Technische Universität Dresden • Faculty of Mathematics

Derivation and study of a non-confluent model for deformable cells

Master's thesis

to obtain the second degree

Master of Science
(*M.Sc.*)

written by

Tim Vogel

(born on June 9, 2002 in Finsterwalde)

Day of submission: December 2, 2025

Supervised by Jun.-Prof. Dr. Markus Schmidtchen
(Institute of Scientific Computing)

Statement of authorship

I hereby declare that I have written this thesis (Derivation and study of a non-confluent model for deformable cells) under the supervision of Jun.-Prof. Dr. Markus Schmidtchen independently and have listed all used sources and aids. I am submitting this thesis for the first time as part of an examination. I understand that attempted deceit will result in the failing grade „not sufficient“ (5.0).

Tim Vogel

Dresden, December 2, 2025

Technische Universität Dresden

Matriculation Number: 4930487

Abstract

Building upon the foundations laid in the Bachelor's thesis [Vog23], significant improvements were made, including bug fixes, code parallelisation, and reformulation of forces for stable large scale simulations (detailed in Sections 3 and 4).
Chapter 5 presents a rigorous validation by recreating the diffusion dynamics of the point particle and hard sphere models from [BC12] through extensive Monte Carlo simulations with varying hardness ($h \in \{0, 0.5, 1\}$), representing soft, intermediate, and hard cell behaviours.
Subsequently, Chapter 6 analyses the resulting density dynamics. By connecting to the frameworks of [BC12, BCR17, BSY$^+$18], this work provides a unified perspective on cell dynamics spanning rigid to deformable regimes. Finally, the thesis concludes with a summary of findings and an outlook on potential future extensions and applications of the DF model.

Acknowledgements

Danke für alles Opa :)

Contents

List of Figures

List of Tables

Nomenclature

abbreviation	description
PDE	A partial differential equation is an equation that contains unknown multivariable functions and their partial derivatives.
SDE	A stochastic differential equation is a differential equation in which one or more of the terms is a stochastic process, resulting in a solution that is also a stochastic process.
DF model	The discrete form model is a vertex cell model that is defined by a list of all wall points.

Mathematical conventions

symbol	description
$\vec{v}$	Superscript arrows denote multidimensional variables, e.g. $\vec{v} = (v^x, v^y) \in \mathbb{R}^2$
$\vec{v} \cdot \vec{w}$	Dot between two vector structures is scalar product, e.g. $\vec{v} \cdot \vec{w} = v^x w^x + v^y w^y$
$\frac{\mathrm{d}f}{\mathrm{d}x}$	Derivative of a function f with respect to its single argument x
$\frac{\partial f}{\partial x}$	Partial derivative of a multivariate function f with respect to x
$\nabla_{\vec{v}} f$	Gradient $\nabla_{\vec{v}} f = (\frac{\partial f}{\partial v_1}, \ldots, \frac{\partial f}{\partial v_n})^T$, where f is a scalar function that (partially) depends on $\vec{v} = (v_1, \ldots, v_n)^T$
$\nabla_{\vec{v}} \cdot F$	Divergence $\nabla_{\vec{v}} \cdot F = \frac{\partial F_1}{\partial v_1} + \ldots + \frac{\partial F_n}{\partial v_n}$, where $F = (F_1, \ldots, F_m)^T$ is a vector valued function
Δf	Laplacian of a scalar function f, i.e. $\Delta f = \nabla \cdot (\nabla f)$
$D_{\vec{v}} F$	Jacobian matrix $D_{\vec{v}} F = \begin{pmatrix} \frac{\partial F_1}{\partial v_1} \cdots \frac{\partial F_1}{\partial v_n} \\ \vdots \\ \frac{\partial F_m}{\partial v_1} \cdots \frac{\partial F_m}{\partial v_n} \end{pmatrix}$,
$2e-3$	Scientific notation for $2 \times 10^{-3} = 0.002$
$\mathrm{d}\vec{B}_i(t)$	Two-dimensional Brownian motion applied to cell i at time t in SDEs
I_d	Identity matrix in $\mathbb{R}^{d \times d}$
$\mathbb{R}_+$	Positive real numbers including 0, i.e. $[0, \infty)$
$\mathbb{R}_-$	Negative real numbers excluding 0, i.e. $(-\infty, 0)$
$\mathbb{N}$	Natural numbers excluding 0
$\mathbb{N}_0$	Natural numbers including 0
$N_C \in \mathbb{N}$	Number of cells in a model
$N_V \in \mathbb{N}$	Number of vertices of each cell in a model
$C^i = (\vec{v}_1^i, \ldots, \vec{v}_{N_V}^i)$	DF cell $1 \leqslant i \leqslant N_C$
$\vec{v}_j^i = (v_j^{i,x}, v_j^{i,y})^T \in \mathbb{R}^2$	Vertex $1 \leqslant j \leqslant N_V$ of DF cell $1 \leqslant i \leqslant N_C$
$P(t; \vec{x}_1, \ldots, \vec{x}_{N_C})$	Joint probability density function
$\rho(t; \vec{x})$, $\rho^{N_C}(t; \vec{x})$	First marginal function
$\bar{\rho}(t; \vec{x})$	Density in the mean field limit

1 TODOs

Abstract

- add what was done in the end

df model dynamics

- explain what the k does to the forces (k=1 -¿ discrepance between current and desired state does not influence scaling of force, k=2 -¿ it does influence it linearly)
- explain that we use k=1 only for deforming overlap and k=2 for shape preserving forces
- explain that each figure representing a force has every parameter chosen to be equivalent to the big monte carlo simulations in the next chapter: cell size, desired states (perfect 6eck with distance 0.005 from each vertex to the cell centre), force scalings, time step size
- explain that we tried to use unfavourable initital states to test the forces stability
- change some of the 1e2 type numbers (eg 1e-1 -¿ 0.1)
- write down the cell SDE nicely
- check this force and the computations in the proof, especially the neighboring vertices part
- add an example figure with 2 DF cells having two overlaps and write down what $\Omega_{1,2}$ is and what their overlaps are
- change from A_D^k to $(A_D)^k$ in proof of deforming overlap force

Density comps

- write in sentences
- check the signs in needle sim and mean field pde
- think about mean field pde with interaction

Conclusion

- summarise all results from the thesis

Outlook

- gather input
- structurise
- add bridges to chapters

- formulate in sentences

general things

- control indices
- check eqref for Equation ....
- british spelling: colour, z, behaviour, modelling, optimisation, analyse, centre, fibre, manoeuvre, metre, organise, specialise, tumour
- add references to introduction papers to all chapters
- do a spell check to only use UK english
- insert correct gif links

2 Introduction

Collective cell migration represents a fundamental process underpinning various biological phenomena, including embryonic development, tissue regeneration, wound healing, and the invasive potential of certain cancers. The collective nature of cell movement has been acknowledged for over a century, with early observations recognising its importance in developmental and regenerative processes [AT20, Hol14, Her32, VT66]. However, the underlying mechanisms driving this coordinated behaviour remained contentious, with competing hypotheses suggesting roles for pressure [Her32], surface tension [AT20], or active forces generated by cells [Hol14].
Following a period where research emphasis shifted towards molecular and genetic details, the field has witnessed a resurgence of interest in understanding the physical principles governing collective cell migration. This revival is largely attributed to recent advances in experimental techniques [RCCT17, DRSB$^+$05, TWA$^+$09], enabling direct measurements of mechanical forces exerted by cells, and the development of new conceptual frameworks in biophysics and active matter physics [MJR$^+$13, PJJ15, JGS18], which challenged purely reductionist perspectives [GT18]. These developments coincided with a growing recognition of the critical role of collective cell migration not only in physiological processes but also in the progression of malignant diseases [FNW$^+$95].
The diverse manifestations of collective cell migration depend heavily on the specific biological context and tissue type [FG09]. For instance, epithelial cells often migrate as cohesive sheets on the extracellular matrix (ECM) during morphogenesis, wound closure, and regeneration. Snapshots from a cell wound healing process, illustrating the dynamics of cell migration, are shown in Figure 1.
In contrast, cancer cells often invade tissues as sheets, strands, or clusters, navigating a complex three-dimensional extracellular matrix (ECM) environment [FG09, CCP$^+$14, CV15].
While cell migration occurs extensively in three dimensions, modelling these complex processes remains a significant challenge. Consequently, much current scientific work focuses on two-dimensional systems as a more tractable approach to understand the underlying physical principles.
Throughout this thesis, we study cell dynamics in a bounded two-dimensional domain $\Omega \subset \mathbb{R}^2$. The domain is taken to be a square centred at the origin with side length $2L$, where $L > 0$, and is defined as

$$\Omega = [-L, L]^2.$$

In this work, the total number of cells present within the domain Ω is denoted by $N_C \in \mathbb{N}$.
In two dimensions, cell monolayers serve as fundamental model systems to study cell behaviour and tissue function. These systems, comprising a single layer of cells grown on a surface, can be mathematically and computationally modelled in either a confluent or non-confluent manner. A confluent cell model depicts a continuous, tightly packed layer where cells cover the entire surface without gaps, whereas a non-confluent monolayer represents a state with spaces and gaps between individual cells or cell clusters that have not yet achieved full surface coverage.

Figure 1: A figure from the paper [AT20] illustrating cell migration during the wound healing process. (a) Vector field representation of cellular traction forces in cells as they close a wound, with color intensity reflecting the radial component (positive values indicate outward-directed forces). Panels labelled i and ii present magnified views of the regions marked by arrows in panel (a). (b) Velocity vectors (green) and monolayer stress ellipses (red), depicting the principal stress directions and magnitudes, in a growing cell colony (phase contrast).

Recent years have seen growing interest in understanding the principles governing collective cell migration in confluent cell monolayers and epithelial tissues, which exhibit remarkable patterns and correlations in both structural arrangements and actively driven flows [WV21]. Experimental studies on model systems have revealed phenomena such as unjamming transitions, spontaneous vortex formation, topological defects, and active turbulence. A key challenge is linking this macroscopic behaviour to the properties of individual cells and their interactions, leading to a diverse range of modelling approaches spanning different levels of coarse graining, from subcellular lattices and multiphase field models to vertex, Voronoi, particle, and continuum models. The systematic comparison of these diverse cell models is crucial for selecting appropriate methods for future studies and enabling predictive simulations of patterns and correlations in cell colonies.
Understanding the collective behaviour of cells, particularly in processes such as wound healing, tissue regeneration, and embryonic development, is a central challenge in computational and mathematical biology. These phenomena arise from the interplay of intrinsic cell dynamics and extrinsic cues, including chemical gradients, mechanical forces, and the physical constraints of the environment. A key aspect of this behaviour is the diffusion like motion of cells, which governs how populations

spread, organize, and respond to perturbations.
To gain insight into these processes, we focus on the diffusion dynamics of cells as a fundamental tool for characterizing and comparing different modelling approaches. In this thesis, we investigate how distinct mathematical cell models, ranging from point particles to extended, deformable cells, give rise to different diffusion behaviours. By analysing the underlying mechanisms that drive cell movement and spatial spreading, we aim to uncover the role of cell shape, interaction dynamics, and mechanical properties in shaping collective migration patterns, ultimately providing a deeper understanding of biological processes such as wound closure and tissue morphogenesis.

2.1 Review of cell model literature

Each cell model employs distinct dynamical rules and interaction mechanisms, leading to qualitatively and quantitatively different diffusion behaviours. These differences arise from how the models represent cell shape, internal structure, cell-cell interactions, and response to external cues. As a result, the emergent collective dynamics, such as the speed of spreading, the formation of clusters or gaps, and the sensitivity to crowding, can vary significantly across models.
To establish a comprehensive understanding of these variations, we now provide an overview of the most widely used cell modelling approaches. Starting from the simplest representation, we progress through increasingly complex frameworks, highlighting their underlying assumptions, mathematical structure, and implications for cell diffusion. This comparative analysis sets the foundation for our investigation into how specific model dynamics influence the macroscopic behaviour of cell populations.

