Incorporation of Nucleus in Phase-Field and Sharp-Interface Models of Cell Migration in Monolayers

Yeeren I. Low

Master of Science

Physics

McGill University

Montreal, Quebec

February 2019

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Science in Physics

ACKNOWLEDGEMENTS

I thank my family for their love and support throughout the years.

I thank my supervisor, Professor Martin Grant, for his experienced guidance of this project and very helpful advice outside of it.

I thank Yony Bresler for useful discussions on the research project.

I thank Professor Paul François of McGill University for reviewing this thesis.

I would also like to thank my brothers and sisters in Christ who have prayed for me during these past few years.

Finally, I thank the God who controls all things, who has helped me through difficulties, uncertainties, and decisions I faced during the time of my M.Sc. program at McGill University. From all that has happened in this period, I believe that it is because of His support of this undertaking that I have been able to successfully complete it to the present point, beyond what I would be able to accomplish myself.

ABSTRACT

We use a phase-field model of a cell monolayer consisting of a cancer cell surrounded by normal cells, where the cancer cell is softer than the normal cells. We extended the model to incorporate a nucleus which is suitably confined within the cytoplasm and constrained to the cell centroid. We use sharp-interface models derived from these phase-field models and simulate them to explore the consequences of an elasticity mismatch, extending previous studies. More accurate numerics have been performed and have somewhat clarified the "pinching" problem [arXiv:1807.07836 (2018)]. Inclusion of a nucleus in the model continued to achieve some resemblance to the speed bursts observed in experiments [*Biophys. J.*, 102:2731–2741 (2012)], but faithful reproduction of the experimental phenomenology has yet to be accomplished. The softening of the cytoplasm from previous models is seen to facilitate motion through the cell packing, beyond that seen in the experiments for the values of the simulation parameters used here.

ABRÉGÉ

Nous utilisons le modèle de champ de phase d'une monocouche constituée d'une cellule cancéreuse entourée de cellules normales, où la cellule cancéreuse est plus souple que les cellules normales. Nous avons étendu le modèle pour incorporer un noyau conformément confiné dans le cytoplasme, et contraint au centroïde de la cellule. Nous utilisons des modèles à interface raide provenus des modèles de champ de phase, et les simulons pour explorer les conséquences du décalage d'élasticité, développant ainsi les études précédentes. Des calculs plus précis ont été effectués et ont pu plus ou moins éclaircir le problème de "pincement" [arXiv:1807.07836 (2018)]. L'inclusion d'un noyau dans le modèle a continué à résulter d'une certaine ressemblance aux pointes de vitesse observées dans les expériences [*Biophys. J.*, 102:2731–2741 (2012)], mais la phénoménologie expérimentale reste à être reproduite. Pour les valeurs des paramètres de simulation utilisés ici, on a observé que l'assouplissement du cytoplasme des modèles précédents facilitait le mouvement, à travers l'agrégat compact de cellules, au-delà de ce que nous avons observé dans les expériences.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Biological Background

Collective cell migration plays an important role in phenomena such as biological development, tissue repair, and cancer [1]. In general, it involves individual cell active propulsion and interactions such as physical contact, adhesion, and chemical signaling [2]. Interactions can be mediated by a flexible substrate [3]. Other types of multicellular translocation that exist in biological tissues include intercalation (rearrangement) and expansive growth (arising from cell proliferation) [1], and collective rotation [2].

Cancer is a disease where cells overstep normal limits on growth, proliferation, and death. A cancer can become metastatic after cells become invasive, acquiring the ability to degrade material and penetrate surrounding tissues. Cancers most commonly arise from epithelial tissues, which consist of layers (possibly a single layer) of tightly adhered cells sitting atop and adhered to the basement membrane, which separates the epithelium from the stroma. The stroma consists mainly of the extracellular matrix, a three-dimensional fibrous mesh secreted by the stromal cells, as well as the stromal cells themselves (mainly fibroblasts) and blood or lymph vessels. Invasiveness confers the ability to degrade the basement membrane and the extracellular matrix. Cancer cells can travel through the stroma individually, in small clusters, or in strands. Metastasis can occur when a cancer cell penetrates cell–cell junctions of blood or lymph vessel walls, travels within the vasculature,

and squeezes out of it into a foreign tissue, starting formation of a secondary tumor [4, 5, 6, 1]. Accumulating evidence indicates that metastatic dissemination can occur before formation of a primary tumor (i.e., a carcinoma *in situ*) [7].

The study of cancer from a physical sciences viewpoint has been an active area of research [5, 6]. Physical interactions are relevant in every step of the metastatic cascade. For example, during invasion a cancer cell must travel through the extracellular matrix, interacting with it. Cancer cells must undergo large deformations in order to squeeze between blood vessel endothelial cells. Cancer cells are softer than normal cells, with mechanical compliance correlating positively with metastatic capability. They also exhibit reduced cell–cell adhesion [5] and higher motility [8].

In vitro experiments studied the consequences of mismatched physical properties in a confluent epithelial-like monolayer where a small density of invasive breast cancer cells (MDA-MB-231) was interspersed among normal breast cells (MCF10A) [9]. This was to study the migration of cancer cells prior to invasion. The MDA-MB-231 cells had dramatically higher motility than those either in isolation or in a homogeneous sheet of like cancer cells. This is due to a pulsating mode of migration in which their trajectories showed bursts of rapid, large displacements interlaced with slow, more random movements. The fast motion is induced by crowding of surrounding normal cells which deforms the cancer cell body and nucleus into an elongated, nonconvex ("dog-bone") shape. The cancer cell then quickly escapes the crowding, recovers a more rounded shape, and slows down. The effect was abrogated when reducing adhesion between normal cells or using noninvasive cancer cells (MCF7) which exhibit adhesion. These implicate mismatched mechanical and

adhesive properties of the cells in enhanced motility. The bursty migration of the soft invasive cancer cell after impingement is also thought to be facilitated by actin assembly and actomyosin contractility via activation of guanosine triphosphatases [9]. We seek to better understand this phenomenon through modeling, specifically the effect of mismatched elasticities. As is often the case with this type of experiment, control of parameters is limited [10].

1.1   The cytoskeleton

Eukaryotic cells contain a cytoskeleton, which is a dynamic protein filament network that spans the cytoplasm of the cell. It is responsible for cell shape, mechanical resistance, and spatial organization of the cell. It consists of three types of filaments: actin filaments, microtubules, and intermediate filaments. All these are formed from individual protein subunits that self-assemble into a helical geometry. Actin filaments and microtubules are made of the proteins actin and tubulin, respectively, whereas the proteins that make up intermediate filaments can vary widely. Cytoskeletal filaments are linked to each other as well as to other components of the cell such as organelles and the plasma membrane. They can span the entire length of a cell, from tens to hundreds of micrometers. Accessory proteins regulate filament assembly and geometry. Motor proteins walk along filaments, transporting organelles or e.g. sliding them past each other as in muscle contraction [4].

The two ends of an actin filament grow at different rates. This is caused by conformational changes in a subunit upon addition to a filament. Additionally, a free actin subunit contains a tightly bound molecule of adenosine triphosphate (ATP)

which is hydrolyzed soon after addition to a filament. This reduces its binding affinity for neighboring subunits, which enables a nonequilibrium phenomenon known as *treadmilling* where, at steady state, subunits are accumulated at the *plus end* while being lost at the *minus end* if the ambient monomer concentration is in the appropriate range. This enables cell protrusion, where actin filaments push against the cell membrane. However, actin filaments are thin (diameter ~ 8 nm) and flexible, and their ability to apply sufficient force (rather than buckling) is due to stiffening from aggregation. Various structures can be formed, regulated by actin-binding proteins. Dendritic structures are formed by the Arp2/3 complex, which mediates nucleation on the side of an existing filament at a 70° angle. Some other actin-binding proteins can be classified as *bundling proteins* or *gel-forming proteins*. In the former case, actin filaments are aggregated with parallel polarities (*parallel bundle*) or antiparallel polarities (*contractile bundle*). With gel-forming proteins, the filaments form high-angle intersections, resulting in a web-like structure. Yet other actin-binding proteins promote or inhibit filament disassembly or continued assembly. These proteins can be spatially organized to produce different structures in different parts of the cell. For example, contractile bundles can function as stress fibers that attach to sites of adhesion to the substratum, and dendritic structures are found in the cell cortex, the peripheral layer lying just beneath the plasma membrane. Additionally, actin can interact with the motor protein myosin to apply contractile forces, as in cytokinesis [4]. In animal cells, myosin also interacts with an actin network lying immediately beneath and attached to the plasma membrane to apply contractile forces (*cortical*

4

tension) that give strength to the cell membrane and control cell shape [11]. Actin filaments are most highly concentrated at the cell periphery [4].

Microtubules are stiff, hollow tubes formed by a ring of 13 tubulin molecules with a diameter of 25 nm. Phosphate hydrolysis in this case leads to dynamic instability, where a microtubule can alternate between periods of growth and rapid shrinking, probing the three-dimensional volume of a cell. As with actin, nucleation is the rate-limiting step for filament formation. Microtubule nucleation is mediated by the γ-tubulin ring complex, which exists in the cytoplasm but is most enriched in a microtubule-organizing center (MTOC). Most animal cells have a single MTOC called the centrosome. Microtubules can position the centrosome at the cell center [4]. However, in epithelia, formation of adherens junctions (cell–cell adhesive sites associated with actin filaments) triggers rearrangement of microtubules into a non-centrosomal array, instead being aligned along the apical–basal axis [12]. Motor proteins walk along microtubules, transporting organelles and vesicles [4].

Intermediate filaments are strong, flexible, ropelike structures (diameter ~ 10 nm) that enable a cell to withstand mechanical stress. This is achieved partly by anchorage to sites of cell–cell (desmosomes) or cell–matrix (hemidesmosomes) contact. Cross-linked keratin networks are responsible for the toughness of skin, hair, and nails [4].

1.2 Single-cell migration

Almost all cell locomotion in animals occurs by crawling. The first step in locomotion is *protrusion*, where a protrusive structure forms at the plasma membrane, often produced by actin polymerization. Next, during *attachment*, adhesion sites are

5

formed that anchor the protrusion to the underlying substratum. Finally, during *traction*, the rear of the cell de-adheres from the substratum and the cell body is pulled forward, using adhesion sites for traction [4].

Various protrusive structures can be formed, depending on actin cross-linking proteins [4]. *Filopodia* have a spike-like shape and are generated by parallel actin bundles, thought to advance using a treadmilling mechanism [13]. They are found in neurons and some types of fibroblasts. *Lamellipodia* are sheet-like structures parallel to the substratum, formed by dendritic networks of actin filaments. They are found in epithelial cells and fibroblasts, as well as some neurons. A third type of actin-based protrusion, comprising *invadopodia* and *podosomes*, extends in three dimensions [4] and enables a cell to degrade components of the extracellular matrix [14] and penetrate surrounding tissue, as in cancer invasion. An alternative mechanism of motility, *blebbing*, is also observed, where the plasma membrane locally advances ahead of, and detaches from, the actin cytoskeleton, enabled by hydrostatic pressure in the cytoplasm [4]. Two-dimensional migration is relevant in an epithelial layer, whereas three-dimensional migration applies when e.g. a cell moves through the extracellular matrix.

As epithelial cells undergoing two-dimensional migration is most relevant here, we will briefly describe lamellipodia-based motility. Cells can migrate at a speed two orders of magnitude faster than a treadmilling actin filament *in vitro* at steady state under physiological conditions. In a cell, a high concentration of actin monomers is maintained by binding of monomers to thymosin, a protein that blocks addition to a filament. Actin monomers also bind to profilin, which prevents self-nucleation

as well as addition to the minus end of a filament. Thus, competition between thymosin and profilin can regulate actin polymerization. The Arp2/3 complex is activated at the front of the cell, inducing formation of a dendritic network. Capping proteins keep filaments short and stiff, decreasing their tendency to buckle and saving monomers for new filaments, in order to maintain generation of protrusive force. Further back, cofilin promotes filament disassembly by debranching, severing, and depolymerization, enabling actin turnover [15]. At the rear of a lamellipodium, myosin filaments reorient actin filaments to be nearly parallel to the cell membrane, gathering in the sides of the cell and preventing sideways protrusion as the cell moves forward. The ability of the actin filaments to generate locomotion depends on their attachment to and the formation of adhesive sites at the front of the cell, which anchor the cell to the underlying substrate and provide traction that enables the cell to advance forward. Actin filaments disassemble at the rear of a cell [4].

Cell migration in a particular direction requires the cell to be *polarized*: an asymmetry generated by orientated organization of polar filaments (actin filaments and microtubules), or by asymmetrical distributions of regulatory molecules [12]. Cell polarity can also refer to apical–basal asymmetry, but here, the distinction between front and back is most relevant. It can arise from external stimuli (in e.g. chemotaxis) or spontaneously [16]. Proteins of the Rho family have been implicated in cell polarity. Cdc42 is active toward the front of the cell [13] and promotes Rac activation, the combination of which promotes formation of lamellipodia via activation of the Arp2/3 complex by proteins of the WASP/WAVE family. Rac activity correlates inversely with RhoA activity, which organizes actomyosin bundles

and prevents protrusion, defining the cell rear [16]. The formation of new adhesions at the leading edge reinforce high Cdc42 and Rac activity, such that locomotion tends to continue in the same direction and turning occurs gradually [13]. Microtubules and their interactions with actin filaments have also been implicated in persistence of cell polarity [12]. In fact, proper cell migration is dependent on all three cytoskeletal filament systems, their interactions with each other, and their connections to the nucleus [4].

Cell migration then has a persistent random walk character, with speed and persistence varying with cell type, dependent on cell–substratum adhesion and generation of protrusive force [17]. A variety of velocity distributions have been observed: Brownian-like, peaked at zero; persistent, peaked at non-zero speeds; and intermittent, a mixture of the two where a cell alternates between slow and fast speeds [18].

Cell migration through three-dimensional (3D) extracellular matrices also plays an important role in cancer [5]; however, in this case the trajectory does not follow a random walk [19]. Additionally, the mechanisms important for 3D migration are different from in 2D. For example, in 3D, wide lamellipodia are typically not seen, and few actomyosin stress fibers are present. Thus, results for 2D migration cannot be simply extrapolated to 3D [5].

1.3 Variations in volume and surface area of the cell and nucleus

Animal cells are separated from the extracellular fluid by a so-called *plasma membrane* which is composed of a ∼5-nm-thick lipid bilayer punctuated by proteins that serve functions such as water and ion transport. Generally, the outer and inner layers have different compositions. The lipid molecules are free to move within the

bilayer, so that it forms a two-dimensional fluid [4]. This means it has elastic moduli for dilatation and bending, but a negligible shear modulus [20]. However, the mechanical properties of the plasma membrane are modified by a dynamic cytoskeletal structure sometimes referred to as the cell cortex that underlies it and attaches to it via its embedded proteins. This network actively generates contractile forces that dominate the bending forces of the lipid bilayer itself [21]. It gives the plasma membrane an effective "surface tension" and controls cell shape [11]. Such tension can be measured using the technique of micropipette aspiration [20]. The lipid bilayer may form many folds on top of the cell cortex [20]. This leads to a difference between the actual plasma membrane area and the apparent cell surface area, which does not take these folds into account [21]. Thus a cell can undergo large changes in apparent surface area without commensurate expansion of the plasma membrane or addition of lipid bilayer [20]. A lipid bilayer can only withstand $\sim 4\%$ expansion before lysis (rupture) [22]. An approximation of constant surface area is used in micropipette aspiration experiments of red blood cells, which are unusual in that its membrane is pulled into a biconcave shape and does not form folds [20]. Membrane tension is spatially homogeneous on time scales relevant to cell migration [23].

Cell volume is primarily determined by osmotic balance [21]. However, the plasma membrane is not a simple "semipermeable membrane" that allows flow of water and prohibits flow of solute. It contains pores, ion transporters, and pumps that mediate transport of water and small ions. Moreover, large charged macromolecules reside in the cytoplasm and give rise to an electrochemical gradient for small ionic species [4]. This complicates the picture of osmotic balance and renders

it a steady state rather than a true equilibrium [24]. Cell volume is thought to remain approximately constant during migration [21, 25, 26]. Such an assumption is also made in micropipette aspiration experiments of neutrophils (a type of white blood cell) and red blood cells [20].

During interphase, the cell must grow to twice its size. Membrane growth also occurs during this time [21]; it is determined by the balance between exocytosis and endocytosis [27], which respond to changes in membrane tension [28]. Exocytosis is the process in which the membrane of intracellular vesicles fuses with the plasma membrane to deliver materials to the extracellular space or the plasma membrane; endocytosis is the reverse process [4].

The nuclear envelope is composed of two lipid bilayer membranes and an underlying nuclear lamina. The nuclear membranes join at nuclear pore complexes [29], which allow free passage of water [30]. The nuclear lamina is a dense protein meshwork connected to the inner nuclear membrane [29]. Micropipette aspiration experiments show buckling and crumpling of the nuclear envelope, demonstrating resistance to shear forces, which is characteristic of solid-like elastic behavior. In contrast, the nuclear interior is largely aqueous and its volume reduction of the nucleus could reach 60–70% in such experiments [31]; however, no significant reduction in nuclear volume was seen during migration of whole cells through narrow constrictions [32]. Similarly to cell volume, osmotic pressure differences across the nuclear envelope have been implicated in nuclear volume changes [33]. Unlike the plasma membrane, the nuclear lamina is very expansible although not very compressible

[30]. Nuclei in adherent cells show a somewhat flattened shape due to compressive forces from the cytoskeleton [33].

In many cells, nuclei occupy around 10% of the total cell volume [4], although volume fractions of 30–50% are also seen [34].

1.4 Cell and nuclear mechanics

Cells and their nuclei exhibit viscoelastic rheological responses that can be quantified by frequency-dependent viscoelastic moduli, combined into a complex modulus $G^*(\omega)$, where ω is angular frequency. Creep compliance is another measurement, related to $G^*(\omega)$ via a Fourier transform. The viscoelastic properties of cells are dominated by the actin cytoskeleton. The viscous response arises from dynamic cross-linking (cytoskeletal remodeling) and controls the rate of cell migration [35]. The rheology of cells is not well described by any simple synthetic model, in part due to active motions, constant cytoskeletal remodeling, and inhomogeneities of the cell [36]. Additionally, cortical tension can give rise to an apparent elastic response [11], and transmission of forces by stress fibers violate predictions of continuum models [37]. Still, the cell rheological response has a contribution that can be interpreted as the response of purified cytoskeletal components [36]. Many studies are consistent with a sum of two power laws:

$$G^*(\omega) = Ae^{\pi i\beta/2}\omega^\beta + Be^{3\pi i/8}\omega^{3/4},\tag{1.1}$$

where A and B are constants, and β is a small exponent, usually between 0.1 and 0.3 [38, 39, 40]. The ω^β term dominates at low frequencies or long times, while the $\omega^{3/4}$ term dominates at shorter time scales than are relevant for cell migration [38].

11

The elasticity of a cell can depend on many factors, such as the underlying substrate [41, 42], contact with other cells [43], and migration state [44]. Additionally, in the last case a migrating cell can exhibit nonuniform mechanical properties [44]. Cells also have nonlinear elasticity, such that the stiffness can be significantly altered by internal prestresses or external stress [36].

Micropipette suction (mentioned in the previous section) can be used to measure elastic constants of solid-like cells or nuclei, as well as to distinguish between solid-like and liquid-like behaviors [20, 30]. It can also measure viscoelastic response to some extent [45]. Methods capable of rheological measurement ($G^*(\omega)$) include [38] magnetic twisting cytometry, atomic force microscopy (AFM) [40], and particle-tracking microrheology [35, 46]. Among these, the last is different in that it probes local and length-scale-dependent viscoelastic properties of the cell interior. All "global" measurements probe a combined response of cytoplasm and nucleus, e.g. whether an AFM tip is placed vertically above the nucleus or away from it [35] (as in Ref. [9]). This distinguishes internal and external techniques, which differ in the measured value of the exponent β. They have been confirmed to probe distinct structures within the cell. A problem common to both categories is that the measured overall stiffness can be affected by unknowns in linkages between probes and the cell [38].

Typical values of measured elastic moduli of cells on a time scale of seconds range from 0.1 to 10 kPa [36], and AFM measurements show that 1 μm indentations of cells relax within seconds [40, 47].

When cells squeeze through constrictions, the deformation of the nucleus can be the rate-limiting step as the nucleus can be an order of magnitude stiffer than

the cytoplasm [48]. Both the nuclear envelope and nuclear interior contribute to the mechanical properties of the nucleus. The nucleus is the stiffest organelle in the cell, being 2 to 10 times stiffer than the cytoskeleton and having typical measured elastic moduli ranging from 0.1 to 10 kPa, depending on cell type and experimental method [29]. The nuclear interior exhibits solid-like behavior on length scales of ~ 100 nm and time scales of seconds [46]. On time scales relevant to cell migration (minutes), the nucleus is primarily elastic [49]. The nucleus cannot move freely inside the cell. It is anchored to cytoskeletal filaments, and these connections contribute to cell mechanical properties comparably to the cytoskeletal filaments themselves [35, 50]. Accordingly, deformation of the cell elicits deformation of the nucleus [51]. Additionally, the nucleus is tightly linked with the centrosome, which is generally located near the geometric cell center [52]. During migration, the nucleus is usually positioned behind the centrosome, but in some cases, it can be in front [49].

1.5 Cell monolayers

Cell layers exhibit a number of properties of physical interest. Biological tissues are resistant to stress, yet allow plastic deformations by remodeling. Different cell types sort out as in a mixture of two immiscible fluids [53]. Cells in epithelial monolayers are polygon-shaped, and their geometry and topology can be studied. In addition, they intercalate (exchange neighbors) using processes that involve shrinking of edges between two cells and subsequent appearance of new edges [54]. Layers of motile cells can exhibit collective migration, where coherence of a group of cells is preserved and their motions are correlated [1]. Upon increasing density of cells, a transition from collective relaxation to a glass-like state where motions of cells are

13

highly localized and constrained. Dynamic heterogeneities in speed [55] and large-scale swirling patterns [56] have also been observed.

Various types of models have been put forth to describe behavior of collections of motile cells. Physics-based models are essentially based on mechanical interactions within a cell (e.g. elasticity), between cells (e.g. volume exclusion, adhesion), and between cell and substrate (traction). Models can differ with respect to single-cell locomotion, cell–cell interaction, and level of description of the cell. Single-cell locomotion is described using a polarity vector which signifies the in-plane polarization of the cell, i.e., the "attempted" direction of motion of the cell. The cell–cell interaction can affect the cell polarity. For example, the phenomenon of contact inhibition of locomotion directs a cell's polarization away from a contact with another cell [10]. (Protrusions still form in confluent monolayers, beneath neighboring cells, known as "cryptic lamellipodia" [57].) Neighbor-aligning interactions are sometimes used, although it has no experimental basis. The description of the cell can range from point-like particles to polygons (used particularly for confluent monolayers) to an arbitrarily-shaped region described by phase-field models, or several interacting particles [10].

Self-propelled particle models formulate motion using interparticle interactions that include repulsion and attraction (adhesion). However, mechanical properties of individual cells are not modeled. Nevertheless, these have had some success when compared with experiment. A flocking transition with increasing density was obtained using a model in which cell polarization aligned to cell velocity, with a force function incorporating both repulsion and attraction [58]. A different particle model

14

with neighbor-aligning interactions succeeded in reproducing statistical quantities such as spatial and temporal correlations of the velocity field [59].

Particle models can be considered to model the cell as an isotropic/circular shape [10]. However, in a confluent monolayer, cells rather have a polygonal shape, where edges are junctions between two cells, and vertices are where three or more cells come into contact [54]. These are often modeled using a vertex model, where the polygonal boundaries of cells are specified, or a Voronoi model, in which the cell location is considered as a single point and the cells are modeled as the Dirichlet tessellation [10]. These models can incorporate cell volume or area, adhesion, active contractility, and membrane tension [60, 61]. They have had experimental support in predicting jamming transitions induced by a balance between cortical tension and cell–cell adhesion, both in nonmotile [60] and motile [61] cells.

