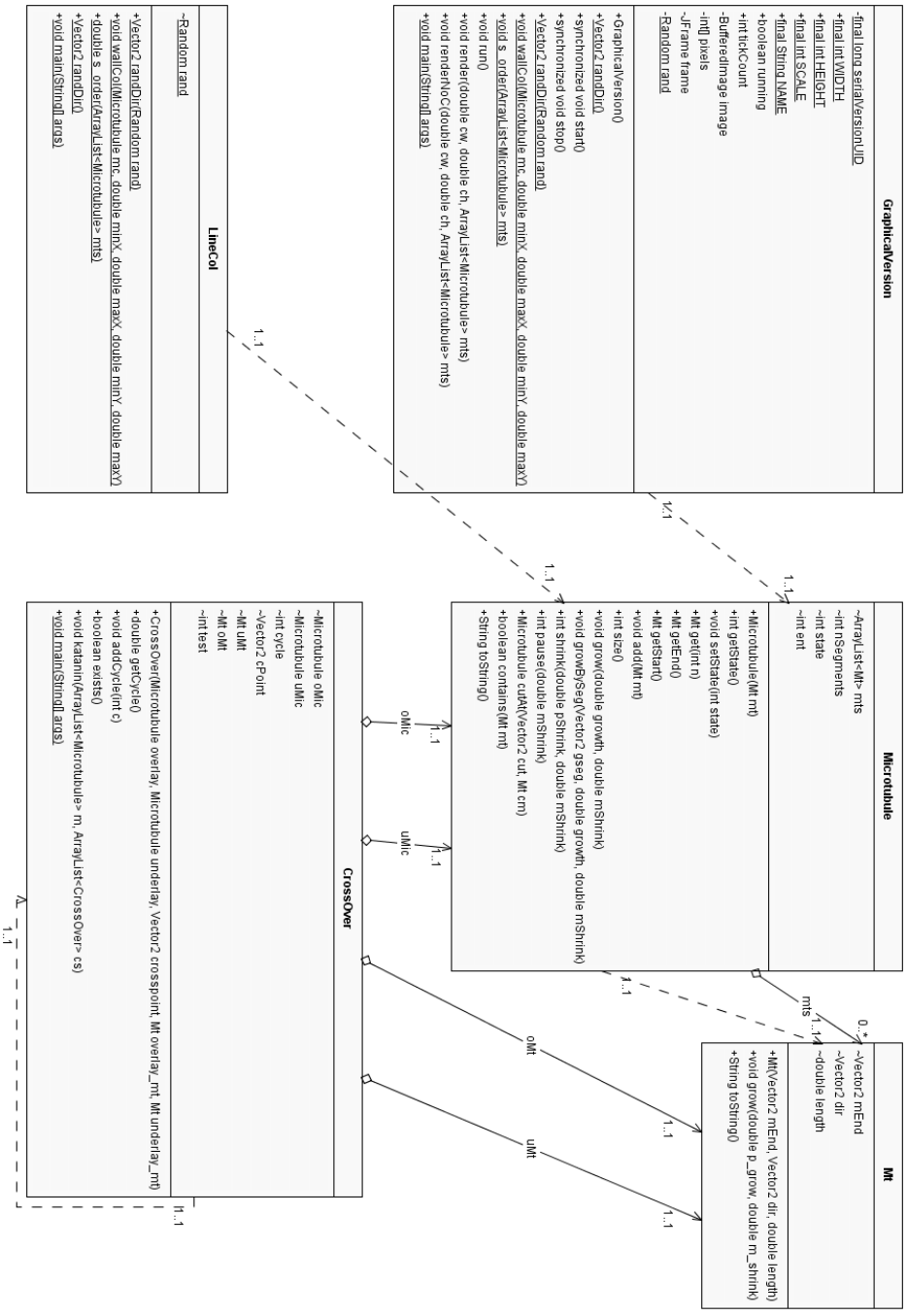


Figure 4.5: The cortical microtubule model class diagram.

The design logic of the simulation is shown in the flowchart in Figure 4.6 with microtubules entering the model through random nucleation throughout the cortex and being removed once their length is less than or equal to zero. In addition to the logic shown in the flow chart, katanin may act on the overlying microtubule present in a crossover (which may be either growing, shrinking or paused), severing it at the crossover site and adding the upstream element of severing site as a new microtubule into the shrinking population and the downstream element, now shorter, remains in its original state.

4.3.3 Parallelised Simulation Model Design

As we did not believe it feasible to produce PDE models of the cortical microtubules we had to parallelise the simulation model in order to perform parameter searches. As the cortical microtubule model is more computationally expensive than the MDCK model this would limit the scale at which the parameter searches could be performed. There were two target areas for parallelisation and our choice was influenced by the nature of our cluster hardware. Firstly we could parallelise the parameter search itself and for each parameter set run each simulation multiple times in series. This could be counter productive as although it would allow us a larger parameter search scope the longer time it would take for each parameter set to run would put it on the long queue within the cluster and make it more vulnerable to congestion particularly when running parameter sets that increase the microtubule count and thus the runtime. Alternatively we could parallelise a given parameter set performing 200 repeats (on the short queue of the cluster) before cluster throttling would begin. Initial experiments suggested an approximate run time of 5-10 minutes for a single model of a simulated 60 minute duration. By parallelising 200 parameter searches and repeating each set 200 times in series we would be looking at an approximate run time of 1000-2000 minutes. By parallelising each parameter set 200 times and performing a smaller set of say 20 parameter sets in series we would be looking at a much lower estimated time of 100-200 minutes. The latter was the approach that we used as we did not believe that much would be gained by performing very high resolution parameter searches and that parallelisation of the repeats would generate results

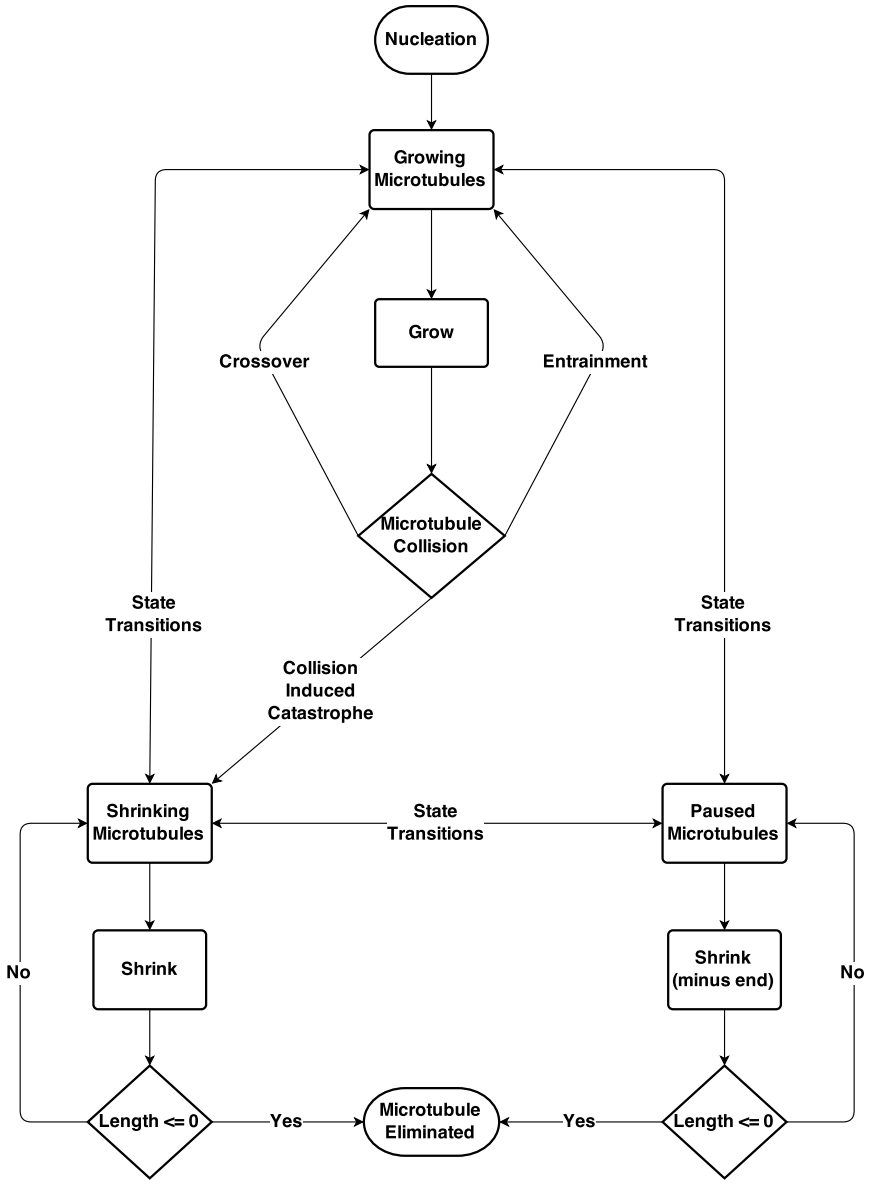


Figure 4.6: A flowchart of the behaviour of the cortical simulation model. The state transitions indicate spontaneous switches between states that are not induced by interactions with another microtubule.

faster.

4.3.4 Programming Language

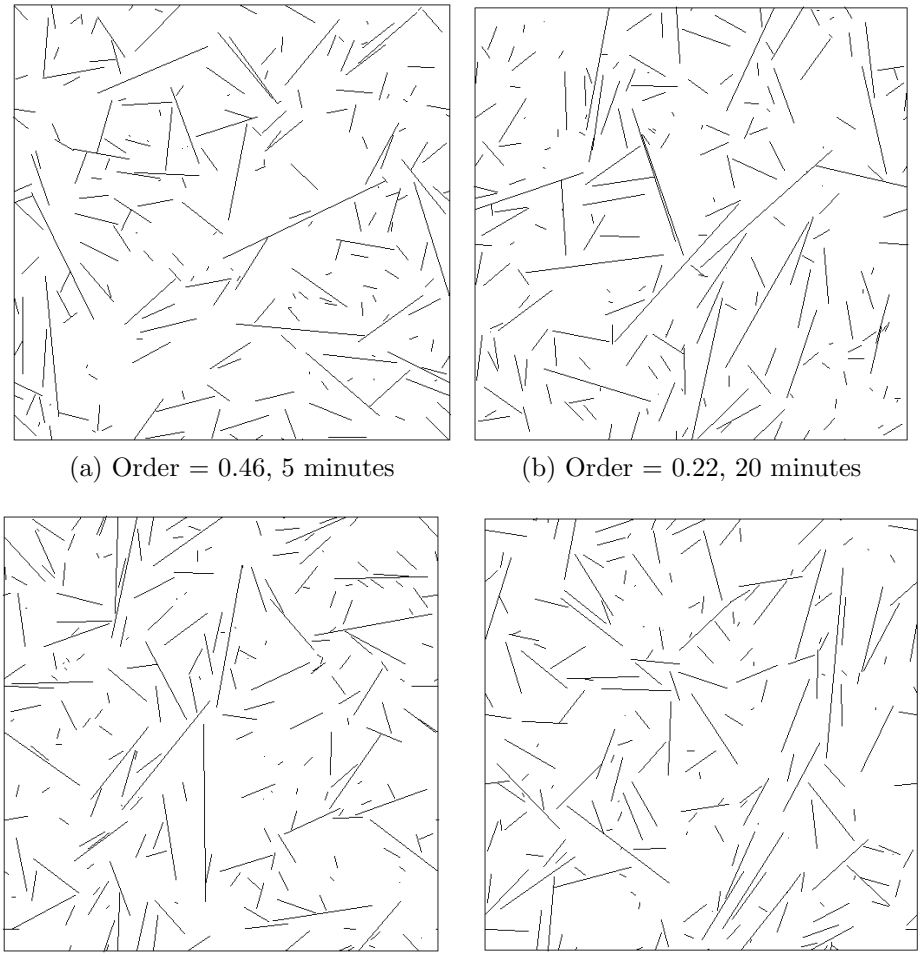
For this software it would not be beneficial to use C# as the parameter searches would have to be run on the UEA cluster which does not support C# and it would be an inefficient use of time to code the graphical version in one language and the cluster version in another language especially it would just be the graphical version minus the graphics. In addition a requirement of the GUI is that it can be used by other researchers in the field and the platform dependence C# has with the Windows operating system would be disadvantageous. Therefore, for this software, we decided to use the Java programming language which is supported on the UEA cluster. Like C# it is a good object orientated language that we were already familiar with. It has the benefit of being platform neutral and will run on any system with a Java Runtime Environment installed which would greatly increase the potential audience of our software. Whilst Java does have OpenGL bindings for graphical output we will only be representing a 2D environment and therefore the native functions in Java will be sufficient .

4.3.5 Implementation and Testing

4.3.5.1 IDEs

As Java was used as the programming language we used the NetBeans IDE for the cortical model. This IDE was used due to author familiarity and also the prevalence of plug-ins for the automatic generation of class diagrams.

We developed the model using a procedural approach and used the graphical output and the order parameters for validation. Firstly we implemented the model without entrainment or crossover to observe whether the microtubules were growing correctly and that the collision detection code was working as intended. A sample test run of a 60 minute simulation is shown in Figure 4.7. There are no crossovers observed and all microtubules are within the cell environment showing the collision detection code has been implemented correctly. An ordered array is not expected as collision based catastrophe is not sufficient to create ordered



(a) Order = 0.46, 5 minutes

(b) Order = 0.22, 20 minutes

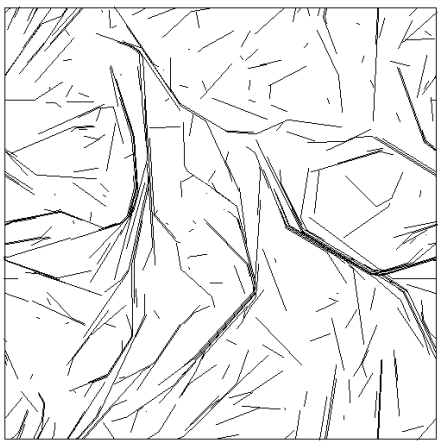
(c) Order = 0.12, 40 minutes

(d) Order = 0.51, 60 minutes

Figure 4.7: Testing the cortical microtubule implementation without entrainment or crossovers at 5, 20, 40 and 60 minutes into the simulation. Order ranges from 0 to 1 where 1 is perfectly ordered. The microtubules are not expected to reach an ordered state.

arrays.

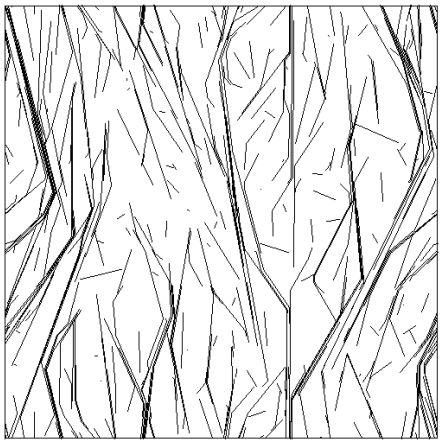
Next we enabled entrainment at an angle of 40 degrees whilst still preventing crossovers to ensure that the collision detection is not compromised and a sample test run is shown in Figure 4.8. We observed no crossovers and clear entrainment as bundles of microtubules form. We observed an increase in order as the simulation progresses leading to a single dominant domain by 60 minutes (Figure 4.8d).