Point particle model
We consider the point particle model on the square $\Omega \subset \mathbb{R}^2$, where N_C point particles are located in. Initially, these particles are randomly distributed in Ω. We impose the reflective boundary condition on $\partial\Omega$.
Let

$$\vec{x}_i(t) \in \Omega, \quad 1 \leqslant i \leqslant N_C,$$

be the location of the particle i at time $t > 0$. In this model, cells are considered to be point particles with no real size. There is also no particle interaction, as there is no possibility of collision.
The particles' dynamics are governed solely by Brownian motion. Brownian motion is a random and unpredictable motion that occurs in the real world when particles are suspended in a fluid and collide with surrounding molecules.
In mathematics, we model Brownian motion using stochastic differential equations (SDEs), which are equations that describe the motion of a particle over time in a random and unpredictable manner. SDEs are a powerful tool for modelling complex phenomena in physics, finance, and other fields, and are characterized by the presence of random terms that capture the uncertainty of the system.
The particle movement can be modelled using the following SDE, which describes

the random motion of particles over time

$$\mathrm{d}\vec{x}_i(t) = \sqrt{2D}\,\mathrm{d}\vec{B}_i(t), \quad 1 \leqslant i \leqslant N_C,$$

where the constant $D > 0$ represents the diffusion coefficient which proportionally scales the speed of the particle movements by scaling the random fluctuations.
At the domain boundary, the reflective boundary condition is used.
The term $\mathrm{d}\vec{B}_i(t)$ represents the stochastic increment of a two-dimensional Wiener process (Brownian motion) associated with particle i. It models the random, unpredictable fluctuations in the particle's position over time.
For infinitesimally small time intervals $\mathrm{d}t$, the increment $\mathrm{d}\vec{B}_i(t)$ is normally distributed with mean zero and covariance $\mathrm{d}t \cdot I_2$, where I_2 denotes the 2×2 identity matrix.
This dynamic is fundamental and widely used in SDEs across various fields, including modelling particle diffusion [BC12], biological systems such as cell migration and also financial markets.

The stochastic dynamics of individual particles, governed by Brownian motion and external forces, naturally give rise to a probability density function that characterises the collective behaviour of the entire system through a mean field description. This density function serves as the fundamental mean field that encapsulates the averaged influence of all particles, enabling a seamless transition from microscopic particle trajectories to a macroscopic description of cell distribution where individual fluctuations are replaced by deterministic evolution of the ensemble density.
When analysing the diffusion behaviour of a cell model, our focus shifts to understanding the evolution of this density. The first marginal density function $\rho(t;\vec{x})$ serves as a central quantity in this analysis, representing the probability of finding any particle at position $\vec{x} \in \Omega$ and time $t > 0$. It provides a coarse grained view of the system, capturing how the spatial distribution of cells evolves over time and revealing key features of collective migration, such as spreading rates and pattern formation.
In order to compute the first marginal, we have to start with the joint probability density function. This function, denoted by

$$\begin{aligned} P : \mathbb{R}_+ \times \Omega^{N_C} &\longrightarrow \mathbb{R}_+, \\ (t;\vec{x}_1,\ldots,\vec{x}_{N_C}) &\longmapsto P(t;\vec{x}_1,\ldots,\vec{x}_{N_C}). \end{aligned}$$

$P(t;\vec{x}_1,\ldots,\vec{x}_{N_C})$ tells the probability that the whole cell configuration is in the given state $\vec{x}_1,\ldots,\vec{x}_{N_C} \in \Omega$ at time t. It is defined on the high-dimensional configuration space $\mathbb{R}_+ \times (\Omega)^{N_C} \subset \mathbb{R} \times (\mathbb{R}^2)^{N_C}$ and fulfils

$$\int_{\Omega^{N_C}} P(t;\vec{x}_1,\ldots,\vec{x}_{N_C})\,\mathrm{d}\vec{x}_1 \ldots \vec{x}_{N_C} = 1,$$

for all $t > 0$.
We define $\vec{X} = (\vec{x}_1,\ldots,\vec{x}_{N_C})^T$. The derivation of the high-dimensional diffusion equation is based on the principle of probability conservation in the configuration space. Starting from the stochastic dynamics of N_C independent particles undergoing pure Brownian motion, the evolution of the joint probability density $P(t;\vec{X})$ is

governed by the continuity equation, which states that the rate of change of probability within a region is equal to the net flux of probability across its boundary.
In the absence of external forces, the probability flux is purely diffusive, proportional to the gradient of P, with diffusion coefficient D.
Applying the divergence theorem in the $2N_C$-dimensional configuration space Ω^{N_C}, the surface integral of the flux is converted into a volume integral, leading to the Laplacian operator acting on P. This results in the final form of the equation:

$$\frac{\partial P}{\partial t} = D\Delta_{\vec{X}} P, \quad \vec{X} \in \Omega^{N_C},\ t > 0,$$

where D is the diffusion constant from the particle dynamic and $\Delta_{\vec{X}}$ is the Laplacian operator with respect to $\vec{X}$.
Having obtained the joint probability function $P(t; \vec{X})$, we can continue by deriving the first marginal $\rho(t; \vec{x})$. In contrast to the joint probability function that tells probabilities of a whole cell configuration $\vec{X}$, the first marginal function is a probability density that describes the likelihood of finding any one particle at position $\vec{x} \in \Omega$ and time $t > 0$.
The first marginal still fulfils the condition

$$\int_\Omega \rho(t; \vec{x})\, \mathrm{d}\vec{x} = 1, \quad t > 0.$$

We can obtain $\rho(t; \vec{x})$ by integrating the joint probability density $P(t; \vec{x}_1, \ldots, \vec{x}_{N_C})$ over the positions of all particles except the first, i.e., over $\vec{x}_2, \ldots, \vec{x}_{N_C}$

$$\rho(t; \vec{x}_1) = \int_{\Omega^{N_C - 1}} P(t; \vec{X})\, \mathrm{d}\vec{x}_2 \ldots \mathrm{d}\vec{x}_{N_C}.$$

This marginalisation process effectively averages out the influence of the other particles, yielding the probability density $\rho(t; \vec{x})$ for finding any one particle at position $\vec{x}$. Due to the indistinguishability of the particles, this marginal is the same for any particle and represents the macroscopic distribution of the system. By integrating out all particles but one, we find ρ acting on a much lower-dimensional space

$$\begin{aligned} \rho : \mathbb{R}_+ \times \Omega &\longrightarrow \mathbb{R}_+, \\ (t; \vec{x}) &\longmapsto \rho(t; \vec{x}_1). \end{aligned}$$

In the setting of our point particle model, the first marginal ρ satisfies the partial differential equation:

$$\frac{\partial \rho(t; \vec{x})}{\partial t} = D\Delta_{\vec{x}} \rho(t; \vec{x}), \quad \vec{x} \in \Omega,\ t > 0,$$

where $\Delta_{\vec{x}}$ is the Laplacian operator with respect to the spatial variables.
This equation represents the classic diffusion equation, a cornerstone of physics and mathematics. The same diffusion constant $D > 0$ is used in the SDE for particle movement and the PDE for the probability density function ρ.

If we wish to model additional dynamics for our point particles, such as directed motion towards a food source or response to a chemical gradient, we must incorporate an external force into the system. This force is modelled by a vector valued function $f(\vec{x})$, which depends on the particle's position $\vec{x}$ and represents the influence of external cues, such as chemotactic gradients or mechanical drag

$$\begin{aligned} f : \Omega &\longrightarrow \mathbb{R}^2, \\ \vec{x} &\longmapsto f(\vec{x}). \end{aligned}$$

This force modifies the stochastic dynamics by introducing a deterministic drift component, enabling the particles to exhibit non-random, directed movement in response to their environment. This yields a new particle dynamic

$$\mathrm{d}\vec{x}_i(t) = f(\vec{x}_i(t))\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}\vec{B}_i(t), \quad 1 \leqslant i \leqslant N_C.$$

We define a multidimensional function

$$\begin{aligned} F : \Omega^{N_C} &\longrightarrow \left(\mathbb{R}^2\right)^{N_C}, \\ \vec{X} \longmapsto F(\vec{X}) &= (f(\vec{x}_1), \ldots, f(\vec{x}_{N_C}))^T, \end{aligned}$$

that gathers the external dynamics for all particles.
With that new external force, the diffusion behaviour changes. For our point particle model with included external effects, the joint probability is given by a high-dimensional Fokker-Planck equation

$$\frac{\partial P(t;\vec{X})}{\partial t} = \nabla_{\vec{X}} \cdot [D\nabla_{\vec{X}} P(t;\vec{X}) - P(t;\vec{X})F(\vec{X})], \quad \vec{X} \in \Omega^{N_C},\ t > 0,$$

where $\nabla_{\vec{X}}\cdot$ denotes the divergence operator with respect to $\vec{X}$ and $\nabla_{\vec{X}}$ is the corresponding gradient.
The Fokker-Planck equation is just as fundamental as the diffusion equation, because it provides a deterministic description of the evolution of the probability density for a stochastic system, enabling the analysis of macroscopic behaviour — such as diffusion, drift, and pattern formation — from microscopic particle dynamics.
It is derived using the continuity equation (or conservation of probability), which states that the rate of change of probability in a region is equal to the net flux of probability across its boundary, similarly to the model before.
This leads to a flux expression involving both advective (drift) and diffusive components. The divergence theorem is applied in the $2N_C$-dimensional configuration space Ω^{N_C} to convert surface integrals into volume integrals. This is the step where the additional term comes from, yielding the final form of the Fokker-Planck equation.
To transition from the joint probability density function to the first marginal density function in the presence of external forces, we consider the evolution of the system at the macroscopic level by integrating out the degrees of freedom of all but one particle. This process, which is analogous to the case of pure Brownian motion, relies on the symmetry of the system: since the particles are identical and non-interacting,

the joint density is invariant under permutations of particle labels.
The first marginal of the point particle model with external forces and Brownian motion is given by

$$\frac{\partial \rho(t;\vec{x})}{\partial t} = D\Delta_{\vec{x}}\rho(t;\vec{x}) - \nabla_{\vec{x}} \cdot (f(\vec{x})\rho(t;\vec{x})), \quad \vec{x} \in \Omega,\ t > 0. \tag{1}$$

The structure of the resulting Equation (1) for ρ mirrors that derived in the absence of external forces, with the key difference being the inclusion of a drift term due to the external force field $\mathbf{f}(\mathbf{x})$. This similarity in derivation underscores the robustness of the marginalisation procedure, which consistently reduces the high-dimensional dynamics to a one-particle description, regardless of whether the motion is driven solely by diffusion or by a combination of diffusion and drift.

Hard sphere model
Next, we consider models that add a real size to the particles and introduce particle interactions. With the inclusion of a real size, the particles cannot overlap, resulting in exclusion effects. To account for this, we introduce a new interaction dynamic that ensures the particles do not overlap. This leads to a more complex and realistic model that captures the behaviour of particles with a real size and interactions.
We model particles as spherical in shape, with a diameter $0 < \epsilon \ll 1$. Since particles cannot overlap, the domain $\Omega_\epsilon^{N_C} \subset \Omega^{N_C}$, that encodes where each particle centre can be located, must exclude the areas where

$$\|\vec{x}_i - \vec{x}_j\|_2 \leqslant \epsilon, \quad 1 \leqslant i < j \leqslant N_C.$$

This is due to the fact that particles cannot occupy the same space simultaneously. How this kind of exclusion works is visualized in an one-dimensional example in Figure 2.
To define the new domain, we first describe the set of illegal configurations as

$$\mathcal{B}_\epsilon = \{(\vec{x}_1, \ldots, \vec{x}_{N_C})^T \in \Omega^{N_C} \mid \exists i \neq j :\ \|\vec{x}_i - \vec{x}_j\|_2 \leqslant \epsilon\}.$$

Then we can define the 'hollow' domain

$$\Omega_\epsilon^{N_C} = \Omega^{N_C} \backslash \mathcal{B}_\epsilon. \tag{2}$$

Figure 3 illustrates how a hard sphere cell configuration looks like compared to a point particle configuration.