Phase-field models are in a sense the most flexible, allowing an arbitrary shape of interface and able to incorporate a curvature-dependent term in the energy and additional structure such as relevant chemical concentrations [62, 63] or a nucleus [62]. By adjusting parameters, they have been able to reproduce experimental observations in single cells [64, 65] as well as small collections of cells in confined regions [62, 63]. However, phase-field models are computationally much more expensive and are suitable for relatively small (< 100) collections of cells [10].

Models for cells on a substrate are usually formulated in 2D. Cells are often modeled as having a constant volume [60] or area [64]; however, in confluent mono-layers, volume fluctuations can reach 20%, suggesting that cell volume regulation in monolayers is different than in isolated cells, possibly involving intercellular fluid

15

exchange through gap junctions [66]. Energetics of the cell boundary can include interfacial tension [60, 64], active contractility of the cell cortex [60], or bending energy [64].

In this thesis, we use the phase-field model to simulate cells of different elasticities, interacting via volume exclusion. We also incorporate a nucleus phase-field with an associated elasticity. We use a simple surface-tension model for the membrane energy and do not consider active contractility or more sophisticated models for the bending energy.

CHAPTER 2
Introduction to Phase-Field Models

The phase-field method is now an established approach in materials science to model microstructures relevant in material properties such as mechanical strength or dynamic processes such as solidification of pure materials or binary alloys. It has been made possible by modern parallel computing. The phase-field is a continuum field that is homogeneous in each bulk phase (e.g. solid or liquid) and interpolates rapidly (but continuously) between them in a diffuse boundary layer. Its dynamics is based on thermodynamic free energy considerations, as explained below. However, numerical simulation of phase-field models can be costly, requiring not only high spatial resolution within the diffuse interface but also concomitantly small time-steps for numerical stability (see discussions in subsequent chapters). Thus, sharp-interface limits have been analyzed in order to simulate dynamics of the interfaces alone [67].

2.1 Landau–Ginzburg free energy

The conventional prescription of statistical mechanics is that the thermodynamic state of a system (e.g. a container of fluid) is characterized by a set of thermodynamic variables (e.g. number of particles, volume, and temperature for the canonical ensemble), and from the Hamiltonian one can calculate the appropriate thermodynamic potential and expectation values of physical quantities. Such calculations will respect symmetries of the Hamiltonian. However, systems can sometimes be in states

17

of broken symmetry, where macroscopic quantities violate symmetries of the Hamiltonian. Examples include crystals (broken translational and rotational invariance) and ferromagnets with non-zero net magnetization in the absence of an external magnetic field (broken rotational invariance of magnetization). In these cases, additional variables are needed for a complete thermodynamic description. They are the order parameters that describe the symmetry breaking: for example, net magnetization in a ferromagnet. Symmetry breaking can accompany a phase transition, e.g. the paramagnetic-to-ferromagnetic transition upon cooling through the Curie temperature. Landau theory aims to describe properties of a system in the vicinity of such transitions. It is also applied when symmetry breaking is not involved, such as liquid–gas transitions [68].

To introduce the phenomenological Landau free energy, we take as a specific example the Ising model, consisting of N classical "spins" s_i placed on a lattice. The spins can point either up or down, encoded by the values $s_i = \pm 1$. The Hamiltonian is:

$$H = -J \sum_{\langle i,j \rangle} s_i s_j, \tag{2.1}$$

where $\langle i, j \rangle$ denote nearest-neighbor pairs, and $J > 0$ for a ferromagnetic interaction. The order parameter is the average magnetization: $\phi := \left(\sum_i s_i \right) / N$. We work in the canonical ensemble, so we want to find the free energy $F(T, N, \phi)$ (where $T =$ temperature). The entropy of mixing is (k_{B} is the Boltzmann constant):

$$\frac{S}{k_{\mathrm{B}} N} = \ln 2 - \frac{1}{2} \left[(1 + \phi) \ln(1 + \phi) + (1 - \phi) \ln(1 - \phi) \right]. \tag{2.2}$$

18

In Bragg–Williams mean-field theory, the energy is approximated as:

$$E = \langle H \rangle \approx -J \sum_{\langle i,j \rangle} \langle s_i \rangle \langle s_j \rangle = -\frac{JNz}{2} \phi^2, \tag{2.3}$$

where z is the coordination number of the lattice. The free energy per spin can be readily found as $f(T, \phi) = (E - TS)/N$. The equilibrium value(s) of ϕ are found by minimizing f, which requires $\partial f / \partial \phi = 0$ [68].

For high T, the function $f(T, \phi)$ is found to have a single minimum at $\phi = 0$, while for low T, it has two minima at $\phi = \pm \phi_{\mathrm{eq}}$. The crossover point is the critical temperature $T_c = zJ/k_B$. For $T \approx T_c$, the order parameter is small and f can be expanded in a power series:

$$f(T, \phi) = -k_B T \ln 2 + \frac{k_B}{2}(T - T_c)\phi^2 + \frac{k_B T}{12}\phi^4 + O(\phi^6), \tag{2.4}$$

where including the ϕ^4 term is necessary for stability, and it can be derived that $\phi_{\mathrm{eq}} \sim (T_c - T)^{1/2}$. Asymmetry can be introduced by applying an external field, adding a term $-h\phi$ to $f(T, \phi)$.

For more complicated systems, a formula for the free energy is not so easily written down. More easily, near the critical point, a phenomenological Landau free energy is written down as a power series in the order parameter, where each term respects the symmetries of the system. It applies if the order parameter is small near the critical point. For the Ising model, such a free energy would look like:

$$f(T, \phi) = \frac{a(T)}{2}\phi^2 + \frac{c(T)}{4}\phi^4, \tag{2.5}$$

where there is no ϕ^3 term because of symmetry $\phi \to -\phi$, the coefficient $c(T) > 0$, and the coefficient $a(T)$ changes sign at T_c [68].

So far, we have been restricting to the case that the order parameter is spatially uniform: $\langle s_i \rangle = \phi$ for every site i. To build a theory for a spatially varying order parameter $\phi(\mathbf{x})$, we add an energy cost for spatial variations. This leads to a phenomenological Landau–Ginzburg free energy functional:

$$F[\phi] = \int d^d\mathbf{x}\, f(\phi(\mathbf{x})) + \frac{1}{2}C \int d^d\mathbf{x}\, |\nabla\phi(\mathbf{x})|^2, \tag{2.6}$$

where $C > 0$, and $f(\phi)$ now represents a free energy density. The continuum field $\phi(\mathbf{x})$ represents a coarse-grained description. The squaring of the gradient preserves spatial isotropy, and neglecting higher-order gradient terms is justified for spatial variations of ϕ that are gradual on a microscopic scale [68]. In the Ising model, this form can be shown to have a microscopic origin, as:

$$-\sum_{\langle i,j \rangle} s_i s_j = \frac{1}{2}\sum_{\langle i,j \rangle} \left[(s_i - s_j)^2 - s_i^2 - s_j^2\right]$$

$$= -N + \frac{1}{2}\sum_{\langle i,j \rangle}(s_i - s_j)^2, \tag{2.7}$$

where the second term gives rise to the gradient energy [68, 67]. The thermodynamic driving force is defined as $\delta F/\delta\phi$, and it is zero in equilibrium [67].

The local order parameter in the Ising model is usually a nonconserved field when a ferromagnet is modeled. An example of a conserved order parameter is concentration in a binary mixture, in which local modifications are necessarily coupled to nearby fluxes. The phenomenological free energy in such a system is identical

in form to the nonconserved case, but different conditions exist for thermodynamic stability and metastability, as in (bulk) phase separation and the spinodal curve, respectively. In this case, a chemical potential field can be derived as $\delta F/\delta\phi$, and its gradient is the thermodynamic driving force for fluxes.

For a symmetric Landau free energy density, a stationary planar interface between the two phases $\pm\phi_{\text{eq}} := \pm\sqrt{a(T)/c(T)}$ for $T < T_{\text{c}}$ can be solved for:

$$\phi(x) = \phi_{\text{eq}} \tanh\left(\frac{x}{2\xi}\right), \tag{2.8}$$

where $\xi = \sqrt{C/(2a)}$ (the dependence on temperature T has been dropped) [69]. The above form can be used to calculate interfacial tension σ between the two bulk phases. For constant a/c, the proportionality is $\sigma \propto 1/\xi$.

2.2 Nonequilibrium dynamics

We consider a Landau–Ginzburg free energy (Eq. 2.6), with

$$f(\phi) = -h\phi - \frac{a}{2}\phi^2 + \frac{c}{4}\phi^4, \tag{2.9}$$

where $a > 0$, $c > 0$. To model dynamics of a nonconserved order parameter, we use the equation:

$$\frac{\partial\phi}{\partial t} = -M\frac{\delta F}{\delta\phi} = M\left(C\nabla^2\phi + h + a\phi - c\phi^3\right), \tag{2.10}$$

where M is a mobility, emulating overdamped Langevin dynamics. A white noise term can be added to simulate thermal fluctuations. For a conserved order parameter, the flux density $\mathbf{j} = -M\nabla\left(\delta F/\delta\phi\right)$, and assuming M is constant:

$$\frac{\partial\phi}{\partial t} = -\nabla\cdot\mathbf{j} = M\nabla^2\frac{\delta F}{\delta\phi}. \tag{2.11}$$

21

In this case, a white noise can be added to $\mathbf{j}$ to give a suitable noise for ϕ. The nonconserved and conserved dynamics above are known as Model A and Model B, respectively, in the Hohenberg–Halperin classification [67].

For Model A, a diffusion coefficient $D := MC$ can be extracted. To obtain sharp-interface dynamics in 2D, we consider the limits where $\xi \ll 1/|K|$ where K is the local curvature of the interface, and where the asymmetry in $f(\phi)$ is small. Concretely, the latter means that $f(\phi)$ has two minima $\phi_{\mathrm{eq}}^{\pm}$ close to $\pm\sqrt{a/c}$, and the bulk free energy density difference $\Delta f := f(\phi_{\mathrm{eq}}^{+}) - f(\phi_{\mathrm{eq}}^{-})$ is small compared to the barrier height. We also assume that the interface velocity V is low enough for diffusional relaxation, the Péclet number $\xi|V|/D \ll 1$. The normal velocity of the interface can then shown to be [67, 69]:

$$V = D\left(-K + \frac{\Delta f}{\sigma}\right),\tag{2.12}$$

where the normal $\hat{\mathbf{n}}$ points $\phi_{\mathrm{eq}}^{-} \to \phi_{\mathrm{eq}}^{+}$, and the sign of K is chosen compatibly, as $d\hat{\mathbf{t}}/ds = -K\hat{\mathbf{n}}$ where s is an arc-length variable at the time-point of evaluation, and $\hat{\mathbf{t}}$ the unit tangent vector of the (sharp) interface.

We consider an interface enclosing a homogeneous region of a bulk phase. The curvature K satisfies a Gauss–Bonnet formula:

$$\oint ds\, K = 2\pi,\tag{2.13}$$

where now the sign of K is chosen compatibly with an outward normal. It can be seen from $K = d\theta/ds$ where $\hat{\mathbf{t}} = (\cos\theta, \sin\theta)$. This can be used to calculate the rate of change of the enclosed area in the absence of asymmetry ($\Delta f = 0$). Traversing

22

the interface counterclockwise gives:

$$\frac{d}{dt} \oint ds\, x\frac{\partial y}{\partial s} = \oint ds\, \left(\frac{\partial x}{\partial t}\frac{\partial y}{\partial s} + x\frac{\partial^2 y}{\partial s \partial t}\right)$$

$$= \oint ds\, \left(\frac{\partial x}{\partial t}\frac{\partial y}{\partial s} - \frac{\partial x}{\partial s}\frac{\partial y}{\partial t}\right)$$

$$= -D \oint ds\, K\left(n_x t_y - n_y t_x\right)$$

$$= -2\pi D,$$

(2.14)

where we used $\hat{\mathbf{n}} = (t_y, -t_x)$. This shows that curvature-driven interface dynamics tends to shrink confined phases, partly responsible for domain coarsening. The quantitative value can be easily confirmed when initializing with a circular domain [70].

The interface dynamics of Eq. 2.12 can also be derived from a free energy of a closed interface defined as $\bar{F}[\mathbf{R}] := \sigma P + A\Delta f$, where P and A are the perimeter of and area enclosed by the interface $\mathbf{R}(s)$. Then $d\mathbf{R}/dt = -(D/\sigma)\delta\bar{F}/\delta\mathbf{R}$. The variation in perimeter is:

$$\delta \oint ds\, \left|\frac{\partial \mathbf{R}}{\partial s}\right| = \oint ds\, \frac{\partial \mathbf{R}/\partial s}{|\partial \mathbf{R}/\partial s|} \cdot \delta\left(\frac{\partial \mathbf{R}}{\partial s}\right)$$

$$= -\oint ds\, \frac{\partial \hat{\mathbf{t}}}{\partial s} \cdot \delta\mathbf{R}$$

$$= \oint ds\, K\hat{\mathbf{n}} \cdot \delta\mathbf{R},$$

(2.15)

and the variation in area is derived similarly to dA/dt above.

Sharp-interface limits for Model B have also been derived [67, 69], but they are more involved and we will not review them here.

CHAPTER 3
Phase-Field Model for Cell Monolayers

Computer simulation studies were done by Palmieri *et al.* [71] where cells were modeled using a phase-field description where normal and cancer cells were identical except for their elasticities. The results qualitatively reproduce many features experimentally observed by Lee *et al.* [9], e.g. speed bursts preceded by deformation, including the amplitude and frequency of the bursts. However, the cell was defined by a single elastic constant with no distinction between cytoplasm and nucleus. In reality, the cytoplasm is about an order of magnitude softer than the nucleus [5], and so the nuclei play an important role in the phenomenon of cancer cell migration. Therefore, we extend the model to include the cell nuclei.

3.1   Cytoplasm-only model

We model cells as droplets described by a phase-field model with a phenomenological cell–cell repulsion term as in Ref. [67]. We follow Ref. [71] in using Model A dynamics with a Lagrange multiplier term to control the cell area. This was used because Model B dynamics had proved to require too much computational resources.

We use and extend model the in Ref. [71], in which more details and discussion can be found (particularly in the Supplemental Information). The free energy consists of a Landau–Ginzburg term with a double-well potential for each cell, penalization

24

of area deviation, and cell–cell repulsion:

$$\mathcal{F}_n = \gamma_n \int d^2\mathbf{r} \left(|\nabla\phi_n|^2 + \frac{30}{\lambda^2}\phi_n^2(1-\phi_n)^2 \right) + \frac{\mu_n}{\pi R^2} \left(\int d^2\mathbf{r}\, \phi_n^2 - \pi R^2 \right)^2$$

$$\mathcal{F} = \sum_n \mathcal{F}_n + \frac{30\kappa}{\lambda^2} \sum_{n,m\neq n} \int d^2\mathbf{r}\, \phi_n^2\phi_m^2,$$

(3.1)

where m and n are cell indices, ϕ_n is the field of cell n, γ_n is its elasticity which penalizes interface, λ is the interface width, R is the nominal radius of the cell (all cells have identical nominal size), μ_n is a Lagrange multiplier that controls the cell area, κ controls cell–cell repulsion, $\mathcal{F}_n$ is the free energy associated with cell n, and $\mathcal{F}$ is the total free energy. The interface dynamics of an isolated cell is independent of λ when $\lambda \ll 1/|K|$ where K is the local curvature of the interface [69], if μ is taken proportional to $1/\lambda$ (see Ref. [72] and also next chapter, "Sharp-Interface Model for Cell Monolayers"). So, it is chosen that "$\lambda \ll R$". We use the values in Ref. [71] unless otherwise specified (Table 3–1). In particular, the elasticity ratio between normal and soft cells is chosen based on the experimental measurements [9]. Also, a thinner interface would require a finer mesh for numerical simulation (see section "With nucleus") and, to keep it computationally feasible, would possibly require an adaptive mesh. This would involve a significantly more complicated algorithm. Also, we wish to compare our results to the studies of Refs. [71, 72].

We choose κ and μ large enough so that when cells are contacted by other cells, they will deform instead of (significantly) overlap or become (significantly) smaller than πR^2 in area measure (defined as $\int d^2\mathbf{r}\, \phi^2$). In the presence of a neighboring cell, the "effective" Landau-free-energy-density ($\nabla\phi$-independent) component for a

Table 3–1: List of model parameters for cytoplasm-only simulations. Taken from Ref. [71].

	Normal	Soft
γ_n	1	0.35
λ	7	
R	49	
μ_n	1	
κ	10	

cell field becomes an asymmetric double-well (favoring $\phi = 0$) or eventually single-welled. The squaring of ϕ in the repulsion potential prevents it from preferring to be negative. The squaring of ϕ in the area term might be considered to have the same effect; however, the surface tension drives down the area (see previous chapter). Because the interfaces are discouraged from overlapping, they also become thinner, particularly that of the soft cell. Thus a correct measure of interface perimeter involves integrating $|\nabla\phi|$ rather than $|\nabla\phi|^2$.

For the equation of evolution, the cells are endowed with a velocity $\mathbf{v}_n$, so that:

$$\frac{\partial\phi_n}{\partial t} + \mathbf{v}_n(t) \cdot \nabla\phi_n = -\frac{1}{2}\frac{\delta\mathcal{F}}{\delta\phi_n}, \tag{3.2}$$

where the field mobility (aside from elasticity γ_n) is assumed to be the same for all cells and absorbed into the time-unit. The $\mathbf{v}_n(t)$ has two contributions, an "active" term $\mathbf{v}_{n,\mathrm{A}}(t)$ representing the attempted crawling of the cell, and an "inactive" term $\mathbf{v}_{n,\mathrm{I}}(t)$ which effectively modifies the "active" term in response to contact with other cells. The latter must be consistent with thermodynamics (i.e., non-increasing free-energy with time in the absence of the active term) and is taken to have the form

[71]:

$$\mathbf{v}_{n,\mathrm{I}}(t) = \frac{60\kappa}{\lambda^2 \xi_n} \int d^2\mathbf{r}\, \phi_n \nabla \phi_n \sum_{m \neq n} \phi_m^2, \qquad (3.3)$$

where ξ_n^{-1} is a coefficient that is possibly 0. This is obtained by considering the change $\phi_n(\mathbf{r}) \to \phi_n(\mathbf{r} - d\mathbf{r})$ and calculating the associated free energy change. In Ref. [71], $\xi_n = 1500$ is chosen.

The active term $\mathbf{v}_{n,\mathrm{A}}(t)$ has a constant magnitude of $v_{\mathrm{A}} = 10^{-2}$ and has a direction which is piecewise constant in time, changing at discrete reorientation events following a Poisson process with rate $\tau_r^{-1} = 10^{-4}$, with the new direction chosen uniformly. The choices of γ_n and v_{A} satisfy separation of time scales for motility and shape relaxation [71]. The choice of $v_{\mathrm{A}}\tau_r \approx 2R$ is made to agree with video microscopy [9], although experimentally the direction of cell motion changes continuously, with a persistence time characterizing the magnitude of the random fluctuations.

The functional derivative of the free energy is given by [71]:

$$\begin{aligned}
\frac{1}{2}\frac{\delta \mathcal{F}}{\delta \phi_n} &= -\gamma_n \nabla^2 \phi_n + \frac{30}{\lambda^2}\left[\gamma_n \phi_n(1-\phi_n)(1-2\phi_n) + 2\kappa\phi_n \sum_{m \neq n} \phi_m^2\right] \\
&\quad + \frac{2\mu_n}{\pi R^2}\left(\int d^2\mathbf{r}\, \phi_n^2 - \pi R^2\right)\phi_n.
\end{aligned} \qquad (3.4)$$

It suffices to work with values of μ_n and ξ_n that are the same for all cells, so we hereafter omit the indices for these variables.

No random noise is used in the dynamics of the fields themselves, because random driving is already provided by stochastic cell locomotion.

We study a system of $N_{\text{cells}} = 72$ cells in a square simulation box with periodic boundary conditions. The density of cells is defined as $\rho := N_{\text{cells}} \pi R^2 / L^2$ where L is the length of the simulation box. One cell is "soft" while the rest have "normal" stiffness. The density measure ρ does not take the interface width into account. We use $\rho = 0.9$ as in Ref. [71] unless otherwise specified.

3.1.1 Numerical algorithm

We use a Euler time-stepping algorithm on a discrete uniform square mesh, $\Delta x = \Delta y =: \Delta$. We use central differences for spatial derivatives and a 9-point stencil for the Laplacian [67]:

$$\frac{\partial}{\partial x} f(x, y) = \frac{f(x + \Delta, y) - f(x - \Delta, y)}{2\Delta} + O(\Delta^2), \tag{3.5}$$

and analogously for $\partial/\partial y$, and

$$\nabla^2 f(x, y) = \frac{1}{\Delta^2} \left(\frac{f(x + \Delta, y) + f(x - \Delta, y) + f(x, y + \Delta) + f(x, y - \Delta)}{2} \right.$$
$$+ \frac{f(x+\Delta, y+\Delta) + f(x+\Delta, y-\Delta) + f(x-\Delta, y+\Delta) + f(x-\Delta, y-\Delta)}{4}$$
$$\left. - 3f(x, y) \right) + O(\Delta^2). $$
$$\tag{3.6}$$

It can be seen that more weight is put on the closer points. The length of the simulation box must be an integer multiple of the mesh spacing: $L = N_{\text{mesh}} \Delta$.

To facilitate computation of the Laplacian, we keep in memory each field as a $(N_{\text{mesh}} + 2) \times (N_{\text{mesh}} + 2)$ array, extended from the $N_{\text{mesh}} \times N_{\text{mesh}}$ simulation box by one mesh unit in the $-x$, $+x$, $-y$, and $+y$ directions, according to periodic boundary

conditions, as described in Ref. [67]. The central $N_{\mathrm{mesh}} \times N_{\mathrm{mesh}}$ subarray (in fact, only part of it; see below) is updated first according to the Euler scheme, and then the remainder is updated according the periodic boundary conditions.

Because each cell takes up only a small fraction of the total area of the simulation box, we only update each cell within a small rectangle "enclosing the cell" (clarified below), which may wrap around the edges of the simulation box. This rectangle is updated over time, on the order of once per time-unit. It is constructed by first forming the smallest rectangle containing all the points at which the cell field exceeds 0.05. It is then extended in both directions in each dimension by a margin that we take as equal to the interface width, $\lambda = 7$. The cell field is enforced to be 0 outside this rectangle when it is updated. Before calculating repulsion from another cell, it is checked that their rectangles overlap, and calculation done only in their intersection.

Integrals were calculated simply by summing all the integrand values inside the bounding rectangle and dividing by Δ^2, of accuracy $O(\Delta^2)$ by equivalence with a 2D trapezoidal rule because the cell field supposedly vanishes outside the bounding rectangle.

An equilibrium interface has the profile [72]:

$$\phi^*(x) = \frac{1 + \tanh(\alpha x)}{2}, \quad \alpha = \frac{\sqrt{15/2}}{\lambda}. \tag{3.7}$$

From this it can be estimated that the method senses cells that are 3 interface widths away, measuring between the centers of the profiles.

The reorientation events were determined using a time-step of $\Delta t_r = 0.06$ with reorientation probability $1 - e^{-\Delta t_r / \tau_r}$ for each cell. The probability of selecting the

29

next direction at time $n\Delta t_r$ follows a geometric distribution:

$$P(n) = e^{-(n-1)\Delta t_r/\tau_r} \left(1 - e^{-\Delta t_r/\tau_r}\right), \quad n = 1, 2, 3, \ldots. \tag{3.8}$$

A single set of reorientation events was generated for the 72-cell system and used for all the runs for reproducibility.

The algorithm was coded in C++ with OpenMP. Random number generation in parallel cannot be done with `rand()`; instead, it requires thread-safe random number generators such as that provided by the GNU Scientific Library [73]. (Parallelization of random number generation is not necessary in this case, since we are not additionally adding noise to the field dynamics.)