(a) Order = 0.15, 5 minutes

(b) Order = 0.32, 20 minutes

(c) Order = 0.68, 40 minutes



(d) Order = 0.75, 60 minutes

Figure 4.8: Testing the cortical microtubule implementation with entrainment and no crossovers at 5, 20, 40 and 60 minutes into the simulation. Order ranges from 0 to 1 where 1 is perfectly ordered. The microtubules are expected to reach an ordered state within 60 minutes.

Once we were satisfied with the graphical version and the microtubule behaviour, the graphical components were simply removed and the code was ready to be run on the cluster with no additional testing required. A script was written in Python to collate and average the results from each cluster run of a particular parameter set.

4.3.6 Graphical Model Evaluation

The model we developed has the following feature set which would make it very useful for use in future studies.

- A 3 state model with customisable dynamic instability parameters. The model can be reverted to a 2 state model simply by zeroing the transition parameters to the paused state.
- Coded in an object orientated language so easier to add new features.
- Platform independence with good cluster suitability.
- View the simulation in real time.
- Modifiable entrainment.
- Modifiable CIC/crossover rates.
- Tracking of microtubule crossovers.
- Katanin action at crossover sites.

The model does, however, have some limitations. For example the model does not currently offer microtubule dependent nucleation (branched/parallel) although this could be a straightforward future addition. Our model also simulates the plant cortical membrane as a single two dimensional area with periodic boundaries at all four edges. The is an adequate and accepted approximation of a three dimensional cell [Allard et al., 2010]. Eren et al. [2010] did propose a three dimensional cylindrical model, however, their claim that it is three dimensional is questionable given that there is only one growth face (the side of the cylinder - the top and bottom of the cylinder act as catastrophe inducing or reflective boundaries) and therefore it appears it is really a two dimensional model which is then mapped onto a cylinder when visualised. The same could be emulated on our model through periodic boundaries at the sides and catastrophe/reflective boundaries at the top and bottom. However, a clear advantage of a multi-faced three dimensional model, such as used in [Ambrose et al., 2011] would be if you

want to simulate different growth parameters and structures in different locations of the cortex (such as at the top versus the base) which would be difficult in our model.

Chapter 5

Modelling the *Bellett* hypothesis of side wall formation in MDCK cells

5.1 Introduction

In this chapter we present our research in modelling the *Bellett* hypothesis using the mathematical and simulation models described in Chapter 4. We begin by defining the relevant geometry of the MDCK cell and the dynamic instability parameters used. We then describe briefly the behaviour of the simulation and our partial differential equation (PDE) implementation. We present the results of parameter searches on tubulin concentration dependent dynamic instability and side wall dependent centrosomal nucleation rate. We find that a side wall dependent nucleation rate is the best fit for the *Bellett* hypothesis and propose that the reason for the shrinking of the centrosomal array is due to the loss of ninein to the side wall.

5.2 Cell Geometry

The cell is modelled as a cylinder containing a centrosome, a spherical nucleus and an adherens belt which functions as the side wall attachment zone. The

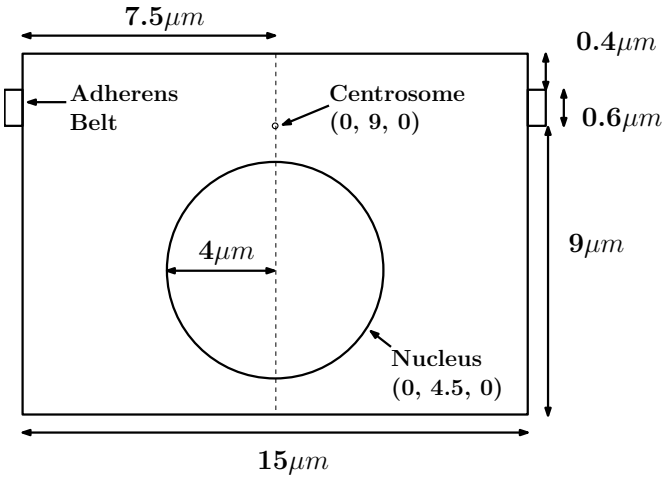


Figure 5.1: The dimensions and positions of the MDCK cell and its components used in our model.

dimensions of the cell and the positions and sizes of the components are shown in Figure 5.1. All microtubules nucleate at the centrosome and undergo catastrophe upon collision with the cell membrane or nucleus. Using the dimensions shown in Figure 5.1, the maximum length of a centrosomal microtubule is 8.43µm and the maximum length of a side wall microtubule is 9.6µm due to restriction from the geometry.

5.3 Model Parameters

The nucleation rate of the model is a constant 50 min^{-1}. In the absence of growth/rate data specifically for MDCK cells we use the microtubule growth and shrinkage equations and values proposed in [Walker et al., 1988] and [Janulevicius et al., 2006] to generate equations for the growth speed (v^+), shrinkage speed (v^-), centrosomal spontaneous catastrophe (r_{cc}) and rescue (r_{cr}) rates and side wall spontaneous catastrophe (r_{sc}) and rescue (r_{sr}) rates.

$$v^+ = ((k_a * [T]) - k_d) \quad (5.1)$$

$$v^- = k_s \quad (5.2)$$

$$r_{cc} = (C_c * [T]) + b_c \quad (5.3)$$

$$r_{cr} = (C_r * [T]) + b_r \quad (5.4)$$

$$r_{sc} = (S_c * [T]) + b_c \quad (5.5)$$

$$r_{sr} = (S_r * [T]) + b_r \quad (5.6)$$

where [T] is concentration of free tubulin. The units of v^+ is in tubulin molecules gained per minute and the units of v^- is tubulin molecules lost per minute. Both speeds are multiplied by 0.008/13 (length of tubulin divided by the number of tubulin molecules in a ring) to convert the speeds into µm min^{-1}. The unit of the rates is min^{-1} and the units and values and description of the remaining variables are shown in Table 5.1.

Table 5.1: The constants used to determine the growth, shrinkage, catastrophe and rescue rates, converted into min^{-1} from [Walker et al., 1988] and [Janulevicius et al., 2006].

Description	Constant	Value	Unit
Tubulin association (growth)	k_a	534	$\mu M^{-1} min^{-1}$
Tubulin disassociation (growth)	k_d	2640	min^{-1}
Tubulin disassociation (shrink)	k_s	43980	min^{-1}
Centrosomal catastrophe	C_c	-0.0348	$\mu M^{-1} min^{-1}$
Side wall catastrophe	S_c	-0.0348	$\mu M^{-1} min^{-1}$
Catastrophe	b_c	0.552	min^{-1}
Centrosomal rescue	C_r	0.3	$\mu M^{-1} min^{-1}$
Side wall rescue	S_r	0.3	$\mu M^{-1} min^{-1}$
Rescue	b_r	-1.8	min^{-1}

We estimate starting tubulin concentration of our MDCK cell to be 13.89 µM (all microtubules deploymerised) by taking the steady state concentration in [Janulevicius et al., 2006] and adding the tubulin of 150 depolymerised 9.3 µm side wall microtubules which we would estimate to be present at our steady state. This gives an initial (and max) v^+ of 2.94 µm min^{-1} and v^-, which is in-

dependent of tubulin concentration, is 27.06 µm min^{-1}. Whilst v^+ is within the bounds observed in MDCK cells [Bré et al., 1990], v^- appears to be unrealistically high as v^- observed in MDCK cells is approximately double the v^+ [Bowen et al., 2011]. Therefore v^- was set to 6 µm min^{-1}. v^+ and v^- are the same for both the centrosomal and side wall populations. The total simulated time of each model is 120 minutes and each model starts with all tubulin existing in an unbound, de-polymerised state. It should be noted that depending on the tubulin concentration both the rescue and catastrophe rates have the potential to be negative. At base C_c and S_c catastrophe rates become negative at tubulin concentrations greater than 15.86 µM so this is not an issue as our maximum tubulin concentration is 13.89µM. At base C_r and S_r rescue rates become negative when the tubulin concentration is less than 6µM which could happen as the microtubule count increases. However, as we decrease the catastrophe and rescue coefficients the more likely that the catastrophe and rescue rates will become negative. Should either the centrosomal or side wall rescue rate become negative it is treated as being zero.

5.4 Simulation

In the simulation microtubules are divided into four populations; centrosomal growing (m_c^+), centrosomal shrinking (m_c^-), side wall growing (m_s^+) and side wall shrinking(m_s^-). Microtubules are added to m_c^+ according to the nucleation rate, with an initial length of zero. Each time step ($\delta t = 0.001$ min) each microtubule undergoes a single event. m_c^+ microtubules either undergo spontaneous catastrophe at rate r_{cc} and are transferred to the m_c^- population or grow at speed v^+. If, after growth, they collide with the nucleus or cell membrane they undergo collision dependent catastrophe and are transferred to the m_c^- population. However if they collide with the attachment zone they are transferred to the m_s^+ population. The transfer to the m_s^+ population causes the minus end of the microtubule to move from the centrosome to the collision point and the plus end to be pulled vertically down towards the base of the cell so that the length of the microtubule is conserved. If the length of the microtubule becomes less than zero it is removed from the model. m_s^+ microtubules either catastrophe at rate r_{sc} and are

transferred to the m_s^- population or grow at speed v^+. Basal membrane proteins, which interact with the plus ends of microtubules, have been identified which may stabilise the side wall microtubules once they reach the cell base [Hotta et al., 2010; Mogensen et al., 2002]. Therefore if the m_s^+ microtubule collides with the base of the cell, the microtubule stops growing and is considered to be fully stable and no longer subject to spontaneous catastrophe for the duration of the simulation. m_s^- microtubules either rescue back to the m_s^+ population at rate r_{sr} or shrink at speed v^- and are removed from the simulation if their length becomes less than zero.

According to the *Bellett* hypothesis, once a centrosomal microtubule has attached to the side wall, ninein needs to be transferred along the microtubule before it is then pulled down onto the side wall. This adds complexity to the model as one would need to consider a pause period for the ninein to arrive, the rate at which the microtubule is pulled by the dynein motor on the side wall and whether the minus end of the microtubule should be subject to dynamic instability once it becomes exposed from the centrosome. Therefore to simplify the model we have the microtubule instantly pulled down onto the side wall once it attaches. We do not believe that this will greatly influence the results as dynein has a motor speed of 5-10 μms$^-$1 [Cross, 2004] which is $\sim$200 times faster than the maximum growth speed of the microtubules in our model. Even if we consider that the motor may only operate at a tenth of its speed when pulling the microtubule there is still unlikely to be a sizeable change in the microtubule length in the time it takes to transfer to the side wall.

5.5 PDE implementation

Similar to the simulation, the partial differential equation system is derived by considering four separate microtubule populations, the growing and shrinking populations attached to the centrosome and the growing and shrinking populations attached to the side wall. This formulation is related to [Tindemans and Mulder, 2010] where one independent growing and shrinking population was considered without spatial restriction.

The coupled system is written as follows,

$$\frac{\partial m_c^+}{\partial t} = -v^+\frac{\partial m_c^+}{\partial l} - r_c m_c^+ + r_r m_c^- \tag{5.7}$$

$$\frac{\partial m_c^-}{\partial t} = v^-\frac{\partial m_c^-}{\partial l} + r_c m_c^+ - r_r m_c^- \tag{5.8}$$

$$\frac{\partial m_s^+}{\partial t} = -v^+\frac{\partial m_s^+}{\partial l} - r_c m_s^+ + r_r m_s^- + \Phi_{att} \tag{5.9}$$

$$\frac{\partial m_s^-}{\partial t} = v^-\frac{\partial m_s^-}{\partial l} + r_c m_s^+ - r_r m_s^- \tag{5.10}$$

where m_c^+ is the growing population attached to the centrosome, m_c^- the shrinking population attached to the centrosome, m_s^+ the growing side wall population and m_s^- the shrinking side wall population. Parameters which govern the behaviour of the system are the growth speed, v^+, the shrink speed , v^-, the spontaneous catastrophe and rescue rates, r_c and r_r. Φ_{att} represents the transfer of microtubules onto the side wall through the *Bellett* hypothesis.

As we model the MDCK cell as a cylinder we can calculate the length of a microtubule when it collides with the cell membrane/nucleus/attachment zone based on the nucleation angle of the microtubule, θ, where $\theta \in [0, \pi]$. Therefore for any given value of θ, $l_0(\theta)$ denotes the length of the microtubule at collision and $\theta \in [\theta_1, \theta_2]$ denotes the angles at which this collision occurs with the attachment zone. Therefore we can define Φ_{att} as:

$$\Phi_{att} = m_c^+ v^+ \delta(l - l_0(\theta))(\theta \in [\theta_1, \theta_2]) \tag{5.11}$$

To model collisions that occur outside the attachment zone and thus result in CIC we use the following two boundary conditions:

$$m_c^-(l_0,t) = \frac{v^+}{v^-} m_c^+(l_0,t), \theta \notin [\theta_1,\theta_2] \tag{5.12}$$

$$m_c^-(l_0,t) = 0, \theta \in [\theta_1,\theta_2] \tag{5.13}$$

The first shows the movement of microtubules to the shrinking population at their collision length when their nucleation angle would not cause them to collide with the attachment zone. This supplemented by the second equation which shows that there is no movement to the centrosomal shrinking population when their nucleation angle would put them into collision with the attachment zone as these collisions would result in the microtubules moving into the side wall population. In addition, we have the following boundary conditions:

$$m_c^+(0,t) = \frac{r_n}{v^+} \tag{5.14}$$

$$m_c^-(0,t) = 0 \tag{5.15}$$

$$m_s^{+/-}(0,t) = 0 \tag{5.16}$$

showing microtubules enter the centrosomal growing population at length zero according to the nucleation rate r_n and that in the centrosomal shrinking and both side wall populations there are no microtubules with zero length.

We also need to account for microtubules in the side wall population stabilising once they reach the base of the cell. We define this length of the side wall microtubule, which is dependent on the angle of the microtubule at nucleation, $l_{sb}(\theta)$ as:

$$l_{sb}(\theta) = l_{sw} + l_r(\cot(\theta) - \cot(\theta_2)) \tag{5.17}$$

where l_{sw} is the length between the attachment zone and the base of the cell and l_r is the radius of the cell. The count of side wall microtubules that stabilise, $m_s^0(t)$, can be defined as:

$$m_s^0(t) = v^+ \int_0^t m_s^+(l_{sb,t})dt \tag{5.18}$$

As the microtubules can only enter the side wall shrinking population through spontaneous catastrophe from the side wall growing population it follows that there can be no shrinking side wall microtubules of length greater than or equal to l_{sb}. Therefore:

$$m_s^-(l_{sb}, t) = 0 \tag{5.19}$$

5.5.1 Numerical solution

The coupled partial differential equation system 5.7-5.10 is solved numerically using a finite difference scheme. For each equation we discretise the microtubule length in terms of a uniform grid with N mesh points spanning from the minimum to the maximum microtubule length thus,

$$x_i = \frac{l(i-1)}{N-1}, \qquad i = 1, ..., N \tag{5.20}$$

Partial derivatives in x are written in terms of the three-point central difference formula,

$$\frac{\partial m}{\partial x} \approx \frac{m(x_{i+1}) - m(x_{i-1})}{2h} \tag{5.21}$$

where $h = x_i - x_{i-1}$, the separation between mesh points. At x_0 and x_{N-1} the central difference cannot be calculated and so the forward and backward differences are used respectively.