Under this circumstances, we will get a new dynamic compared to the point particle model. In the work of Bruna et al. [BC12], a hard sphere particle model is examined. All particles in this model are distinct and can be distinguished from one another. A diffusion constant, $D > 0$, is again included in the model to scale the Brownian motion.
The dynamics of the particles are governed by the SDE

$$\mathrm{d}\vec{x}_i(t) = f(\vec{x}_i(t))\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}\vec{B}_i(t), \quad \vec{x}_i \in \Omega_\epsilon^{(i)},\ 1 \leqslant i \leqslant N_C,$$

where $\Omega_\epsilon^{(i)} \subset \Omega$ is the excluded volume domain for cell i, i.e.

$$\Omega_\epsilon^{(i)} = \{\vec{y} \in \Omega \mid \forall j \in \{1, \ldots, N_C\} \backslash \{i\} : \|\vec{x}_j - \vec{y}\|_2 > \epsilon\}.$$

The particles are initially randomly distributed in $\Omega_\epsilon^{N_C}$, ensuring that no overlap occurs between the particles.
The dynamics of hard spheres share significant similarities with those of point particles, featuring both an external force f and Brownian motion. The primary difference lies in the constrained movement: hard spheres are confined to a domain that excludes the volume occupied by other spheres.

Figure 2: This plot illustrates the feasible regions for cell 2 in one-dimensional space, dependent on the position of cell 1. The domain of interest is restricted to the one-dimensional interval $[-L, L]$, with the two cells having a diameter of ϵ. The red boundaries indicate the excluded regions, where neither cell can occupy a position closer than $\frac{\epsilon}{2}$ to the boundary. The red diagonal line represents the occupied region by cell 1.

Figure 3: An illustration from [BC12] of the point particles on the left and the hard sphere particles on the right. We can see areas where the centre of the particles cannot be located, as they would overlap with other particles being marked red.

The model enforces this constraint using reflective boundary conditions on the collision surfaces whenever there are two cells with

$$\|\vec{x}_i - \vec{x}_j\|_2 = \epsilon, \quad 1 \leqslant i < j \leqslant N_C.$$

Due to the presence of numerous holes (occupied by other cells), these reflections create a unique dynamic where cells effectively bounce off each other, distinct from the unhindered motion of point particles.

Similarly as in the point particle model, we are now interested in the macroscopic view of this model. The structure for computing the first marginal of the hard sphere model is quite analogous to the computation for the point particles.
The joint probability density function P of the N_C particles satisfies the high-dimensional Fokker-Planck equation

$$\frac{\partial P}{\partial t} = \nabla_{\vec{X}} \cdot [D\nabla_{\vec{X}} P - PF], \quad \vec{X} \in \Omega_\epsilon^{N_C},\ t > 0,$$

where $\nabla_{\vec{X}}$ and $\nabla_{\vec{X}}\cdot$ denote the gradient and divergence operators with respect to the N_C particle position vector $\vec{X}$ and we again use $F(\vec{X}) = (f(\vec{x}_1, \ldots, f(\vec{x}_{N_C})))^T$.
The joint probability density function for hard spheres differs from that of point particles solely in its domain of definition. While the point particle model is defined on the full configuration space Ω^{N_C}, the hard sphere model is governed by the same Fokker-Planck equation but restricted to the excluded volume domain $\Omega_\epsilon^{N_C}$. This domain restriction encodes the physical impenetrability of finite sized particles while preserving the mathematical structure of the evolution equation.
Using the method of matched asymptotic expansions, the authors also derived the probability density function ρ of finding a single particle at time t and position $\vec{x}$, which satisfies the equation

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= \nabla_{\vec{x}} \cdot \{D\nabla_{\vec{x}}[\rho + \tfrac{\pi}{2}(N_C - 1)\epsilon^2\rho^2] - f\rho\} \\ &= D\Delta_{\vec{x}}\rho + D\tfrac{\pi}{2}(N_C - 1)\epsilon^2\Delta_{\vec{x}}(\rho^2) - \nabla_{\vec{x}} \cdot (f\rho), \quad \vec{x} \in \Omega,\ t > 0. \end{aligned} \tag{3}$$

Comparing Equation (3) (the first marginal distribution for hard spheres) with Equation (1) (the first marginal distribution for point particles), we observe an additional term in Equation (3). This term accounts for the enhanced diffusion rate resulting from the excluded volume effects of the hard spheres. As the particle diameter $\epsilon \to 0$, Equation (3) converges to Equation (1), recovering the dynamics of point particles.
It is important to note that while the joint probability density function for hard spheres is defined on the restricted configuration space $\Omega_\epsilon^{N_C}$, the first marginal density function $\rho(t, \mathbf{x})$ remains defined on the original domain Ω.
This distinction arises because $\rho(t; \vec{x})$ is obtained by integrating the joint density over all other particle positions, effectively averaging out the excluded volume constraints.
For any position $\vec{x} \in \Omega$, there exist valid configurations where the first particle occupies $\vec{x}$ while all other particles are positioned to avoid overlap. Consequently, the

marginal density remains well defined across the entire domain Ω, despite the joint density being restricted to $\Omega_\epsilon^{N_C}$.
In the derivation of Equation (3), the divergence theorem introduces additional boundary terms arising from the excluded volume holes in the configuration space $\Omega_\epsilon^{N_C}$. These boundary contributions, localized near particle-particle collision surfaces, ultimately generate the non-linear diffusion correction term in the first marginal density equation for hard spheres. This term, proportional to the particle volume fraction, captures the excluded volume effects that enhance collective diffusion compared to the point particle model. Bruna and Chapman [BC12] demonstrated that the excluded volume effects of hard spheres lead to enhanced diffusion within the system.
This is also shown in Figure 25 from the original paper, which compares the diffusion behaviour of the point particle model and the hard sphere model. The figure confirms the enhanced diffusion behaviour of the hard sphere system, as the density profile exhibits greater spatial spreading across Ω compared to the point particle model. It uses the Monte Carlo method for the point particles and the hard sphere in order to give a numerical approximation of the density dynamic. The Monte Carlo method involves running the simulation many times ($M = 10\,000$). For each discrete subsquare of our domain, we count how many particles were found at that location across all runs at time $t = 0.05$. We then divide by the total number of runs to get the estimated probability density at that place and time. This estimated density converges to the true density as the number of runs increases.

Soft sphere model
Next, we consider an extension of the model by introducing deformable soft spherical particles. This new model incorporates the effect of deformation and interaction between particles through a potential energy function that depends on the distance between the particles. E.g. the paper [BCR17], written by Bruna, Chapman and Robinson, analyses the diffusion properties of such a model.
The hard sphere model enforces rigid, non-deformable cell boundaries through reflective boundary conditions at a fixed separation distance, leading to abrupt, instantaneous collisions without any interface deformation.
Cells are still represented solely by their centre points $\vec{x}_i \in \Omega$, $1 \leqslant i \leqslant N_C$. Consequently, this model does not account for shape deformations, as no explicit cell boundary is included in the system.
In contrast to the hard sphere model, the soft sphere model does not enforce rigid reflective boundaries. Instead, it allows the centres of two cells to temporarily approach each other to a distance less than ϵ. The system is driven to resolve such overlaps through a repulsive interaction potential, $u : \mathbb{R}_+ \longrightarrow \mathbb{R}$ that assigns an energy penalisation for a given intercellular distance. For repulsive short range potentials, u is characterised by a rapid decay from a high value at small separations to zero at larger distances.
The transition from reflective boundary conditions on neighbouring cell walls to interaction potentials renders the use of a hollow domain impractical. Therefore, we return to the original domain Ω, within which all cells are located.
There are different approaches for modelling a specific function for u, that of course

also change the cell system dynamics. The interaction potentials discussed in Bruna, Chapman, and Robinson [BC12], represent the energy of interaction between two particles separated by distance r. The cell diameter $\epsilon > 0$ used in the hard sphere model now represents the range of interaction in the soft sphere setting. Nevertheless, it can still be interpreted as an effective cell diameter, since regions with $r = \epsilon$ mark the transition between overlapping and non-overlapping cells.
The simplest repulsive pair potential is the soft sphere (SS) potential, which assumes the form

$$u_{\mathrm{SS}}(r) = \left(\frac{\epsilon}{r}\right)^{\nu},$$

where $\nu > 0$ is the hardness parameter that controls the steepness of the potential. As ν increases, the potential becomes more hard sphere like, with the hard sphere limit ($u(r) = \infty$ for $r \leqslant \epsilon$ and $u(r) = 0$ for $r > \epsilon$) approached as $\nu \to \infty$.
Another common form is the exponential (EX) potential

$$u_{\mathrm{EX}}(r) = e^{-\frac{r}{\epsilon}},$$

which features a smooth, rapidly decaying repulsion.

Once an interaction potential has been chosen, the dynamics of the soft sphere cell model for cell i can be defined as follows

$$\mathrm{d}\vec{x}_i(t) = f(\vec{x}_i(t))\,\mathrm{d}t - \sum_{j \neq i} \nabla_{\vec{x}_i} u(\|\vec{x}_i(t) - \vec{x}_j(t)\|_2)\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}\vec{B}_i(t),$$

for $1 \leqslant i \leqslant N_C$ and $t > 0$.

The authors derive the joint probability density for the soft sphere system as part of their many-particle Fokker-Planck formulation. They begin from the stochastic Langevin dynamics for N_C interacting particles subject to pairwise soft repulsive potentials, and then obtain the corresponding Fokker-Planck equation governing the evolution of the joint probability density as

$$\frac{\partial P}{\partial t} = \nabla_{\vec{X}} \cdot [D\nabla_{\vec{X}} P + P\nabla_{\vec{X}}\Phi], \quad \vec{X} \in \Omega^{N_C},\ t > 0,$$

where

$$\begin{aligned} \Phi : \Omega^{N_C} &\longrightarrow \mathbb{R}, \\ \vec{X} &\longmapsto \Phi(\vec{X}) = \sum_{i \neq j} u\big(\tfrac{1}{\epsilon}\|\vec{x}_i - \vec{x}_j\|_2\big), \end{aligned}$$

is total interaction potential energy of the N_C particle system.

The authors continued by deriving the first marginal function, by integrating the joint probability density function of all particles over the positions of all other particles. They then used matched asymptotic expansions to obtain a closed equation for the first marginal, which is a non-linear advection diffusion equation, given by

$$\frac{\partial \rho}{\partial t} = D\Delta_{\vec{x}}\rho - \nabla_{\vec{x}} \cdot (\rho f) + \nabla_{\vec{x}} \cdot (\alpha_u \epsilon^2 (N_C - 1)\rho \nabla_{\vec{x}}\rho), \quad \vec{x} \in \Omega,\ t > 0, \tag{4}$$

where α_u depends on the interaction potential u and $0 < \epsilon \ll 1$ is the interaction range of the particles. Crucially, the parameter

$$\alpha_u = \int_\Omega 1 - e^{-u(\epsilon\|x\|)}\, dx,$$

quantifies the strength of the non-linear diffusion effect, with larger values corresponding to stronger excluded volume effects that enhance the overall diffusion rate of the system.
When comparing the first marginals of the soft sphere model Equation (4) and the hard sphere model Equation (3), we can see that they are similar in structure, with the difference being the coefficients of the non-linear diffusion terms. We can not clearly say which model diffuses faster, as this is dependent on the modelling of the soft interaction.
The influence of the cell hardness to the diffusion rate of the cell system will be investigated in this thesis. We even introduce a parameter that can continuously change the cell hardness from hard to soft.
While these models are powerful, they are limited to spherical particles and do not account for the complex shapes and deformations observed in biological cells.