Due to the combination of a simple update rule and relatively high memory requirements, this is a memory-bound algorithm [74], as confirmed by modifying the update rule to a trivial one which is "artificial". Two arrays are used which alternate storing the field values, so that updates are not written to an array during computation of its Laplacian, and cache misses are not incurred by unnecessary copying.

To display the cells, we use a discrete RGB (red-green-blue) scheme where field values ϕ are partitioned into categories of $\phi > 0.95$, $0.05 < \phi \leq 0.95$, and $\phi \leq 0.05$. This is the primary way that results of simulations are saved. Full values of the cell fields are saved sparingly (≤ 1 in 30 RGB images) because of the forbidding memory requirements. In the active simulations (except the collision experiments), an RGB image is stored every 180 time-units. Videos were typically watched at 5400 time-units per second.

3.1.2 Numerical stability

We neglect the area conservation term because the fractional area deviation $\left(\int d^2\mathbf{r}\, \phi^2 \right) / (\pi R^2) - 1$ is small (see subsection "Active simulations"). We consider the numerical stability of the Euler time-stepping procedure for the (now local) equations of motion linearized around the bulk value $\phi_n = 0$. We do not consider the velocity term here. The requirement for the time-step Δt in the absence of cell–cell repulsion has been calculated in Ref. [75] for a 5-stencil Laplacian. A similar calculation for the 9-stencil Laplacian gives:

$$\Delta t < \left(\frac{2\gamma_n}{\Delta^2} + \frac{15\gamma_n}{\lambda^2} \right)^{-1}, \tag{3.9}$$

which applies to linearization around both bulk values $\phi_n = 0$ and $\phi_n = 1$. The term arising from the Laplacian dominates for $\Delta = 1$.

The repulsion term is considered with the worst-case value of the neighbor $\phi_m = 1$; it is then modified to:

$$\Delta t < \left(\frac{2\gamma_n}{\Delta^2} + \frac{15\gamma_n}{\lambda^2} + \frac{30\kappa}{\lambda^2} \right)^{-1}. \tag{3.10}$$

The term arising from cell–cell repulsion now dominates. For our model parameters, this evaluates to $\Delta t < 0.119$ for normal cells.

3.1.3 Initialization and equilibration

We study a system with $\rho = 0.9$, $L = 777$. Cell centers were placed in sequence at random points in the simulation box, where cell centers are forbidden to come within $1.5R$ of each other and if that occurs, a new location is chosen at random.

Each cell is initialized as a Heaviside step function $\Theta(R-r_n)$, where r_n is the distance between the field point and the chosen center of cell n.

For equilibration, evolution is done without the active velocity term. The time-step was taken to be $\Delta t = 0.001$ for $0 \leq t < 1815$ and then $\Delta t = 0.01$ increasing to $\Delta t = 0.06$ in discrete jumps for $1815 \leq t \lesssim 40000$ (Fig. 3–1). The oval-shaped empty spaces between adjacent cells at $t = 15$ could be due to a time-step that is not small enough, but this was not investigated as it was considered unimportant.
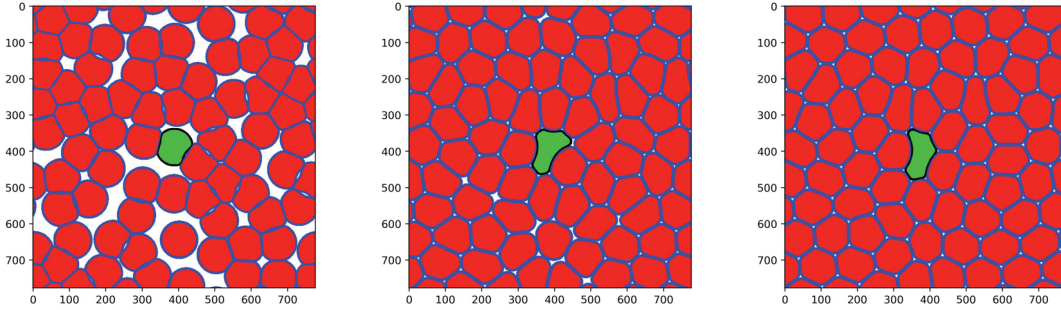

Figure 3–1: Equilibration of cytoplasm-only cell monolayer. From left to right: $t = 15$, $t = 1815$, $t \approx 40000$.

3.1.4 Active simulations

Initially, it was thought to compare the trajectories with and without the nucleus. Thus most of the cytoplasm-only simulations were initialized from equilibration of the models with the nucleus. Two such initial conditions were used, described in the section "Model with nucleus". They are qualitatively similar to the equilibrated cytoplasm-only system.

Using initial condition I: Comparing $\Delta t = 0.06$ and $\Delta t = 0.01$, by $t = 16200$ the soft cell already took a different trajectory and the cells around it had rearranged

differently. It seems that at a certain time, the error reaches a "tipping point" at which the cells choose a different rearrangement to undergo. Thus numerical error does not accumulate in a steady fashion. This showed that $\Delta t = 0.06$ was too large to be acceptable. Such a change was not observed decreasing to $\Delta t = 0.003$ running to $t = 35640$. However, this is an insufficient length of simulation time to validate numerical parameters. Accuracy may be easier to achieve during the first ≈ 50000 time-units because of the equilibrated initial conditions. Thinning of the interface of the soft cell can be seen, as well as sharp corners on its boundary (Fig. 3–2).
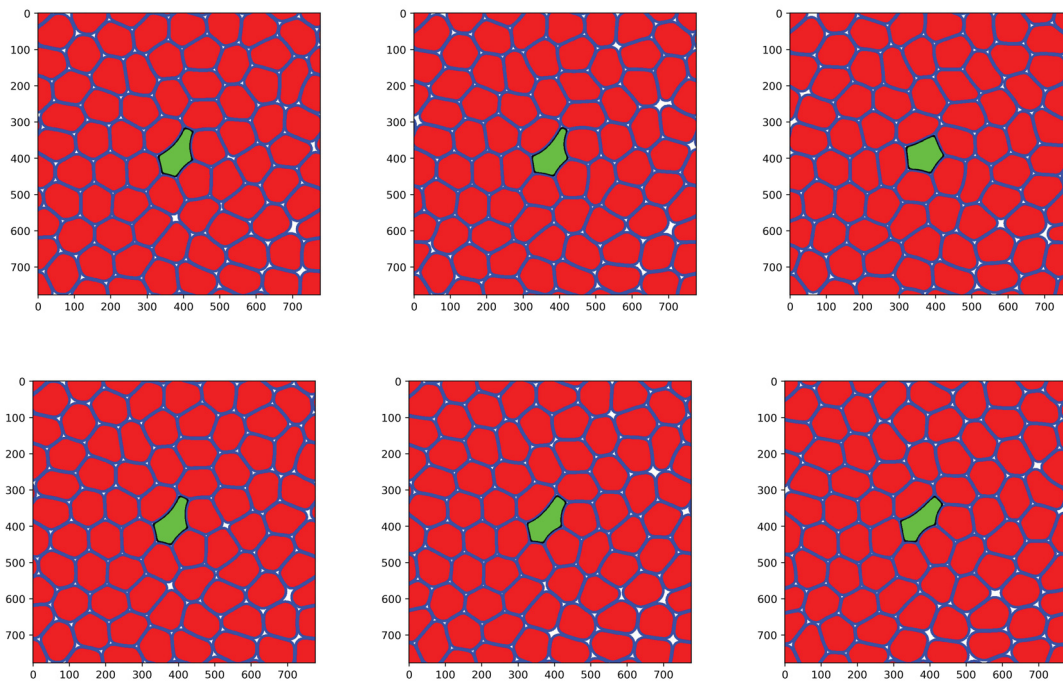

Figure 3–2: Snapshots of active simulations of cytoplasm-only cell monolayer, with inactive velocity. Top row: $\Delta t = 0.06$. Bottom row: $\Delta t = 0.01$. From left to right: $t = 16200$, $t = 18900$, $t = 21600$.

33

The inactive velocity reached at most a few percent of the active velocity in these very short runs. However, in longer runs, it could reach up to 10% [76].

Using initial condition II, we initially set $\xi = \infty$ (no inactive velocity). Thus far, the value of Δ had not been validated. Decreasing Δ gave changes in the simulations, and suggested further decreases of Δt (Fig. 3–3). Obtaining accurate cell rearrangement dynamics may require $\Delta \leq 0.6$ and $\Delta t \leq 0.0025$. Possible reasons for this difficulty (if true) may be the thinning of the interface as well as large curvatures (see Fig. 3–4), and propagation of collision forces. Using $\xi = 1500$ and $\xi = 500$ gave differences noticeably larger than those obtained by changing numerical parameters. Length of videos ranged from approximately 43200 to 81000 time-units. For all the simulations with the nucleus (next section), the inactive velocity was discarded as it seemed unnecessary.

We explored increasing κ and μ by a factor of 4. As expected, this discourages empty space (Fig. 3–5; compare Fig. 3–2). However, due to possible numerical inaccuracy, the true effect might not quite be correctly captured.

In all the simulations of this model, the area measure $\int d^2\mathbf{r}\,\phi^2$ typically deviated by no more than $\sim 1\%$ from πR^2, and did not seem to go over πR^2. However, deviations of the cell field from its equilibrium value (which is > 1; see Supplemental Information of Ref. [71]), and corresponding deviations in the "geometrical area" of the cell, were more substantial (see Fig. 3–4). Within the model we use, there is not an additional parameter we can use to control this.
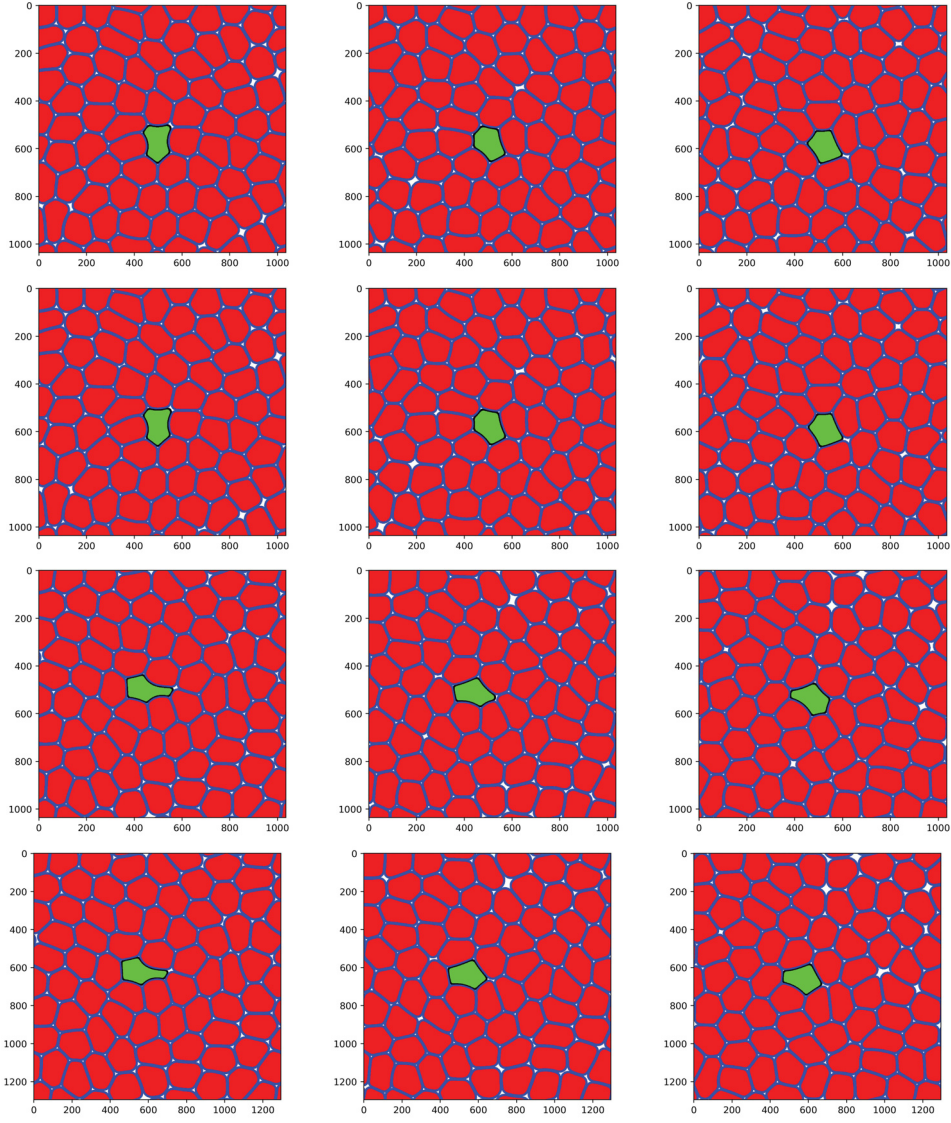
Figure 3–3: Snapshots of active simulations of cytoplasm-only cell monolayer, without inactive velocity. Top row: $\Delta t = 0.005$, $\Delta = 0.75$; (L–R) $t = 54000$, $t = 59400$, $t = 64800$. 2$^{\text{nd}}$ row: $\Delta t = 0.002$, $\Delta = 0.75$; (L–R) $t = 54000$, $t = 59400$, $t = 64800$. 3$^{\text{rd}}$ row: $\Delta t = 0.01$, $\Delta = 0.75$; (L–R) $t = 32400$, $t = 37800$, $t = 43020$; nearly identical to $\Delta t = 0.002$. Bottom row: $\Delta t = 0.01$, $\Delta = 0.6$; (L–R) $t = 32400$, $t = 37800$, $t = 43020$; nearly identical to $\Delta t = 0.0025$.
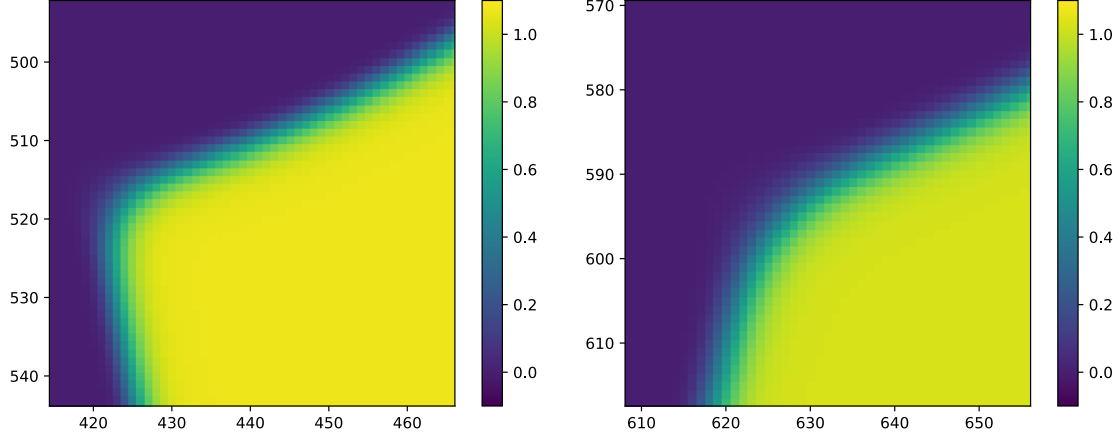
Figure 3–4: Cell field profiles in active simulations of cytoplasm-only cell monolayer. Used initial condition II, without inactive velocity, with $\Delta t = 0.002$ and $\Delta = 0.75$ at $t = 16200$. A single pixel is 1 mesh point. (L) The soft cell; the field value in its interior is ≈ 1.07. (R) A normal cell; the field value in its interior is ≈ 1.03.

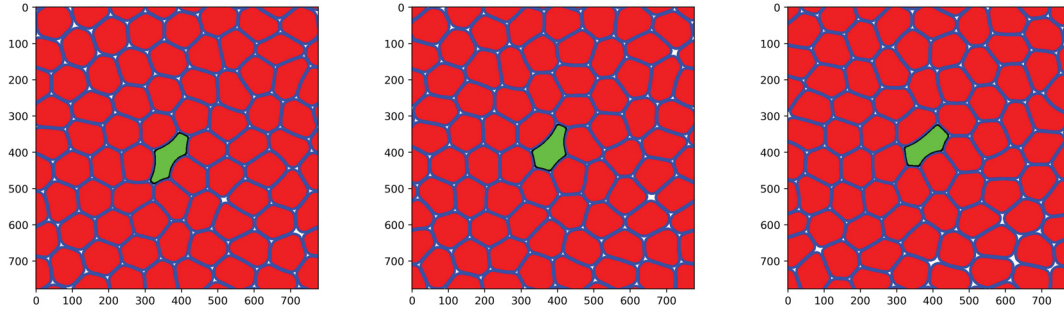Figure 3–5: Snapshots of active simulations of cytoplasm-only cell monolayer, without inactive velocity, with $\kappa = 40$ and $\mu = 4$. (L–R) $t = 10800$, $t = 16200$, $t = 21600$.

36

3.1.5 Collision experiments

We collide 2 cells head-on with the same active velocity $v_A = 10^{-2}$, initially circularly shaped with their centers spaced a distance of 130 apart so that they do not overlap. The parameters $\xi = \infty$, $\Delta = 1$, and $\Delta t = 0.01$ were used. For normal cells ($\gamma_n = 1$), the cell RGB images showed almost no change after $t = 9000$ up to $t = 144000$. For soft cells ($\gamma_n = 0.35$), they showed almost no change after $t = 27000$ up to $t \approx 108000$, when the cells start to slide past each other due to numerical error. Changing ξ to 1500, these were still true but attained a very slightly different configuration (Fig. 3–6). The inactive velocity was 2.47×10^{-4}, modifying the active velocity by a few percent. For the $\xi = \infty$ simulation, the term for the inactive velocity calculated using $\xi = 1500$ (although not used in the simulation) evaluated to 2.56×10^{-4}.

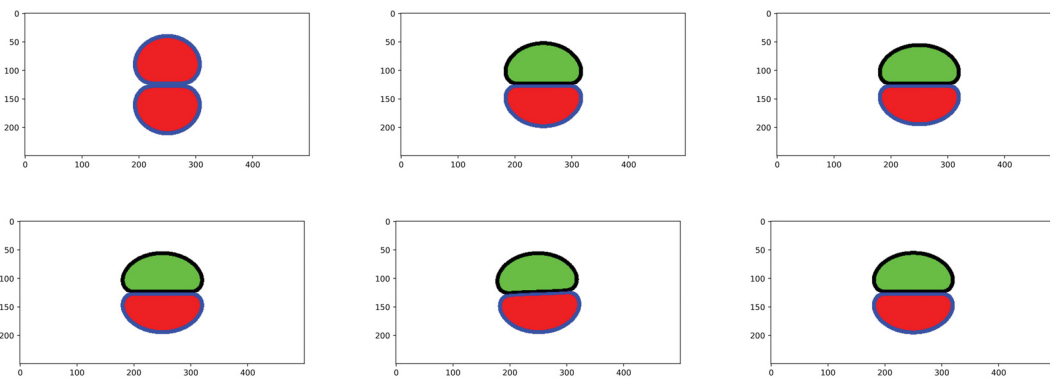
Figure 3–6: Collision experiments in cytoplasm-only model. Left to right, top to bottom: $\xi = \infty$ and $\gamma_n = 1$ at $t = 9000$, $\xi = \infty$ and $\gamma_n = 0.35$ at $t = 9000$, 18000, 27000, 117000, $\xi = 1500$ and $\gamma_n = 0.35$ at $t = 27000$. (Ignore the different colors of the cells.)

For the simulations with nucleus, we reduce the elasticity of the cytoplasm (discussed later). We tried using cells with $\gamma_n = 0.35 \times 0.3 = 0.105$. Again we only use $\xi = \infty$. Using the same numerical parameters, the cells never slowed down but rather, one reached around the other and eventually broke into two (not shown). This behavior persisted when using $\Delta t = 0.003$, except that the other cell instead broke apart (not shown). Using $\Delta = 0.5$ and $\Delta t = 0.01$, the problem was resolved. At $t = 72000$ the cells looked as if they were equilibrating (Fig. 3–7), but then slid past each other. These results suggest that in this case, the coarser resolution of the interface had problems handling either the reduced interface thickness or the regions of large curvature.
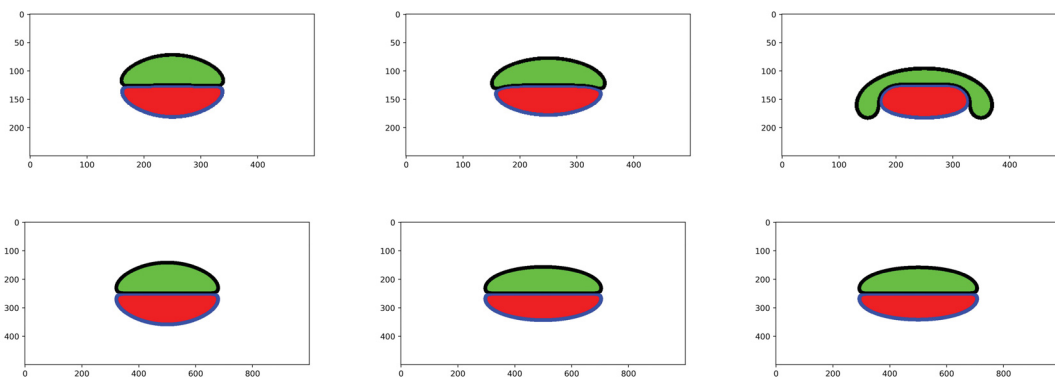
Figure 3–7: Collision experiments in cytoplasm-only model using $\gamma_n = 0.105$. Top row: $\Delta = 1$; (L–R) $t = 18000$, $t = 27000$, $t = 36000$ (initial positions slightly perturbed along the axis of the active velocities). Bottom row: $\Delta = 0.5$; (L–R) $t = 18000$, $t = 45000$, $t = 72000$. (Ignore the different colors of the cells.)

From the first ≈ 1000 time-units, the speed of isolated cells can be measured, traveling parallel to the mesh axes. The cells with $\gamma_n = 0.35$ and $\gamma_n = 0.105$ using $\Delta = 1$ moved at a speed of 9.84×10^{-3}, whereas the cells with $\gamma_n = 0.105$

using $\Delta = 0.5$ moved at a speed of 9.96×10^{-3} (also see next section, subsection "Confinement inside cytoplasm"). At the point when the cells' centers of mass were separated by $2R + \lambda$ (at $t \approx 1250$), their speeds were still 95% and 98% of v_A, respectively. Only after this point did the cells slow down more considerably. The center of mass was calculated as:

$$(\mathbf{r}_\mathrm{CM})_n = \frac{\int d^2\mathbf{r}\,\mathbf{r}\phi^2}{\int d^2\mathbf{r}\,\phi^2}. \tag{3.11}$$

3.2 Model with nucleus

The nucleus is modeled in the same way as the cytoplasm, using a field ψ_n for each cell having a free energy of the same form as the cytoplasm (Eq. 3.1):

$$\mathcal{F}_n^\nu = \gamma_n^\nu \int d^2\mathbf{r} \left(|\nabla\psi_n|^2 + \frac{30}{\lambda_\nu^2}\psi_n^2(1-\psi_n)^2 \right) + \frac{\mu_\nu}{\pi R_\nu^2} \left(\int d^2\mathbf{r}\,\psi_n^2 - \pi R_\nu^2 \right)^2, \tag{3.12}$$

where $R_\nu < R$, and parameters are analogous to Eq. 3.1.

To determine the appropriate value of γ_n^ν, we consider a nucleus with a given mechanical stiffness. Imagining a compression–relaxation experiment, the spring constant of the two-dimensional disk is independent of R_ν. If we suppose that the force per unit length is proportional to the interface velocity, then upon performing a scaling transformation, the interface speed should be preserved. This means that the time-scale should be concomitantly adjusted proportionally to R_ν (faster for smaller nuclei), assuming a given mobility (see Eq. 3.2). Performing both transformations, the result is $\gamma_n^\nu \propto 1/R_\nu$. The other parameters also become modified, but their values are relatively inconsequential as long as they are suitably chosen.