To compute the time evolution of the microtubule populations the right-hand side of the PDEs are integrated using the trapezoidal rule to obtain,

$$m(l, t_{j+1}) = m(l, t_j) + (\text{RHS terms})\,\mathrm{d}t \tag{5.22}$$

where $\mathrm{d}t$ is the separation between time-points.

5.5.2 Exploration of parameter space

Due to the difficulty of determining through experiment the parameters governing the search-and-capture system we proceeded with a thorough investigation of

the parameter space for certain paired combinations four key parameters. Our motivation for doing this was twofold. First we wished to study the influence of changing these parameters upon the overall behaviour of the system and second to see whether the *Bellett* hypothesis was sufficient to explain the phenomena observed within physiologically reasonable parameters.

To perform the parameter search we varied two of the four parameters (C_c, C_r, S_c and S_r) and kept the remaining two at their base values.: C_c vs. C_r, C_c vs. S_c and S_c vs. S_r. We chose the pairings C_c vs. C_r and S_c vs. S_r to maximise/minimise the stabilities of the centrosomal and side wall arrays respectively. C_c vs. S_c was chosen so that the relative stabilities of the two arrays could be investigated. In the two searches where the centrosomal rescue rate was not targeted by parameter search, the centrosomal rescue rate was set to zero so that the faster mathematical model could be used. This condition should be more favourable to the increase of the side wall array/decrease of the centrosomal array as with the base centrosomal catastrophe rate it is likely that the majority of rescue events would be on microtubules that have undergone CIC thus prolonging the existence of microtubules that are not targeted to the attachment zone.

The parameter range used for C_c and S_c was from -0.068208 to 0.0696$\mu M^{-1}min^{-1}$ in increments of -0.001392$\mu M^{-1}min^{-1}$. The parameter range used for S_r was from 0 to 1.188$\mu M^{-1}min^{-1}$ in increments of 0.012$\mu M^{-1}min^{-1}$. In each parameter search there were 100 x 100 models run. There are zero microtubules at initiation and the model was run for a simulated time of 120 minutes and output both the total length and number of microtubules in the centrosomal and side wall populations.

Table 5.2: Comparison between run times using an OpenCL (parallelised) implementation and a standard serial implementation when performing a parameter search with the simulation and PDE models.

Model	OpenCL	Serial Estimate	% Reduction
Simulation	8.3 hours	402 hours	97.9
PDE	1.5 hours	89 hours	98.3

Using the PDE model, all 10000 pairings for a parameter set were completed within a single OpenCL call. For the simulation model, each of the 10000 pairings was run 1000 times to derive a consensus for that pairing. Therefore each call

to OpenCL included 10 pairings with each pairing represented 1000 times (10000 models per call). To reduce the runtime of the simulation model the time step was increased to 0.01 mins. The total runtime for a parameter search using the PDE and simulation models using OpenCL compared to running the parameter search in series is shown in Table 5.2. As the runtime for the PDE model is independent of changes in the catastrophe and rescue rates the serial runtime estimate was obtained by multiplying the run time for single PDE model by 10000. The simulation model runtime, however, is dependent on the catastrophe and rescue rates with runtime increasing as the catastrophe rates decrease and the rescue rates increase and so the serial runtime figures were estimated using the base line parameters. We observe an approximate 98% reduction in run times using parallelisation with OpenCL.

5.6 Results

The primary aim of these investigations is to revisit and perform a thorough analysis of the *Bellett* hypothesis to see whether it holds under physiologically reasonable conditions. We performed these investigations using numerical simulation due to the challenges of making the necessary experimental observations.

It is clear that any model must address the following:

1. The effectiveness of the search-and-capture process for transporting centrosomal microtubules to the side-walls

2. The observed increase and subsequent decrease of the centrosomal microtubule population

3. The time scales involved for the dynamics and stabilisation of the system throughout the entire lifecycle of the search-and-capture process

4. What reasonable mechanisms might exist which account for any discrepancies between a detailed model of the hypothesis and current experimental observations.

5.6.1 Centrosomal Catastrophe Rate versus Centrosomal Rescue Rate

The heat maps in Figure 5.2 show the influence upon the microtubule populations as a function of the centrosomal catastrophe and rescue rates. The size and length of the side wall array is greatest when the centrosomal catastrophe and rescue rates are low (Figures 5.2b and 5.2d) as it increases the cycling of new centrosomal microtubules that are not directed to the side wall without destabilising microtubules that are directed to the side wall. Intuitively increasing the centrosomal catastrophe rate has a negative impact upon both the total length and size of the side wall array. At low centrosomal rescue rates, increasing the centrosomal catastrophe rate decreases both the total length and size of the centrosomal array but at high centrosomal rescue rates, increasing the centrosomal catastrophe rate has the opposite effect. A possible reason for this is that the side wall array decreases and therefore there is more tubulin available to the centrosomal microtubules. Increasing the centrosomal rescue rate increases the number of centrosomal microtubules as it directly prolongs their lifespan and increases the size and total length of the side wall array when the centrosomal catastrophe rate is not low. However, we do observe the total length of the centrosomal array shrinking if the rescue rate is raised too high as less tubulin slows growth rates. At low centrosomal catastrophe rates increasing the centrosomal rescue rate causes a drop then rise in the number of microtubules on the array but it should be noted that the length of the side wall array drops showing that few of the side wall microtubules are able to grow to their full length.

These heat maps highlight the complex interplay between the parameters and how some properties which might be assumed to increase the speed of the search-and-capture process (in this case increasing centrosomal microtubule stability) can actually have the opposite effect. Therefore the effectiveness of a search and capture process is dependent upon all parameters being correctly tuned.

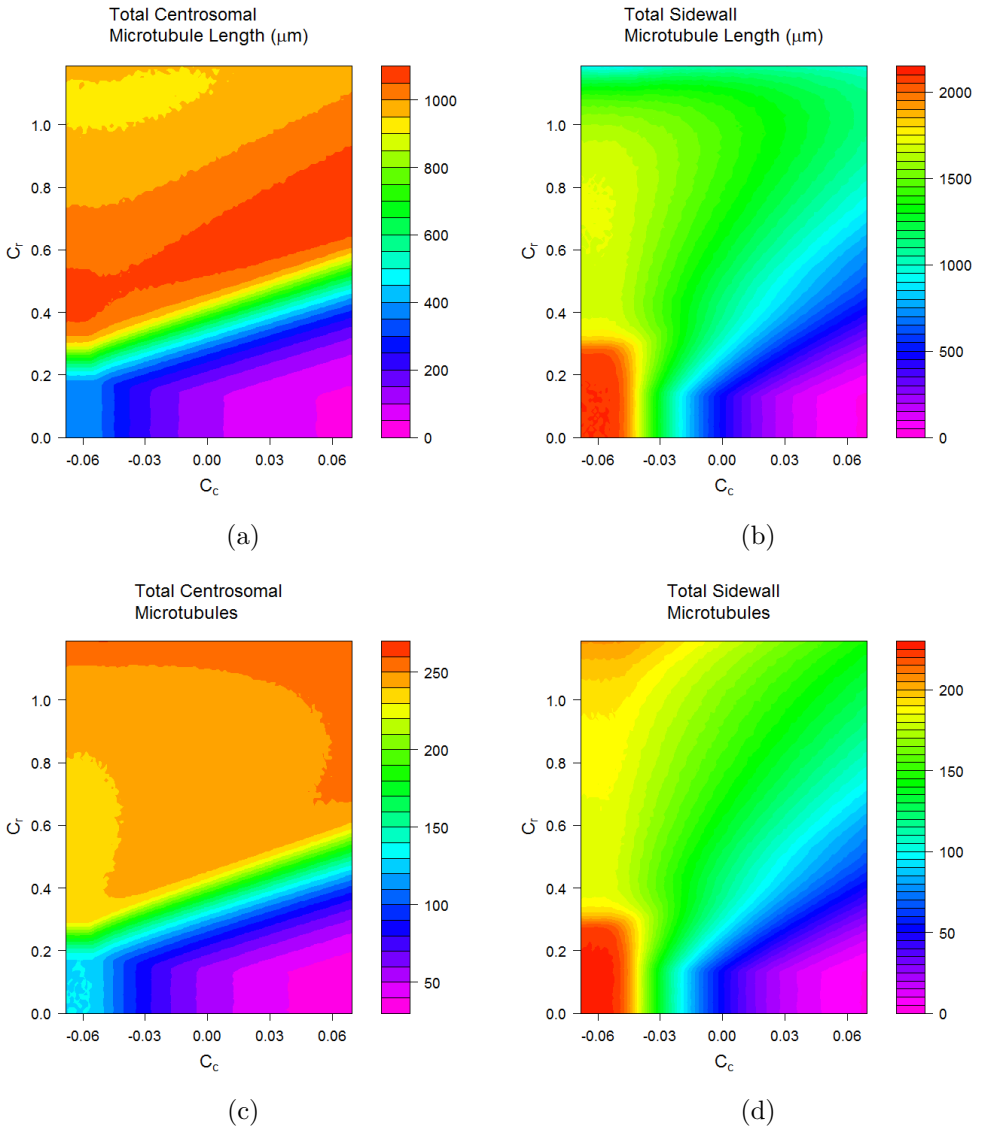


Figure 5.2: Parameter search: Centrosomal catastrophe versus centrosomal rescue after 120 minutes. The value of the colours is shown to the right of each map. Increasing C_r has a positive effect on centrosomal total length but becomes negative at high C_r (a). Increasing C_r has a positive effect on the number of centrosomal microtubules (b). In both centrosomal heat maps decreasing C_c has a positive effect at low C_r but this is the reverse at high C_r. Both the side wall total length (b) and the number of microtubules on the side wall (d) decrease with increasing C_c. In both we observe an increase when C_r is increases at high C_c but a decrease followed by an increase when C_c is low.