Phase field model
A new cell modelling framework is now considered which, unlike the models before, features cell deformation. The representation of a cell shifts from just a two-dimensional point $\vec{x}_i(t) \in \Omega$ to a phase field variable $\phi_i(t, \vec{x}) : \mathbb{R}_+ \times \Omega \longrightarrow [-1, 1]$ that defines it's cell border through the level null set, i.e., points $\vec{x} \in \Omega$ where $\phi_i(t, \vec{x}) = 0$.
The phase field approach exhibits conceptual parallels with the soft sphere model proposed by Bruna, Chapman, and Robinson [BCR17], particularly in how cell-cell interactions are modelled through a continuous repulsive energy that prevents overlap.
In both formulations, interactions are governed by a smooth, short range influence. The soft sphere model derives its interaction dynamics from the potential energy $u(\|\vec{x}_i - \vec{x}_j\|_2)$, which leads to a non-linear diffusion equation featuring a diffusion coefficient that depends on local density.
The phase field model resolves cell overlaps through a free energy producing a repulsive force between cells. This enables gradual, continuous deformation of cell interfaces, thereby avoiding discontinuities in the dynamics that are characteristic of discrete collision models. Thus, the interaction mechanisms operate continuously and locally, ensuring a seamless transition between overlapping and non-overlapping configurations while maintaining physical consistency in both models.
Nevertheless, the phase field model diverges fundamentally in its underlying structure: it is a continuum model based on partial differential equations that explicitly encodes cell morphology and internal organisation through the phase field variable ϕ_i, enabling dynamic shape changes, topological transitions, and coupling to geometric features such as surface curvature.
The free energy functional encodes shape regularization, intercellular interactions, and physical constraints of the system. Unlike the soft sphere model, which treats cells as point like entities interacting via a smooth potential, the phase field model

represents cells as much more complex, spatially extended, continuously structured entities, with their internal state fully described by the evolution of the phase field variable ϕ_i. This enables a natural description of complex cell morphologies, topological transitions such as cell division or fusion, and the integration of cell mechanics with geometric curvature via extrinsic curvature contributions in the free energy.
Phase field variables represent cells as smooth functions $\phi_i(\vec{x}, t) \in [-1, 1]$, with $\phi_i > 0$ in the cell interior and $\phi_i < 0$ in the exterior. The cell wall is denoted by values of $\phi_i = 0$. An illustration of a phase field variable is shown in Figure 4.

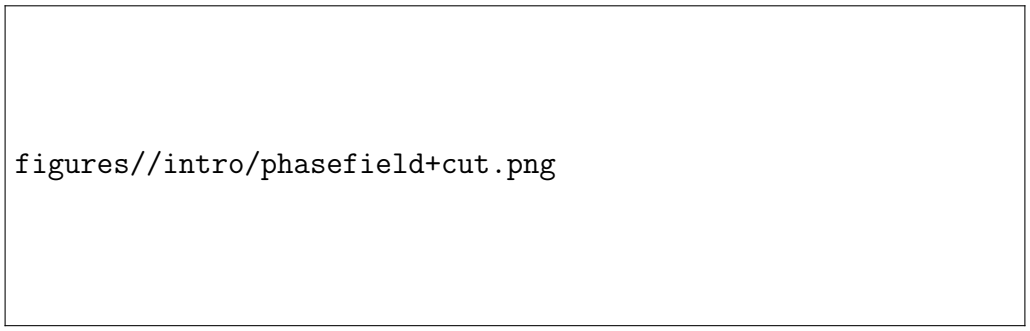


Figure 4: A snapshot from the paper [AT20] illustrating a phase field variable. The cell's inside has value $\phi = 1$ and the outside $\phi = -1$. On the right, we can see a cut of the plot from the left. It also shows the cell wall level set with a dotted line.

The dynamics of a phase field variable ϕ_i can typically be given by a gradient flow of a free energy functional:

$$\frac{\partial \phi_i}{\partial t} + v_0(\vec{v}_i \cdot \nabla_{\vec{x}} \phi_i) = \Delta_{\vec{x}} \frac{\delta F}{\delta \phi_i}, \quad 1 \leqslant i \leqslant N_C,$$

where $\vec{v}_i$ is a vector field used to incorporate activity, with a self propulsion strength v_0, F is a free energy, and $\frac{\delta F}{\delta \phi_i}$ denotes the first variation.
The free energy F arises from a sum of different energies,

$$F = F_{CH} + F_{INT} + F_M.$$

The first energy is a Cahn-Hilliard energy and could look like in [WV21]

$$F_{CH} = \sum_{i=1}^{N} \frac{1}{Ca} \int_\Omega \left(\tfrac{\epsilon}{2} \|\nabla_{\vec{x}} \phi_i\|_2^2 + \tfrac{1}{\epsilon} W(\phi_i) \right) \mathrm{d}\vec{x},$$

where now, ϵ is a small parameter related to the interface thickness, and Ca is a capillary number that scales the relative importance of surface tension. The term $\|\nabla_{\vec{x}} \phi_i\|_2^2$ penalises a long cell wall, as $\nabla_{\vec{x}} \phi_i \neq 0$ only at the cell wall. Thus, the Cahn-Hilliard energy always tries to minimize the area of the cell interface.
$W(\phi_i) = \frac{1}{4}(\phi_i^2 - 1)^2$ is a double-well potential. This energy ensures that each ϕ_i maintains a stable interface of $[-1, 1]$.
The second energy term F_{INT} models cell-cell interactions and could be defined as

in [WV21]

$$F_{INT} = \sum_{i=1}^{N} \frac{1}{Ca} \int_{\Omega} B(\phi_i) \sum_{j \neq i} w(d_j) \, \mathrm{d}\vec{x},$$

where

$$B(\phi_i) = \tfrac{3}{4\sqrt{2}\epsilon}(\phi_i^2 - 1)^2,$$

is an approximation of the delta function of the cell boundary that is non-zero only at the cell wall. The sum in the integral accounts for the interaction with all other cells $j \neq i$ through a short range potential $w(d_j)$, where d_j is the signed distance function to the cell boundary of cell j. When employing this dynamics, the interaction force F_{INT}, acts to reduce or eliminate cell overlaps.
The third energy term F_M differs for different models and incorporates additional mechanical properties of the cells, such as area conservation or bending energy. This energy is dependent on the use case of the model.
It can have a big influence on the dynamic of a phase field model, as analysed in [WV21], where the authors focussed on the influence of microscopic details to incorporate active forces on emerging phenomena.
Four different approaches are considered. One in which the activity is determined by a random orientation, one where the activity is related to the deformation of the cells, and two models with subcellular details to resolve the mechanical interactions underlying cell migration.
The random model determines the direction of motion on the single cell level by a stochastic process. The second model is called elongation model. It identifies the longest axis of the cell's phase field and aligns the direction of motion with this axis. The third and fourth models presented are referred to as the polar model and the nematic model, respectively. Cell motion strength and direction are determined based on subcellular details at the single cell level.
All models are compared with respect to generic features, such as coordination number distribution, cell shape variability, emerging nematic properties, as well as vorticity correlations and flow patterns in large confluent monolayers and confinements. We can see the results in Table 1.

The goal of this paper is a systematic comparison of these approaches and their linkage with statistical observables of experiments to provide a route towards predictive simulations of patterns and correlations in cell colonies. Model predictions are compared with experimental data from various cell cultures. The qualitative differences observed highlight the importance of microscopic details.
A related study by Happel and Voigt [HV23] employs phase field models to investigate cell dynamics, specifically focusing on the influence of curved domains (defined as tori) on collective cell behaviour. The visualization of cells on a torus is shown in Figure 5.
Their work examines emergent phenomena like coordinated rotation on curved surfaces, driven by curvature alignment and self propulsion. Unlike the soft sphere model, which typically assumes spherical symmetry and isotropic interactions, the phase field approach allows for the incorporation of anisotropic effects, such as alignment with principal curvature directions, through geometric coupling terms. This

characteristic	Random	Elongation	Polar	Nematic
Coordination number distribution	(✓)	(✓)	(✓)	(✓)
Shape variability	(✓)	✓	✓	(✓)
Rosette ratio		Differences between models		
Velocity distribution of topological defects		Differences between models		
Correlation between direction of motion and orientation of defect	×	✓	✓	(✓)
Elastic property of + 12 defect	×	Extensile	Contractile	Contractile
Active turbulence	(✓)	(✓)	(✓)	(✓)
Vorticity-vorticity correlation		Similar for all models		
Dependency of defect density on activity	Linear	Linear	Linear	Constant
Rotational motion in circular confinement	×	(✓)	×	×

Table 1: Comparison of the four different phase field models from [WV21] with respect to various characteristics observed in experiments. A check mark ✓ indicates observed agreement, × indicates disagreement and (✓) indicates only qualitative agreement with universal feature. If experimental data are not available or insufficient for a comparison, only similarities or differences of the models are noted.

capability enables the simulation of complex collective behaviours but also increases computational demands compared to simpler point particle models.
The phase field model developed by Happel and Voigt [HV23] highlights the critical role of extrinsic curvature coupling in dictating the alignment of cell elongation with principal curvature directions and the emergence of coordinated rotation in epithelial layers on curved surfaces. By combining a diffuse interface representation with a free energy incorporating both intrinsic and extrinsic geometric terms, their simulations

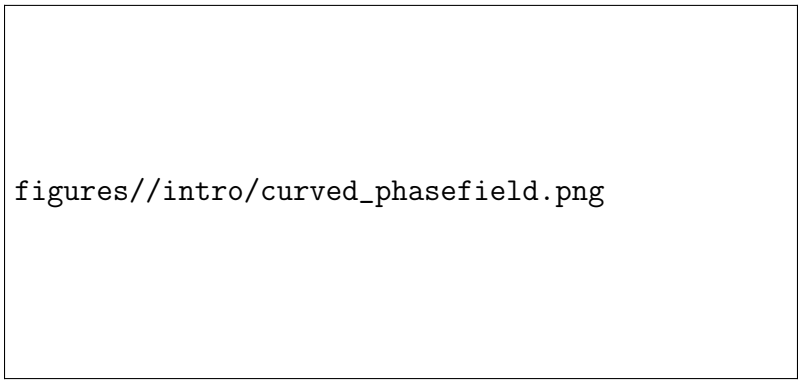


Figure 5: A figure from the paper [HV23] showing a phase field variable ϕ on a curved domain, specifically a torus. In (a) and (b), the surface of the torus from (c) is shown as a parametrization on a rectangle, where the blue edges are identified (glued together) to form the toroidal direction, and the red edges are similarly identified to form the poloidal direction. Red and blue lines represent periodic boundary conditions, which are enforced by gluing the corresponding edges in (b) and (c). The color coding corresponds to the extrinsic curvature parameter E_c (see Eq. (3)): $E_c = 0$ (purple) results in a geodesic circle on both geometries; $E_c > 0$ (green) favours alignment with the direction of maximum absolute curvature; $E_c < 0$ (yellow) favours alignment with the direction of minimal absolute curvature. Cell elongation is highlighted for visibility. On toroidal surfaces, cell shape depends on position due to varying curvature. (c) shows the trajectories, final positions, and shapes of cells over time. The influence of extrinsic curvature is not apparent in the final configuration, as all shapes were obtained by solving Eq. (1) with $v_0 = 0$.

successfully reproduce key experimental observations, including spontaneous rotation on cylindrical surfaces and curvature dependent morphological changes on tori. This work emphasizes the significance of geometric constraints in tissue morphogenesis and offers a framework for investigating how curvature influences collective cell dynamics beyond planar environments.

Vertex model

The last cell model approach is also the approach we choose in this thesis. Vertex models represent a powerful and versatile approach for simulating the mechanical behaviour of biological cells and tissues.

Originating from materials science and solid mechanics, these models represent cells as polygons, where the boundaries are defined by a discrete set of vertices connected by edges. The degrees of freedom in these models are precisely these vertices, meaning that all the cell model dynamics are applied only to the vertices.

The cell dynamic in a vertex model is usually given by the equation

$$\frac{\mathrm{d}\vec{x}_i}{\mathrm{d}t} = F_i^m, \tag{5}$$

where F_i^m is the total force acting on vertex $1 \leqslant i \leqslant N_V$ of cell $1 \leqslant m \leqslant N_C$.