We choose $R_\nu = R/2$ for these simulations.

39

3.2.1   Confinement inside cytoplasm

We prevent the nucleus from intersecting the region outside the cytoplasm by including an interaction between cytoplasm and nucleus fields of each cell, given by the free energy density:

$$f_n^{c\nu} = \frac{60\kappa_\nu}{\lambda_\nu^2}\psi_n^2(c - \phi_n)^2, \tag{3.13}$$

where $c > 1$. The squaring of terms is similar to the cell–cell repulsion (Eq. 3.1), preserving for any value of ϕ_n a local minimum of free energy density at $\psi_n = 0$, and for $\psi_n \approx 1$ a local minimum in the range $1 < \phi_n < c$. The requirement $c > 1$ is used because $\phi_n > 1$ in the cytoplasmic interior; a choice of c that is too low will result in the nucleus being repelled by the cytoplasmic interior and preferring the cell edges (see next subsection, "Constraining the center of mass").

The total free energy functional $\mathcal{F}^{\text{tot}}$ is now the sum of the cytoplasmic (Eq. 3.1), nuclear, and interaction contributions. The nucleus is given the same mobility and active velocity as the cytoplasm. No inactive velocity is used. The evolution is given by:

$$\begin{aligned}
\frac{\partial \phi_n}{\partial t} + \mathbf{v}_{n,\text{A}}(t) \cdot \nabla \phi_n &= -\frac{1}{2}\frac{\delta \mathcal{F}^{\text{tot}}}{\delta \phi_n}, \\
\frac{\partial \psi_n}{\partial t} + \mathbf{v}_{n,\text{A}}(t) \cdot \nabla \psi_n &= -\frac{1}{2}\frac{\delta \mathcal{F}^{\text{tot}}}{\delta \psi_n}.
\end{aligned} \tag{3.14}$$

Initially, parameters used were $\gamma_n^\nu = \gamma_n/2$, $\lambda_\nu = \lambda/2$, $\mu_\nu = 2\mu$, $c = 1.05$ and $\kappa_\nu = 0.5$. This system was (almost) equilibrated similarly to the cytoplasm-only system, for ≈ 80000 time-units (Fig. 3–8). This provides the initial condition I mentioned in the previous section. Active simulations were performed using this initial condition and the same numerical parameters, $\Delta = 1$ and $\Delta t = 0.01$. Motion

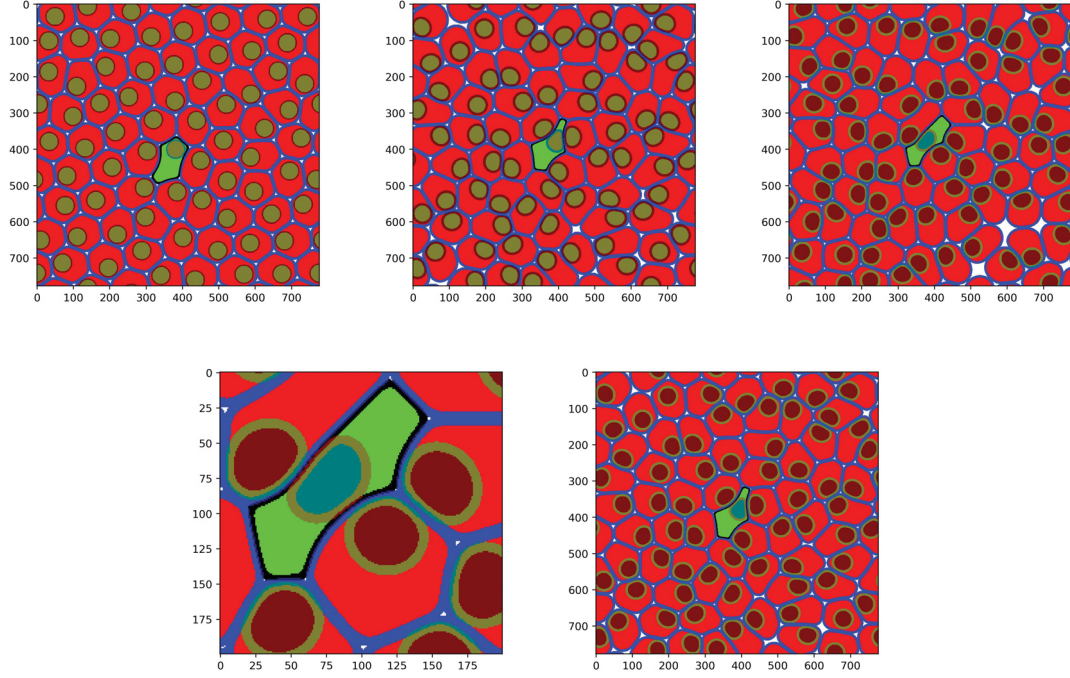Figure 3–8: Simulations with nucleus able to move freely within cytoplasm. Top row: (L–R) Equilibrated configuration (initial condition I); active simulation using $\Delta t = 0.001$ (indistinguishable from $\Delta t = 0.003$): $t = 13140$, 21420. Bottom row: (L–R) Close-up of top right image; active simulation using $\kappa = 50$, $\mu = 5$, $\kappa_\nu = 2.5$, $\mu_\nu = 10$, $\Delta t = 0.003$ at $t = 13140$.

of the nuclei within the cytoplasm was far too slow, presumably caused by a lack of spatial resolution.[1] The problem was resolved by using $\lambda_\nu = \lambda$. A small κ_ν was chosen to try to avoid unnecessary numerical cost as the cytoplasm dynamics now has repulsion from both neighboring cells and its nucleus. The cells "rammed" into each other in such a way as to geometrically emphasize the portions of cytoplasm boundary which were contacted by the nucleus (Fig. 3–8). This reflects the added contributions of elasticity on such portions of interface. However, this model does not take into account that a real nucleus does not move freely within the cytoplasm. Additionally, the nucleus looked like it came very close to the exterior of the cell (Fig. 3–8). So afterwards, we used $\kappa_\nu = 2$ instead.

This model was run on an Intel® Core™ i7-7500U processor using 2 threads (OMP_NUM_THREADS=2), taking ≈ 1 minute per 4000 iterations. There was a small amount of time spent ($\sim 10\%$) outputting RGB values to a CSV (comma-separated values) file. Visualization was done using Matplotlib [77].

When using the cytoplasm–nucleus interaction of Eq. 3.13, the cytoplasm field was inhomogeneous in the interior, taking a value of approximately c in the nuclear region and another outside. In later simulations, the cytoplasm field could reach

---

[1] This might be understood by considering the Fourier transform of the spatially discretized field, $\psi(x,t) = \sum_k a_k(t)e^{ikx}$ in 1D. Solving the equation $\partial\psi/\partial t - v \cdot \partial\psi/\partial x = 0$ using mesh spacing Δ and central differences gives $a_k(t) \propto \exp(-iv\sin(k\Delta)t/\Delta)$. This means that the kth Fourier component travels at a speed $v\sin(k\Delta)/(k\Delta)$. For $k\Delta \sim \pi/2$, this is significantly lower than v. (Actually, in the absence of a diffusive term, the Euler method is numerically unstable; a different one such as the Lax–Friedrichs method must be used.)

values higher than $c = 1.05$, causing the nucleus to stick to the cytoplasm boundary (see subsection "Higher density"). Or, we would need to use an even higher value of c, $\gtrsim 1.1$. This problem was fixed by changing the cytoplasm–nucleus interaction free energy density to:

$$f_n^{c\nu} = \kappa_\nu \psi_n^2 \cdot \Theta(1 - \phi_n) \cdot (1 - \phi_n)^2, \tag{3.15}$$

which also has continuous derivatives with respect to ϕ_n and ψ_n.

A free energy used for eutectic systems [78] was also experimented with, where the free energy density favors $\psi = 0$ or $\psi = 1$ inside the cytoplasm, and $\psi = 1/2$ outside:

$$\mathcal{F} = \mathcal{F}^c + \sum_n \gamma_n^\nu \int d^2\mathbf{r} \left\{ |\nabla \psi_n|^2 + \frac{30}{\lambda_\nu^2} \left[\left(\frac{1}{2} - \phi_n \right) \left(\psi_n - \frac{1}{2} \right)^2 + \left(\psi_n - \frac{1}{2} \right)^4 \right] \right\}$$
$$+ \sum_n \frac{\mu_\nu}{\pi R_\nu^2} \left(\int d^2\mathbf{r}\, \phi_n^2 \psi_n^2 - \pi R_\nu^2 \right)^2, \tag{3.16}$$

where $\mathcal{F}^c$ is the cytoplasmic free energy (Eq. 3.1). During equilibration, when the interfaces of the nucleus and cytoplasm contacted, they became roughly perpendicular to each other, which is not desired (Fig. 3–9). So this model was not used.

3.2.2 Constraining the center of mass

In reality, the nucleus stays close to the cell center. We first accomplished this by adding a spring term to the free energy for each cell:

$$\mathcal{F}_n^{spr} = \frac{k_n^{spr}}{2} |(\mathbf{r}_{CM})_n^c - (\mathbf{r}_{CM})_n^\nu|^2, \tag{3.17}$$
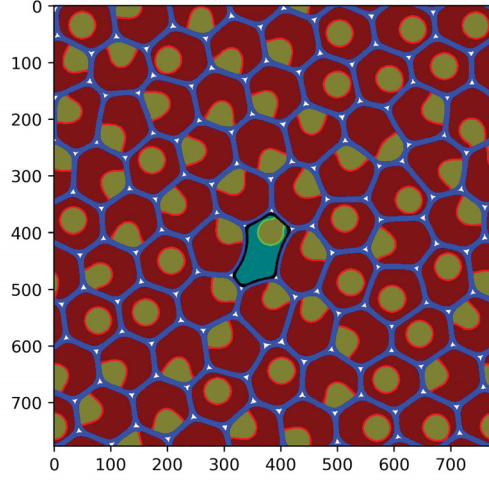
Figure 3–9: Equilibration using a eutectic-like free energy for the nucleus.

where $(\mathbf{r}_{\text{CM}})_n^{\text{c}}$ is the center of mass of the cytoplasm, defined as in Eq. 3.11, and $(\mathbf{r}_{\text{CM}})_n^{\nu}$ is the center of mass of the nucleus, defined analogously. While the area measures are constrained to be approximately constant, their explicit inclusion (via Eq. 3.11) is necessary for translational invariance of the functional derivatives, which are given by:

$$\frac{1}{2}\frac{\delta \mathcal{F}_n^{\text{spr}}}{\delta \phi_n(\mathbf{r}')} = k_n^{\text{spr}}\left((\mathbf{r}_{\text{CM}})_n^{\text{c}} - (\mathbf{r}_{\text{CM}})_n^{\nu}\right) \cdot \frac{\phi_n(\mathbf{r}')}{\int d^2\mathbf{r}\,\phi(\mathbf{r})^2}\left(\mathbf{r}' - (\mathbf{r}_{\text{CM}})_n^{\text{c}}\right), \qquad (3.18)$$

and analogously for ψ_n.

This model was equilibrated starting from initial condition I using $k_n^{\text{spr}} = 3$, providing initial condition II (Fig. 3–10) for the cytoplasm-only simulations. This was also used for the remaining simulations in this subsection. Subsequently, we used $\mu_\nu = 1$.

With the active velocity applied to both the cytoplasm and the nucleus, the cell "ramming" was still present because the cytoplasm tries to keep up with the nucleus (Fig. 3–10). However, removing the active velocity from the nucleus resulted in reduced speed of an isolated cell by 40%. Also, reducing k_n^{spr} to 0.3 seemed to require a smaller time-step, which was puzzling (Fig. 3–10). The distance between the centers of mass of the nucleus and the cytoplasm was $\lesssim 0.1$ for $k_n^{\text{spr}} = 3$ and $\lesssim 1$ for $k_n^{\text{spr}} = 0.3$. From this, it is estimated that the spring contribution to the difference in ϕ_n or ψ_n between time-steps is $\sim 0.002 \cdot \Delta t$, comparable to the differences themselves, whereas the contribution from cell–cell repulsion could be larger by an order of magnitude or more.

Active simulations were then performed for the 72-cell system using $k_n^{\text{spr}} = 3$ and including $\mathcal{F}_n^{\text{spr}}$ in the dynamics for ψ_n only (i.e., no $\delta \mathcal{F}_n^{\text{spr}} / \delta \phi_n$ term in the evolution of ϕ_n) while removing the active velocity of the nucleus ("asymmetric spring"). This more closely resembles the cytoplasm-only simulations. We ran another simulation, multiplying our original values of γ_n and γ_n^ν by 0.3 and 3, respectively, to make the nucleus 10 times stiffer than the cytoplasm. The cytoplasms' elasticities were decreased because in the experimental videos [9], they seemed to have little resistance to mechanical deformation aside from volume regulation. A large amount of white space can be seen when using the latter values, and the interface of the soft cell became very thin (Fig. 3–10). These values were used in the rest of the simulations in this section.
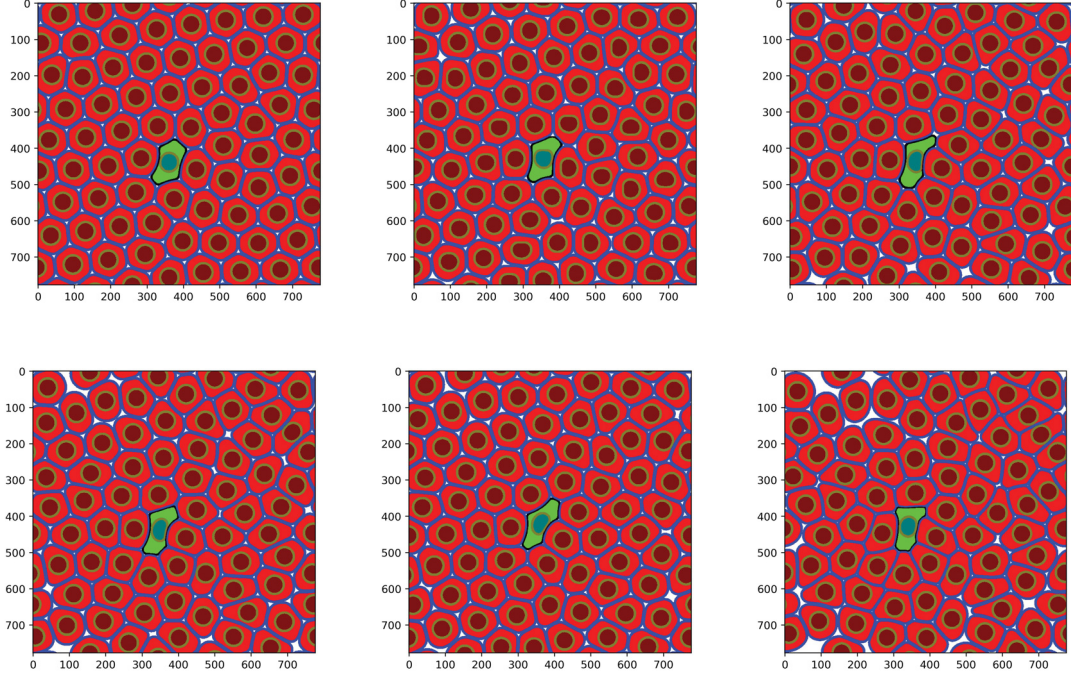
Figure 3–10: Simulations with the nucleus tied to the cell center by a spring-like term. Top row: (L–R) Equilibrated configuration using $k_n^{\mathrm{spr}} = 3$; active simulations using $k_n^{\mathrm{spr}} = 0.3$ at $t = 7740$ using $\Delta t = 0.01$; $\Delta t = 0.003$. Bottom row: Active simulations using $k_n^{\mathrm{spr}} = 3$ at $t = 9720$. (L–R) Dynamics given by Eq. 3.14; "asymmetric spring" simulations using original values of γ_n and γ_n^{ν}; "asymmetric spring" simulations using modified values of γ_n and γ_n^{ν} reflecting stiffer nucleus.

The spring term was subsequently discarded in favor of a simple relaxation to the cytoplasm center:

$$\mathbf{v}_n^\nu = \frac{(\mathbf{r}_{\mathrm{CM}})_n^{\mathrm{c}} - (\mathbf{r}_{\mathrm{CM}})_n^\nu}{\tau_\nu}, \tag{3.19}$$

where τ_ν is a time constant. This replaces $\mathbf{v}_{n,\mathrm{A}}$ in the evolution of ψ_n in Eq. 3.14.

3.2.3 Higher density

To model a more confluent layer, we increased the density to $\rho = 0.97$, using $L = 748$. The initial condition II (described in the previous subsection) was simply shrunk. Running active simulations, the cells also shrank in geometrical size, as the cytoplasmic field values could exceed 1.05 for a normal cell and 1.1 for a soft cell. The reduced area of the soft cell could be seen visually. This is not surprising because reduced values of γ_n amount to a weaker constraint on ϕ_n. As before, the interface of the soft cell became very thin (Fig. 3–11).

From short samples (≈ 5000 time-units), it is inferred that the distance between the centers of mass of the nucleus and the cytoplasm was $\lesssim 0.1$ for $\tau_\nu = 100$ and $\lesssim 0.03$ for $\tau_\nu = 1$. A lower density of $\rho = 0.95$ ($L = 756$) was also used, but cells were still shrunk and more white space formed (Fig. 3–11).

Also shown in Fig. 3–11 is an undesired situation where the nuclear field obeys dynamics using the cytoplasm–nucleus interaction given by Eq. 3.13, causing the nucleus to prefer the cytoplasm boundary.

The very thin interfaces call for a smaller mesh spacing, $\Delta \sim 1/3$–$1/2$. This comes with a substantial increase in numerical cost. Additionally, the geometrical area of a cell cannot be well controlled except by modifying the double-well potential

47
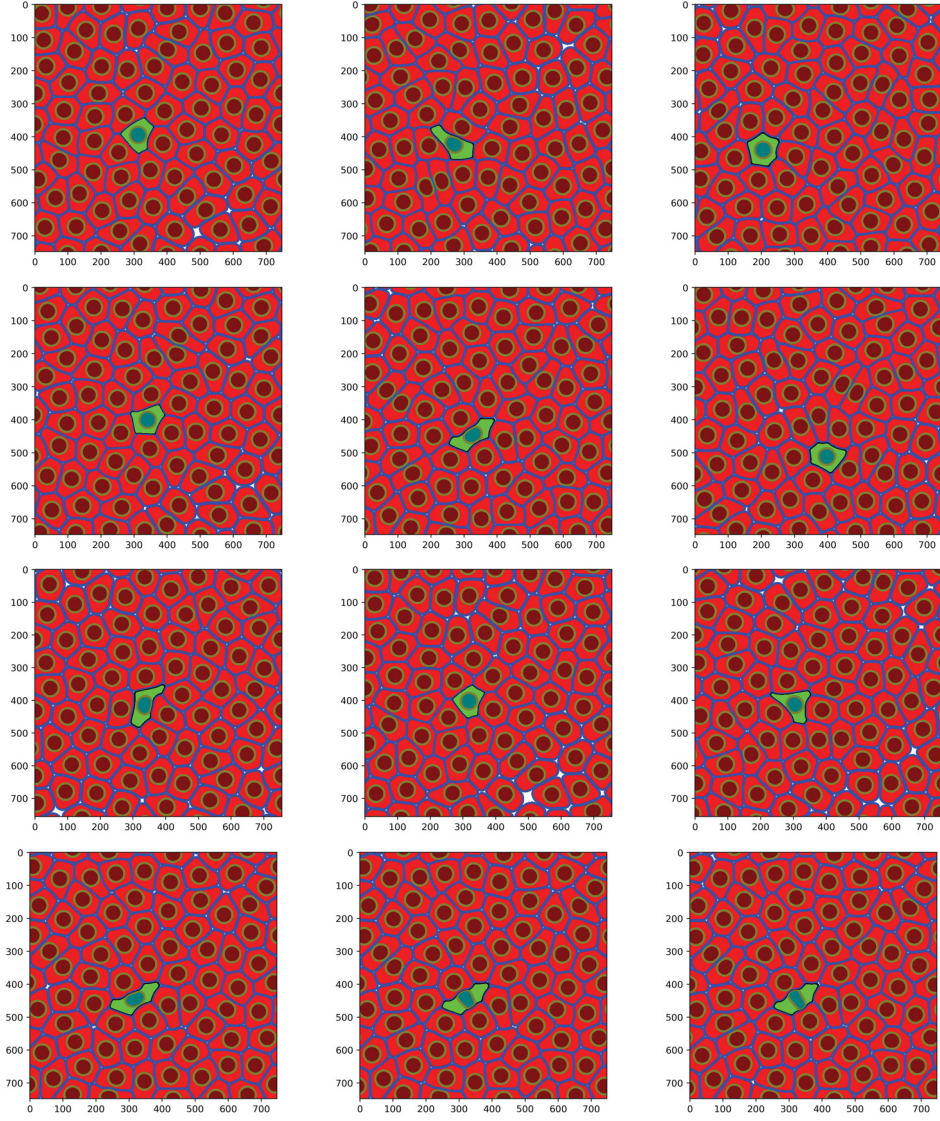
Figure 3–11: Phase-field simulations with nucleus using higher density. Top row: $\rho = 0.97$, $\kappa = 10$, $\kappa_\nu = 2$. (L–R) $t = 25020$, $t = 40500$, $t = 55980$. 2nd row: $\rho = 0.97$, $\kappa = 30$, $\kappa_\nu = 6$. (L–R) $t = 21600$, $t = 54000$, $t = 86400$. 3rd row: $\rho = 0.95$, $\kappa = 30$, $\kappa_\nu = 6$. (L–R) $t = 8100$, $t = 24300$, $t = 35100$. Bottom row: $\rho = 0.97$, $\kappa = 30$, $\kappa_\nu = 6$. (L–R) $t = 51300$, $t = 52200$, $t = 53100$ (see text in the subsection "Confinement inside cytoplasm").

to more strongly discourage field values above 1. Due to these considerations, we switched to a sharp-interface model.

CHAPTER 4
Sharp-Interface Model for Cell Monolayers

Sharp-interface models of the cytoplasm-only cells have been simulated, allowing more extensive studies of velocity distributions [72] and jamming transitions [79]. However, these numerical codes could crash when a soft cell was "pinched" in by two normal cells [72], apparently being less stable than the phase-field simulations. Our hope is that inclusion of a stiff nucleus, as well as more accurate numerics, might help to alleviate this problem. Two important new variables that arise in such an extension are the nuclear size and elasticity. The codes for this chapter were adapted from Yony Bresler's sharp-interface algorithm [72] written in C++ with OpenMP, using a small amount of help from the Boost library. Data processing and visualization codes were also adapted from Bresler's, using Python and MATLAB® (The MathWorks, Inc., Natick, MA, USA).