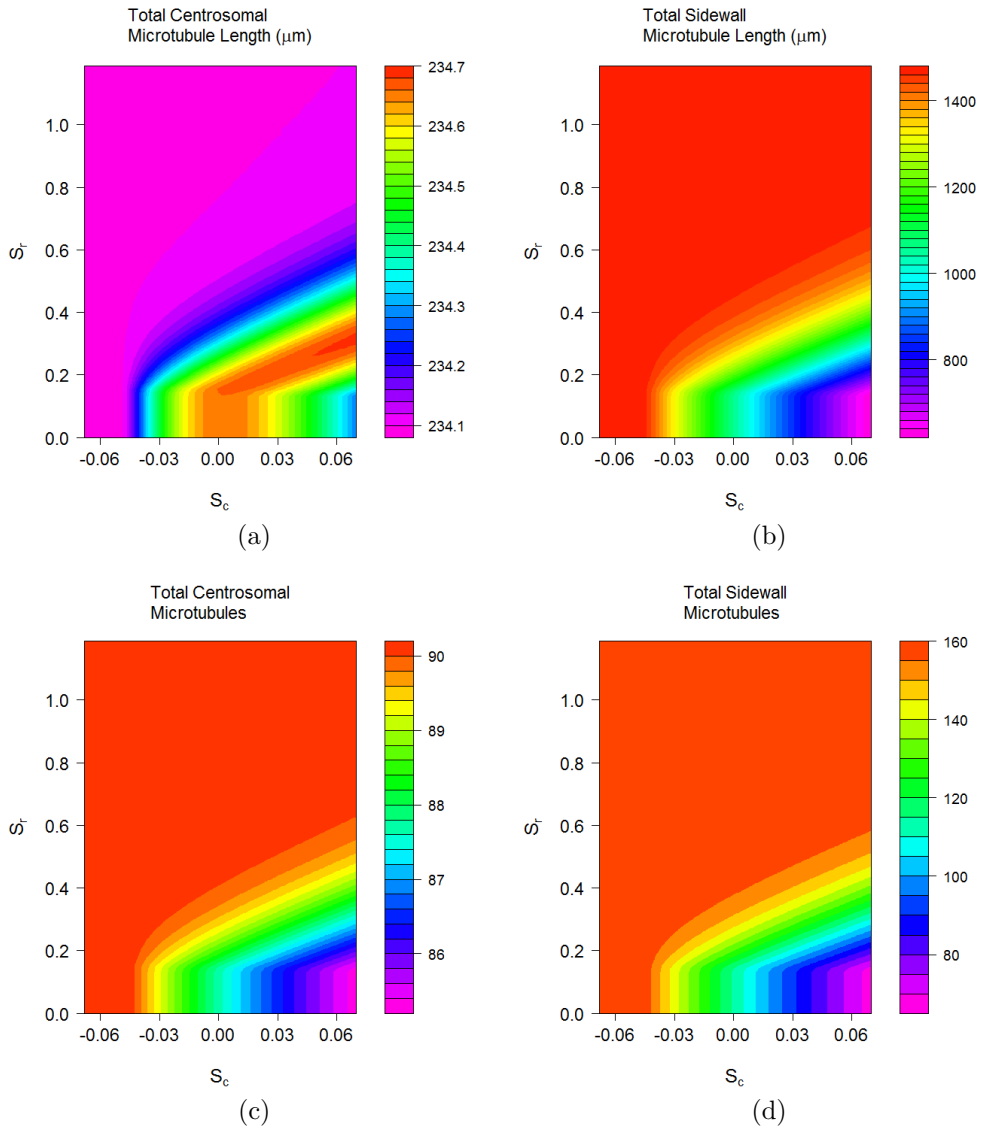


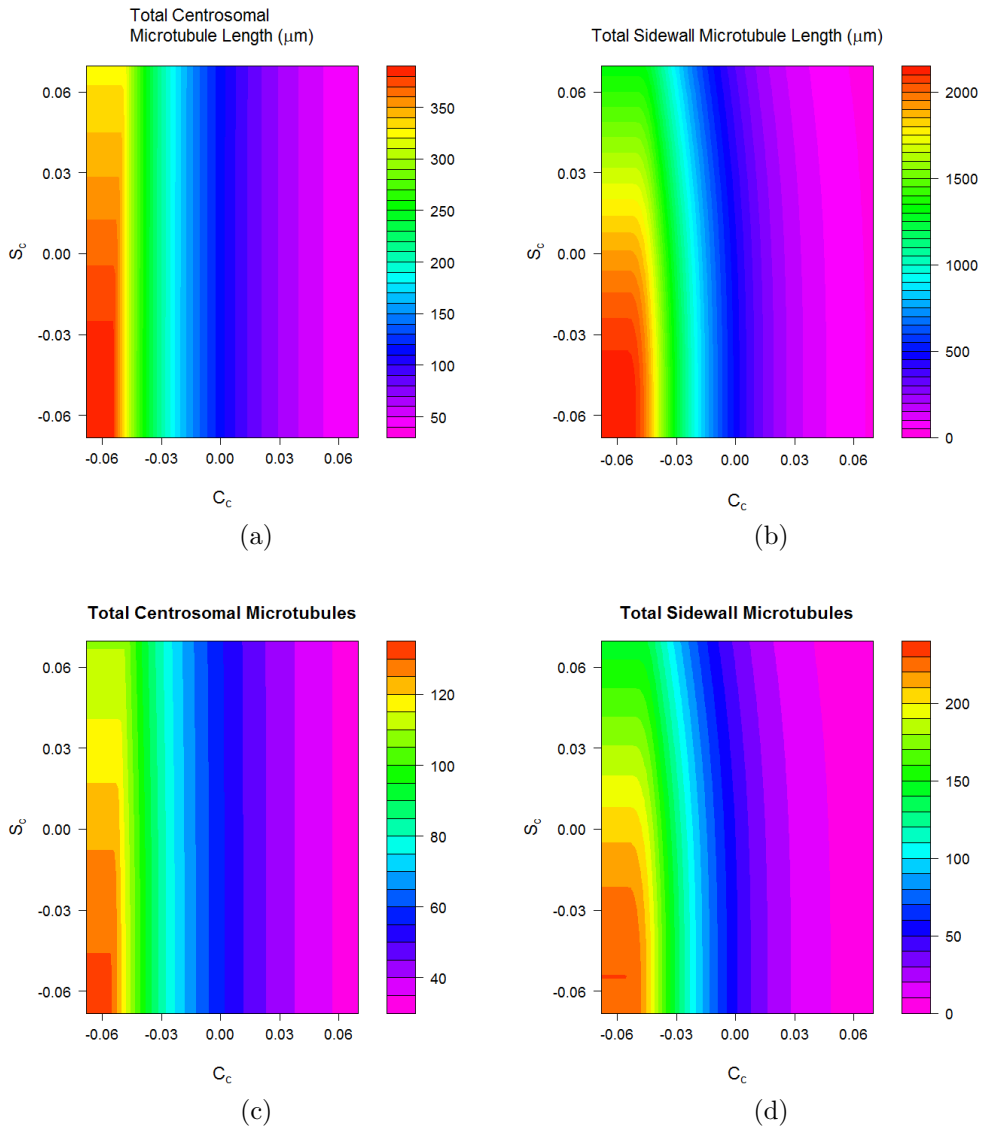
Figure 5.3: Parameter Search: Side wall catastrophe versus side wall rescue after 120 minutes. The value of the colours is shown to the right of each map. The total length of the centrosomal array shows a very low sensitivity to S_c and S_r with a less than 1 µm difference observed across the whole heat map (a). The total length of the side wall population (b), and both the number of centrosomal (c) and side wall (d) microtubules increases as S_c decreases and S_r increases.

5.6.2 Side wall Catastrophe Rate versus Side wall Rescue Rate

Figure 5.3 shows the influence of the side wall catastrophe and rescue rates upon the centrosomal and side wall populations. As might be expected the total length and number of microtubules in the side wall array increases with an increasing side wall rescue rate and a decreasing side wall catastrophe rate (Figures 5.3b and 5.3d). We observe that the positive effect of increasing the side wall rescue rate appears to cap out as increasing it further would have little affect if most catastrophes are already being rescued. We do see a similar relationship in the number of centrosomal microtubules (Figure 5.3c) likely due to a reduced growth rate reducing the chance of CIC but we only observe a difference of $\sim$4 microtubules across the parameter search. The effect on the total length of the centrosomal array is even smaller with less than 1 µm difference in the centrosomal population across the entire heat map (Figure 5.3a). This does appear to indicate that side wall population would not be capable of removing the centrosomal population simply through reducing the free tubulin concentration.

5.6.3 Centrosomal Catastrophe Rate versus Side wall Catastrophe Rate

Figure 5.4 shows the effect of the centrosomal and side wall catastrophe rates on the centrosomal and side wall populations. Both the number of microtubules on the centrosomal and side wall arrays (Figures 5.4c and 5.4d) and the total length of these two arrays (Figures 5.4a and 5.4b) decreases as the centrosomal rate. The effect of the centrosomal catastrophe rate dominates the side wall catastrophe rate as the centrosomal catastrophe rate directly influences the number of microtubules that the side wall catastrophe rate can act upon. Whilst it is intuitive that both the side wall total number of microtubules and total length decrease as the side wall catastrophe rate increases, it is not intuitive that both the centrosomal number of microtubules and total length also decrease. This could be due to high amounts of free tubulin increasing the growth speed and thus the rate of CICs.



(a) (b) (c) (d)

Figure 5.4: Parameter Search of Centrosomal catastrophe versus side wall catastrophe after 120 minutes. The value of the colours is shown to the right of each map. In all four heat maps we observe the same trend. Numbers and length decreases as C_c increases. Increasing S_c, however, causes a reduction in all four heat maps but this is only noticeable when C_c is low.

To investigate parameter sets where the side wall population is largest compared to the centrosomal population we plot the ratio of the total side wall microtubule length to the total centrosomal microtubule length (Figure 5.6.3). We

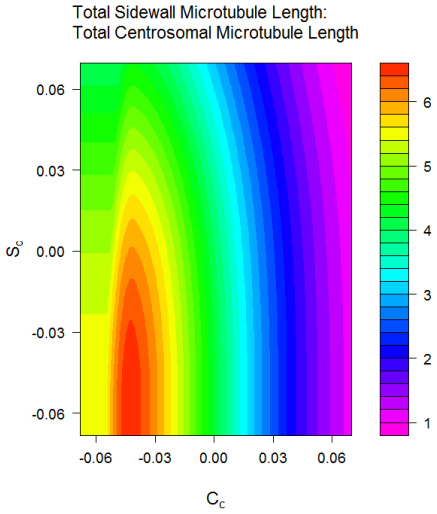


Figure 5.5: The ratio of the total length of the side wall array to the total length of the centrosomal array in the centrosomal catastrophe rate versus side wall catastrophe rate parameter search. The value of the colours is shown to the right of the map.

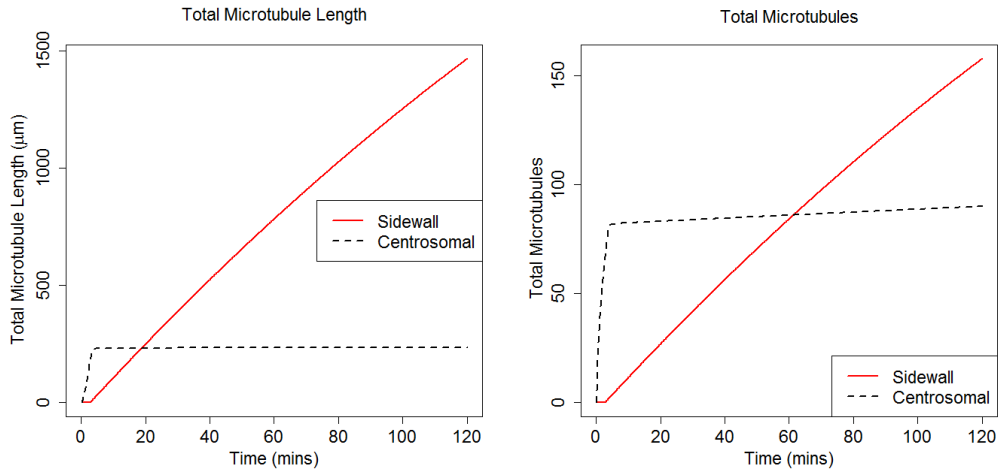
do observe that the side wall population is greatest relative to the centrosomal population when C_c is close to its base value of -0.0348µM^{-1}min^{-1} and S_c is decreased to -0.03828 µM^{-1}min^{-1}. If we use this parameter set whilst leaving S_r at its base value and C_r at zero we can see that the side wall population increases and reaches the ∼150 side wall microtubules that we would expect at 120 minutes but does not appear to be approaching a steady state(Figure 5.6). Also the centrosomal population does not show a decrease throughout the simulated 120 minutes which is at odds with the experimental observations.

5.6.4 Side wall dependent nucleation rate

To investigate the possibility of the centrosomal nucleation rate decreasing as the number of microtubules on the side wall increases we redefine the nucleation rate as:

$$nuc = base\_nuc\_rate - (base\_nuc\_rate * \frac{S_{tot}}{S_{max}}) \tag{5.23}$$

where S_{tot} is the total number of microtubules, shrinking, growing and stable,



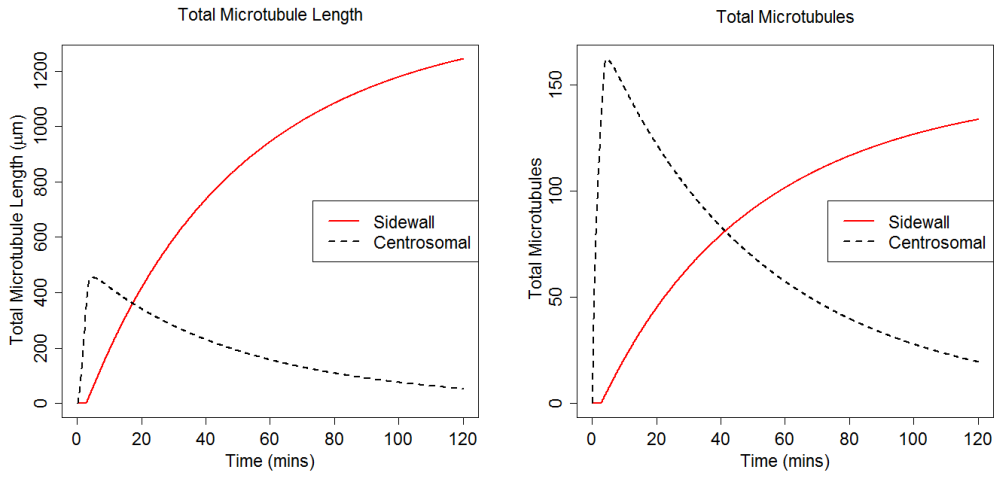
(a) (b)

Figure 5.6: Number and total length of the side wall and centrosomal microtubule populations using the estimated parameters from the parameter search assuming free tubulin concentration only dependence. The total length of the side wall population increases in a linear manner whereas the total length of the centrosomal population briefly increases in the first 5 minutes before flattening out (a). The total number of side wall microtubules increases linearly over time whereas the centrosomal population has an initial sharp increase in the first 5 minutes in the number of microtubules before sustaining a much slower rate of increase over the next 115 minutes (b).

that are present on the side wall and S_{max} is the maximum number of microtubules we expect to see on the side wall. The base nucleation rate is doubled from 50 min^{-1} to account for the decrease in rate as the model progresses and S_{max} is set to 150. This means that microtubule nucleation rate will be zero when we have the expected number of microtubules on the side wall.

We use the side wall dependent nucleation rate with the parameters used in Figure 5.6 in our mathematical model to observe the changes in the microtubule arrays over time (Figure 5.7). We observe a relationship that is very similar to that observed in live cells. There is a rapid increase in the centrosomal microtubule population both in number and length in the first 10 minutes of the model but then it rapidly shrinks as the side wall population increases. As the centrosomal population is now in decline the rate of increase in the side wall pop-

ulation also declines which causes the side wall population to approach a steady state of 134 microtubules and contains over twenty times more tubulin than the centrosomal array after two hours.



(a) (b)

Figure 5.7: Number and total length of the side wall and centrosomal microtubule populations using the estimated parameters from the parameter search using both a microtubule dependent nucleation rate and free tubulin concentration dependence. The total microtubule length of the centrosomal population rapidly increases until microtubules start transferring to the side wall population. At this point the total centrosomal microtubule length decreases towards zero whilst the side wall population increases past 1200 µm (a). This relationship is also observed in the total number of microtubules with the side wall population reaching 133 microtubules and the centrosomal population decreasing past 20 microtubules (b).

5.7 Discussion

5.7.1 Tubulin dependent dynamic instability

Our results suggest that free tubulin dependent dynamic instability is not sufficient to explain the formation of the acentrosomal side wall array in epithelial cells according to the *Bellett* hypothesis. Whilst with certain parameters we do

observe the side wall reaching a suitable size, it does not appear to be approaching a steady state and we do not observe a shrinkage in the centrosomal array. The reason for this is that the side wall array is more sensitive to changes in the stability of the centrosomal array than the centrosomal array is to changes in the side wall. This means that for a suitably sized side wall array to form the centrosomal array must have favourable stability parameters. However, the centrosomal array is then more resistant to the lowering tubulin concentration and therefore there is not sufficient negative feedback. A limitation to this model could be that the tubulin dependent dynamic instability parameters are linear and therefore are unlikely to bring about sudden changes in the parameters that would be required. Whilst tubulin dependent dynamic instability may not be a good overall fit for the *Bellett* hypothesis it does imply that the centrosomal array needs a low spontaneous catastrophe rate for a large side wall array to form. This is consistent with the recruitment of +TIP proteins on the centrosomal array described by Bellett et al. [2009] as they would act to stabilise the microtubule Akhmanova and Hoogenraad [2005].

5.7.2 Side wall dependent nucleation rate

We show that an arbitrary side wall dependent centrosomal nucleation rate is a good fit for the *Bellett* hypothesis. However, what is not clear is why an increased side wall microtubule population should reduce the nucleation rate at the centrosome. A possible reason for this is the transference of ninein from the centrosome to stabilise the minus end of the side wall microtubules. There is strong evidence that this is the case. Ou et al. [2002] show that the injection of ninein specific antibodies disrupts the centrosome's ability to function as an organising centre. In a recent paper, Shinohara et al. [2013] observe that the downregulation of ninein through a *Pax6* mutant and/or the direct knock out of ninein through RNA silencing causes a significant impairment of the centrosomal microtubule array in rat apical progenitor cells. Therefore we show that through computational modelling that the *Bellett* hypothesis for the formation of the side wall array can work in biologically reasonable conditions but we propose that the reason for the shrinkage of the centrosomal array is due to the transference of

ninein to the side wall.

It would be beneficial for future studies to investigate the nucleation rate of a MDCK centrosome as the cell goes through differentiation to observe whether the shrinkage of the centrosomal array is due to an increased instability of the microtubules or a reduction in nucleation rate.

5.7.3 Benefits of OpenCL

In this work we also highlight the potential benefits of using OpenCL for parallelisation even when running non-mathematical models, achieving run time reductions of ∼98% compared to running in serial. More optimisation could potentially reduce this further. However, it is important to note that the simulation model was small without a high degree of complexity. For more complex simulations it may be more time efficient to use a cluster for parallelisation even if the amount of concurrency is less.