Like in the phase field model, F_i^m is a sum of different forces that define the cell behaviour, such as the cell flexibility or the interaction with other cells. The forces

participating in F_i^m or often found as negative gradients of energies that shall be minimized. This method called gradient flow is also used for phase field models.
Due to their discrete nature, vertex models are computationally efficient compared to continuum models, while still providing rich insights into the interplay between cell mechanics and collective behaviour, making them invaluable for understanding fundamental principles governing cell organization and tissue dynamics across diverse biological contexts.
This geometric representation allows for the incorporation of key cellular properties, such as surface tension, cortical tension, adhesion, and local shape constraints, into an energy functional. Minimizing this energy through computational simulation enables the capture of emergent phenomena like tissue morphogenesis, cell migration, wound healing, and pattern formation.

The paper [FOBS14] developed a sophisticated vertex based model to simulate the dynamic behaviour of confluent epithelial cell sheets, representing tissues where cells completely cover the available space without gaps. Their model aims to capture key biological processes observed in real epithelial tissues.
Conservation of cell area and perimeter ensures cells maintain their size and shape during deformation. Junctional rearrangements, including neighbour exchange and vertex/edge merging, enable cell migration and tissue remodelling. Cell division is modelled by creating a rosette structure from multiple adjacent vertices and cell growth and death simulate changes in individual cell size. A detailed exploration of the implementation and specific consequences of these confluent specific mechanics is beyond the scope of this work. While these mechanisms are essential for understanding the complex dynamics of confluent tissues, the focus of this thesis lies on non-confluent cell models, where the presence of interstitial space and variable packing density leads to distinct physical behaviours.

Boromand, Merkel, and Manning [BSY$^+$18] investigated the jamming transition in a system of deformable cells using a vertex model. Of particular interest is their development of a non-confluent cell model meaning that gaps between the cells are allowed by the model. This model is versatile and can be applied to simulate cells, foams, emulsions, and other soft particulate materials.
It uniquely combines the ability to represent individual deformable particles with the established shape energy function of the vertex model. This shape energy function incorporates terms that penalize deviations from a target area (a) and perimeter (p), along with repulsive interparticular forces to prevent overlap.
The model defines polygons with $N_V \in \mathbb{N}$ vertices, where the bond vector $\vec{l}_{mi}$ connects vertex $\vec{v}_{m,i}$ to $\vec{v}_{m,i+1}$, i.e. vertex i to vertex $i+1$ of cell m. The complete shape energy function, which governs the particle dynamics, integrates these various contributions:

$$E = E_{\text{contract}} + E_{\text{compress}} + E_{\text{line tension}} + E_{\text{bending}} + E_{\text{interaction}}.$$

The first energy reads

$$E_{\text{contract}} = \frac{k_l N_V}{2} \sum_{m=1}^{N_C} \sum_{i=1}^{N_V} (l_{m,i} - l_0)^2,$$

where k_l is the spring constant and l_0 is the equilibrium length of the edges. It penalises contractions or expansion of cell edges and makes the edges return to their desired length l_0.
Similarly, the second force E_{compress} stabilises the cells area to a equilibrium area a_0

$$E_{\text{compress}} = \frac{k_a}{2} \sum_{m=1}^{N_C} (a_m - a_0)^2,$$

where k_a is compressibility constant.
The line tension energy is another energy working with the cell edges. It sanctions long edges with the formula

$$E_{\text{line tension}} = \gamma \sum_{m=1}^{N_C} \sum_{i=1}^{N_V} l_{m,i},$$

where γ is the line tension coefficient.
The last shape energy is given by the bending energy. It acts to resist changes in the angles between adjacent edges of the polygonal cells. This energy encourages the cells to maintain smooth, relatively straight boundaries and discourages sharp bends or kinks.

$$E_{\text{bending}} = \frac{k_b}{2N_V} \sum_{m=1}^{N_C} \sum_{i=1}^{N_V} \left(\frac{2(\hat{l}_{m,i} - \hat{l}_{m,i+1})}{l_{m,i} - l_{m,i+1}} \right)^2,$$

where k_b is the bending rigidity constant, $\hat{l}_{m,i} = \frac{\vec{l}_{m,i}}{\|\vec{l}_{m,i}\|_2}$ is the unit vector of $\vec{l}_{m,i}$.
These four energies are conserving a specific cell shape.

There are two different methods to model the repulsive interaction between two deformable polygons called rough surface (RS) and smooth surface (SS) method. Both methods are illustrated in Figure 6.
In the Rough Surface (RS) method, the discrete nature of the polygonal cell representation is emphasized. Each vertex of a cell is treated as the center of a rigid disk with a fixed diameter, set to $\delta = l_0$ (equilibrium bond length). Repulsive interactions between cells are then modelled as linear spring forces that arise when these disks overlap. This approach effectively simulates a rough surface interaction, characterized by discrete points of repulsion localized at the cell vertices, rather than a continuous interaction across the cell boundary.

$$E_{\text{RS}} = \sum_{m=1}^{N_C} \sum_{n>m}^{N} \sum_{j=1}^{N_V} \sum_{k=1}^{N_V} \frac{k_r}{2} (\delta - |\vec{v}_{m,j} - \vec{v}_{n,k}|)^2 \times \Theta(\delta - |\vec{v}_{m,j} - \vec{v}_{n,k}),$$

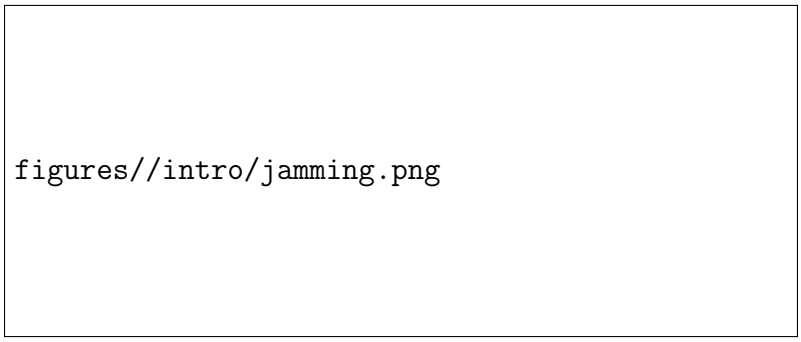


Figure 6: A snapshot from the paper [BSY+18] illustrating a configuration of vertex based deformable cells. Schematic of deformable polygons with $N_V = 34$ vertices (where the position of vertex j in the polygon m is denoted by $\vec{v}_{m,j}$), area a_m, and perimeter p_m. The edge $l_{m,j} = p_m/N_V$ represents the line segment connecting vertices j and $j+1$ in polygon m. Two methods are used to model the edges of deformable polygons: (a) and (b) show the RS method, where disks of diameter δ are centred at the polygon vertices; (c) and (d) show the SS method, where polygon edges are modelled as circulo-lines of width δ. The quantity $d_{\min}$ denotes the minimum distance between the line segments $l_{m,j}$ and $l_{n,k}$.

where k_r is the repulsive constant and Θ is the Heaviside step function, that is either 1 for a positive argument or 0 otherwise.
In contrast to the RS method, the Smooth Surface (SS) method adopts a smoother representation, modelling polygon edges as circulo-lines, essentially line segments with a finite width δ. Repulsive forces are then calculated based on the minimum distance $d_{\min}$, between these circulo-line segments belonging to different polygons. The interaction energy is then formulated using this minimum distance $d_{\min}$.

$$E_{\mathrm{SS}} = \sum_{m=1}^{N_C} \sum_{n \neq m} \sum_{j=1}^{N_V} \sum_{k=1}^{N_V} \frac{1}{2} k_r \left(\delta - d_{\min}\right)^2 \Theta\left(\delta - d_{\min}\right).$$

This method provides a smoother, more continuous repulsion, better approximating the behaviour of soft, continuous interfaces.
Despite the different interaction mechanisms, both methods yield similar structural and mechanical properties at jamming onset. This indicates that the overall jamming behaviour is robust to the specific choice of interaction model.

Discrete Form model

This thesis introduces the Discrete Form (DF) cell model, a vertex based framework inspired by [FOBS14, BSY+18] for simulating cell dynamics in non-confluent systems, similar to [BSY+18], but typically using fewer vertices (e.g., $N_V = 6$).
The DF model incorporates shape preserving forces (area, edge, interior angle energies) and interaction forces (deforming and bounce overlap forces) derived as gradient flows of energy functionals, allowing for cell deformation like in [BSY+18, FOBS14, WV21, HV23] and ensuring evolution towards lower energy configurations. This allows representation of a wide range of desired cell shapes.

The core contribution of this thesis is a non-confluent DF model that systematically investigates how cellular deformability, controlled by a hardness parameter $h \in [0, 1]$, influences the overall diffusivity of the cell system, enabling a transition from hard disc like to soft disc like behaviour.

3 DF cell model

The following two sections are a recap of the DF cell model and its dynamics that were introduced in my Bachelor's thesis [Vog23].
We are considering cells in the two-dimensional space $\mathbb{R}^2$. Here, cells are considered to be polygons.

Definition 3.1. Polygon
A polygon is a closed geometric figure in $\mathbb{R}^2$, constructed by joining a finite number of straight line segments end to end. It can be described by a sequence of its vertices $(\vec{v}_1, \dots, \vec{v}_N)$. The following properties characterise a polygon:

1. A polygon is **simple** if no two line segments cross each other.
2. A polygon has a **positive orientation** if the vertices are ordered counterclockwise.
3. A polygon has a **negative orientation** if the vertices are ordered clockwise.

Having established this definition, we are now ready to define our cell model.

Definition 3.2. Discrete form (DF)
A cell in its discrete form (**DF**) is given by an ordered sequence of its vertices $C = (\vec{v}_1, \dots, \vec{v}_N)$ if the resulting polygon when connecting every vertex with its neighbours and $\vec{v}_1$ with $\vec{v}_N$ is simple and positively orientated. We set $\vec{v}_{N+1} = \vec{v}_1$ and $\vec{v}_0 = \vec{v}_N$ to enable periodic indexing, which simplifies the computation of the upcoming forces a lot.

In this thesis, DF cells may also be called discrete cells. In our model, the cell vertices are denoted by $\vec{v}$. Thus, the character v refers to vertex positions and not to velocity. The term velocity is not used throughout this thesis as vertex dynamics are entirely given by the upcoming forces and a cell wise computed Brownian motion.

The next step is to describe the setup of a DF simulation.

Definition 3.3. DF simulation
A DF simulation considers $N_C \in \mathbb{N}$ cells. Each cell has the same amount of $N_V \in \mathbb{N}$ vertices. Thus, the notation of all cells and their vertices is given by

$$C^i = (\vec{v}_1^{\,i}, \dots, \vec{v}_{N_V}^{\,i}), \quad 1 \leqslant i \leqslant N_C.$$

The complete set of all cells is represented by

$$\vec{C} = (C^1, \dots, C^{N_C}),$$

which also contains all vertices from all cells.
The simulation's dynamics are defined on all cell vertices via the stochastic differential equation (SDE):

$$\mathrm{d}\vec{v}_j^{\,i}(\vec{C}(t), t) = \mathbf{F}_j^i(\vec{C}(t))\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}\vec{B}_i(t), \quad 1 \leqslant i \leqslant N_C,\ 1 \leqslant j \leqslant N_V.$$

where $\mathbf{F}_j^i$ describes the total interaction force on vertex $\vec{v}_j^i$ caused by the current cell system $\vec{C}$ and $\sqrt{2D}\mathrm{d}\vec{B}_i(t)$ models the two-dimensional standard Brownian motion of cell i with diffusion coefficient D. Note, that all vertices of cell i perform the same Brownian motion such that the whole cell i moves in the direction of $\mathrm{d}\vec{B}_i(t)$.

The simulation domain is always a square around the origin that is defined by $L > 0$ via

$$\Omega_L = [-L, L]^2.$$

How the interaction force $\mathbf{F}$ can be modelled will be shown the next chapter.