4.1   The models

The sharp-interface models are derived from the corresponding phase-field models in the limit $\lambda \ll 1/K$; however, this condition is often not true in the simulations. Additionally, no details of the interface profile are resolved during cell–cell contact. Thus, the two are not directly comparable. We assume the form Eq. 3.7, extended to allow for the curved closed interface C_n of a cell, regarded as the locus of points

where the field value is $1/2$:

$$\phi_n^*(\mathbf{r}) := \frac{1 - \tanh(\alpha d_n(\mathbf{r}))}{2} = \frac{1}{1 + e^{2\alpha d_n(\mathbf{r})}}, \tag{4.1}$$

where $d_n(\mathbf{r})$ is a signed distance to the boundary of cell n:

$$d_n(\mathbf{r}) := \left(\min_{\mathbf{P} \in C_n} |\mathbf{r} - \mathbf{P}| \right) \cdot \begin{cases} +1 & \text{if } \mathbf{r} \text{ is outside } C_n \\ -1 & \text{if } \mathbf{r} \text{ is inside } C_n. \end{cases} \tag{4.2}$$

We calculate the interface velocity starting from the model Eq. 3.1 and Eq. 3.2. The Landau–Ginzburg free energy density of each cell is already known to give a contribution $-\gamma_n K \hat{\mathbf{n}}$, where $\hat{\mathbf{n}}$ is the local outward normal to C_n and the sign of K is chosen compatibly. Locally, letting u be a signed distance to the interface with $u > 0$ being outside the cell, the interface velocity is calculated as:

$$\frac{\partial \mathbf{R}_n}{\partial t} = -\frac{(\partial \phi_n/\partial t)|_{u=0}}{(\partial \phi_n/\partial u)|_{u=0}} \hat{\mathbf{n}} + \mathbf{v}_A(t). \tag{4.3}$$

The numerator is calculated from the dynamics Eq. 3.2, while the denominator is calculated from the form Eq. 3.7, evaluating to $-\alpha/2$. For the numerator, the contribution from cell–cell repulsion in Eq. 3.4 is:

$$-\frac{60\kappa}{\lambda^2} \phi_n^*(\mathbf{r}) \sum_{m \neq n} \phi_m^*(\mathbf{r})^2 = -\frac{30\kappa}{\lambda^2} \sum_{m \neq n} \frac{1}{\left(1 + e^{2\alpha d_m(\mathbf{r})}\right)^2}. \tag{4.4}$$

The area term remains (we do not use an inactive velocity term). For the integral $\int d^2\mathbf{r}\, \phi_n^2$, there is a bulk area term, $A_n = \int d^2\mathbf{r}\, \Theta(-d_n(\mathbf{r})) = \oint_{C_n} dy\, x$. However, there is also a contribution from the perimeter P_n arising from integration of ϕ_n^2 as

opposed to ϕ_n, approximated as:

$$\oint_{C_n} ds \int_{-\infty}^{\infty} du \left[\frac{1}{(1 + e^{2\alpha u})^2} - \Theta(-u) \right] = -\frac{P_n}{2\alpha}. \tag{4.5}$$

Since this is of order λ^1 rather than λ^0, we consider it permissible to discard. For the parameters used here, it is a fraction $\geq 1/(\alpha R) = 5\%$ of the cell area, which could be considered very significant as we are working at densities $\sim 90\%$. However, the cell interactions are not directly analogous to the phase-field models. Also, the interface thinning in the phase-field model could reduce the magnitude of the above contribution.

Unlike the phase-field models, there is no interface thinning upon cell–cell contact. This makes repulsion stronger, and a large κ becomes unsuitable. For a head-on collision experiment, the distance between the inner interfaces in equilibrium can be estimated from Eq. 4.4, matching to the active velocity magnitude, v_{A}. This ignores contribution from the area deviation, which gives an outward push in the cytoplasm-only model and renders the obtained distances an upper bound. Equilibrium in general is given by $\hat{\mathbf{n}} \cdot \partial \mathbf{R}_n / \partial t = 0$. For $\lambda = 7$, this calculation gives a distance of 5.12 for $\kappa = 10$ and 4.67 for $\kappa = 5$. For the model with nucleus, there is additionally the (outward) pushing from the nucleus. We mainly work with $\kappa = 5$ (see section "Simulations").

The model with nucleus is derived using Eq. 3.15 and Eq. 3.19. The cytoplasm–nucleus interaction gives the same form as Eq. 4.4, except for a sign change for the cytoplasm and a coefficient change $\kappa \to \kappa_\nu$, assuming $\lambda_\nu = \lambda$. The center of

mass can be calculated as:

$$\mathbf{r}_{\text{CM}} = \left(\frac{1}{A} \oint dy \, \frac{x^2}{2}, -\frac{1}{A} \oint dx \, \frac{y^2}{2} \right), \tag{4.6}$$

where A is the area. The center-of-mass relaxation of the nucleus is again given by Eq. 3.19.

4.2 Numerical algorithm

The evolution equation for the sharp-interface model has basic similarities with the phase-field model, involving a first derivative in time and a second derivative in space. Explicit time-stepping is also used here, since an implicit method would require iterative solving. The essential new step is calculation of the repulsion term, involving finding the closest point on a neighboring cell interface. The same kinds of issues are present, such as numerical stability and spatial resolution. As before, we work in a square simulation box of length L with $N_{\text{cells}} = 72$ and again work at $\rho = 0.9$. The active velocities are determined identically to in the previous chapter, except using $\Delta t_r = 0.1$.

4.2.1 Spatial discretization

The sharp interface is described by a fixed number N_{pts} (for the nucleus, N_{pts}^{ν}) of points $\mathbf{R}_n[i] = (X_n[i], Y_n[i])$ distributed along the curve, going counterclockwise. The distances Δs between adjacent points is regulated as described later; it should be kept as "$\Delta s \ll \lambda$". Imagining the curve to be parameterized by a variable ζ with points at equally spaced parameter values ζ_i, a 5-point stencil is used for the finite

difference method to estimate derivatives and curvature:

$$\mathbf{R}'_n(\zeta_i) = \frac{-\mathbf{R}_n[i+2] + 8\mathbf{R}_n[i+1] - 8\mathbf{R}_n[i-1] + \mathbf{R}_n[i-2]}{12\Delta\zeta} + O((\Delta s)^4)$$

$$\hat{\mathbf{n}}_n(\zeta_i) = \frac{(Y'_n(\zeta_i), -X'_n(\zeta_i))}{|\mathbf{R}'_n(\zeta_i)|}$$

$$\mathbf{R}''_n(\zeta_i) = \frac{-\mathbf{R}_n[i+2] + 16\mathbf{R}_n[i+1] - 30\mathbf{R}_n[i] + 16\mathbf{R}_n[i-1] - \mathbf{R}_n[i-2]}{12(\Delta\zeta)^2} + O((\Delta s)^4)$$

$$K_n(\zeta_i) = \frac{X'_n(\zeta_i)Y''_n(\zeta_i) - X''_n(\zeta_i)Y'_n(\zeta_i)}{|\mathbf{R}'_n(\zeta_i)|^3},$$

$$(4.7)$$

where $\Delta\zeta := \zeta_{i+1} - \zeta_i$ and periodic boundary conditions are used ($i \rightarrow i \pm N_{\text{pts}}$). The factor $\Delta\zeta$ cancels out in all relevant quantities. The above expressions correspond to those of a quartic Lagrange polynomial in ζ interpolated between the 5 points. (Notice that the $\mathbf{R}_n[i \pm 2]$ contribute oppositely than the $\mathbf{R}_n[i \pm 1]$.) The estimates might be made more accurate by using a $\Delta\zeta$ dependent on the distances between adjacent points, and using finite differences for a non-uniform grid (see Appendix). We try to keep the spacing between the points relatively uniform at each instant of time (discussed later). For the normal vector specifically, it may make more sense to use a 3-point stencil (codes provided by Yony Bresler) as it better reflects the local geometry. The periodic boundary conditions are implemented similarly to in the previous chapter, except that an extra 2 points are replicated at each end of the (now 1D) array.

54

The area, perimeter, and center of mass can be most simply approximated as those of the polygon joining the sampled points:

$$A_n = \oint dy\, x = \sum_i \frac{X_n[i+1] + X_n[i]}{2} (Y_n[i+1] - Y_n[i]) + O((\Delta s)^2)$$

$$\oint dy\, \frac{x^2}{2} = \sum_i \frac{(X_n[i])^2 + X_n[i]X_n[i+1] + (X_n[i+1])^2}{6} (Y_n[i+1] - Y_n[i]) + O((\Delta s)^2)$$

$$P_n = \sum_i |\mathbf{R}_n[i+1] - \mathbf{R}_n[i]| + O((\Delta s)^2).$$

$$(4.8)$$

Similarly to the phase-field models, the fractional area deviation was observed to be $\lesssim 1.5\%$ in magnitude. The perimeter serves as a measure of cell deformation, and it could range from $2\pi R$ to occasionally exceeding $1.5 \times 2\pi R$. Formulas accurate to $O((\Delta s)^4)$ can be obtained by using a piecewise cubic interpolation in conjunction with Gaussian quadrature (or direct integration for the area and center of mass). Such an approximation could be a Lagrange interpolating polynomial, or a cubic Hermite spline where the derivatives $\mathbf{R}'_n[i]$ (i.e., tangents) are specified at each sampled point [80]:

$$\mathbf{R}_n(\zeta) = (1 - \bar{\zeta})\mathbf{R}_n[i] + \bar{\zeta}\mathbf{R}_n[i+1] + (\zeta - \zeta_i)(1 - \bar{\zeta})^2 (\mathbf{R}'_n[i] - \Delta\mathbf{R}_n[i])$$
$$- \bar{\zeta}^2(\zeta_{i+1} - \zeta)(\mathbf{R}'_n[i+1] - \Delta\mathbf{R}_n[i]) + O((\Delta s)^4), \quad \zeta \in [\zeta_i, \zeta_{i+1}],$$

$$(4.9)$$

where $\bar\zeta := (\zeta - \zeta_i)/\Delta\zeta$ and $\Delta\mathbf{R}_n[i] := \mathbf{R}_n[i+1] - \mathbf{R}_n[i]$. The $O((\Delta s)^4)$ error can be bounded above by a term proportional to $(\zeta - \zeta_i)^2(\zeta_{i+1} - \zeta)^2$ [80].[1] While this might not be of much interest in calculating areas and perimeters, as the area is not strictly conserved, it is of more interest in calculating repulsion forces (see following subsections).

The interface dynamics does not preserve distances between points, so the points have to be redistributed along the curve occasionally. We do not allow adjacent points to be too far apart or too close. The former leads to inaccuracy while the latter leads to numerical instability (see subsection "Time-stepping scheme"). Later on, we also do not allow the distance between adjacent points to vary too quickly along the interface. We follow Ref. [72] in utilizing a centripetal Catmull–Rom spline whenever one of the stipulated conditions are violated after a time-step update. The Catmull–Rom splines are a family of cubic Hermite splines governed by the specifications that each segment of the curve passes through its two endpoints, and the tangents (i.e., derivatives) are given by:

$$\widetilde{\mathbf{R}}_n'(\widetilde{\zeta}_i) = \frac{\mathbf{R}_n[i+1] - \mathbf{R}_n[i-1]}{\widetilde{\zeta}_{i+1} - \widetilde{\zeta}_{i-1}}, \tag{4.10}$$

where $\widetilde{\mathbf{R}}_n(\widetilde{\zeta})$ is the spline curve, interpolated at the parameter values $\widetilde{\zeta}_i$. (The above is different from the second-order finite difference formula for a non-uniform spacing;

---

[1] The error being $O((\Delta s)^4)$ depends on the derivatives $\mathbf{R}_n'[i]$ being $O((\Delta s)^4)$ accurate. However, if some lower-order approximation to $\mathbf{R}_n'[i]$ (e.g. as used for Catmull–Rom splines) is considered as being the true tangent, then the error can still be considered $O((\Delta s)^4)$.

however, the differences $\Delta\widetilde{\zeta}_i := \widetilde{\zeta}_{i+1} - \widetilde{\zeta}_i$ become more uniform for a high density of sampled points, so the expression is $O((\Delta s)^2)$ accurate.) The different Catmull–Rom splines are distinguished by the choice of $\widetilde{\zeta}_i$; the centripetal Catmull–Rom spline is given by $\widetilde{\zeta}_{i+1} - \widetilde{\zeta}_i \propto |\mathbf{R}_n[i+1] - \mathbf{R}_n[i]|^{1/2}$. It has some special properties among Catmull–Rom splines, one of them being that it does not form cusps or self-intersections [81] (though admittedly, this is of limited relevance when we are sampling the curve well). It also discourages overshooting. After the spline is fit, new points are chosen equally spaced in the parameter $\widetilde{\zeta}$. The Catmull–Rom splines have continuous first derivatives but discontinuous second derivatives. To obtain a cubic interpolant with continuous second derivatives requires solving a cyclic tridiagonal system of N_{pts} linear equations for each space dimension [80]. A discontinuous second derivative is not a concern because it is not used to calculate curvature, and accurate estimation of curvature would benefit more from a higher density of sampled points. Also, it can be mentioned that using a cubic Lagrange interpolating polynomial in each segment produces a tangent of accuracy $O((\Delta s)^3)$, generally discontinuous at the sampled points. The Catmull–Rom splines have continuous tangents of accuracy $O((\Delta s)^2)$. Of course, these asymptotics are meaningful only for sufficiently small Δs, which may be smaller than what we use.

To control the distance between adjacent sampled points, we originally used the approach of Ref. [72] where a fixed minimum and maximum are imposed. Later, because of the longer perimeters that could be occasionally obtained and also to try to achieve a more even spacing, we subsequently required for $\Delta s_n[i] := |\mathbf{R}_n[i+1] - \mathbf{R}_n[i]|$ that (1) $0.8 \times 2\pi R/N_{\text{pts}} \leq \Delta s_n[i] \leq (0.8 P_n + 0.4 \times 2\pi R)/N_{\text{pts}}$, which allows P_n up to

$4\pi R$, and (2) $(\Delta s_n[i+1])^2 - (\Delta s_n[i])^2 \leq 0.1 \times (2\pi R/N_{\text{pts}})^2$, which imposes changes in consecutive distances below $\approx 5\%$. As N_{pts} increases, the condition (1) remains unchanged, while condition (2) becomes less frequent. Using this scheme with $N_{\text{pts}} = 320$ and using the Euler method with $\Delta t = 0.1$, around 15000 Catmull–Rom splines were fit in a period of 54000 time-units.

4.2.2 Repulsion calculation

The repulsion calculations are the most time-consuming part ($\sim 90\%$ of run-time currently) of the evolution update. The other parts (computation of unit normal vector, curvature, area, and possibly perimeter) take $\sim N_{\text{pts}}$ operations, while computing the closest point on a neighboring cell interface to every point on a given cell interface naïvely takes $\sim N_{\text{pts}}^2$ operations. Computation can be sped up using parallelization and vectorization [74], but the specific cell geometries in this system can also be taken advantage of.

We used N_{pts} ranging from 150 to over 1000. Three basic categories of precision can be defined when calculating the minimum distance from a discretized curve to a point: (1) taking the closest sampled point, (2) taking the closest point on the polygon defined by the sampled points (the "interface polygon"), and (3) using a higher-order approximation. While the first two theoretically both have accuracy $O((\Delta s)^2)$, we consider the first method to be too crude, since the "ideal" condition $\Delta s \ll \lambda$ does not really hold well.

Typically, most of the cell interface is relatively flat or has relatively low curvature, similar to that of a relaxed cell. However, the regions of higher and more rapidly varying curvature (can reach $K \gtrsim 5/R$ for soft cell in cytoplasm-only model,

and $\sim 10/R$ in model with nucleus) occur in places where three or more cells join, which are important sites of cell rearrangement and dynamics in a confluent monolayer. While it seems sensible to try to use a variable distance between points based on the local curvature, we did not yet attempt such an approach.

The cell interfaces were never observed to overlap, except during equilibration, which was carried out in the same way as in the previous chapter. The value $N_{\text{pts}} = 150$ was used and a Catmull–Rom spline was fit if a larger N_{pts} was used for the active simulations. Detection of a negative signed minimum distance (Eq. 4.2) from a point $\mathbf{P}$ to a cell m was done by looking at the sign of the dot product $\hat{\mathbf{n}}_m(\zeta)\cdot(\mathbf{P}-\mathbf{R}_m(\zeta))$, where the (outward) unit normal is evaluated at the deemed closest point $\mathbf{R}_m(\zeta)$ of cell m's interface. Regarding the interface as a differentiable curve, the two arguments of the dot product are then parallel or antiparallel. This method sufficed for our purposes, not requiring a more elaborate general point-in-polygon algorithm.

Active cytoplasm-only simulations could reach a minimum distance between cells of ~ 3.3 for $\kappa = 5$ and ~ 4 for $\kappa = 10$. The first step in the repulsion calculation is to specify a maximum cutoff minimum-distance d_{cut}, beyond which the contribution is neglected. Initially, a value of 1.4λ was used, but comparing with 2.5λ and 3λ indicated that this might be too low. Also, the interface profiles still overlap considerably (Fig. 4–1). The choice of 2.5λ appeared to be sufficient and gave visually indistinguishable results as 3λ in one simulation for 54000 time-units (although this is not long; see next section, "Simulations"). While this may seem excessively far when compared to the repulsion strength at smaller distances, the cells travel more slowly in such situations.

With $\kappa = 5$, the repulsion force (contribution to $\partial \mathbf{R}_n / \partial t$) $f^{\mathrm{rep}}(d)$ from a single cell is 10^{-6} at a distance $d = 1.5\lambda$ and 2×10^{-11} at $d = 2.5\lambda$. Using a time-step $\Delta t = 0.1$, the latter one approaches the maximum precision for a double-precision floating-point number ≥ 512, which is $\approx 10^{-13}$. The former one may appear negligible at first sight, contributing an error of 10^{-2} in a reorientation time $\tau_r = 10^4$; however, this contributes back to an error in $f^{\mathrm{rep}}(d)$ through an inaccuracy in d. The maximum value M of $|(f^{\mathrm{rep}})'(d)|$ attained is $M \approx 0.1$, corresponding to $d \approx 3.3$. This leads to an upper bound for inaccuracy in position of 5 space-units in a time τ_r, though this is certainly an overestimate. In fact, this error propagation procedure can continue indefinitely, in the limit approaching an upper bound of $\Delta v \cdot (e^{Mt} - 1)/M$ after a time t for an inaccuracy Δv in the velocity; a similar bound exists for the global truncation error of a numerically solved differential equation [80, 82] (see next subsection "Time-stepping scheme"). However, it becomes a gross overestimate on relevant time-scales.
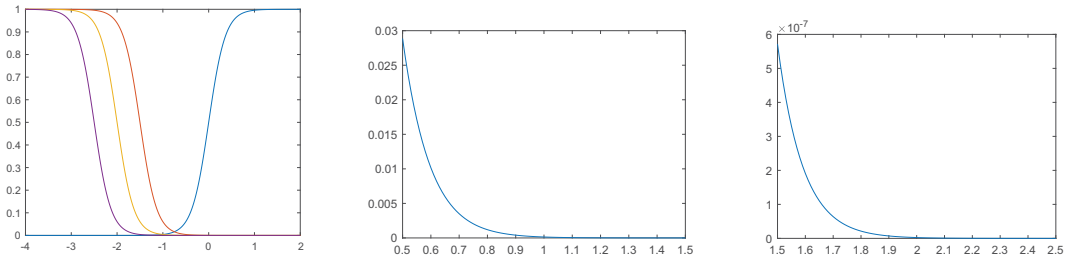
Figure 4–1: Interface overlap and repulsion. The horizontal axis is distance in units of λ. Left: Antiparallel opposing equilibrium interfaces (Eq. 3.7) at center-to-center distances of 1.5λ, 2λ, and 2.5λ. Middle and right: Repulsion force f^{rep} for $\kappa = 5$.

For each cell n, its center of mass $(\mathbf{r}_{\mathrm{CM}})_n$ is tracked and its farthest distance $r_n^{\max}$ from the interface is maintained. At each time-step, the possible neighbors considered are restricted to the cells $m \neq n$ such that $|(\mathbf{r}_{\mathrm{CM}})_n - (\mathbf{r}_{\mathrm{CM}})_m| \leq r_n^{\max} + r_m^{\max} + 4\lambda$ (the

choice 4λ is somewhat arbitrary, as long as it is $> d_{\mathrm{cut}}$), where periodic boundary conditions of the simulation box are taken into account. Then, for each point $\mathbf{R}_n[i]$ on the interface of cell n, the neighbors considered for interaction are further restricted to $|\mathbf{R}_n[i] - (\mathbf{r}_{\mathrm{CM}})_m| \leq r_m^{\mathrm{max}} + 4\lambda$. Here, we consider finding the closest point on cell m from each point $\mathbf{R}_n[i]$ separately; a more efficient algorithm might exploit spatial proximity to use results of nearby points.

The simplest algorithm to calculate the minimum distance to a cell m from a point $\mathbf{R}$ is obtained by regarding the interface C_m as a polygon, analytically calculating the minimum (squared) distance to each of its line segments, and taking the smallest value. Alternatively, the closest sampled point could be calculated, and the line segments on both sides of it examined. However, by taking advantage of the constrained spacing between points and the "smooth" cell geometries, the search can be made more efficient by first considering a "coarsened" version of the cell, evenly subsampling N_{coarse} points (where N_{coarse} divides N_{pts}) and locating the closest point on the resulting polygon. Then, one examines the $N_{\mathrm{pts}}/N_{\mathrm{coarse}}$ line segments corresponding to the coarsened line segment found[2], as well as the $N_{\mathrm{pts}}/N_{\mathrm{coarse}}$ line segments on each side of it. It is not sufficient to calculate the closest of the N_{coarse} points because of cases where opposite sides of a cell come close to each other. (This was in fact done until incorrect results in "pinched" configurations were noticed.) For a suitable choice of N_{coarse} (we use $N_{\mathrm{coarse}} \geq 20$ for cytoplasm-only simulations

---

[2] If the closest point on the coarsened polygon happens to be a vertex, it may be associated with the coarsened line segment on either side of it.

and $N_{\text{coarse}} \geq 40$ for the cytoplasm when including nucleus), the algorithm is correct for the cell shapes encountered because only relatively small distances are measured, and also because any inwardly curved regions are gentle ($|K| \lesssim 1/R$) and contacted by a neighbor throughout its length (Fig. 4–2). We tend to choose N_{pts}, N_{coarse}, and $N_{\text{pts}}/N_{\text{coarse}}$ values that are multiples of 4 to take advantage of Intel® AVX2 vectorization.[3]

The described computation results in jump discontinuities of size $O(\Delta s)$ in $\partial d_m(\mathbf{P})/\partial\mathbf{P}$. A way to try to remove this is to use a cubic interpolation such as Eq. 4.9 and compute the shortest distance to it, using the Newton–Raphson method initialized with $\bar{\zeta}$ corresponding to the result for the line segment. Sometimes, the resulting closest point will lie outside the original segment considered; if the same (non-piecewise) cubic-polynomial curve is still used for coding ease or computational efficiency, errors and jump discontinuities in $d_m(\mathbf{P})$ of size $O((\Delta s)^4)$ could be incurred (although it might actually be higher-order because the polygonal approximation improves as well).[4] Using this scheme with Lagrange interpolation for simulations with

---

[3] However, using the `#pragma ivdep` directive, the Intel compiler chooses 2-way vectorization over 4-way, apparently because of a lower estimated cost than when using the `#pragma simd vectorlength(4)` directive.

[4] In the case of a cubic Hermite polynomial, this would amount to an extrapolation in addition to an interpolation. This might argue for preferring a Lagrange interpolation for smaller N_{pts}. Additionally, the derivative $\mathbf{R}'_m(\zeta)$ suffers an $O((\Delta s)^3)$ error in Hermite interpolation as with Lagrange interpolation. Ultimately, the Hermite interpolation with tangents specified from a 5-point stencil computation may itself be somewhat undesirable because with the large range of influence, it may not well reflect the local geometry.
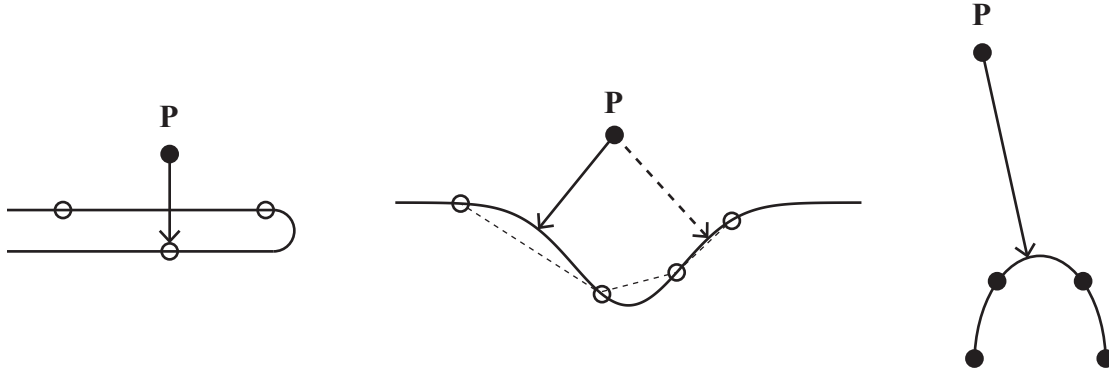
Figure 4–2: Schematics of distance calculations. The solid curved line represents a cell interface. The point **P** is a query point on a neighboring cell interface, at a distance less than 3λ away. The other filled dots are sampled points of the first cell interface. The hollow dots are the subsampled points. The solid line with an arrow shows the closest point. Left: Example showing the inadequacy of taking the closest subsampled point. This also applies to cells in a "pinched" configuration. Middle: An impossible situation possibly resulting in an incorrect result. The dashed lines conencting the hollow dots are the coarsened line segments making up the coarsened polygon, and the dashed line with an arrow shows the closest point found by the algorithm. The spacing between the subsampled points is very non-uniform, but distortions to the curve could be considered instead. Right: Example showing a closest point lying outside the line segment associated with the closest point on the interface polygon; here, the line segment is taken to be the one connecting the two leftmost points shown.