Chapter 6

Modelling the role of catastrophe, crossover and katanin in the self organisation of cortical microtubules

6.1 Introduction

In this chapter we present our research in modelling the role of sponaenous catstrophe, collision induced catastrophe(CIC) and katanin in the formation of the plant cell acentrosomal cortical array. Part of this chapter (not including the katanin modelling) has been published in [Mace and Wang, 2014]. An expanded version of the this paper, which includes the katanin modelling, has been submitted to an IET Systems Biology special edition journal 'Computational Models & Methods in Systems Biology & Medicine' and has been provisionally accepted subject to revisions.

We begin by describing the dynamic instability parameters used and the types of microtubule interactions. We show that the observed catastrophe and rescue rates in live cells produce an ordered model but the order is sensitive to changes from these rates. We show that, with entrainment, reducing CIC below 0.5 causes the cortical array to lose order. We show that katanin can restore order to the

Table 6.1: Wild type microtubule dynamic instability parameters used in the simulation. Data from [Allard et al., 2010; Kawamura and Wasteneys, 2008].

Parameter	Value
v_g^p	$6.50\,\mu\text{m min}^{-1}$
v_s^p	$12.0\,\mu\text{m min}^{-1}$
v_s^m	$0.53\,\mu\text{m min}^{-1}$
f_{gp}	$0.380\ \text{min}^{-1}$
f_{gs}	$1.590\ \text{min}^{-1}$
f_{sp}	$0.440\ \text{min}^{-1}$
f_{sg}	$1.990\ \text{min}^{-1}$
f_{pg}	$1.400\ \text{min}^{-1}$
f_{ps}	$0.700\ \text{min}^{-1}$

array but argue that at low CIC probabilities katanin would have to act infeasibly fast to be solely responsible for an ordered array.

6.2 Methodology

To model the microtubules we use a similar three state model (growing, shrinking and paused) proposed in [Allard et al., 2010]. Growing microtubules grow at their plus-end at rate v_g^p and switch to paused or shrinking states at frequencies f_{gp} and f_{gs} respectively. Shrinking microtubules shrink at their plus-end at rate v_s^p and switch to paused or growing states at frequencies f_{sp} and f_{sg} respectively. Paused microtubules remain static at their plus-end and switch to growing or shrinking states at frequencies f_{pg} and f_{ps} respectively. All three states of microtubule shrink at rate v_s^m at their minus-ends. The values used for these parameters are taken from data used in a simulation by [Allard et al., 2010] who obtained them from microtubules observed in wild type (WT) *Arabidopsis thaliana* cultured at 31°C [Kawamura and Wasteneys, 2008] and are shown in Table 6.1. It should be noted that there is an apparent conflict between the value of f_{gs} as 1.590 is used in [Allard et al., 2010] but the data in [Kawamura and Wasteneys, 2008] appears to suggest 0.590 with no apparent explanation for the discrepancy. For consistency we use the value presented in [Allard et al., 2010] in our model but also consider the effect of a significantly lower f_{gs} in our analysis.

6.2.1 Microtubule Collisions

The outcome of a microtubule collision is dependent on the angle of the collision, θ_c. If $\theta_c <= 40°$ then the incident microtubule entrains and begins growing parallel to the barrier microtubule with a distance between them of 25 nm [Chan et al., 1999]. If $\theta_c > 40°$, then either a collision induced catastrophe (CIC) occurs according to the CIC rate and the incident microtubule immediately enters a shrinking state or the incident microtubule grows over the barrier microtubule creating a crossover. When katanin severing is investigated we assume that it is uniformly spread throughout the cell, in a non-rate inhibiting amount, that it targets the overlying microtubule, and that the time it takes to reach and sever all crossover sites is constant.

6.2.2 Microtubule Order

To observe the order of the cortical microtubule array we use the order parameter S proposed in [Allard et al., 2010]:

$$S = \Sigma_i l_i(cos^2(\theta_i - \Omega) - sin^2(\theta_i - \Omega))/\Sigma_i l_i \tag{6.1}$$

As $cos^2(x) - sin^2(x) = cos(2x)$ we simplify this parameter to:

$$S = \Sigma_i l_i(cos(2(\theta_i - \Omega)))/\Sigma_i l_i \tag{6.2}$$

where i indexes each microtubule in the microtubule array and ranges from 1 to n where n is the total number of microtubules. Ω is the dominant angle of the microtubule array and θ_i the angle of the i^{th} microtubule and l_i the length of the microtubule. The dominant angle is defined as the angle which has the minimum total difference from the angles of all the microtubules weighted by the microtubule length.

In the case of an entraining microtubule, the microtubule is first divided into multiple microtubules at the points of entrainment before the order is calculated (a microtubule that has entrained twice becomes three microtubules each with a single growth direction). S is between 0 and 1 with 1 indicating all microtubules sharing the same direction.

6.2.3 Simulation Environment

The plant cell is modelled as a 10 μm by 10 μm square. The cell walls are considered periodic boundaries, such that a microtubule that reaches the cell wall will continue growing from the opposite side of the cell maintaining the same state and direction. Microtubules are nucleated in a growing state, with a random growth direction, in a random location within the cell at a rate $k_0 = 10\,\mu\text{m}^{-2}\text{min}^{-1}$ [Allard et al., 2010]. In this model we do not consider branched or microtubule dependent nucleation as it does not appear to enhance the cortical array [Allard et al., 2010]. To advance the simulation, we use a fixed time step of 0.001 min. Each time step, new microtubules are added according to the nucleation rate and existing microtubules change states according to the state change rates. Microtubules that do not change state are then grown or shrunk or paused at their plus-end depending on their current state. All microtubules are then shrunk from their minus-end and removed from the simulation if their length is less than or equal to zero. Microtubule collision detection is the most computationally expensive part of our model as each microtubule must be compared against all other microtubules to see if a collision has occurred and the more computation that is required, which may slow down the simulation. Therefore as a compromise between accuracy and speed, collision detection and crossover existence is resolved every five time steps instead of every time step. Throughout the simulation we assume that there is an abundance of free tubulin and therefore the dynamic instability parameters do not alter as the size of the microtubule array increases. We also assume that the microtubule nucleation rate is not limited and that the γ-tubulin rings are immediately removed once growth starts to coincide with shrinkage beginning at the microtubule minus end.

To investigate the effects of the state change rates we apply a sequence of multipliers to the target rate ranging from 0 to 2.0 in a step size of 0.2. All other parameters are kept at their WT value and the CIC rate was set to 1.0. When investigating the CIC to crossover rate and also the effects of katanin, the microtubule dynamic instability parameters were kept at their WT value.

Due to the high degree of randomness that is present in the model (microtubule nucleation location, microtubule growth direction, growth state changes)

a single run of the model would not be likely to produce a reliable result for a particular parameter set. Therefore the simulation was run 200 times for 60 minutes for each value of the parameter and the mean of the microtubule order calculated and presented in the Results section.

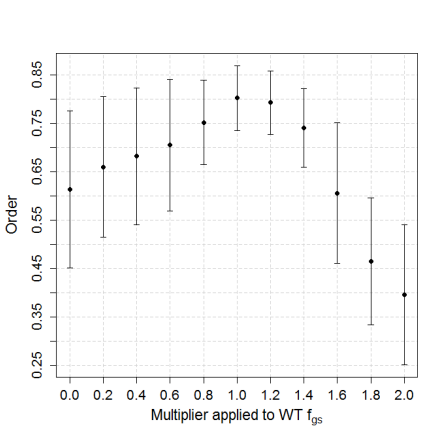
6.2.4 Software

The simulation model was implemented in Java and the UEA high performance computing cluster was used to run multiple simulations of the same parameter set. A separate version of the model which includes a graphical output was used for visual confirmation of the results.

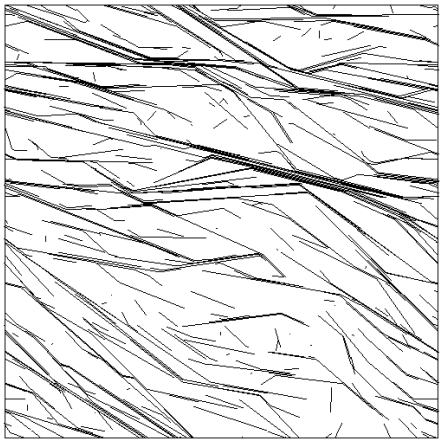
6.3 Results

6.3.1 The role of growing to shrinking frequency

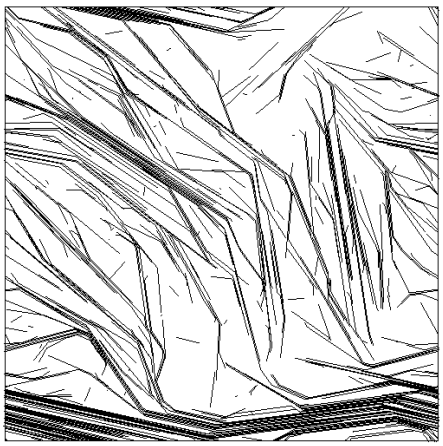
We observe that the order of the microtubule array is greatest when f_{gs} is at the WT value (Figure 6.1a). From the simulation snapshot (Figure 6.1b) we observe a clear single growth domain and evidence of microtubule bundling. Decreasing f_{gs} from its WT value causes the order to decrease with the rate of decline slowing as f_{gs} approaches 0. We observe that at $0{f_{gs}}^{WT}$ (Figure 6.1c) there are not only more microtubules than at the WT value of f_{gs} but also denser bundles. This is due to the average lifetime of the microtubule increasing and thus increases the chance of microtubules growing long enough to interact with each other. However, there is not a single dominant direction, with horizontal, diagonal and vertical domains present. A probable cause for this is that in the absence of f_{gs}, a larger proportion of rescue events will occur on microtubules that have undergone CIC (some catastrophe events will still occur spontaneously by going to the pause then shrinking state) and therefore by increasing the lifetime of microtubules that have conflicting directions, the more differing domains have the chance to occur. Were we to use the f_{gs} as indicated in [Kawamura and Wasteneys, 2008] (0.59), this would be equal to a multiplier of 0.37 which would result in an approximate order of 0.68 and would likely have multiple domains and as such would not be



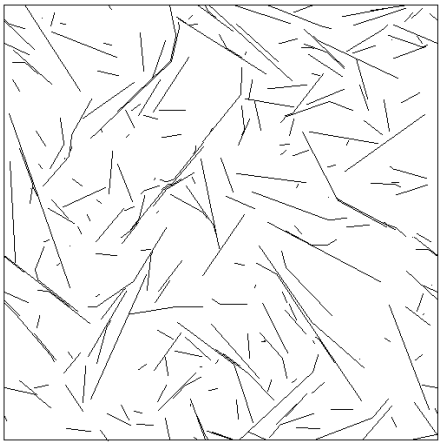
(a)



(b) S = 0.83, $f_{gs} = {f_{gs}}^{WT}$



(c) S = 0.57, $f_{gs} = 0{f_{gs}}^{WT}$



(d) S = 0.31, $f_{gs} = 2{f_{gs}}^{WT}$

Figure 6.1: The change in order of the microtubule array as a result of change in the growing to shrinking frequency, f_{gs}. (a) the mean order of 200 simulations at 60 minutes when applying a range of multipliers to f_{gs}. Error bars indicate $\pm$ standard deviation. The greatest order is observed when f_{gs} is at its WT value. Increasing f_{gs} past the WT value causes a much more rapid decline in order compared to decreasing f_{gs} past the WT value. (b) A snapshot of the model at 60 minutes using ${f_{gs}}^{WT}$. A clear single dominant growth direction is observed. (c) A snapshot of the model at 60 minutes using $0{f_{gs}}^{WT}$. Instead of a single dominant growth direction there appear to be two. (d) A snapshot of the model at 60 minutes using $2{f_{gs}}^{WT}$. The microtubule array is very sparse and does not have a clear dominant growth direction.

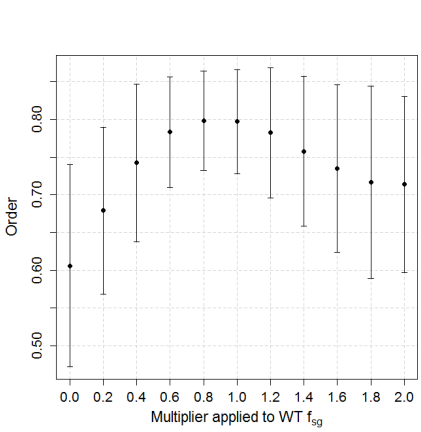
biologically reasonable. Increasing f_{gs} from the WT value causes a rapid drop in order. This due to the lifetime of the microtubules being decreased and therefore microtubules have reduced length and there are less interactions between them. This leads to sparse, unordered arrays as shown in Figure 6.1d where f_{gs} is double the WT value.

6.3.2 The role of shrinking to growing frequency

We observe a similar relationship with f_{sg} as we did with f_{gs} where increasing or decreasing the rate past a certain point causes the order of the microtubule array to decrease (Figure 6.2a). However, in the case of f_{sg} the highest order is not observed at the WT value but at ${0.8f_{sg}}^{WT}$. The snapshot for ${0.8f_{sg}}^{WT}$ shows a clear dominant vertical domain (Figure 6.2b). Lowering f_{sg} from ${0.8f_{sg}}^{WT}$ towards ${0f_{sg}}^{WT}$ causes a greater drop in microtubule ordering compared to increasing f_{sg} towards ${2f_{sg}}^{WT}$. This is due the spontaneous catastrophe rates no longer being opposed and as such the microtubule lifetime decreases leading to fewer interactions as evidenced by the snapshot for ${0f_{sg}}^{WT}$ (Figure 6.2c) where, although there are some areas of order, the array is sparse. The effects of decreasing f_{sg}, however, are not as pronounced as increasing f_{gs}. This is due to v_g^p being close to half v_s^p and therefore f_{sg} has a smaller time frame to act upon the microtubule. Increasing f_{sg} causes a decline in order as the life time of microtubules increases prolonging conflicting domains. At ${2f_{sg}}^{WT}$ we observe a greater increase in microtubule count and cell coverage although a less uniform microtubule direction. The loss in order in the ${2f_{sg}}^{WT}$ model is less than the ${0f_{sg}}^{WT}$ model as a proportion of the increased f_{sg} still acts to counteract spontaneous catastrophe rates.