4 DF model dynamics

We characterise the interaction force $\mathbf{F}$ as the sum of gradient flows of energies.
A gradient flow describes how a system changes over time in a way that always reduces a given energy $E(\vec{C})$. To obtain the gradient flow of this energy on vertex $\vec{v}$, we must add the term $-\nabla_{\vec{v}} E(\vec{C})$ to $\mathbf{F}$. Since all our energy terms are positive, the lowest possible value is zero. So, the gradient flow moves the system step by step toward this minimum, always trying to decrease the energy until, ideally, it reaches zero. This is how we guide the motion of our cells: by letting them follow the gradient flow of each energy so that their shapes and vertex positions gradually adjust to reduce the total energy.
In [Vog23], the area, edge, interior angle, and overlap energies were introduced. The first three energies are responsible for maintaining the shape of each cell. All of these three according forces act on each cell in a vacuum based only on its own current cell shape.
Unlike in [Vog23], where each cell was assigned an individual desired state, we now assume a common desired state for all cells. This simplification allows for a more controlled analysis of the system's deformability and its influence on the collective dynamics. We assume that all cells are initially given in their desired states in order to prevent system instabilities right from the beginning.
Additionally, we introduce slight modifications to the energy formulation: rather than being defined locally on vertices or edges, the energies are now defined over entire cells. This adjustment provides a more coherent basis for deriving cell-level forces and ensures consistency with the global dynamic framework introduced in this study.
Interactions between different cells just arise from the overlap force, which acts to resolve overlaps and to prevent cell interpenetration. In the process of resolving overlaps, the shape of the cells will change. Once the overlap is resolved, the first three forces act to restore the cell's original shape.
The central question we aim to investigate in this thesis is how the deformability of individual cells influences the overall diffusivity of the cell system. But first, let us introduce each of the mentioned forces.
We define our energies as

$$E_k(x) = \tfrac{1}{k} |x_{\text{desired}} - x_{\text{current}}|^k,$$

where $k \in \mathbb{N}$ is a positive integer parameter specific to each energy term. Using different values of k allows us to model various types of energies and their corresponding forces, resulting in distinct dynamical behaviors that reflect different aspects of cell physics.
In order to compute the forces arising from these energy functions, we require the gradient ∇E. This leads us to compute derivatives of the form

$$\frac{\mathrm{d}}{\mathrm{d}x} |x|^k.$$

While $|x|^k$ is not classically differentiable at $x = 0$, it is weakly differentiable for all $k \in \mathbb{N}$. There exists a locally integrable function

$$x \mapsto k \operatorname{sgn}(x) |x|^{k-1} \in L^1_{loc}(\mathbb{R}),$$

such that for all $\phi \in C_C^\infty(\mathbb{R})$:

$$\begin{aligned}\int_{\mathbb{R}} |x|^k \phi'(x)\,\mathrm{d}x &= \int_{\mathbb{R}_+} x^k \phi'(x)\,\mathrm{d}x + \int_{\mathbb{R}_-} (-x)^k \phi'(x)\,\mathrm{d}x \\ &= [x^k \phi(x)]_0^\infty - \int_{\mathbb{R}_+} k x^{k-1} \phi(x)\,\mathrm{d}x \\ &\quad + [(-x)^k \phi(x)]_0^\infty - \int_{\mathbb{R}_-} k(-x)^{k-1} \phi(x)\,\mathrm{d}x \\ &= -\int_{\mathbb{R}_+} k x^{k-1} \phi(x)\,\mathrm{d}x - \int_{\mathbb{R}_-} k(-x)^{k-1} \phi(x)\,\mathrm{d}x \\ &= -\int_{\mathbb{R}} k\,\mathrm{sgn}(x)|x|^{k-1} \phi(x)\,\mathrm{d}x.\end{aligned}$$

Thus, $x \mapsto k\,\mathrm{sgn}(x)|x|^{k-1}$ is the weak derivative of $x \mapsto |x|^k$. We will use this weak derivative for all of our force computations.

4.1 Area force

The area force is designed to maintain each cell's area close to a preferred target value. In order to compute a cells area, which is the area of a positively orientated polygon, we can use the Shoelace formula from [Sho14].

Proposition 4.1. ***Shoelace formula for DF cells***
Let $C = (\vec{v}_1, \ldots, \vec{v}_N)$ be a DF cell with $\vec{v}_j = (v_j^x, v_j^y)^T$ for $j = 1, \ldots, N$. We determine the area A_C of C by applying the Shoelace formula

$$A_C = \tfrac{1}{2} \sum_{j=1}^{N} (v_j^x v_{j+1}^y - v_{j+1}^x v_j^y),$$

where $\vec{v}_{N+1} = \vec{v}_1$.
Proof.
An illustration supporting the proof is provided in Figure7, which is where the idea of the proof comes from. Without loss of generality, we may assume that all coordinates are positive. If this is not initially the case, the entire polygon can be translated into the positive quadrant without affecting its area.
For each $1 \leqslant j \leqslant N$ the edge $\overline{\vec{v}_j\,\vec{v}_{j+1}}$ is associated with the area T_j of the trapeze that arises when connecting the line segment vertically with the x-axis. The signed trapeze area of T_j can be computed with

$$T_j = \tfrac{1}{2}(v_j^y + v_{j+1}^y)(v_j^x - v_{j+1}^x).$$

The area T_j has a positive sign if $v_j^x \geqslant v_{j+1}^x$ (green arrow in Figure 7) and a negative sign otherwise (red arrow). As depicted in the figure, the negatively signed areas precisely cancel the excess portions that would result from summing only the positively signed trapezoids. Thus the total polygon's area is equal to the sum of all trapezes

$$A_C = \sum_{j=1}^{N} T_j = \tfrac{1}{2} \sum_{j=1}^{N} (v_j^y + v_{j+1}^y)(v_j^x - v_{j+1}^x) = \tfrac{1}{2} \sum_{j=1}^{N} (v_j^x v_{j+1}^y - v_{j+1}^x v_j^y).$$

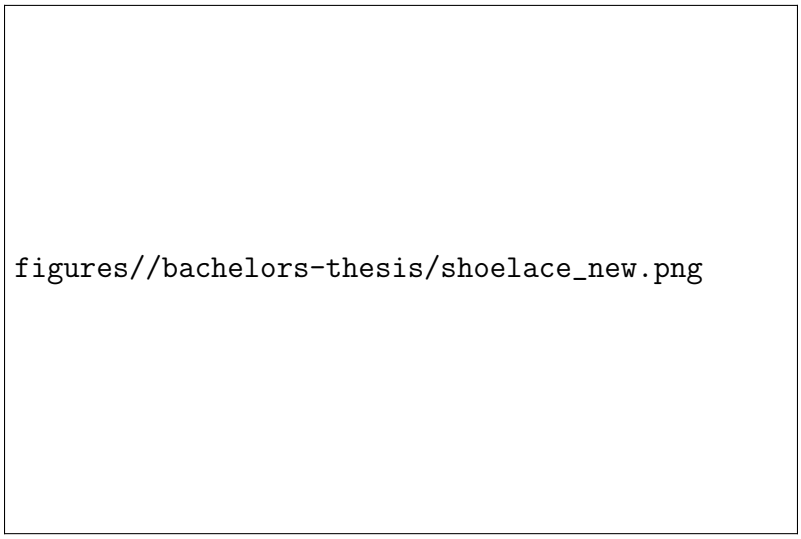


Figure 7: This figure shows a geometrical interpretation of the shoelace formula. The green arrows, which point from right to left, represent positive trapezoidal areas that contribute positively to the total area of the polygon. In contrast, the red arrows point from left to right and represent negative areas that are subtracted in the computation. The vertical black lines divide the plot into subregions. Within each subregion, green arrows are counted with plus signs and red arrows with minus signs. We observe that the subregions lying outside the polygon contain an equal number of plus and minus signs, indicating that their net contribution to the area is zero. In contrast, the subregions inside the polygon always have one more plus sign than minus signs, meaning their area is counted exactly once in the total. Overall, this illustrates that the method correctly computes the area of the polygon. Source: [Sho22]

□

With the Shoelace formula we are able to easily compute all cell areas at all times in the simulation. This enables us to implement the gradient flow over the area energy.

Definition 4.2. Area energy
The energy $A_k : (\mathbb{R}^2)^{N_V} \to \mathbb{R}_{\geqslant 0}$ for $k \in \mathbb{N}$, used to keep the cells at a constant volume, reads

$$A_k(C) = \tfrac{1}{k}|A_C - A_d|^k, \tag{6}$$

where A_d is the desired cell area of all cells and A_C is the current area of cell C.

To maintain the cell area during the simulation, we evaluate the gradient flow of the area energy which indicates the direction of motion for each vertex for preserving the cell area.

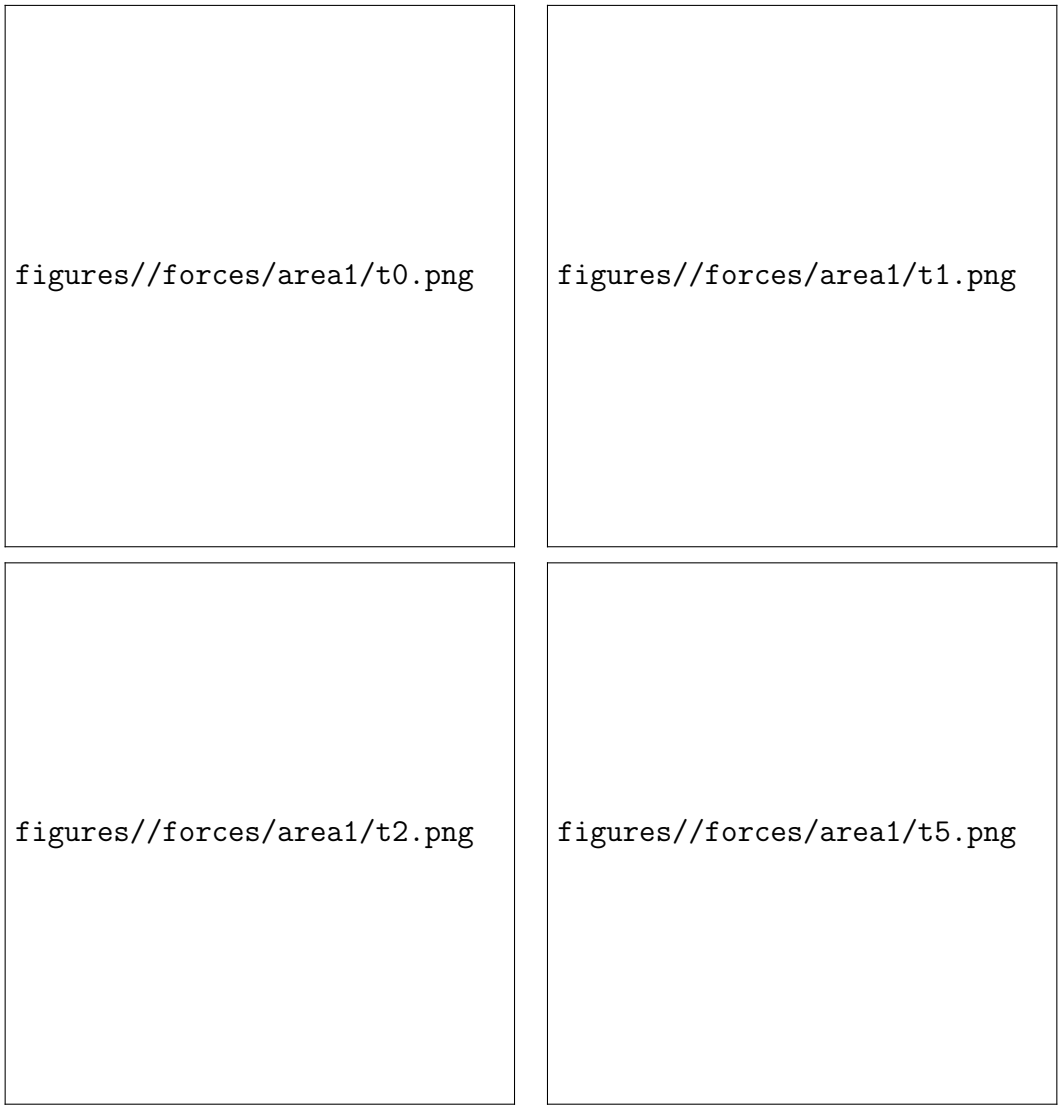


Figure 8: The top four plots show the evolution of a DF cell influenced solely by the area force, with $k = 2$ applied to the vertices and a force scaling of 4×10^8, at times $t \in \{0, 1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}\}$.
Thus, we have $\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = -4 \times 10^8 \nabla_{\vec{v}} A_2(C)$ for all vertices. The initial cell area is $A_C = 1 \times 10^{-5}$, while the desired cell area is set to 6.5×10^{-5}.
We deliberately chose an irregular cell shape, since small vertical changes to the vertices lead to large changes in area. Click *here* to view the corresponding animation (GIF).
The area force successfully restores the desired cell area after 5 time steps. This reduction in energy is likewise reflected in Figure 9.