63

$N_{\text{pts}} = 150$, the observed solutions for $\bar{\zeta}$ deviated by $< 7\%$ from the original segment (which is $\bar{\zeta} \in [0, 1]$) when using $d_{\text{cut}} = 1.4\lambda$. Meanwhile, using $d_{\text{cut}} = 2.5\lambda$ and Lagrange interpolation for $N_{\text{pts}} = 320$ and $N_{\text{pts}} = 512$, the deviations were below and up to $\approx 50\%$ and $\approx 30\%$, respectively. The largest deviations occur when calculating a large distance to a tightly curved section (Fig. 4–2). In a single time-step, this scheme would incur an error of $O((\Delta s)^4 \Delta t)$ in the estimated solution, which is the same order as that due to the approximation of $d_m(\mathbf{P})$ itself.

The quadratically converging Newton–Raphson method [80] is applied to the equation $(\mathbf{R}_m(\zeta) - \mathbf{P}) \cdot \mathbf{R}'_m(\zeta) = 0$. The distance behaves quadratically in the vicinity of the closest point, so we do not need extreme precision in calculating the solution for ζ. For a line segment of length Δs a distance d away, the squared distance is $d^2 + (\Delta s)^2(\bar{\zeta} - \bar{\zeta}^*)^2$ where $\bar{\zeta}^*$ is the $\bar{\zeta}$-value at the closest point, so that the distance has a quadratic coefficient of $(\Delta s)^2/(2d)$ in its vicinity. For a cubic interpolation, we expect a result of the same order. Using $N_{\text{pts}} = 150$ with Lagrange interpolation with $d_{\text{cut}} = 1.4\lambda$, in most cases, 4 iterations are sufficient for consecutive $\bar{\zeta}$ values to differ by less than 10^{-8}, and another two iterations are used. After 6 iterations, in most cases, the consecutive differences in $\bar{\zeta}$ range from 10^{-15}–10^{-14}, showing that we are indeed approaching the solution. This is approaching maximum precision as the minimum resolution attainable for a double-precision floating-point number ≥ 0.5 is $\approx 10^{-16}$. Combined with the quadratic behavior of the distance with $\bar{\zeta}$, this means that the computation is for all purposes exact and the last 2 iterations are unnecessary. However, the above could be violated in extreme situations when a

64

cell "tail" exhibits very high curvature as it narrowly escapes or avoids a "pinched" configuration (see next section, "Simulations").

Using $N_{pts} = 160$ and Lagrange interpolation with a larger $d_{cut} = 3\lambda$ for 6×54000 time-units, there were ≈ 600 instances in which 5 iterations were necessary to achieve consecutive differences in $\bar{\zeta}$ of under 10^{-8}. After 6 iterations, they did not exceed 10^{-13}. In this case, the last 2 iterations are still unnecessary.

Using $N_{pts} = 512$ with Lagrange interpolation and $d_{cut} = 2.5\lambda$ for 12×54000 time-units, the consecutive differences in $\bar{\zeta}$ did not exceed 3×10^{-12} after 4 iterations, with a few exceptions above 10^{-10}, some of which required 5 iterations for the consecutive differences to fall below 10^{-8}. The latter cases involved an unusually high curvature of the soft cell, recovering from a "pinched" configuration.

Using $N_{pts} = 480$ with Lagrange interpolation and $d_{cut} = 3\lambda$ for 3×54000 time-units (very short; see next section, "Simulations"), starting from a different initial configuration, the consecutive differences in $\bar{\zeta}$ again did not exceed 3×10^{-12} after 4 iterations, and the maximum difference between the calculated minimum distance and the polygonal result was ≈ 0.02.

For $N_{coarse} = 40$ and $N_{pts} = 160$ for the cytoplasm-only simulations, the coarsening procedure reduced computation time by a factor of 2 in non-vectorized code using the polygonal scheme. Running on an Intel® Core™ i7-7500U processor using 2 threads, about 200 time-steps of the Euler method per second could be calculated. Keeping $N_{coarse} = 40$ fixed and increasing N_{pts} to 960, the computation time increased roughly proportional to N_{pts}, taking 7 times longer. For $N_{pts} = 160$, the cubic interpolation reduced the speed to ≈ 150 time-steps of the Euler method per

second. There are significantly more floating-point operations but its effect seems to be partially masked by costs associated with the memory hierarchy [74]. For relatively small N_{pts}, the increased computational cost could be considered merited because it does not well hold that $\Delta s \ll d$ (d being the distance between cells), and the repulsion force at the small distances encountered varies strongly. While the interfaces are rather flat at the smallest distances, they are not exactly far apart in highly curved regions.

The more complicated schemes (i.e., than polygonal) would be relevant only when N_{pts} is relatively low, e.g. $\lesssim 320$ for the cytoplasm. Ultimately, we use larger values for N_{pts} (see next section, "Simulations") and the cubic-polynomial fits become somewhat irrelevant.

When incorporating a nucleus into the sharp-interface model, errors from strict correctness could arise when using the coarsening procedure in situations illustrated schematically in Fig. 4–2, but these can arise only for relatively far closest-distances and the deviation from the correct value is expected to be small.

We might expect the sharp-interface model to require less spatial resolution than the phase-field model since the interface profile is not modeled.

When running on Compute Canada machines, the runtime did not speed up anywhere near proportionally to the specified number of threads, for a reason that we did not identify yet. The results of the repulsion calculation are stored in a separate array and are completed before executing the time-step update, so this should not be a source of cache thrashing between cores.

66

4.2.3 Time-stepping scheme

For a given N_{pts}, the equations for $\partial \mathbf{R}_n / \partial t$ can be considered as a system of ordinary differential equations (ODEs) in $2 N_{\text{pts}} N_{\text{cells}}$ real variables. By the Picard–Lindelöf theorem, an ODE $d\mathbf{x}/dt = \mathbf{g}(\mathbf{x}, t)$ with an initial value condition has a unique solution when $\mathbf{g}(\mathbf{x}, t)$ is continuous in t and uniformly Lipschitz continuous in $\mathbf{x}$ in some suitably defined regions of "space" and time, i.e.:

$$|\mathbf{g}(\mathbf{x}_1, t) - \mathbf{g}(\mathbf{x}_2, t)| \leq c|\mathbf{x}_1 - \mathbf{x}_2|, \quad \mathbf{x}_1, \mathbf{x}_2 \in \Omega, \ t \in [T_1, T_2], \tag{4.11}$$

where c is a constant [80]. (The active velocity is discontinuous in time but we are not concerned with this.) The above condition holds whenever $\mathbf{g}(\mathbf{x}, t)$ is continuously differentiable in $\Omega \times [T_1, T_2]$. The initial value problem can be numerically solved using the Euler method with a time-step Δt, obtaining a local truncation error of $O((\Delta t)^2)$ with each time-step if $\mathbf{x}(t)$ is twice continuously differentiable [82]. Each jump discontinuity in $d\mathbf{x}/dt$ incurs an error $O(\Delta t)$, proportional to the size of the discontinuity. If $\mathbf{x}$ is twice differentiable with its second derivative bounded above in magnitude by M_2, then the local truncation error is at most $M_2(\Delta t)^2/2$ [82]. If $d^2\mathbf{x}/dt^2$ has only jump discontinuities, the same holds considering it as a limit of twice continuously differentiable functions. These would arise from jump discontinuities in the first derivatives of $\mathbf{g}(\mathbf{x}, t)$. If $\mathbf{x}(t)$ is three times continuously differentiable except with possible jump discontinuities, the midpoint method can be used to obtain an $O((\Delta t)^3)$ local truncation error [82]:

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \Delta t \cdot \mathbf{g}\left(\mathbf{x}(t) + \frac{\Delta t}{2}\mathbf{g}(\mathbf{x}, t), t + \frac{\Delta t}{2}\right). \tag{4.12}$$

The global truncation error after a time T is one order less accurate in Δt than the local truncation error, because $T/\Delta t$ time-steps need to be taken. It can be bounded above by an exponential function in T [80, 82]. The accumulated error results not only from the local truncation error but also the error in calculating $\mathbf{g}(\mathbf{x}, t)$ for the next time-step.

Catmull–Rom splines are fit only after the entire time-step update is completed. The need to fit such splines renders the system more appropriate for one-step methods (e.g. of the Runge–Kutta family) than multistep methods (e.g. Adams–Bashforth).

Local truncation errors can be estimated by halving (or equivalently, doubling) the time-step, strictly applicable when the solution is sufficiently differentiable. Applying this to the Euler method for this system anyway, we estimate after 2 time-steps an error of at most 10^{-7} for $\Delta t = 0.005$ with $N_{\text{pts}} = 320$, and at most 2.5×10^{-7} for $\Delta t = 0.01$ with $N_{\text{pts}} = 640$, using the polygonal method and running from $t = 0$ to 6×54000.

For numerical stability with respect to the curvature term, we consider an approximately flat interface with sampled points equally spaced by Δx. Such an interface can be described by the equation

$$y(x) = A \sin(qx), \tag{4.13}$$

where $A \ll \Delta x$. To leading order in A, the numerical curvature calculated using the 5-point stencil is:

$$K^{[5]}(x) = \frac{-y(x - 2\Delta x) + 16y(x - \Delta x) - 30y(x) + 16y(x + \Delta x) - y(x + 2\Delta x)}{12(\Delta x)^2}$$

$$= \frac{A}{(\Delta x)^2} \left[-\frac{1}{6} \cos(2q\Delta x) + \frac{8}{3} \cos(q\Delta x) - \frac{5}{2} \right] \sin(qx)$$

$$=: \Gamma^{[5]}(q) \cdot y(x).$$

(4.14)

Using the Euler method, after each time-step, the amplitude A is multiplied by the factor $1 + \gamma_n \Delta t \cdot \Gamma^{[5]}(q)$; the condition for numerical stability is then $\gamma_n \Delta t < 3(\Delta x)^2/8$. Using the midpoint method, the amplitude is modified over time as:

$$A(t + \Delta t) = A(t) \left\{ 1 + \gamma_n \Delta t \cdot \Gamma^{[5]}(q) \left[1 + \gamma_n \frac{\Delta t}{2} \cdot \Gamma^{[5]}(q) \right] \right\}, \qquad (4.15)$$

which leads to the same stability condition. Using a 3-point stencil, $K^{[3]}(x) = \Gamma^{[3]}(q) \cdot y(x)$ with $\Gamma^{[3]}(q) = [2\cos(q\Delta x) - 2]/(\Delta x)^2$, which leads to $\gamma_n \Delta t < (\Delta x)^2/2$.

Numerical stability analysis can also be applied to the repulsion and area terms. A cell with an active velocity v_{A} colliding into a fixed, flat perpendicular interface will equilibrate at a distance d from it, given by:

$$f^{\mathrm{rep}}(d) := \frac{30\kappa}{\alpha\lambda^2} \left(1 + e^{2\alpha d} \right)^{-2} = v_{\mathrm{A}}. \qquad (4.16)$$

The equation of motion in the vicinity of d can be linearized to $d(\delta x)/dt = (f^{\mathrm{rep}})'(d)\delta x$, where $\delta x(t) := x(t) - d$ with $x(t)$ the (time-dependent) interface position of the active cell measured from the fixed interface. Using either the Euler method or the

69

midpoint rule, this gives a stability condition $\Delta t < 2/|(f^{\mathrm{rep}})'(d)|$. For κ ranging from 1 to 10, the calculated $|(f^{\mathrm{rep}})'(d)| \sim 0.15$.

A similar calculation can be done with the area term, competing with curvature. Considering dilational perturbations of a perfect circle, the equilibrium radius R_{eq} satisfies:

$$-\frac{\gamma_n}{R_{\mathrm{eq}}} - \frac{2\mu}{\alpha}\left(\frac{R_{\mathrm{eq}}^2}{R^2} - 1\right) = 0 \tag{4.17}$$

(neglecting the perimeter term). The derivative of the left-hand side with respect to R_{eq}, evaluated at the equilibrium point, is:

$$\frac{\gamma_n}{R_{\mathrm{eq}}^2} - \frac{4\mu}{\alpha}\frac{R_{\mathrm{eq}}}{R^2} \quad = \quad \frac{\gamma_n}{R_{\mathrm{eq}}^2} - \frac{2}{R_{\mathrm{eq}}}\left(\frac{2\mu}{\alpha} - \frac{\gamma_n}{R_{\mathrm{eq}}}\right). \tag{4.18}$$

The only significant term is $4\mu/(\alpha R_{\mathrm{eq}}) \approx 4\mu/(\alpha R) = 0.21$ for a choice of $\mu = 1$. So neither repulsion for area conservation are of concern with respect to numerical stability.

We use double precision throughout to minimize concern about intrinsic precision limits of the computer, and techniques such as compensated summation [82] are not used. Single precision may be insufficient for accurate simulations of both the phase-field and sharp-interface models.

4.3 Simulations

For the simulation videos, normally the configurations were saved every 180 time-units[5] , and typically watched at 60–120 frames per second. The active velocities

---

[5] Every 360 time-units works fine as well.

were also drawn for each cell, at their respective centers of mass. The velocity magnitudes (blue) and perimeters (red) of the cells were tracked, as well as the number of "neighbors" which we define as being within a distance of λ or 2λ (both definitions are used). These properties for the soft cell are plotted together with the simulation animation, with the perimeter transformed such that values of $2\pi R$ and $4\pi R$ are plotted as 0 and 0.02, respectively.

The number of neighbors defined as being within a distance of 2λ appeared to better capture neighbor exchanges, whereas the definition of being within a distance of λ could have spurious fluctuations not reflective of actual neighbor exchanges. Because of the non-confluence of the monolayer (at $\rho = 0.9$), the restriction that the cells must have six nearest neighbors on average (see e.g. Ref. [60]) does not hold. It is interesting that the nominal density $\rho = 0.9$ is very close to that of a hexagonal packing of perfect circles. As the cells can become highly deformed and elongated, counting using a Voronoi construction is not always suitable.

4.3.1 Model and numerical parameters and "pinching" problem

Equilibration was done for $\approx 2 \times 10^5$ time-units, obtaining the configuration shown in Fig. 4–3. "Pinched" configurations are also shown in Fig. 4–3. The close approach of the opposite sides of the interface is enabled by stretching in the perpendicular direction. It is not entirely certain what causes this to sometimes wreck the simulation (values diverging), but it may start from the crossing of the opposing sides of the interface of a single cell. This firstly causes the described coarsening algorithm to fail (underestimating the repulsion force), but also the "repulsion" from the neighbor on the "wrong" side (i.e., the "right" side of a cell interface, now at

the left, could be "repulsed" by a neighboring cell to the left of it) could reinforce the crossing. Even if such "backward" interactions are forbidden, the crossings are a problem in and of themselves because as the cell relaxes from the configuration, its tail may form a knot (observed in some tests).

Interface crossings could be prevented by adding a self-repulsion term, e.g. of the same form as f^{rep}. The easiest way this could be done would be to draw a line through the point in question, locally perpendicular to the interface, and find the closest opposing point. This would be amenable to the coarsening procedure described. However, a way more similar to the repulsion between cells would be to find an opposing point of local minimum distance (if it exists), which might be more involved. We did not try either of these additions to the model.

Also, we do not exclude repulsion force on a point at the "front" of a cell interface exerted by a cell positioned "behind" it. This is gives continuity in space and time in situations such as when a cell's "tail" retracts.

For the cytoplasm-only model, using $d_{\mathrm{cut}} = 1.4\lambda$ and $N_{\mathrm{pts}} = 160$, differences seemed to remain when decreasing from $\Delta t = 0.1$ to $\Delta t = 0.0075$ using the Euler method and the polygonal scheme. Then using $d_{\mathrm{cut}} = 3\lambda$, experiments were done using Δt values from 0.005 to 0.02, as well as 0.05 and 0.1, using the Euler method, while N_{pts} was varied among 160, 320, 640, and 960. A simulation time of $54000 \times 6 \approx 3 \times 10^5$ seemed sufficient to observe differences. Convergence did not seem to be reached yet. For $N_{\mathrm{pts}} = 320$, using even the smallest time-steps ($\Delta t = 0.005$) still resulted in highly "pinched" configurations of the soft cell within 3×10^5 time-units.
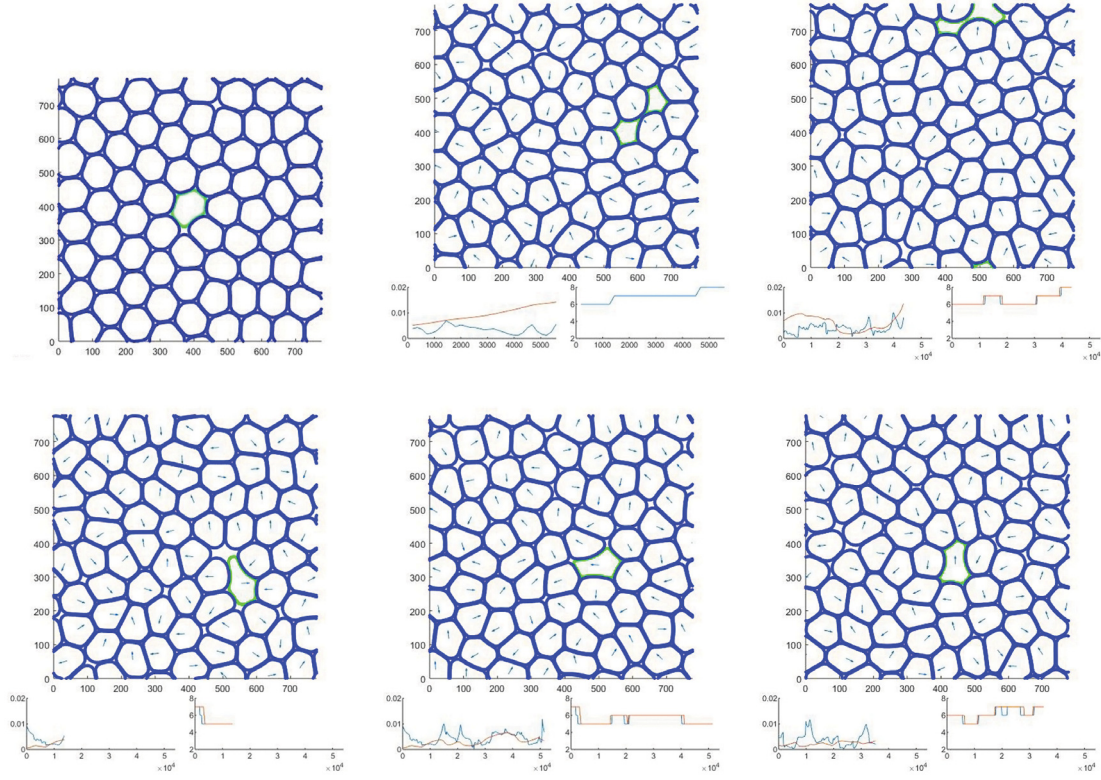
Figure 4–3: Samples of cytoplasm-only sharp-interface model. Time axes are offset. Top left: Equilibrated configuration. Top middle: Pinched configuration using $\Delta t = 0.1$ with Euler method and $N_{\mathrm{pts}} = 150$ with polygonal method and $d_{\mathrm{cut}} = 1.4\lambda$; simulation does not survive. Top right: Pinched configuration using $\Delta t = 0.09$ with midpoint method and $N_{\mathrm{pts}} = 512$ with Lagrange interpolation and $d_{\mathrm{cut}} = 2.5\lambda$; simulation survives with soft cell forming a "tail" that retracts rapidly. Bottom row: Simulation using $\Delta t = 0.04$ with midpoint method and $N_{\mathrm{pts}} = 400$ with Lagrange interpolation and $d_{\mathrm{cut}} = 2.5\lambda$ at $t = 553500$, $t = 591300$, and $t = 629100$.

73

The effect of a "low" N_{pts} could be confounded with insufficiently accurate time-stepping.

Higher-order Runge–Kutta methods were tried [82] (still in the polygonal approximation), although they suppose higher-order differentiability. Using the fourth-order Runge–Kutta 3/8-rule ("RK4-3/8") with $N_{\text{pts}} = 160$, using $\Delta t = 0.05$ crashed within $\approx 2 \times 10^5$ time-units due to pinching, becoming extremely elongated, while using $\Delta t = 0.1$ did not have a problem up to 5.4×10^5 time-units. Using the Euler method with $N_{\text{pts}} = 150$ and $\Delta t = 0.1$ also got a crash due to pinching at around 6.5×10^5 time-units. This shows how disordered the system is, and motivates a more careful examination of numerical parameters/schemes.

Using a Δt too large gives an inaccurate effect of the forces on the interface points, while a too small N_{pts} sacrifices representation of the cell interface and curvature estimation. These become apparent when different values are compared (Fig. 4–4). Very large perimeters ($\gtrsim 3.5\pi R$) can be a sign of numerical inaccuracy. Obtaining numerics accurate to within subtle but (reasonably) humanly detectable differences in the dynamical patterns can add a factor of $\gtrsim 10$–15 to the computational cost. This has not been previously done for this model; also, we are not modeling large amounts of simulation time as in Refs. [72, 79].