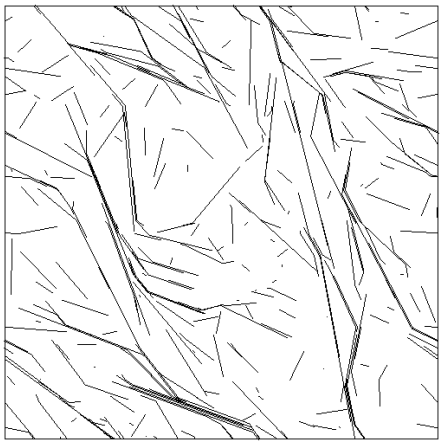
6.3.3 The role of collision induced catastrophe (CIC)

It has been reported that microtubules are able to form an ordered array in the absence of CIC but in the presence of entrainment [Allard et al., 2010]. To investigate this, we ran simulations using the WT parameters but altering the proportion of collisions greater than the entrainment angle that result in catastrophe or cause the microtubule to cross over the barrier microtubule with no

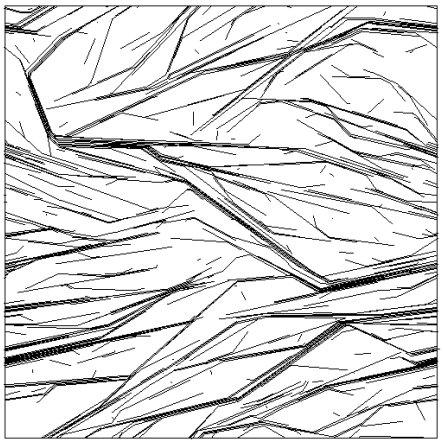


(a)

(b) S = 0.89, $f_{sg} = 0.8{f_{sg}}^{WT}$

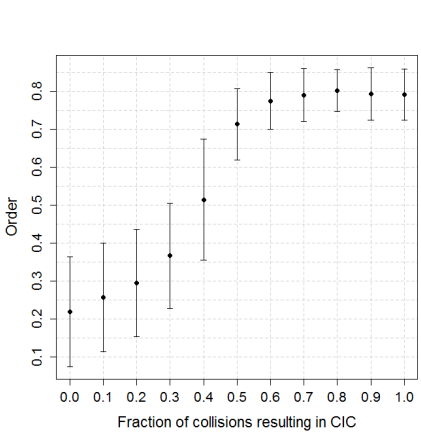


(c) S = 0.63, $f_{sg} = 0{f_{sg}}^{WT}$

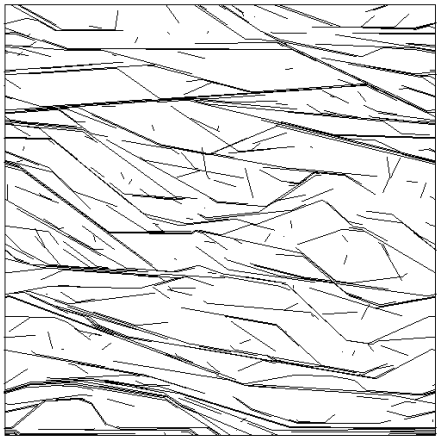


(d) S = 0.70, $f_{sg} = 2{f_{sg}}^{WT}$

Figure 6.2: The change in order of the microtubule array according to changes in the shrinking to growing frequency, f_{sg}. (a) The mean order of 200 simulations at 60 minutes when applying a range of multipliers to ${f_{sg}}^{WT}$. Error bars indicate $\pm$ standard deviation. Increasing f_{sg} past the WT value causes a decline in the order. There is a slight increase in order by decreasing f_{sg} to 0.8 of the WT value but decreasing f_{sg} further causes a sharp decline in the order. (b) A snapshot of the model at 60 minutes using $0.8{f_{sg}}^{WT}$. A single dominant growth direction is observed. (c) A snapshot of the model at 60 minutes using $0{f_{sg}}^{WT}$. The microtubule array is more sparse than the $0.8{f_{sg}}^{WT}$ model and a dominant growth direction less pronounced. (d) A snapshot of the model at 60 minutes using $2{f_{sg}}^{WT}$. The microtubule array is thicker than the $0.8{f_{sg}}^{WT}$ model but a single dominant growth direction is less clear.

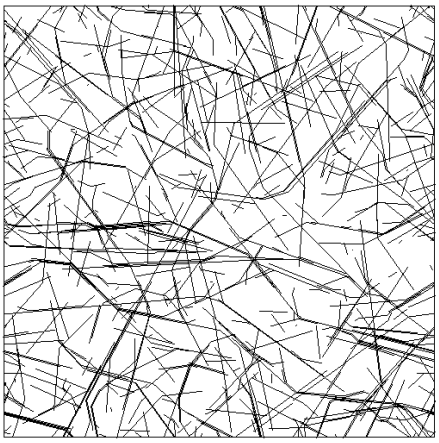


(a)



(b) S = 0.84, 0.8 CIC

(c) S = 0.63, 0.4 CIC



(d) S = 0.10, 0.1 CIC

Figure 6.3: (a) The mean order of 200 simulations at 60 minutes when changing the fraction of collisions greater than the entrainment angle that result in a CIC. Error bars indicate ± standard deviation. (b) A snapshot of the model at 60 minutes using 0.8 CIC. A clear single dominant direction is observed. (c) A snapshot of the model at 60 minutes using 0.4 CIC. Whilst the dominant direct appears to be vertical, there are many microtubules growing across it. (d) A snapshot of the model at 60 minutes using 0.1 CIC. There is no clear dominant direction of microtubule growth.

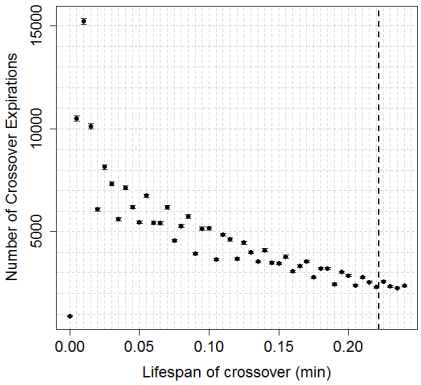
structural consequences. We observe that at high proportions of CIC (0.5 to 1.0) the simulation generates highly ordered microtubule arrays (Figure 6.3a). The highest order is observed at 0.8 CIC which could indicate a positive effect in

permitting some cross over events but the difference is unlikely to be significant. It is possible that, at high values of CIC, the chance for a microtubule to cross over an existing bundle of microtubules is very low and this is countered by the ability of microtubule bundles to pass random small barrier microtubules. With the exception of a few visible cross-over events, a snapshot of a 0.8 CIC simulation at 60 minutes is very similar to those observed in Figures 6.1b and 6.2b showing a single dominant direction domain.

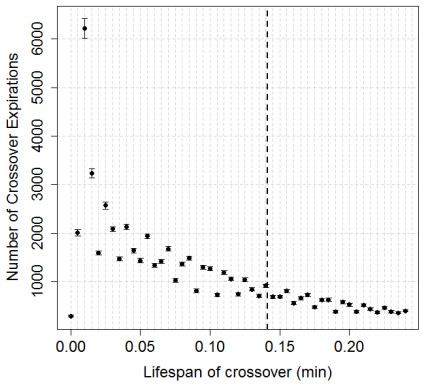
However, as the fraction of CIC decreases below 0.5 we observe a rapid decrease in order as CIC approaches 0 contrary to the findings in [Allard et al., 2010]. At 0.4 CIC (Figure 6.3c) there are signs of a dominant diagonal direction but also an increased presence of long, unbundled, conflicting crossover microtubules. At 0.1 CIC (Figure 6.3d) there is little evidence of any order at all. A reason for this is that low probabilities of CIC greatly reduce the level of interaction between microtubules as the incident microtubule can simply pass through conflicting barrier microtubules. Also the lifetime of the microtubule greatly increases as the spontaneous state change rates become the predominant/only method of entering a shrinking state.

6.3.4 Katanin and Crossovers

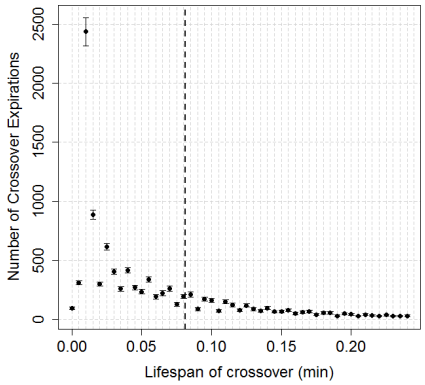
Figure 6.4 shows the distribution of natural crossover expirations (those that cease to exist solely due to the growth/shrinkage of the underlying/overlying microtubules and not through katanin severing) at CIC values of 0.1, 0.4 and 0.8 using the same parameters as Figure 6.3. We chose these CIC values as 0.1 is similar to the rate observed in [Wightman and Turner, 2007] in live experiments, 0.4 as it is here that the order drops rapidly compared to higher rates and 0.8 as we observed the greatest order there (Figure 6.3). As CIC decreases we observe both a higher number of crossover expirations and a greater mean time to expiration as such conditions increase the number of microtubules within the cell and also their individual life spans. In all three cases, the distribution of expirations is skewed to the left and the most frequent expiration time is 0.01 minutes. A contributing factor to this is that within this simulation, 0.01 minutes is the time it takes for a microtubule to crossover a barrier microtubule but then collide



(a) CIC 0.1, Mean 0.222



(b) CIC 0.4, Mean 0.141

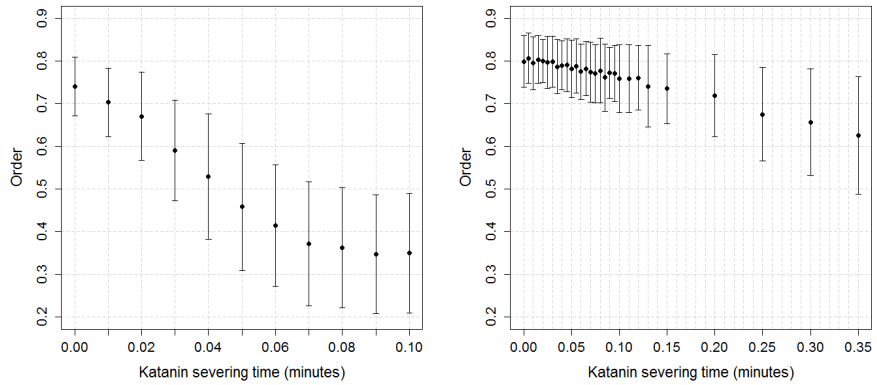


(c) CIC 0.8, Mean 0.081

Figure 6.4: The mean count of crossover expirations of 200 simulations at 60 minutes at CIC values of 0.1, 0.4 and 0.8 using the same parameters as Figure 6.3. Error bars indicate ± standard deviation and the black vertical dotted line shows the mean crossover expiration time. Only the first 49 time intervals are shown on the graphs.

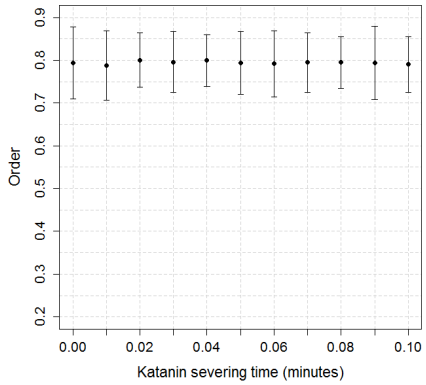
with an entraining microtubule and subsequently shrink back past the crossover point. Therefore we observe a higher frequency of 0.01 minute expirations relative to all other expirations as we increase the CIC proportion as it becomes more likely that a microtubule will undergo catastrophe if it collides with an entraining microtubule barrier microtubule post crossover.

Note that there are also crossovers showing an expiration time of 0 minutes.



(a) CIC 0.1

(b) CIC 0.4



(c) CIC 0.8

Figure 6.5: The mean order of 200 simulations at 60 minutes at CIC values of 0.1, 0.4 and 0.8 using the same parameters as Figure 6.3, but also simulating the role of katanin. We define katanin severing time as the time it takes for the katanin protein to localise to and sever a microtubule crossover. Error bars indicate ± standard deviation.

As crossover events and expirations are observed in 5 time step intervals as part of collision detection, then it is possible that if a microtubule nucleates very close to an existing microtubule, due to minus end shrinkage and plus end growth, the microtubule can completely pass over the barrier microtubule causing the crossover event to cease to exist in the same iteration.

We observe that at CIC 0.8 (Figure 6.5c) enabling katanin does not appear to

affect the order of the microtubule array even if the katanin was able to instantly sever nascent crossovers. This is because a highly ordered array already forms without the need for katanin and as such any microtubules growing contrary to that array will be swiftly removed through collision induced catastrophe, greatly dwarfing the effect of crossover severance.

On the other hand, at 0.1 CIC (Figure 6.5a) the katanin must localise and sever the crossover almost immediately (0-0.02 minutes) for it to create an ordered array. Longer than 0.08 minutes severing time gives the microtubule enough time to be extended far from the crossover point that its disruptive effect on the order of the cortical array is not mitigated by the severing. At 0.4 CIC (Figure 6.5b) we again observe an increase in order by enabling katanin but again observe a steady decline in order as the severing time increases. The rate of decline is much slower than at 0.1 CIC, with an order of greater than 0.7 still being observed with a severing time of 0.2 minutes as the higher CIC contributes to order through more induced catastrophes.

However is it biologically feasible that katanin is able to localise and sever a crossover site with 0.2 minutes (12 seconds)? The mean time for katanin to sever a crossover site has been observed at 41 ± 14 seconds [Zhang et al., 2013]. Whilst a direct comparison is difficult as that study was conducted at room temperature so one would expect lower kinetics compared to that of 31°C and it is not clear what the CIC rate was in that study it would imply at least that the 0.02 minutes (1.2 seconds) required for a high order at 0.1 CIC is unrealistic and additional mechanisms to katanin are required to create order at very low CIC rates.

6.4 Discussion

6.4.1 Catastrophe versus rescue rate

We conclude that the formation of an ordered cortical array is more sensitive to changes in catastrophe rate compared to the rescue rate. We observe that using the state change parameters as per [Allard et al., 2010] causes the simulated model to generate an ordered array within a biologically reasonable time frame. However, a less ordered array is formed when using data direct from [Kawamura

and Wasteneys, 2008]. This shows that computational modelling can be a valuable tool in confirming observed microtubule dynamics. We show that there is a balance between the catastrophe and rescue rates where changing one and not the other to compensate can result in a loss of order. However, as both the catastrophe and rescue rates also affect the length and number of microtubules this must also be considered in addition to maximising the order parameter as there should still be a biologically reasonable number of microtubules in the cell. As the order of the microtubules appears to be more sensitive to the catastrophe rate, it suggests that allowing the microtubule to survive long enough to interact with others is an important part in establishing an ordered microtubule array. Also in a 3-state model there are four other state change parameters that could affect the formation of the cortical array and therefore further work could be to observe how each parameter interacts with the others. Also the state change rates may not be uniformly spread throughout all the microtubules - for example a microtubule in a bundle may be made more stable due to cross linking proteins [Van Damme et al., 2004]. It should be noted that when calculating the mean order of the simulations that the standard deviation appears to be quite large compared to the mean across most of our results. This is an effect of the randomness inherent within the simulation as mentioned in Section 6.2.3 and highlights the need for many simulations to be run for each parameter set. However, the standard deviation is lower relative to the mean when the parameter set encourages a high order as would be expected compared to parameter sets which lead to disorder.