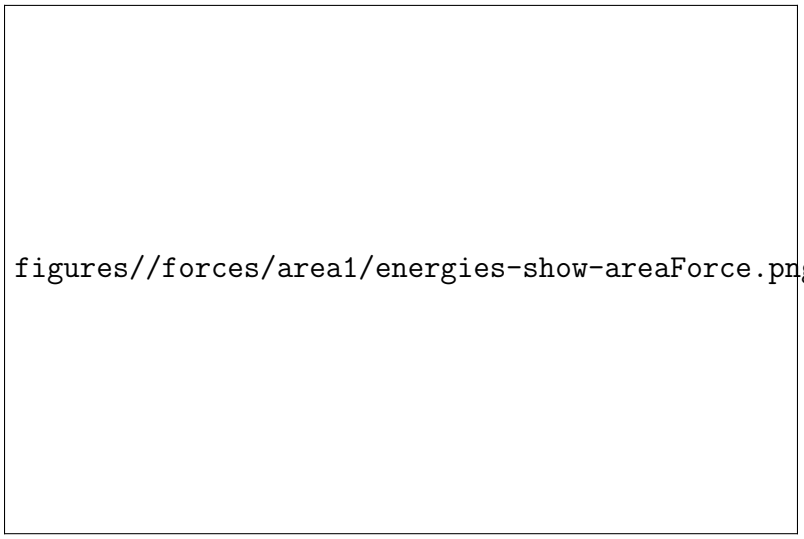


Figure 9: The area force successfully restores the desired cell area after 5 time steps as this energy diagram shows.

Proposition 4.3. ***Area force***
The area force $F_k^{(A)} : (\mathbb{R}^2)^{N_V} \to (\mathbb{R}^2)^{N_V}$ *that gets applied on cell* C *is given by*

$$F_k^{(A)}(C) = -(\nabla_{\vec{v}_1} A_k(C), \dots, \nabla_{\vec{v}_{N_V}} A_k(C))^T,$$

where the gradient $\nabla_{\vec{v}_j} A_k(C)$ *with respect to* $\vec{v}_j = (v_j^x, v_j^y)^T$ *is given by*

$$\nabla_{\vec{v}_j} A_k(C) = \frac{1}{2} \operatorname{sgn}(A_C - A_d)|A_C - A_d|^{k-1} \begin{pmatrix} v_{j+1}^y - v_{j-1}^y \\ v_{j-1}^x - v_{j+1}^x \end{pmatrix}, \tag{7}$$

for all $1 \leqslant j \leqslant N_V$.

Proof.
Choose $1 \leqslant j \leqslant N_V$.

$$\begin{aligned}
\nabla_{\vec{v}_j} A_k(C) &= \tfrac{1}{k} \nabla_{\vec{v}_j} |A_C - A_d|^k \\
&= \operatorname{sgn}(A_C - A_d)|A_C - A_d|^{k-1} \nabla_{\vec{v}_j} (A_C - A_d) \\
&= \operatorname{sgn}(A_C - A_d)|A_C - A_d|^{k-1} \nabla_{\vec{v}_j} A_C \\
&= \operatorname{sgn}(A_C - A_d)|A_C - A_d|^{k-1} \nabla_{\vec{v}_j} \left(\frac{1}{2} \sum_{k=1}^{N} (v_k^x v_{k+1}^y - v_{k+1}^x v_k^y) \right) \\
&= \frac{1}{2} \operatorname{sgn}(A_C - A_d)|A_C - A_d|^{k-1} \begin{pmatrix} \partial_{v_j^x} (v_j^x v_{j+1}^y - v_j^x v_{j-1}^y) \\ \partial_{v_j^y} (v_{j-1}^x v_j^y - v_{j+1}^x v_j^y) \end{pmatrix} \\
&= \frac{1}{2} \operatorname{sgn}(A_C - A_d)|A_C - A_d|^{k-1} \begin{pmatrix} v_{j+1}^y - v_{j-1}^y \\ v_{j-1}^x - v_{j+1}^x \end{pmatrix}.
\end{aligned}$$

Remember that A_d is just an independent constant. □

It is also valid to write $F_j^{(A)}(\vec{C})$ instead of $F_j^{(A)}(C)$, since C is included in $\vec{C}$. Figure 8 illustrates how the area force acts on a cell to either expand or contract it toward the desired target area.

4.2 Edge force

The next force we would like to model is the edge force. It acts on the cells' edges and aims to maintain their lengths. We define the edge j as $\overline{\vec{v}_j\,\vec{v}_{j+1}}$ and use the operator

$$E_C^j = \|\vec{v}_j - \vec{v}_{j+1}\|_2,$$

to compute the length of the edge.
The according energy for this edge is:

Definition 4.4. Edge energy
The energy $E_k : (\mathbb{R}^2)^{N_V} \to \mathbb{R}_{\geqslant 0}$, used to keep the edges at a constant length, reads

$$E_k(C) = \sum_{j=1}^{N_V} \tfrac{1}{k} |E_C^j - E_d^j|^k, \tag{8}$$

where E_C^j is the current edge length and E_d^j is the desired edge length of edge j.

Since each vertex $\vec{v}_j$ influences exactly the edge lengths of the edges e_j and e_{j-1}, we get the total edge force on $\vec{v}_j$ with: An isolated application of the edge force can be seen in Figure 10.

Figure 10: The top figures illustrate the evolution of a DF cell governed exclusively by the edge force, with $k = 2$ applied to the vertices and a force scaling of 3×10^4, at times $t \in \{0, 1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}\}$.
Accordingly, we have $\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = -3 \times 10^4 \nabla_{\vec{v}} E_2(C)$ for all vertices.
The initial edge length of the top edge is $E_C^2 = 1 \times 10^{-3}$, while the desired edge lengths are all set to 5×10^{-3}.
Click *here* to view the associated animation (GIF).
The edge force nearly restores the desired edge lengths after 5 time steps, which can also be observed in Figure 11.

Figure 11: This energy diagram shows that the edge force nearly restores the desired edge lengths after 5 time steps.

Proposition 4.5. ***Edge force***
The edge force $F_k^{(E)} : (\mathbb{R}^2)^{N_V} \to (\mathbb{R}^2)^{N_V}$ *that gets applied on cell* C *is given by*

$$F_k^{(E)}(C) = -(\nabla_{\vec{v}_1} E_k(C), \ldots, \nabla_{\vec{v}_{N_V}} E_k(C))^T,$$

where the gradient $\nabla_{\vec{v}_j} E_k(C)$ *with respect to* $\vec{v}_j = (v_j^x, v_j^y)^T$ *is given by*

$$\begin{aligned} \nabla_{\vec{v}_j} E_k(C) = \operatorname{sgn}(E_C^{j-1} - E_d^{j-1}) \frac{|E_C^{j-1} - E_d^{j-1}|^{k-1}}{E_C^{j-1}} \begin{pmatrix} v_j^x - v_{j-1}^x \\ v_j^y - v_{j-1}^y \end{pmatrix} \\ + \operatorname{sgn}(E_C^{j} - E_d^{j}) \frac{|E_C^{j} - E_d^{j}|^{k-1}}{E_C^{j}} \begin{pmatrix} v_j^x - v_{j+1}^x \\ v_j^y - v_{j+1}^y \end{pmatrix}, \end{aligned} \tag{9}$$

for all $1 \leqslant j \leqslant N_V$.

Proof.
We consider

$$\begin{aligned} \nabla_{\vec{v}_j} E_k(C) &= \nabla_{\vec{v}_j} \sum_{j=1}^{N_V} \tfrac{1}{k} |E_C^j - E_d^j|^k \\ &= \tfrac{1}{k} \nabla_{\vec{v}_j} |E_C^{j-1} - E_d^{j-1}|^k + \tfrac{1}{k} \nabla_{\vec{v}_j} |E_C^j - E_d^j|^k. \end{aligned}$$

For the first summand, we can compute

$$\begin{aligned}
\tfrac{1}{k}\nabla_{\vec{v}_j}|E_C^{j-1}-E_d^{j-1}|^k &= \operatorname{sgn}(E_C^{j-1}-E_d^{j-1})|E_C^{j-1}-E_d^{j-1}|^{k-1}\nabla_{\vec{v}_j}(E_C^{j-1}-E_d^{j-1})\\
&= \operatorname{sgn}(E_C^{j-1}-E_d^{j-1})|E_C^{j-1}-E_d^{j-1}|^{k-1}\nabla_{\vec{v}_j}E_C^{j-1}\\
&= \operatorname{sgn}(E_C^{j-1}-E_d^{j-1})|E_C^{j-1}-E_d^{j-1}|^{k-1}\cdot\\
&\qquad \nabla_{\vec{v}_j}\big[(v_{j-1}^x-v_j^x)^2+(v_{j-1}^y-v_j^y)^2\big]^{\frac{1}{2}}\\
&= \operatorname{sgn}(E_C^{j-1}-E_d^{j-1})|E_C^{j-1}-E_d^{j-1}|^{k-1}\cdot\\
&\qquad \left(\frac{1}{2E_C^{j-1}}\nabla_{\vec{v}_j}\big[(v_{j-1}^x-v_j^x)^2+(v_{j-1}^y-v_j^y)^2\big]\right)\\
&= \operatorname{sgn}(E_C^{j-1}-E_d^{j-1})|E_C^{j-1}-E_d^{j-1}|^{k-1}\cdot\\
&\qquad \nabla_{\vec{v}_j}\big[(v_{j-1}^x-v_j^x)^2+(v_{j-1}^y-v_j^y)^2\big]^{\frac{1}{2}}\\
&= \operatorname{sgn}(E_C^{j-1}-E_d^{j-1})|E_C^{j-1}-E_d^{j-1}|^{k-1}\cdot\\
&\qquad \left(\frac{1}{2E_C^{j-1}}\nabla_{\vec{v}_j}\big[(v_{j-1}^x-v_j^x)^2+(v_{j-1}^y-v_j^y)^2\big]\right)\\
&= \operatorname{sgn}(E_C^{j-1}-E_d^{j-1})\frac{|E_C^{j-1}-E_d^{j-1}|^{k-1}}{2E_C^{j-1}}\begin{pmatrix}\partial_{v_j^x}(v_{j-1}^x-v_j^x)^2\\ \partial_{v_j^y}(v_{j-1}^y-v_j^y)^2\end{pmatrix}\\
&= \operatorname{sgn}(E_C^{j-1}-E_d^{j-1})\frac{|E_C^{j-1}-E_d^{j-1}|^{k-1}}{2E_C^{j-1}}\begin{pmatrix}-2(v_{j-1}^x-v_j^x)\\ -2(v_{j-1}^y-v_j^y)\end{pmatrix}\\
&= \operatorname{sgn}(E_C^{j-1}-E_d^{j-1})\frac{|E_C^{j-1}-E_d^{j-1}|^{k-1}}{E_C^{j-1}}\begin{pmatrix}v_j^x-v_{j-1}^x\\ v_j^y-v_{j-1}^y\end{pmatrix}.
\end{aligned}$$