Some time may be needed for the simulations to advance to a point where more "difficult" configurations are reached. Using the midpoint method and Lagrange interpolation, the choices of $\Delta t = 0.03$ with $N_{\text{pts}} = 320$ vs. $\Delta t = 0.09$ with $N_{\text{pts}} = 512$ (in the latter case, restricting the distance between adjacent points to be $\geq 0.9 \times 2\pi R/N_{\text{pts}}$ for numerical stability) gave similar results for the first 2.5×10^5
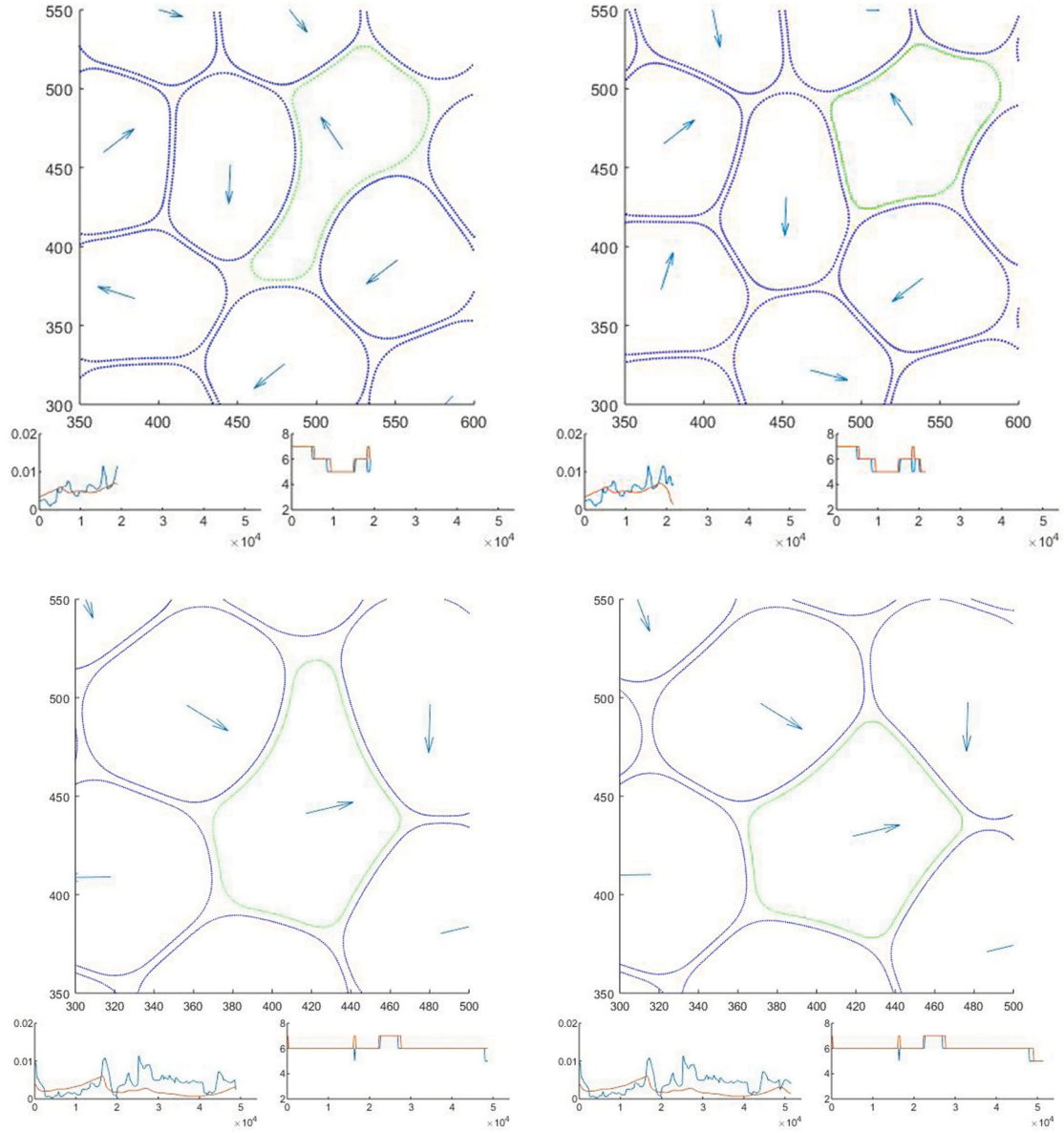
Figure 4–4: Close-ups of cytoplasm-only sharp-interface model. Time axes are offset. Here, $d_{\text{cut}} = 3\lambda$ was used. Top row: $N_{\text{pts}} = 160$ with $\Delta t = 0.1$ using Euler method around $t = 2.4 \times 10^5$. Bottom row: $N_{\text{pts}} = 320$ with $\Delta t = 0.1$ using Euler method around $t = 2.1 \times 10^5$.

75

time-units. Comparing the latter choice with the combination of $\Delta t = 0.06$ and $N_{\text{pts}} = 640$, the results were similar for the first 1.5×10^5 time-units. The latter of those showed improved rearrangements while cell representation seemed to remain equally as good. From a run done with $\Delta t = 0.04$ and $N_{\text{pts}} = 400$, the effect of a smaller time-step and finer cell representation can be seen. Pinching problems are possibly reduced; however, the small system size ($L = 777$) may be a limiting factor. The simulation may suffer from finite-size effects that might somewhat disrupt the continuous smooth flow expected in a liquid-like state. Each cell experiences a force from its neighbors that has a contribution propagated from itself, possibly hindering its motion. An amount of anisotropy could be noticed as well.

For $N_{\text{pts}} = 480$ and $N_{\text{coarse}} = 40$ with polygonal method, using 8 cores and 16 cores (Intel® Xeon® E5-2683 v4) could achieve ≈ 80 and ≈ 180 iterations per second of the midpoint method, respectively.

Pinching tests were performed using 3 cells, one soft cell in the middle with normal cells on both sides of it compressing it, with velocities of magnitude v_{A} along the x-axis. The x-coordinates of their centers were initially placed 100 units apart, while the y-coordinates of the normal cells were the same and that of the soft cell differed by 0.1 units. The soft cell narrowly escapes the crowding after $\approx 2.5 \times 10^4$ time-units and survives, reaching a very high curvature in the process. Simulations using the phase-field model gave very similar results (Fig. 4–5). While there are no issues in this particular test, the effect of more cells pushing on the soft cell, whether directly in contact or indirectly, can be greater. This renders pinching problems a probabilistic phenomenon, though possibly rare as far as one might be concerned.

Although it may be unlikely for two normal cells to impinge on a soft cell directly for such a duration while neighboring cells are positioned to allow extension of the soft cell in the perpendicular direction, the issue might ultimately be unavoidable in the model.

Methods to alleviate this problem could be to use higher elasticities or a "natural curvature" as in Ref. [72]. Other ways of more strongly penalizing curvature would be to use driving forces such as $-\gamma_n K \left[1 + \beta(\lambda K)^2\right] \hat{\mathbf{n}}$ (β a dimensionless coefficient) or $-\gamma_n K \exp((\lambda K)^2)\hat{\mathbf{n}}$. A theory of 2D fluid elastic membranes leads to a K^3 curvature force when area is conserved [83]. Such reduction of curvature was also thought to possibly facilitate numerics. However, using the exponential form of the driving force gave a very different result for the pinching test (Fig. 4–5), as well as similar differences for the 72-cell system. In this work, we sought to imitate the phase-field model and opted for the originally derived sharp-interface model. Besides, the interface thickness of real cells is much smaller than what would correspond to our value of λ, so the latter is not a suitable length scale to invoke. However, the curvature-driven forces in real cells may differ from the $-\gamma_n K\hat{\mathbf{n}}$ form.

Lowering the value of κ from 10 (used in the phase-field model) to 5 would also seem to reduce the chance of attaining a "pinched" configuration by allowing cells to come closer together and thereby allowing more "room"; the cells appear to be avoiding each other a little too much when using $\kappa = 10$ for the 72-cell simulation, which may encourage opposite sides of a single cell interface to come closer. Additionally, it would be expected to more easily obtain a numerically accurate simulation for lower κ. So we also use $\kappa_\nu = 5$ in the simulations with nucleus.
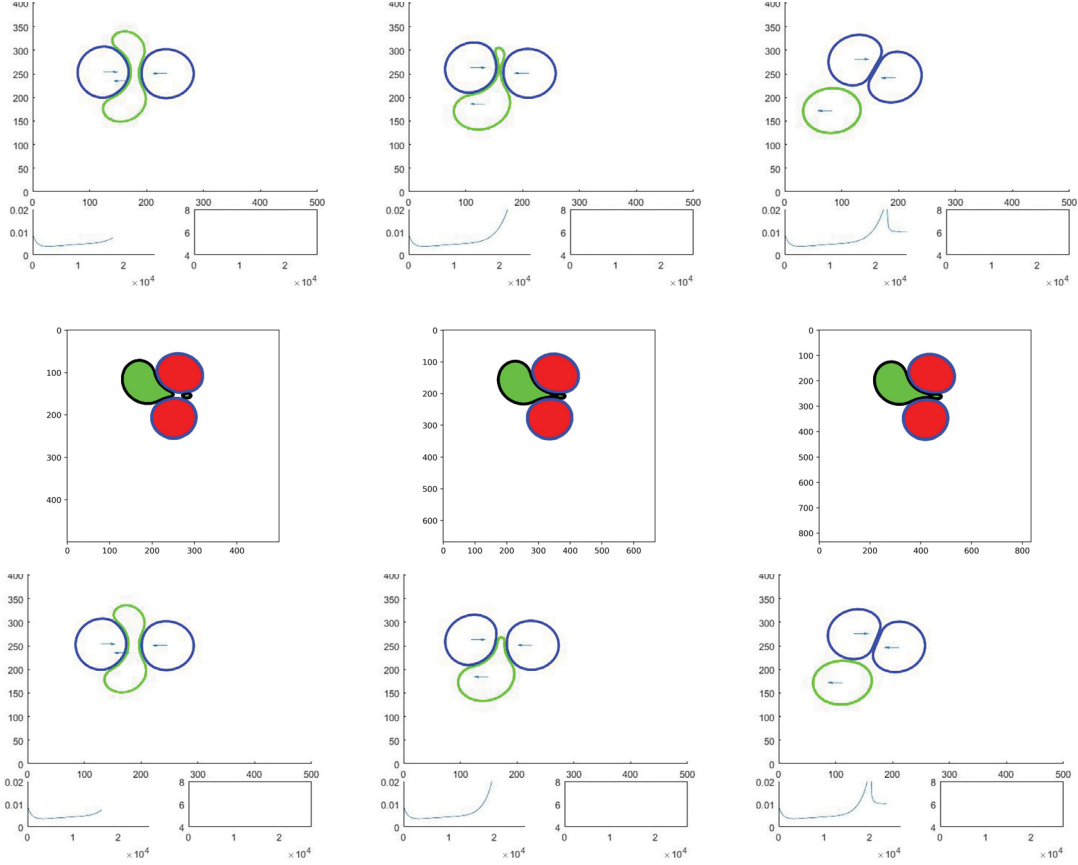
Figure 4–5: "Pinching" tests for cytoplasm-only model. Top row: Sharp-interface model with $\Delta t = 0.01$, $N_{\text{pts}} = 150$, and distances between adjacent points restricted to between $0.8 \times 2\pi R/N_{\text{pts}}$ and $1.75 \times 2\pi R/N_{\text{pts}}$. The result is essentially the same with restrictions on the adjacent differences of distances. Horizontal and vertical axes are not quite equal, and the plotted interface thickness is less than $\lambda = 7$. Middle row: Snapshots from phase-field model. (Left) $\Delta t = 0.006$ and $\Delta = 1$, showing "pinching-off"; (Middle) $\Delta t = 0.006$ and $\Delta = 0.75$, with essentially identical results for $\Delta t = 0.002$; (Right) $\Delta t = 0.0008$ and $\Delta = 0.6$. Bottom row: Sharp-interface model with same numerical and plotting schemes as top row, but with exponential form of curvature penalization.

For the center-of-mass of the nucleus, we choose a value $\tau_\nu \gg \Delta t$ while retaining proximity of the nucleus and cytoplasm centers of mass. If the cell moves at a speed $2v_A$, we can expect the centers of mass to be placed a distance $\lesssim 2v_A\tau_\nu$ apart. We choose $\tau_\nu = 5$.

For the model with nucleus, we initially used $R_\nu = R/2$. Pinching tests were also run with cytoplasm and nucleus γ values (coefficient of K in the curvature-driven force) multiplied from the cytoplasm-only values by $1/2$ and 2, respectively, and also by $1/3$ and 3, respectively. As discussed in the previous chapter ("Phase-Field Model for Cell Monolayers"), the former case would correspond to a factor of 8 between the cytoplasm and nucleus elasticities. The former survived while the latter did not (Fig. 4–6).
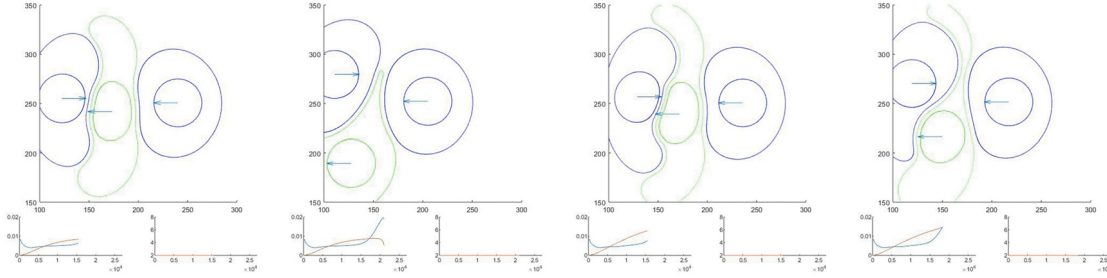
Figure 4–6: "Pinching" tests for model with nucleus. Used $R_\nu = R/2$, $\Delta t = 0.05$ with Euler method, and $N_{\text{pts}} = 320$ and $N_{\text{pts}}^\nu = 160$. Interface thickness is not plotted. Left two: $\gamma_0 = 0.35 \times 2$ and $\gamma_0^\nu = 0.35/2$ for soft cell, and $\gamma_n = 2$ and $\gamma_n^\nu = 1/2$ for normal cells. Right two: $\gamma_0 = 0.35 \times 3$ and $\gamma_0^\nu = 0.35/3$ for soft cell, and $\gamma_n = 3$ and $\gamma_n^\nu = 1/3$ for normal cells. Same result when using $\Delta t = 0.03$ with midpoint method and $N_{\text{pts}} = 480$ and $N_{\text{pts}}^\nu = 240$.

We chose similar values for the time-step and spacing between adjacent points as in the cytoplasm-only simulations. Using 16 cores (Intel® Xeon® E5-2683 v4) with

79

$N_{\text{pts}} = 400$ and $N_{\text{pts}}^{\nu} = 200$ could achieve ≈ 150 iterations of the midpoint method per second, while using 24 cores (Intel® Xeon® Platinum 8160F) with $N_{\text{pts}} = 480$ and $N_{\text{pts}}^{\nu} = 240$ could achieve ≈ 100 iterations of the midpoint method per second.

We also used a larger nucleus with $R_{\nu} = 5R/8$ and $N_{\text{pts}}^{\nu}/N_{\text{pts}} = 7/10$. The nucleus cannot be made too large as the interface thickness is applied to both the cytoplasm and nucleus.

4.3.2 Observations

In the 72-cell simulations with nucleus using $\gamma_n = 0.5$ for the cytoplasm of normal cells (stiffer than the original soft cell), the cytoplasm boundary was seen to still give significant mechanical support to the cell, in contrast with the experimental videos where they seem quite floppy. However, reducing the elasticities further does not seem to be viable while maintaining the elasticity ratio between "soft" and "normal" cells (see above), without including self-repulsion. Indeed, in the experiments, opposing sides of a cell membrane do appear to come very close or even touch.

The diffusion coefficient in 2D is $D = \langle [\Delta \mathbf{r}(t)]^2 \rangle / (4t)$, where $\Delta \mathbf{r}(t)$ is the displacement of the relevant particle (here we take the centers of mass) over a time period t, for sufficiently long times t, and $\langle \cdots \rangle$ denotes an ensemble average. This quantity is non-self-averaging, i.e., in the limit $t \to \infty$, a single sample does not suffice. For example, in a random walk, $\Delta \mathbf{r}(t)$ is expected to be Gaussian with zero mean, and for such a random variable X, a single measurement of X^2 has $\text{Var}(X^2) = 2(\mathbb{E}[X^2])^2$ ($\mathbb{E}$ being expectation). Thus the signal-to-noise ratio remains finite.

An equivalent way of measuring the self-diffusion coefficient is by integrating the velocity autocorrelation function (VACF): $D = (1/2) \int_0^\infty dt' \langle \mathbf{v}(0) \cdot \mathbf{v}(t') \rangle$, where $\langle \mathbf{v}(0) \cdot \mathbf{v}(t') \rangle$ is the velocity autocorrelation function with time delay t'. If a suitable cutoff is used for the integral, this expression now gives a decent estimate for D when a single run is used. For a discrete-time system, the derivation is as follows (taking the time discretization to be 1 unit):

$$[\Delta \mathbf{r}(t+1)]^2 - [\Delta \mathbf{r}(t)]^2 = [\Delta \mathbf{r}(t) + \mathbf{v}(t)]^2 - [\Delta \mathbf{r}(t)]^2$$

$$= \mathbf{v}(t)^2 + 2\mathbf{v}(t) \cdot \sum_{t'=0}^{t-1} \mathbf{v}(t'),$$

(4.19)

so in this case the correct expression for D is actually:

$$D = \frac{\langle \mathbf{v}(0)^2 \rangle}{4} + \frac{1}{2} \sum_{t'=1}^{\infty} \langle \mathbf{v}(0) \cdot \mathbf{v}(t') \rangle.$$

(4.20)

We use a discrete-time treatment here, taking the velocity to be the displacement between saved frames divided by the elapsed time ($\gg \Delta t$).

In this system, we can try to take advantage of the form of the motor driving by separating the VACF $\langle \mathbf{v}(0) \cdot \mathbf{v}(t') \rangle$ into an expectation conditional on $\mathbf{v}_A(0) = \mathbf{v}_A(t')$ (with probability e^{-t'/τ_r} assuming $t' > 0$) or $\mathbf{v}_A(0) \neq \mathbf{v}_A(t')$, i.e., whether the cell's driving has changed direction within the time interval concerned (Fig. 4–7). While the latter case would be less correlated, the cell has stochastic forces from neighbors acting on it which are correlated in time because of their driving or possibly large-scale flows. This results in some correlation of the cell's motion across changes in propulsion direction and some de-correlation within a constant-propulsion interval.

81

The combination of the two results in the appearance of an exponential decay faster than e^{-t'/τ_r} for $t' \lesssim 2\tau_r$ when the VACF is plotted logarithmically (Fig. 4–7).

The system has no inertia, so collisions do not result in rebounding. Thus we might expect both contributions to the VACF to be positive, which is plausible from the data to within statistical error. We might also expect the "unequal" contribution to exponentially decay in time; however, it is known from computer simulation studies of fluids that algebraic decays of the VACF can occur due to hydrodynamic flows whereas an exponential decay would be expected from microscopic interactions [84]. (On the other hand, those systems are not inertia-less.) In any case, our data is very imprecise (Table 4–1) and after a time delay of $3\tau_r$, the VACF is quite small. Additionally, some finite-size effects may be present.

Table 4–1: Contributions to diffusion coefficients for cytoplasm-only simulations. Simulation was divided into 5 disjoint parts, each 6×54000 time-units long (rows). "Equal" and "unequal" contributions are labeled "same" and "chgd" respectively, and normal cells are labeled "nrml". Calculated from VACFs integrated for time delays only up to $\approx 3\tau_r$.

	$D_{\text{soft}}^{\text{same}}$	$D_{\text{soft}}^{\text{chgd}}$	$D_{\text{nrml}}^{\text{same}}$	$D_{\text{nrml}}^{\text{chgd}}$
1	4.3×10^{-2}	-1.9×10^{-2}	5.8×10^{-2}	1.4×10^{-2}
2	4.6×10^{-2}	0.7×10^{-2}	5.0×10^{-2}	0.4×10^{-2}
3	5.4×10^{-2}	-1.1×10^{-2}	5.5×10^{-2}	1.1×10^{-2}
4	4.5×10^{-2}	2.0×10^{-2}	5.3×10^{-2}	0.8×10^{-2}
5	3.2×10^{-2}	3.1×10^{-2}	4.9×10^{-2}	0.3×10^{-2}

While there is some resemblance of the motion of the soft cell in the cytoplasm-only simulation to that in the experimental videos [9], occasionally "bursting" from an elongated "pinched" configuration into a neighboring location, a large amount of its motion occurs similarly to the normal cells. The significantly greater deformations
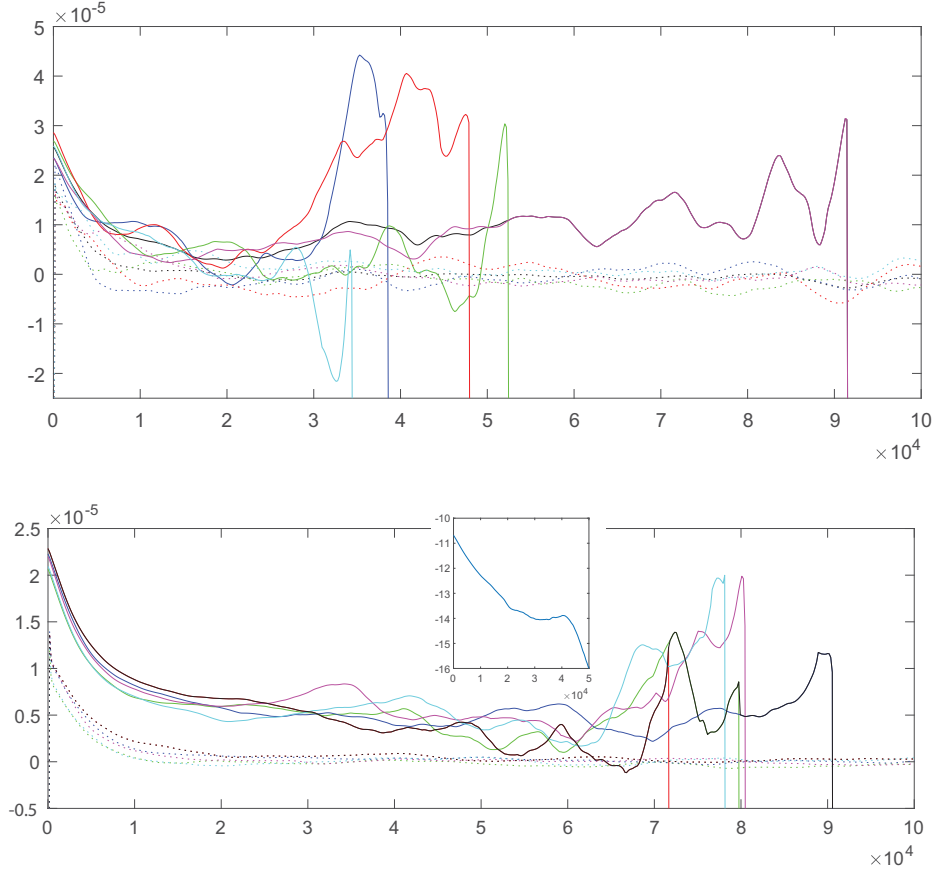
Figure 4–7: Velocity autocorrelation functions for cytoplasm-only model. Time delay is on the horizontal axis. "Equal" (i.e., conditional upon $\mathbf{v}_A(0) = \mathbf{v}_A(t')$ and not weighted by e^{-t'/τ_r}; solid lines) and "unequal" (dotted lines) VACFs for soft cell (top) and normal cells (bottom) in a period of 30×54000 time-units. Fully averaged values are shown (black), as well as for the simulation broken up into 5 chunks of equal length (red, blue, green, magenta, cyan). The lack of a time delay of a given length is designated by being below the plotting region. Inset (bottom): Natural logarithm of normal-cell VACF.

are evident, but with the simulation parameter values used, it is not clear that the soft cell acquires a significantly enhanced ability to move within the cell layer. Steric repulsion appears to restrict the motion of both cell types comparably.

The "bursting" phenomena in the experiments and the cytoplasm-only simulations are not exactly analogous. In the experiments, the nucleus is characteristically impinged upon and then moves to occupy a separate space. In the simulations lacking a nucleus, the impingement often occurs more toward the rear, sometimes amounting to the mere retraction of a "tail" which can drive the instantaneous velocity above v_A but with little resultant net displacement. An enhanced instantaneous velocity can also be caused by the combination of shape relaxation and being pushed by a neighboring cell.