6.4.2 CIC versus crossover

We conclude that through microtubule to microtubule interactions alone an ordered array forms when the CIC rate is above 0.5 but the order rapidly decreases as CIC decreases below that. Our threshold value of 0.5 CIC is not far from the 0.6 observed in tobacco BY2 cells [Dixit and Cyr, 2004] but far larger than the 0.09 observed in *Arabidopsis thaliana* petiole epidermal cells [Wightman and Turner, 2007]. This implies that where there are low levels of CIC, and thus increased cross overs, self organisation through microtubule to microtubule interactions alone is not sufficient and therefore additional microtubule interactions,

such as the severing action of katanin, are required to bring order.

6.4.3 The effect of katanin at low CIC

We conclude that where the CIC rate is too low to create order by itself, katanin is able to create an ordered array through severing, however, at very low CIC rates it is unlikely that katanin is able to act fast enough and additional mechanisms to microtubule to microtubule interactions would be required for an ordered array to be established. If we consider the role of katanin in a microtubule-to-microtubule interaction order model, the difference between CIC and entrainment and katanin is that whilst both entrainment and CIC actively prevent disorder from occurring either by causing the microtubule to grow alongside the array or to be annihilated through rapid catastrophe, katanin acts by limiting but not fully removing an element of disorder (crossover) that has already occurred. To be effective at this role, the plus end growth speed of the microtubule must be sufficiently slow enough that the microtubule does not extend too far from the crossover site before katanin is able to sever it and also the CIC rate must be low enough for crossovers to occur. However, both these requirements act to inhibit order as greater growth speed leads to more microtubule interactions and less crossovers means more meaningful interactions. Therefore we propose that katanin's role in forming ordered cortical arrays is reliant on external factors driving the order/orientation of microtubules as opposed to random microtubule to microtubule interactions. An example of such a factor could be CLASP which permits microtubules across certain catastrophe inducing edges where expressed [Ambrose et al., 2011]. This biases the orientation of microtubules and so contributes to order thus lessening the requirement of microtubule to microtubule interactions. Future work would then be to model this and see if both a realistic severing time and microtubule order could be achieved.

Chapter 7

Conclusions

7.1 Summary of Scientific Contributions

In this thesis we used computational modelling to investigate the formation of ordered acentrosomal microtubule arrays in two cell types; epithelial cells and cortical plant cells. Our scientific contributions are:

1. In epithelial cells we propose that the *Bellett* hypothesis can explain the formation of the acentrosomal array but add our own proposition that it is the loss of ninein from the centrosome that causes the centrosomal array to shrink as the side wall increases.

2. We highlight the potential of OpenCL as an alternative to a high performance cluster when performing biological modelling with a high degree of parallelisation.

3. We show that cortical microtubules are able to generate ordered structures where the CIC probability is $>= 0.5$ when entrainment is permitted and there is no severing of crossovers.

4. At CIC probabilities of $\sim$0.4 we propose that katanin can cause an ordered array but at very low CIC (0.1) katanin alone, whilst necessary, is not sufficient to create an ordered array.

5. We developed simulation software for cortical microtubules that can be used and expanded by the scientific community for future modelling studies.

7.2 Epithelial cell research

Our aim was to investigate the *Bellet* hypothesis as to how to microtubules moved from the centrosome to the side walls in epithelial cells using an MDCK cell as a model cell type. We hypothesised that the concentration of free tubulin and the relative catastrophe and rescue rates of the centrosomal population could explain the shrinking of the centrosomal microtubule population as the side wall population grows. We created both Monte Carlo simulation and PDE models of the MDCK microtubules and parallelised them through OpenCL to perform parameter searches on the centrosomal and side wall catastrophe and rescue rates. Whilst we observed the side wall population reaching an appropriate size within the specified time frame, we did not observe a corresponding decrease in the centrosomal population as side wall population was very sensitive to changes in the centrosomal dynamic instability. Therefore we modelled the loss of nucleation potential from the centrosome, corresponding with ninein being transported to the side wall, by adding a negative feedback term reducing centrosomal nucleation as the side wall population increased. Here we observed an increase in the side wall and decrease in the centrosomal populations that was similar that observed in life cell experiments and therefore we propose that the *Bellett* hypothesis can work but with the added proviso that the nucleation rate of the centrosome drops as the amount of ninein on the side wall increases.

7.3 Cortical microtubule research

Our primary aim was to investigate how catastrophe, both spontaneous and collision induced, affected the formation of ordered cortical microtubule arrays. Our objectives were to investigate the impact of katanin at low rates of collision induced catastrophe (high chance of crossover) and to generate a simulation model for cortical microtubules as there were not any available. We created an intuitive Monte Carlo simulation model in Java that accepted a wide amount of dynamic instability and cellular parameters that could provide a graphical output for individual runs but also a version that could be run on a cluster for param-

eter searches. By performing parameter searches, with microtubule entrainment enabled, we observed that changes in the spontaneous catastrophe rate, both positive and negative, had a negative effect on the formation of the ordered microtubule array. We also observed that an ordered array no longer formed once the probability of CIC was less than 0.5, with peak order being observed at 0.8 probability. Through modelling the role of katanin we observed that at low rates of collision induced catastrophe observed in some cortical cells, katanin would have to act implausibly fast to create an ordered array and therefore we propose that katanin, whilst it is a necessary element of creating ordered arrays, requires additional mechanisms of controlling the order of the microtubules.

7.4 Cortical microtubule software

Our objective was to produce a cortical microtubule model that would assist in future research studies. We believe we have achieved this as our model provides fully changeable dynamic instability parameters in addition to entrainment and cross over severing. It is also platform independent and provides a real time view of the simulation. Being written in an object orientated manner it should also not be overly complex to update with additional features/interactions.

7.5 Future work

Future work would be to expand the cortical microtubule model to a multiple surface three dimensional model. This would allow greater control over spatially distributed factors and also pave the way towards large, tissue based models. As the microtubules themselves do not differ between cell types it should be able to expand the simulation model into different cell types by adding a different geometry and behavioural protocols. Also in order to make the simulation model more complete an OpenCL component could be added to enable parameter searches.

As an alternative to Monte Carlo simulation studies it could be advantageous to instead model microtubules as nematic liquid crystals. Nematic liquid crystals are rod like structures in a fluid whose energetically favourable configuration is to

align in parallel. This is very similar to the observed behaviour microtubules [Hitt et al., 1990]. As there is a large body of mathematical work on liquid crystals, modelling microtubules as them could provide strong analytical solutions and yield further insight into their behaviour. A recent study has used a liquid crystal model of microtubule cytoskeleton aggregates [Fan and Li, 2015].

References

Munshi Aaftab. OpenCL Specification Version 1.2. `http://www.khronos.org/registry/cl/`, 2011. 27, 28

Anna Akhmanova and Casper C Hoogenraad. Microtubule plus-end-tracking proteins: mechanisms and functions. *Current Opinion in Cell Biology*, 17(1): 47–54, 2005. 2, 70

Jun F. Allard, Geoffrey O. Wasteneys, and Eric N. Cytrynbaum. Mechanisms of self-organization of cortical microtubules in plants revealed by computational simulations. *Molecular Biology of the Cell*, 21(2):278–286, 2010. 7, 8, 20, 21, 23, 26, 31, 50, 73, 74, 75, 78, 81, 84

Chris Ambrose, Jun F Allard, Eric N Cytrynbaum, and Geoffrey O Wasteneys. A CLASP-modulated cell edge barrier mechanism drives cell-wide cortical microtubule organization in Arabidopsis. *Nature Communications*, 2:430, 2011. 16, 21, 23, 26, 50, 86

Søren S.L. Andersen. Molecular characteristics of the centrosome. *International Review of Cytology*, 187:51–109, 1999. 1

D. Andrade. OpenCL Optimization I. `http://www.cmsoft.com.br`, 2011. 29

R. Bacallao, C. Antony, C. Dotti, E. Karsenti, E. H. Stelzer, and K. Simons. The subcellular organization of Madin-Darby canine kidney cells during the formation of a polarized epithelium. *The Journal of Cell Biology*, 109(6 Pt 1): 2817–2832, Dec 1989. 2

Kevin J. Barker, Kei Davis, Adolfy Hoisie, Darren J. Kerbyson, Mike Lang, Scott Pakin, and Jose C. Sancho. Entering the petaflop era: the architecture and performance of Roadrunner. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, SC '08, pages 1:1–1:11, Piscataway, NJ, USA, 2008. IEEE Press. ISBN 978-1-4244-2835-9. URL `http://dl.acm.org/citation.cfm?id=1413370.1413372`. 26

Vladimir A. Baulin, Carlos M. Marques, and Fabrice Thalmann. Collision induced spatial organization of microtubules. *Biophysical Chemistry*, 128(2):231–244, 2007. 19, 20, 21, 31

Gemma Bellett, Jane M. Carter, Jennifer Keynton, Deborah Goldspink, Colin James, David K. Moss, and Mette M. Mogensen. Microtubule plus-end and minus-end capture at adherens junctions is involved in the assembly of apico-basal arrays in polarised epithelial cells. *Cell Motility and the Cytoskeleton*, 66 (10):893–908, 2009. 3, 5, 70

J.R. Bowen, D. Hwang, X. Bai, D. Roy, and E.T. Spiliotis. Septin GTPases spatially guide microtubule organization and plus end dynamics in polarizing epithelia. *The Journal of Cell Biology*, 194(2):187–197, 2011. 16, 55

M. H. Bré, T. E. Kreis, and E. Karsenti. Control of microtubule nucleation and stability in madin-darby canine kidney cells: the occurrence of noncentrosomal, stable detyrosinated microtubules. *The Journal of Cell Biology*, 105(3):1283–1296, 1987. 2

M. H. Bré, R. Pepperkok, A. M. Hill, N. Levilliers, W. Ansorge, E. H. Stelzer, and E. Karsenti. Regulation of microtubule dynamics and nucleation during polarization in MDCK II cells. *The Journal of Cell Biology*, 111(6):3013–3021, 1990. 55

A.L. Brittle and H. Ohkura. Mini spindles, the XMAP215 homologue, suppresses pausing of interphase microtubules in drosophila. *The EMBO Journal*, 24(7): 1387–1396, 2005. 13

B. Buendia, M. H. Bré, G. Griffiths, and E. Karsenti. Cytoskeletal control of centrioles movement during the establishment of polarity in Madin-Darby canine kidney cells. *The Journal of Cell Biology*, 110(4):1123–1135, Apr 1990. 2

Jeannette C. Bulinski and Gregg G. Gundersen. Stabilization and post-translational modification of microtubules during cellular morphogenesis. *Bioessays*, 13(6):285–293, 1991. 2

David H Burk, Bo Liu, Ruiqin Zhong, W Herbert Morrison, and Zheng-Hua Ye. A katanin-like protein regulates normal cell wall biosynthesis and cell elongation. *The Plant Cell Online*, 13(4):807–827, 2001. 7

Roy G. Burns. α-, β-, and γ-tubulins: Sequence comparisons and structural constraints. *Cell Motility and the Cytoskeleton*, 20(3):181–189, 1991. 1

M. Caplow and J. Shanks. Evidence that a single monolayer tubulin-gtp cap is both necessary and sufficient to stabilize microtubules. *Molecular Biology of the Cell*, 7(4):663–675, 1996. 13

J.L. Carminati and T. Stearns. Microtubules orient the mitotic spindle in yeast through dynein-dependent interactions with the cell cortex. *The Journal of Cell Biology*, 138(3):629–641, 1997. 12

Jordi Chan, Cynthia G. Jensen, Lawrence C. W. Jensen, Max Bush, and Clive W. Lloyd. The 65-kda carrot microtubule-associated protein forms regularly arranged filamentous cross-bridges between microtubules. *Proceedings of the National Academy of Sciences*, 96(26):14931–14936, 1999. 16, 74

Jordi Chan, Grant M. Calder, John H. Doonan, and Clive W. Lloyd. Eb1 reveals mobile microtubule nucleation sites in arabidopsis. *Nature Cell Biology*, 5(11): 967–971, 2003. 2

Jordi Chan, Adrian Sambade, Grant Calder, and Clive Lloyd. Arabidopsis cortical microtubules are initiated along, as well as branching from, existing microtubules. *The Plant Cell Online*, 21(8):2298–2306, 2009. 14

Denis Chrétien and Richard H. Wade. New data on the microtubule surface lattice. *Biology of the Cell*, 71(1-2):161–174, 1991. 1

R.A. Cross. Molecular motors: Dynein's gearbox. *Current Biology*, 14(9):R355–R356, 2004. 56

A. Desai and T.J. Mitchison. Microtubule polymerization dynamics. *Annual Review of Cell and Developmental Biology*, 13:83–117, 1997. 1

Ram Dixit and Richard Cyr. Encounters between dynamic cortical microtubules promote ordering of the cortical array through angle-dependent modifications of microtubule behavior. *The Plant Cell Online*, 16(12):3274–3284, 2004. 7, 15, 16, 19, 85

Stephen J. Doxsey, Pascal Stein, Louise Evans, Patricia D. Calarco, and Marc Kirschner. Pericentrin, a highly conserved centrosome protein involved in microtubule organization. *Cell*, 76(4):639–650, 1994. 1

D.N. Drechsel and M.W. Kirschner. The minimum gtp cap required to stabilize microtubules. *Current Biology*, 4(12):1053–1061, 1994. 13

Peng Du, Rick Weber, Piotr Luszczek, Stanimire Tomov, Gregory Peterson, and Jack Dongarra. From CUDA to OpenCL: Towards a performance-portable solution for multi-platform GPU programming. *Parallel Computing*, 38(8):391 – 407, 2012. ISSN 0167-8191. doi: 10.1016/j.parco.2011.10.002. URL `http://www.sciencedirect.com/science/article/pii/S0167819111001335`. 27

Joseph D. Dukes, Paul Whitley, and Andrew D. Chalmers. The MDCK variety pack: choosing the right strain. *BMC Cell Biology*, 12(1):43, 2011. 5

Andrey Efimov, Alexey Kharitonov, Nadia Efimova, Jadranka Loncarek, Paul M. Miller, Natalia Andreyeva, Paul Gleeson, Niels Galjart, Ana R.R. Maia, Ian X. McLeod, et al. Asymmetric CLASP-dependent nucleation of noncentrosomal microtubules at the trans-golgi network. *Developmental Cell*, 12(6):917–930, 2007. 14

Ezgi Can Eren, Ram Dixit, and Natarajan Gautam. A three-dimensional computer simulation model reveals the mechanisms for self-organization of plant cortical microtubules into oblique arrays. *Molecular Biology of the Cell*, 21(15): 2674–2684, 2010. 7, 8, 20, 23, 50

Mathieu Erhardt, Virginie Stoppin-Mellet, Sarah Campagne, Jean Canaday, Jérôme Mutterer, Tanja Fabian, Margret Sauter, Thierry Muller, Christine Peter, Anne-Marie Lambert, et al. The plant spc98p homologue colocalizes with γ-tubulin at microtubule nucleation sites and is required for microtubule nucleation. *Journal of Cell Science*, 115(11):2423–2431, 2002. 14

Houfu Fan and Shaofan Li. Modeling microtubule cytoskeleton via an active liquid crystal elastomer model. *Computational Materials Science*, 96:559–566, 2015. 90