More motion occurs in the simulations with nucleus, which is to be expected because the elasticities of the cytoplasms were reduced. Also, while the cell area deviated by at most $\approx 1\%$, larger voids could arise because of the reduced stiffnesses. The nuclear area deviated by at most $\approx 2\%$. Using both nuclear sizes, 1 burst was seen in a time-period of 12×54000 time-units (Fig. 4–8), corresponding to ≈ 65 hours in real time [71]. This would be fewer than in the experiments [9]. While a comparison could be made despite somewhat limited data (only a few cancer cells are shown for a limited time in the experimental videos), it may depend sensitively on model parameters such as density [71]. The cells also appeared more motile than in the experimental videos. This might be because the density value used allows them to easily slip through the packing, which could be related to the low bursting. Also, the presence of adhesion between normal cells in the experiments appears to
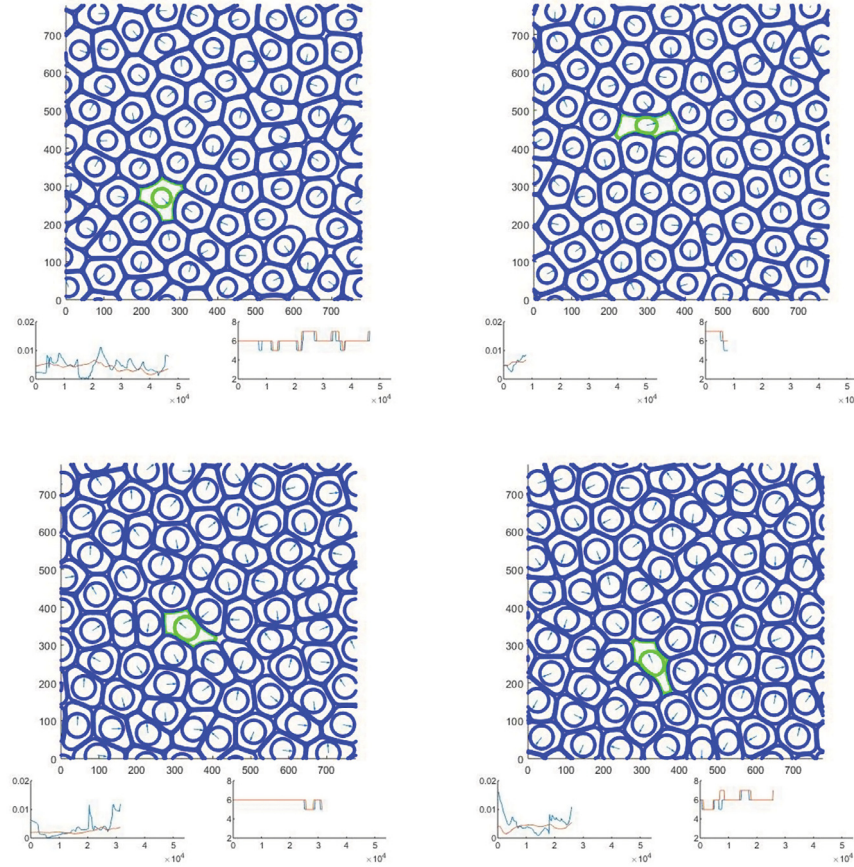
Figure 4–8: Snapshots of simulations of models with nucleus. Midpoint method and polygonal method are used. Time axes are offset. Top row: $R_\nu = R/2$ with $\Delta t = 0.03$, $N_{\text{pts}} = 480$, and $N_{\text{pts}}^\nu = 240$. (L) Still frame showing a large void; (R) the soft cell is about to "burst". Bottom row: $R_\nu = 5R/8$. (L) A soft cell is being pushed by a normal cell behind it ($\Delta t = 0.03$, $N_{\text{pts}} = 320$, $N_{\text{pts}}^\nu = 224$); (R) the soft cell is beginning to "burst", its nucleus having been impinged toward the rear by the normal cells to its bottom left and bottom right ($\Delta t = 0.04$, $N_{\text{pts}} = 400$, $N_{\text{pts}}^\nu = 280$).

85

stabilize their positions, which might help promote bursting. It should be mentioned that the "bursting" behavior seen in the experiments is biologically enhanced and not of purely mechanical origin [9]. As in the cytoplasm-only simulations, with the simulation parameter values used, there are no visibly obvious differences in the motility of the soft and normal cells nor in their primary modes of migration.

CHAPTER 5
Conclusion

We have extended the simulation studies of Refs. [71] and [72] to incorporate a nucleus, as well as obtained more accurate numerics that shed light on the "pinching" problem. In attempting to understand the experimental results of Ref. [9], we have not yet succeeded in finding the right combination of parameter values, such as adhesion, density, and cell mechanical properties. These results clarify the role of a soft cytoplasm and stiff nucleus in our model of cell motility. The effect of cell aggregation has so far hardly achieved closer resemblance to the experimental phenomenology. Further work remains to be done to understand the capabilities and limitations of the model.

Future directions for this project could be the following:

- Studies could be done of the dynamics of the number of neighbors and correlations with cell speed, and differences thereof between the normal and soft cells. A previous study investigated dynamics of topological defects in a magnetic system [85]. Such analysis of the model could also be compared with experiments, and could additionally serve to help validate numerical parameters.

- A larger system size could be experimented with to elucidate possible finite-size effects and anisotropy arising from the periodic boundary conditions. For example, hexagonal boundary conditions could be used.

- Adhesion could be included between normal cells.

- Self-repulsion could be added, enabling lower elasticities.

- The elasticity ratio between "soft" and "normal" cells could be varied. While we have not performed a direct analog to the cancer-cell sheet experiments of Ref. [9], our results for the model with nucleus demonstrate easier movement when using lower elasticities. The two types could be made more unlike to see if a larger separation in motility mechanisms occurs.

- The role of the density parameter could be examined further. While the experiments involve confluent layers, the studies of Ref. [79] revealed a jamming behavior in the high-density regime. Inclusion of adhesion may confound the effect.

- The algorithm for calculating the cell interactions has been improved but still seems very wasteful, starting from scratch for each point on the interface. Intuitively, the continuity of the interface should allow for results of nearby points to be taken advantage of. However, for the model with nucleus, the closest point on the cytoplasm interface from the nucleus can be discontinuous. Also, the parallel structure of modern processors needs to be taken into account to devise a fast algorithm [74].

- Other models for the cell mechanical response could be investigated.

Appendix

The equations for $\mathbf{R}'_n$ and $\mathbf{R}''_n$ for a non-uniform 3-point stencil with $\Delta\zeta_i := \Delta s_n[i]$ are:

$$\mathbf{R}'_n(\zeta_i) = \frac{\Delta s_n[i]\dfrac{\mathbf{R}_n[i] - \mathbf{R}_n[i-1]}{\Delta s_n[i-1]} + \Delta s_n[i-1]\dfrac{\mathbf{R}_n[i+1] - \mathbf{R}_n[i]}{\Delta s_n[i]}}{\Delta s_n[i-1] + \Delta s_n[i]} + O((\Delta s)^2)$$

(A.1)

$$\mathbf{R}''_n(\zeta_i) = \frac{\dfrac{\mathbf{R}_n[i+1] - \mathbf{R}_n[i]}{\Delta s_n[i]} - \dfrac{\mathbf{R}_n[i] - \mathbf{R}_n[i-1]}{\Delta s_n[i-1]}}{(\Delta s_n[i-1] + \Delta s_n[i])/2} + O((\Delta s)^2),$$

where, as before, $\Delta s_n[i] := |\mathbf{R}_n[i+1] - \mathbf{R}_n[i]|$ and $\Delta s \sim \Delta s_n[i]$. In the cytoplasm-only simulations, for $N_{\text{pts}} \lesssim 480$ or so (at least), this appears to work better than the uniform 5-point stencil which may suffer from a too large range of influence. Using these for the unit normal and curvature and the RK4-3/8 method with $\Delta t = 0.008$ and $N_{\text{pts}} = 480$ seemed to work fairly well. Higher accuracy can be obtained by using $\Delta t = 0.006$ and $N_{\text{pts}} = 600$. These parameter values are sensible given the issues with $\Delta t = 0.05$ using the RK4-3/8 method and with $N_{\text{pts}} = 320$. When comparing different numerical parameters, a run may be affected by its initial condition for some time if it was initialized from an active simulation of inferior accuracy.

Higher-order methods (e.g. the fifth-order Dormand–Prince method) were not investigated. For such high orders, the scheme used should be applicable to vector equations in addition to scalar equations [82].

References

[1] P. Friedl and D. Gilmour. Collective cell migration in morphogenesis, regeneration and cancer. *Nat. Rev. Mol. Cell Biol.*, 10:445–457, 2009.

[2] P. Rørth. Fellow travelers: emergent properties of collective cell migration. *EMBO Rep.*, 13:984–991, 2012.

[3] B. Palmieri, C. Scanlon, D. Worroll III, M. Grant, and J. Lee. Substrate mediated interaction between pairs of keratocytes: a tripod traction force model describes their migratory behavior. Submitted for publication.

[4] B. Alberts, A. Johnson, J. Lewis, D. Morgan, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garland Science, 6th edition, 2014.

[5] D. Wirtz, K. Konstantopoulos, and P. C. Searson. The physics of cancer: the role of physical interactions and mechanical forces in metastasis. *Nat. Rev. Cancer*, 11:512–522, 2011.

[6] V. Cristini and J. Lowengrub. *Multiscale Modeling of Cancer: An Integrated Experimental and Mathematical Modeling Approach*. Cambridge University Press, 2010.

[7] C. A. Klein. The metastasis cascade. *Science*, 321:1785–1787, 2008.

[8] A. Wells, Y. L. Chao, J. Grahovac, Q. Wu, and D. A. Lauffenburger. Cell motility in carcinoma metastasis as modulated by switching between epithelial and mesenchymal phenotypes. *Front. Biosci.*, 16:815–837, 2011.

[9] M.-H. Lee, P.-H. Wu, J. R. Staunton, R. Ros, G. D. Longmore, and D. Wirtz. Mismatch in mechanical and adhesive properties induces pulsating cancer cell migration in epithelial monolayer. *Biophys. J.*, 102:2731–2741, 2012.

[10] B. Camley and W.-J. Rappel. Physical models of collective cell motility: from cell to tissue. *J. Phys. D: Appl. Phys.*, 50:113002, 2017.

[11] G. Salbreux, G. Charras, and E. Paluch. Actin cortex mechanics and cellular morphogenesis. *Trends Cell Biol.*, 22:536–545, 2012.

[12] R. Li and G. G. Gundersen. Beyond polymer polarity: how the cytoskeleton builds a polarized cell. *Nat. Rev. Mol. Cell Biol.*, 9:860–873, 2008.

[13] A. J. Ridley, M. A. Schwartz, K. Burridge, R. A. Firtel, M. H. Ginsberg, G. Borisy, J. T. Parsons, and A. R. Horwitz. Cell migration: integrating signals from front to back. *Science*, 302:1704–1709, 2003.

[14] S. Linder. The matrix corroded: podosomes and invadopodia in extracellular matrix degradation. *Trends Cell Biol.*, 17:107–117, 2007.

[15] T. D. Pollard and G. G. Borisy. Cellular motility driven by assembly and disassembly of actin filaments. *Cell*, 112:453–465, 2003.

[16] M. Vicente-Manzanares and A. R. Horwitz. Cell migration: an overview. In C. M. Wells and M. Parsons, editors, *Cell Migration: Developmental Methods and Protocols*, chapter 1, pages 1–24. Humana Press, 2nd edition, 2011.

[17] D. A. Lauffenburger and A. F. Horwitz. Cell migration: a physically integrated molecular process. *Cell*, 84:359–369, 1996.

[18] P. Maiuri *et al.* Actin flows mediate a universal coupling between cell speed and cell persistence. *Cell*, 161:374–386, 2015.

[19] P.-H. Wu, A. Giri, S. X. Sun, and D. Wirtz. Three-dimensional cell migration does not follow a random walk. *Proc. Natl. Acad. Sci. USA*, 111:3949–3954, 2014.

[20] R. Hochmuth. Micropipette aspiration of living cells. *J. Biomech.*, 13:15–22, 2000.

[21] A. G. Clark and E. Paluch. Mechanics and regulation of cell shape during the cell cycle. In J. Z. Kubiak, editor, *Cell Cycle in Development*, chapter 3, pages 31–73. Springer-Verlag, 2011.

[22] M. P. Sheetz, J. E. Sable, and H.-G. Döbereiner. Continuous membrane-cytoskeleton adhesion requires continuous accommodation to lipid and cytoskeleton dynamics. *Annu. Rev. Biophys. Biomol. Struct.*, 35:417–434, 2006.

[23] K. Keren, Z. Pinchus, G. M. Allen, E. L. Barnhart, G. Marriott, A. Mogilner, and J. A. Theriot. Mechanism of shape determination in motile cells. *Nature*, 453:475–480, 2008.

[24] C. M. Baumgarten and J. J. Feher. Osmosis and regulation of cell volume. In N. Sperelakis, editor, *Cell Physiology Sourcebook: Essentials of Membrane Biophysics*, chapter 16, pages 261–301. Academic Press (Elsevier), 4th edition, 2012.

[25] J.-Y. Tivenez, U. Schulze, G. Salbreux, J. Roensch, J.-F. Joanny, and E. Paluch. Role of cortical tension in bleb growth. *Proc. Natl. Acad. Sci. USA*, 106:18581–18586, 2009.

[26] J. Bereiter-Hahn. Mechanics of crawling cells. *Med. Eng. Phys.*, 28:743–753, 2005.

[27] D. McCusker and D. R. Kellogg. Plasma membrane growth during the cell cycle: unsolved mysteries and recent progress. *Curr. Opin. Cell Biol.*, 24:845–851, 2012.

[28] N. C. Gauthier, T. A. Masters, and M. P. Sheetz. Mechanical feedback between membrane tension and dynamics. *Trends Cell Biol.*, 22:527–535, 2012.

[29] K. N. Dahl, A. J. S. Ribeiro, and J. Lammerding. Nuclear shape, mechanics, and mechanotransduction. *Circ. Res.*, 102:1307–1318, 2008.

[30] K. N. Dahl, S. M. Kahn, K. L. Wilson, and D. E. Discher. The nuclear envelope lamina network has elasticity and a compressibility limit suggestive of a molecular shock absorber. *J. Cell Sci.*, 117:4779–4786, 2004.

[31] A. C. Rowat, J. Lammerding, H. Herrmann, and U. Aebi. Towards an integrated understanding of the structure and mechanics of the cell nucleus. *BioEssays*, 30:226–236, 2008.

[32] P. M. Davidson, J. Sliz, P. Isermann, C. Denais, and J. Lammerding. Design of a microfluidic device to quantify dynamic intra-nuclear deformation during cell migration through confining environments. *Integr. Biol.*, 7:1534–1546, 2015.

[33] D.-H. Kim, B. Li, F. Si, J. M. Phillip, D. Wirtz, and S. X. Sun. Volume regulation and shape bifurcation in the cell nucleus. *J. Cell Sci.*, 128:3375–3385, 2015.

[34] V. Nandakumar, J. Kelbauskas, K. F. Hernandez, K. M. Lintecum, P. Senechal, K. J. Bussey, P. C. W. Davies, R. H. Johnson, and D. R. Meldrum. Isotropic 3D nuclear morphometry of normal, fibrocystic and malignant breast epithelial cells reveals new structural alterations. *PLoS One*, 7:e29230, 2012.

[35] D. Wirtz. Particle-tracking microrheology of living cells: principles and applications. *Annu. Rev. Biophys.*, 38:301–326, 2009.

[36] P. A. Janmey and C. A. McCulloch. Cell mechanics: integrating cell responses to mechanical stimuli. *Annu. Rev. Biomed. Eng.*, 9:1–34, 2007.

[37] P. Kollmannsberger and B. Fabry. Linear and nonlinear rheology of living cells. *Annu. Rev. Mater. Res.*, 41:75–97, 2011.

[38] B. D. Hoffman and J. C. Crocker. Cell mechanics: dissecting the physical response of cells to force. *Annu. Rev. Biomed. Eng.*, 11:259–288, 2009.

[39] L. Deng, X. Trepat, J. P. Butler, E. Millet, K. G. Morgan, D. A. Weitz, and J. J. Fredberg. Fast and slow dynamics of the cytoskeleton. *Nat. Mater.*, 5:636–640, 2006.

[40] L. M. Rebêlo, J. S. de Sousa, J. Mendes Filho, J. Schäpe, H. Doschke, and M. Radmacher. Microrheology of cells with magnetic modulation atomic force microscopy. *Soft Matter*, 10:2141–2149, 2014.

[41] D. E. Discher, P. Janmey, and Y. L. Wang. Tissue cells feel and respond to the stiffness of their substrate. *Science*, 310:1139–1143, 2005.

[42] M. Lekka, K. Pogoda, J. Gostek, O. Klymenko, S. Prauzner-Bechcicki, J. Wiltowska-Zuber, J. Jaczewska, J. Lekki, and Z. Stachura. Cancer cell recognition—mechanical phenotype. *Micron*, 43:1259–1266, 2012.

[43] M. Li, D. Dang, L. Liu, N. Xi, and Y. Wang. Atomic force microscopy in characterizing cell mechanics for biomedical applications: a review. *IEEE Transactions on NanoBioscience*, 16:523–540, 2017.

[44] T. P. Kole, Y. Tseng, I. Jiang, J. L. Katz, and D. Wirtz. Intracellular mechanics of migrating fibroblasts. *Mol. Biol. Cell*, 16:328–338, 2005.

[45] F. Guilak, J. R. Tedrow, and R. Burgkart. Viscoelastic properties of the cell nucleus. *Biochem. Biophys. Res. Commun.*, 269:781–786, 2000.

[46] Y. Tseng, J. S. H. Lee, T. P. Kole, I. Jiang, and D. Wirtz. Micro-organization and visco-elasticity of the interphase nucleus revealed by particle nanotracking. *J. Cell Sci.*, 117:2159–2167, 2004.

[47] E. Moeendarbary, L. Valon, M. Fritzsche, A. R. Harris, D. A. Moulding, A. J. Trasher, E. Stride, L. Mahadevan, and G. T. Charras. The cytoplasm of living cells behaves as a poroelastic material. *Nat. Mater.*, 12:253–261, 2013.

[48] P. Friedl, K. Wolf, and J. Lammerding. Nuclear mechanics during cell migration. *Curr. Opin. Cell Biol.*, 23:55–64, 2011.

[49] A. Fruleux and R. J. Hawkins. Physical role for the nucleus in cell migration. *J. Phys.: Condens. Matter*, 28:363002, 2016.

[50] J. S. Lee, C. M. Hale, P. Panorchan, S. B. Khatau, J. P. George, Y. Tseng, C. L. Stewart, D. Hodzic, and D. Wirtz. Nuclear lamin A/C deficiency induces defects in cell mechanics, polarization, and migration. *Biophys. J.*, 93:2542–2552, 2007.

[51] S. B. Khatau, C. M. Hale, P. J. Stewart-Hutchinson, M. S. Patel, C. L. Stewart, P. C. Searson, C. Hodzic, and D. Wirtz. A perinuclear actin cap regulates nuclear shape. *Proc. Natl. Acad. Sci. USA*, 106:19017–19022, 2009.

[52] J. Elric and S. Etienne-Manneville. Centrosome positioning in polarized cells: common themes and variations. *Exp. Cell Res.*, 328:240–248, 2014.

[53] T. Lecuit and P.-F. Lenne. Cell surface mechanics and the control of cell shape, tissue patterns and morphogenesis. *Nat. Rev. Mol. Cell Biol.*, 8:633–644, 2007.

[54] C. Guillot and T. Lecuit. Mechanics of epithelial tissue homeostasis and morphogenesis. *Science*, 340:1185–1189, 2013.

[55] T. E. Angelini, E. Hannezo, X. Trepat, M. Marquez, J. J. Fredberg, D. A. Weitz, and D. Chandler. Glass-like dynamics of collective cell migration. *Proc. Natl. Acad. Sci. USA*, 108:4714–4719, 2011.

[56] T. E. Angelini, E. Hannezo, X. Trepat, J. J. Fredberg, and D. A. Weitz. Cell migration driven by cooperative substrate deformation patterns. *Phys. Rev. Lett.*, 104:168104, 2010.

[57] V. Hakim and P. Silberzan. Collective cell migration: a physics perspective. *Rep. Prog. Phys.*, 80:076601, 2017.

[58] B. Szabó, G. J. Szöllösi, B. Gönci, Z. Jurányi, D. Selmeczi, and T. Vicsek. Phase transition in the collective migration of tissue cells: Experiment and model. *Phys. Rev. E*, 74:061908, 2006.

[59] N. Sepúlveda, L. Petitjean, O. Cochet, E. Grasland-Mongrain, P. Silberzan, and V. Hakim. Collective cell motion in an epithelial sheet can be quantitatively described by a stochastic interacting particle model. *PLoS Comput. Biol.*, 9:e1002944, 2013.

[60] D. Bi, J. H. Lopez, J. M. Schwarz, and M. L. Manning. A density-independent rigidity transition in biological tissues. *Nat. Phys.*, 11:1074–1079, 2015.

[61] D. Bi, X. Yang, M. C. Marchetti, and M. L. Manning. Motility-driven glass and jamming transitions in biological tissues. *Phys. Rev. X*, 6:021011, 2016.

[62] B. A. Camley, Y. Zhang, Y. Zhao, B. Li, E. Ben-Jacob, H. Levine, and W.-J. Rappel. Polarity mechanisms such as contact inhibition of locomotion regulate persistent rotational motion of mammalian cells on micropatterns. *Proc. Natl. Acad. Sci. USA*, 111:14770–14775, 2014.

[63] D. A. Kulawiak, B. A. Camley, and W.-J. Rappel. Modeling contact inhibition of locomotion of colliding cells migrating on micropatterned substrates. *PLoS Comput. Biol.*, 12:e1005239, 2016.

[64] D. Shao, W.-J. Rappel, and H. Levine. Computational model for cell morphodynamics. *Phys. Rev. Lett.*, 105:108104, 2010.

[65] D. Shao, H. Levine, and W.-J. Rappel. Coupling actin flow, adhesion, and morphology in a computational cell motility model. *Proc. Natl. Acad. Sci. USA*, 109:6851–6856, 2012.

[66] S. M. Zehnder, M. Suaris, M. M. Bellaire, and T. E. Angelini. Cell volume fluctuations in MDCK monolayers. *Biophys. J.*, 108:247–250, 2015.

[67] N. Provatas and K. Elder. *Phase-Field Methods in Materials Science and Engineering*. WILEY-VCH, 2010.

[68] P. M. Chaikin and T. C. Lubensky. *Principles of Condensed Matter Physics*. Cambridge University Press, 2000.

[69] K. R. Elder, M. Grant, N. Provatas, and J. Kosterlitz. Sharp-interface limits of phase-field models. *Phys. Rev. E*, 64:021604, 2001.

[70] D. Gruner, R. Kapral, and A. Lawniczak. Nucleation, domain growth, and fluctuations in a bistable chemical system. *J. Chem. Phys.*, 99:3938–3945, 1993.

[71] B. Palmieri, Y. Bresler, D. Wirtz, and M. Grant. Multiple scale model for cell migration in monolayers: Elastic mismatch between cells enhances motility. *Sci. Rep.*, 5:11745, 2015.

[72] Y. Bresler, B. Palmieri, and M. Grant. Effects of cell elasticity on the migration behavior of a monolayer of motile cells: sharp interface model. arXiv:1807.07836, 2018.

[73] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi, and R. Ulerich. GNU Scientific Library: Release 2.5. `https://www.gnu.org/software/gsl/doc/latex/gsl-ref.pdf`, 2018. Accessed 22 October 2018.

[74] S. Chellappa, F. Franchetti, and M. Püschel. How to write fast numerical code: a small introduction. In R. Lämmel, J. Visser, and J. Saraiva, editors, *Generative and Transformational Techniques in Software Engineering II. GTTSE 2007*, pages 196–259. Springer-Verlag, 2008.

[75] K. Elder, H. Gould, and J. Tobochnik. Langevin simulations of nonequilibrium phenomena. *Comput. Phys.*, 7:27–33, 1993.

[76] Private discussion with Yony Bresler.

[77] J. D. Hunter. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.*, 9:90–95, 2007.

[78] F. Drolet, K. R. Elder, M. Grant, and J. M. Kosterlitz. Phase-field modeling of eutectic growth. *Phys. Rev. E*, 61:6705–6720, 2000.

[79] Y. Bresler, B. Palmieri, and M. Grant. Near the jamming transition of elastic active cells: a sharp-interface approach. arXiv:1807.10318, 2018.

[80] E. Süli and D. F. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, 2003.

[81] C. Yuskel, S. Schaefer, and J. Keyser. Parameterization and applications of Catmull–Rom curves. *Comput. Aided Des.*, 43:747–755, 2011.

[82] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations.* Wiley, 3$^{\text{rd}}$ edition, 2016.

[83] W. Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Z. Naturforsch. C*, 28:693–703, 1973.

[84] J.-P. Boon and S. Yip. *Molecular Hydrodynamics.* Dover, 1991. Reprinted from McGraw-Hill, Inc., 1980.

[85] C. Sagui and R. C. Desai. Kinetics of topological defects in systems with competing interactions. *Phys. Rev. Lett.*, 71:3995–3998, 1993.