J. Fang, A.L. Varbanescu, and H. Sips. A comprehensive performance comparison of CUDA and OpenCL. In *Parallel Processing (ICPP), 2011 International Conference on*, pages 216–225. IEEE, 2011. 27

Charles R. Gaush, Walter L. Hard, and Thomas F. Smith. Characterization of an established line of canine kidney cells (MDCK). *Experimental Biology and Medicine*, 122(3):931–935, 1966. 5

N.R. Gliksman, R.V. Skibbens, and E.D. Salmon. How the transition frequencies of microtubule dynamic instability (nucleation, catastrophe, and rescue) regulate microtubule dynamics in interphase and mitosis: analysis using a monte carlo computer simulation. *Molecular Biology of the Cell*, 4(10):1035–1050, 1993. 17

S.P. Gross. Hither and yon: a review of bi-directional microtubule-based transport. *Physical Biology*, 1(2):R1–R11, 2004. 12

L.T. Haimo and J.L. Rosenbaum. Cilia, flagella, and microtubules. *The Journal of Cell Biology*, 91(3):125–130, 1981. 13

A.R. Hardham and B.E. Gunning. Structure of cortical microtubule arrays in plant cells. *The Journal of Cell Biology*, 77(1):14–34, 1978. 6, 15

Yan He, Franto Francis, Kenneth A. Myers, Wenqian Yu, Mark M. Black, and Peter W. Baas. Role of cytoplasmic dynein in the axonal transport of microtubules and neurofilaments. *The Journal of Cell Biology*, 168(5):697–703, 2005. 2

Regina Himmelspach, Richard E. Williamson, and Geoffrey O. Wasteneys. Cellulose microfibril alignment recovers from dcb-induced disruption despite microtubule disorganization. *The Plant Journal*, 36(4):565–575, 2003. 6

A.L. Hitt, A.R. Cross, and R.C. Williams. Microtubule solutions display nematic liquid crystalline structure. *Journal of Biological Chemistry*, 265(3):1639–1647, 1990. 90

Jörn Hoffmann, Karim El-Laithy, Frank Güttler, and Martin Bogdan. Simulating biological-inspired spiking neural networks with OpenCL. In Konstantinos Diamantaras, Wlodek Duch, and Lazaros Iliadis, editors, *Artificial Neural Networks ICANN 2010*, volume 6352 of *Lecture Notes in Computer Science*, pages 184–187. Springer Berlin / Heidelberg, 2010. ISBN 978-3-642-15818-6. URL `http://dx.doi.org/10.1007/978-3-642-15819-3_23`. 31

A. Hotta, T. Kawakatsu, T. Nakatani, T. Sato, C. Matsui, T. Sukezane, T. Akagi, T. Hamaji, I. Grigoriev, A. Akhmanova, Y. Takai, and Y. Mimori-Kiyosue. Laminin-based cell adhesion anchors microtubule plus ends to the epithelial cell basal cortex through LL5alpha/beta. *The Journal of Cell Biology*, 189(5): 901–917, May 2010. 56

Marcel E. Janson, E. Mathilde, and Marileen Dogterom. Dynamic instability of microtubules is regulated by force. *The Journal of Cell Biology*, 161(6): 1029–1034, 2003. 15

Albertas Janulevicius, Jaap van Pelt, and Arjen van Ooyen. Compartment volume influences microtubule dynamic instability: A model study. *Biophysical Journal*, 90(3):788 – 798, 2006. 5, 17, 53, 54

Fanny Jaulin and Geri Kreitzer. KIF17 stabilizes microtubules and contributes to epithelial morphogenesis by acting at mt plus ends with eb1 and apc. *The Journal of Cell Biology*, 190(3):443–460, 2010. 17

Eiko Kawamura and Geoffrey O. Wasteneys. MOR1, the arabidopsis thaliana homologue of xenopus MAP215, promotes rapid growth and shrinkage, and suppresses the pausing of microtubules in vivo. *Journal of Cell Science*, 121 (24):4114–4123, 2008. 8, 73, 76, 84

P.J. Keller, F. Pampaloni, G. Lattanzi, and E.H.K. Stelzer. Three-dimensional microtubule behavior in *Xenopus* egg extracts reveals four dynamic states and state-dependent elastic properties. *Biophysical Journal*, 95(3):1474–1486, 2008. 13

Gaurav Khanna and Justin McKennon. Numerical modeling of gravitational wave sources accelerated by OpenCL. *Computer Physics Communications*, 181(9): 1605 – 1611, 2010. ISSN 0010-4655. doi: 10.1016/j.cpc.2010.05.014. URL `http://www.sciencedirect.com/science/article/pii/S0010465510001682`. 30

K. Komatsu, K. Sato, Y. Arai, K. Koyama, H. Takizawa, and H. Kobayashi. Evaluating performance and portability of OpenCL programs. In *The Fifth International Workshop on Automatic Performance Tuning (iWAPT2010)*, 2010. 28

Alex Mace and Wenjia Wang. Modelling the role of spontaneous and collision induced catastrophe in the self organisation of cortical microtubules. In *BIBM*, pages 8–13, 2014. 72

Robert W. Mays, Kenneth A. Beck, and W. James Nelson. Organization and function of the cytoskeleton in polarized epithelial cells: a component of the protein sorting machinery. *Current Opinion in Cell Biology*, 6(1):16–24, 1994. 3

Francis J. McNally and Ronald D. Vale. Identification of katanin, an ATPase that severs and disassembles stable microtubules. *Cell*, 75(3):419–429, 1993. 7

Tim Meads and Trina A. Schroer. Polarity and nucleation of microtubules in polarized epithelial cells. *Cell Motility and the Cytoskeleton*, 32(4):273–288, 1995. 2

Yuko Mimori-Kiyosue, Ilya Grigoriev, Gideon Lansbergen, Hiroyuki Sasaki, Chiyuki Matsui, Fedor Severin, Niels Galjart, Frank Grosveld, Ivan Vorobjev, Shoichiro Tsukita, et al. CLASP1 and CLASP2 bind to EB1 and regulate microtubule plus-end dynamics at the cell cortex. *The Journal of Cell Biology*, 168(1):141–153, 2005. 16

T. Mitchison and M. Kirschner. Dynamic instability of microtubule growth. *Nature*, 312(5991):237–242, 1984. 1, 13

M. M. Mogensen, J. B. Tucker, J. B. Mackie, A. R. Prescott, and I. S. Näthke. The adenomatous polyposis coli protein unambiguously localizes to microtubule plus ends and is involved in establishing parallel arrays of microtubule bundles in highly polarized epithelial cells. *The Journal of Cell Biology*, 157(6):1041–1048, Jun 2002. 56

Mette M. Mogensen, John B. Mackie, Stephen J. Doxsey, Tim Stearns, and John B. Tucker. Centrosomal deployment of γ-tubulin and pericentrin: Evidence for a microtubule-nucleating domain and a minus-end docking domain in certain mouse epithelial cells. *Cell Motility and the Cytoskeleton*, 36(3): 276–290, 1997. 3

Mette M. Mogensen, Azer Malik, Matthieu Piel, Veronique Bouckson-Castaing, and Michel Bornens. Microtubule minus-end anchorage at centrosomal and non-centrosomal sites: the role of ninein. *Journal of Cell Science*, 113(17): 3013–3023, 2000. 3

David K. Moss, Gemma Bellett, Jane M. Carter, Mirjana Liovic, Jennifer Keynton, Alan R. Prescott, E. Birgitte Lane, and Mette M. Mogensen. Ninein is released from the centrosome and moves bi-directionally along microtubules. *Journal of Cell Science*, 120(17):3064–3074, 2007. 2, 3

Takashi Murata, Seiji Sonobe, Tobias I. Baskin, Susumu Hyodo, Seiichiro Hasezawa, Toshiyuki Nagata, Tetsuya Horio, and Mitsuyasu Hasebe. Microtubule-dependent microtubule nucleation based on recruitment of γ-tubulin in higher plants. *Nature Cell Biology*, 7(10):961–968, 2005. 2, 14

Masayoshi Nakamura, David W. Ehrhardt, and Takashi Hashimoto. Microtubule and katanin-dependent dynamics of microtubule nucleation complexes in the acentrosomal Arabidopsis cortical array. *Nature Cell Biology*, 12(11):1064–1070, 2010. 14

John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with CUDA. *Queue*, 6(2):40–53, March 2008. ISSN 1542-7730. doi: 10.1145/1365490.1365500. URL `http://doi.acm.org/10.1145/1365490.1365500`. 27

Geoffrey O. Wasteneys and Richard E. Williamson. Reassembly of microtubules in nitella tasmanica: assembly of cortical microtubules in branching clusters and its relevance to steady-state microtubule assembly. *Journal of Cell Science*, 93(4):705–714, 1989. 14

Berl R. Oakley, C.Elizabeth Oakley, Yisang Yoon, and M.Katherine Jung. γ-tubulin is a component of the spindle pole body that is essential for microtubule function in *Aspergillus nidulans*. *Cell*, 61(7):1289–1301, 1990. 13

C. Elizabeth Oakley and Berl R. Oakley. Identification of γ-tubulin, a new member of the tubulin superfamily encoded by mipa gene of aspergillus nidulans. *Nature*, 338:662–664, 1989. 13

Young Y. Ou, Gary J. Mack, Meifeng Zhang, and Jerome B. Rattner. Cep110 and ninein are located in a specific domain of the centrosome associated with centrosome maturation. *Journal of Cell Science*, 115(9):1825–1835, 2002. 70

PubMed. PubMed. `http://www.ncbi.nlm.nih.gov/pubmed`, 2015. Accessed: 27/04/2015. 5

Amy Reilein, Soichiro Yamada, and W. James Nelson. Self-organization of an acentrosomal microtubule network at the basal cortex of polarized epithelial cells. *The Journal of Cell Biology*, 171(5):845–855, 2005. 18, 31

Antonina Roll-Mecak and Ronald D. Vale. Making more microtubules by severing: a common theme of noncentrosomal microtubule arrays? *The Journal of Cell Biology*, 175(6):849–851, 2006. 7, 18

David J. Sharp, Wenqian Yu, Lotfi Ferhat, Ryoko Kuriyama, David C. Rueger, and Peter W. Baas. Identification of a microtubule-associated motor protein essential for dendritic differentiation. *The Journal of Cell Biology*, 138(4):833–843, 1997. 2

Sidney L. Shaw, Roheena Kamyar, and David W. Ehrhardt. Sustained microtubule treadmilling in arabidopsis cortical arrays. *Science*, 300(5626):1715–1718, 2003. 2, 6

Guochun Shi, V. Kindratenko, F. Pratas, P. Trancoso, and M. Gschwind. Application acceleration with the cell broadband engine. *Computing in Science Engineering*, 12(1):76 –81, jan.-feb. 2010. ISSN 1521-9615. doi: 10.1109/MCSE.2010.4. 26

Tomoyoshi Shimobaba, Tomoyoshi Ito, Nobuyuki Masuda, Yasuyuki Ichihashi, and Naoki Takada. Fast calculation of computer-generated-hologram on AMD HD5000 series GPU and OpenCL. *Opt. Express*, 18(10):9955–9960, May 2010. doi: 10.1364/OE.18.009955. URL `http://www.opticsexpress.org/abstract.cfm?URI=oe-18-10-9955`. 31

Hiroshi Shinohara, Nobuyuki Sakayori, Masanori Takahashi, and Noriko Osumi. Ninein is essential for the maintenance of the cortical progenitor character by anchoring the centrosome to microtubules. *Biology Open*, 2(7):739–749, 2013. 70

Tim Stearns, Louise Evans, and Marc Kirschner. γ-tubulin is a highly conserved component of the centrosome. *Cell*, 65(5):825–836, 1991. 1

J.E. Stone, D. Gohara, and Guochun Shi. Opencl: A parallel programming standard for heterogeneous computing systems. *Computing in Science Engineering*, 12(3):66 –73, may-june 2010. ISSN 1521-9615. doi: 10.1109/MCSE.2010.69. 27

Daniel B. Szymanski and Daniel J. Cosgrove. Dynamic coordination of cytoskeletal and cell wall systems during plant cell morphogenesis. *Current Biology*, 19 (17):R800–R811, 2009. 6

Simon H. Tindemans and Bela M. Mulder. Microtubule length distributions in the presence of protein-induced severing. *Phys. Rev. E*, 81:031910, 2010. 18, 23, 56

Jennifer S. Tirnauer and Barbara E. Bierer. EB1 proteins regulate microtubule dynamics, cell polarity, and chromosome stability. *The Journal of Cell Biology*, 149(4):761–766, 2000. 16

Daniël Van Damme, Kris Van Poucke, Emmanuel Boutant, Christophe Ritzenthaler, Dirk Inzé, and Danny Geelen. In vivo dynamics and differential microtubule-binding activities of MAP65 proteins. *Plant Physiology*, 136(4): 3956–3967, 2004. 85

R. A. Walker, E. T. O'Brien, N. K. Pryer, M. F. Soboeiro, W. A. Voter, H. P. Erickson, and E. D. Salmon. Dynamic instability of individual microtubules analyzed by video light microscopy: rate constants and transition frequencies. *The Journal of Cell Biology*, 107(4):1437–1448, 1988. 53, 54

Geoffrey O. Wasteneys and J. Christian Ambrose. Spatial organization of plant cortical microtubules: close encounters of the 2D kind. *Trends in Cell Biology*, 19(2):62 – 71, 2009. ISSN 0962-8924. 6, 14

Raymond Wightman and Simon R. Turner. Severing at sites of microtubule crossover contributes to microtubule alignment in cortical arrays. *The Plant Journal*, 52(4):742–751, 2007. ISSN 1365-313X. 7, 15, 81, 85

Wenqian Yu and Peter W. Baas. Changes in microtubule number and length during axon differentiation. *The Journal of Neuroscience*, 14(5):2818–2829, 1994. 1

Quan Zhang, Erica Fishel, Tyler Bertroche, and Ram Dixit. Microtubule severing at crossover sites by katanin generates ordered cortical microtubule arrays in Arabidopsis. *Current Biology*, 23(21):2191 – 2195, 2013. ISSN 0960-9822. 7, 84

Yixian Zheng, Mei Lie Wong, Bruce Alberts, and Tim Mitchison. Nucleation of microtubule assembly by a gamma-tubulin-containing ring complex. *Nature*, 378(6557):578–583, 1995. 13

Jun Zhou and Paraskevi Giannakakou. Targeting microtubules for cancer chemotherapy. *Current Medicinal Chemistry - Anti-Cancer*

Agents, 5(1):65–71, 2005. doi: doi:10.2174/1568011053352569. URL `http://www.ingentaconnect.com/content/ben/cmcaca/2005/00000005/00000001/art00006`. 12

Sabina Zimmerman, P.T. Tran, Rafael R. Daga, Osami Niwa, and Fred Chang. Rsp1p, a j domain protein required for disassembly and assembly of microtubule organizing centers during the fission yeast cell cycle. *Developmental Cell*, 6(4): 497–509, 2004. 2