For the second summand, we get:

$$\begin{aligned}
\tfrac{1}{k}\nabla_{\vec{v}_j}|E_C^{j}-E_d^{j}|^k &= \operatorname{sgn}(E_C^{j}-E_d^{j})|E_C^{j}-E_d^{j}|^{k-1}\nabla_{\vec{v}_j}E_C^{j}\\
&= \operatorname{sgn}(E_C^{j}-E_d^{j})|E_C^{j}-E_d^{j}|^{k-1}\cdot\\
&\qquad \left(\frac{1}{2E_C^{j}}\nabla_{\vec{v}_j}\big[(v_j^x-v_{j+1}^x)^2+(v_j^y-v_{j+1}^y)^2\big]\right)\\
&= \operatorname{sgn}(E_C^{j}-E_d^{j})|E_C^{j}-E_d^{j}|^{k-1}\cdot\\
&\qquad \left(\frac{1}{2E_C^{j}}\nabla_{\vec{v}_j}\big[(v_j^x-v_{j+1}^x)^2+(v_j^y-v_{j+1}^y)^2\big]\right)\\
&= \operatorname{sgn}(E_C^{j}-E_d^{j})\frac{|E_C^{j}-E_d^{j}|^{k-1}}{2E_C^{j}}\begin{pmatrix}\partial_{v_j^x}(v_j^x-v_{j+1}^x)^2\\ \partial_{v_j^y}(v_j^y-v_{j+1}^y)^2\end{pmatrix}
\end{aligned}$$

$$= \operatorname{sgn}(E_C^j - E_d^j)\frac{|E_C^j - E_d^j|^{k-1}}{2E_C^j}\begin{pmatrix}2(v_j^x - v_{j+1}^x)\\ 2(v_j^y - v_{j+1}^y)\end{pmatrix}$$

$$= \operatorname{sgn}(E_C^j - E_d^j)\frac{|E_C^j - E_d^j|^{k-1}}{E_C^j}\begin{pmatrix}v_j^x - v_{j+1}^x\\ v_j^y - v_{j+1}^y\end{pmatrix}.$$

Equation (9) *arises when adding both summands.*

□

4.3 Interior angle force

The combined application of the area and edge forces revealed instabilities in unfavorable configurations, where self intersections of the cell edges occurred. Simulations without this energy sometimes can also result in constrictions at certain vertices, where the interior angle approaches 360°. To address this issue, we introduce the interior angle energy.
The first challenge is to consistently determine the interior angle at a given vertex throughout the simulation. Although we could apply the law of cosines and use arccos to compute the angle, this method would suffer from poor stability as the angle approaches 180°. A better alternative is to use the arctan2 function, as it remains reliably stable at all angles.

Definition 4.6. arctan2
The function

$$\operatorname{arctan2} : \mathbb{R}^2/\{0\} \to (-\pi, \pi],$$

is defined by:

$$\operatorname{arctan2}(\vec{v}) = \begin{cases} \arctan(\frac{v^y}{v^x}) & v^x > 0, \\ \arctan(\frac{v^y}{v^x}) + \pi & v^x < 0, v^y > 0, \\ \arctan(\frac{v^y}{v^x}) - \pi & v^x < 0, v^y < 0, \\ \pi & v^x < 0, v^y = 0, \\ \dfrac{\pi}{2} & v^x = 0, v^y > 0, \\ -\dfrac{\pi}{2} & v^x = 0, v^y < 0. \end{cases}$$

The arctan2($\vec{v}$) function computes the angle of a vector $\vec{v}$ with respect to the positive x-axis.
With this, we can compute the angles

$$\theta_1 = \operatorname{arctan2}(\vec{v}_{j-1} - \vec{v_j}),$$
$$\theta_2 = \operatorname{arctan2}(\vec{v}_{j+1} - \vec{v_j}),$$

between the positive x-axis and the vectors from $\vec{v}_j$ to its neighboring vertices $\vec{v}_{j-1}$ and $\vec{v}_{j+1}$. We get the searched angle at $\vec{v}_j$ by subtracting $\theta_1 - \theta_2$. To ensure that the angle lies within the interval $[0, 2\pi)$, we use the modulo operator $[\cdot]_{[0,2\pi)}$, which repeatedly adds or subtracts 2π from the angle until it falls within the desired range. Thus, our interior angle operator is:

$$I_C^j = [\text{arctan2}(\vec{v}_{j-1} - \vec{v_j}) - \text{arctan2}(\vec{v}_{j+1} - \vec{v_j})]_{[0,2\pi)}.$$

With that, we can define our interior angle energy.

Definition 4.7. Interior angle energy
The energy $I : (\mathbb{R}^2)^{N_V} \to \mathbb{R}_{\geqslant 0}$ associated with preserving the cell interior angles is given by

$$I_k(C) = \sum_{j=1}^{N_V} \tfrac{1}{k} |I_C^j - I_d^j|^k, \tag{10}$$

where I_d^j is the desired interior angle at vertex j and I_C^j is the current interior angle at vertex j of the considered cell.

We continue by computing the resulting force. The arctan2 function is partly defined and not truly differentiable. We still want to compute a gradient to use it for our interior angle force. It is

$$\text{arctan2}(\vec{v}) = \arctan\left(\frac{v^y}{v^x}\right) + \text{constant},$$

almost everywhere, just not on areas with measure zero. We just compute the gradient of $\arctan(\frac{v^y}{v^x})$ instead.
Another problem is the modulo operator $[\cdot]_{[0,2\pi)}$, which is not differentiable at the interval limits. However, we just neglect the modulo operator as it does not affect the dynamics of the gradient.

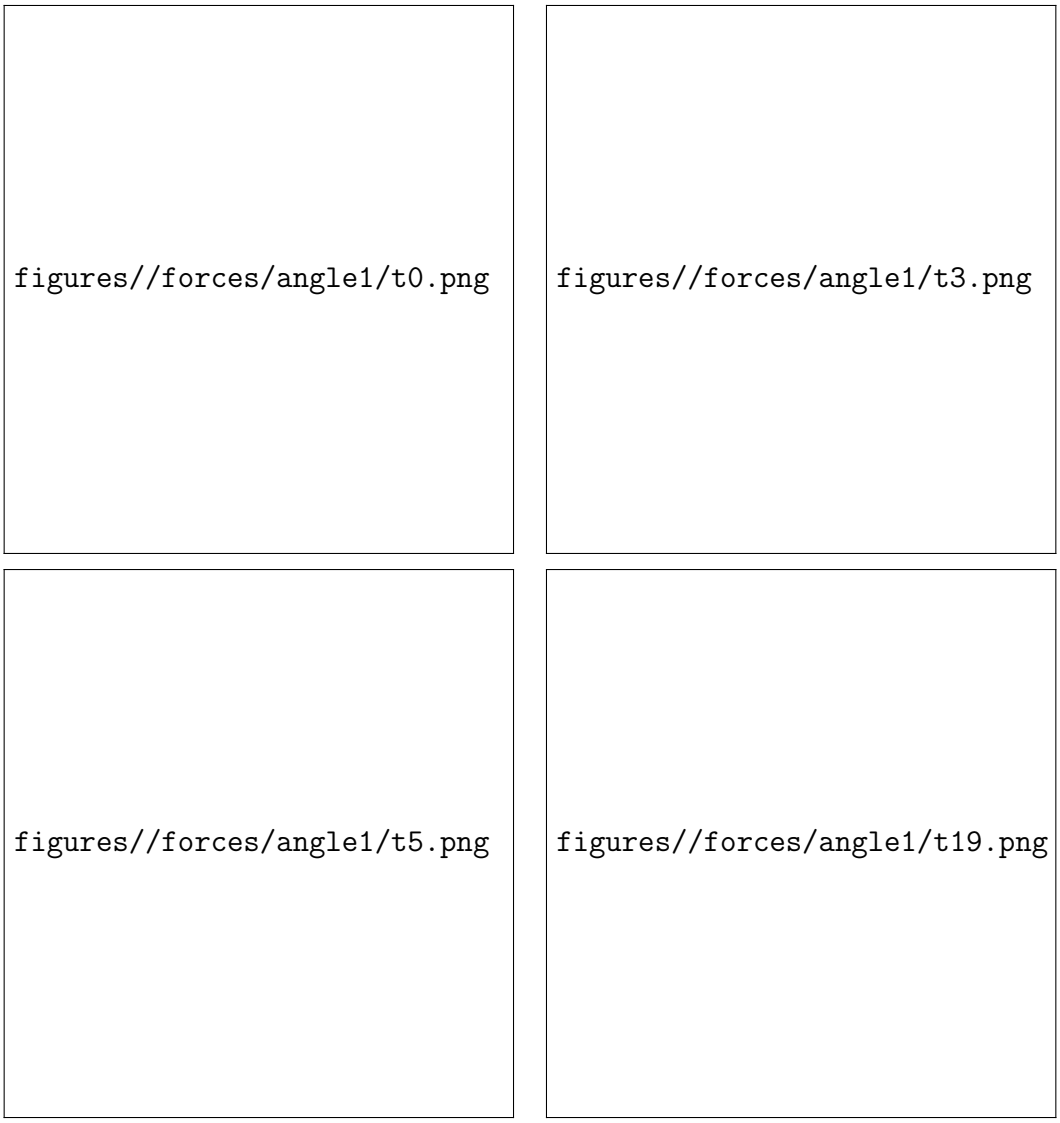


Figure 12: The top four plots depict the evolution of a DF cell subject only to the interior angle force with $k = 2$ applied to the vertices and a force scaling of 1×10^{-1}, at times $t \in \{0, 3 \times 10^{-5}, 5 \times 10^{-5}, 19 \times 10^{-5}\}$.
In this case, we have $\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = -1 \times 10^{-1} \nabla_{\vec{v}} I_2(C)$ for all vertices.
At the vertex with initial position $\vec{v}_1 = (0.003, 0.0)^T$, the starting interior angle is 306.9, while all desired interior angles are 120.
Click *here* to view the corresponding animation (GIF).
As seen in Figure 13, the interior angle force requires more time to reach the target configuration. This slower convergence is due to the reduced force scaling, which was necessary to avoid stability issues encountered at higher force scaling values.

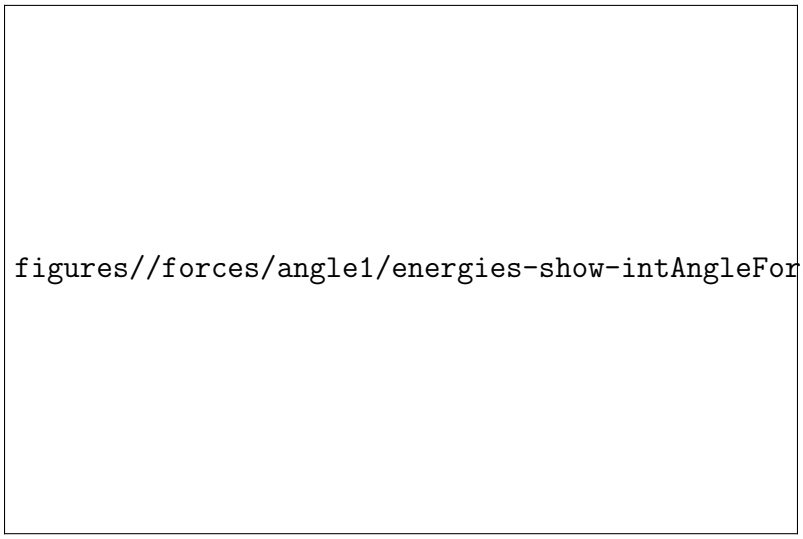


Figure 13: The interior angle force is able to minimize the interior angle energy over time.

Proposition 4.8. ***Interior angle force***

The interior angle force $F_k^{(I)} : (\mathbb{R}^2)^{N_V} \to (\mathbb{R}^2)^{N_V}$ *that gets applied on cell* C *is given by*

$$F_k^{(I)}(C) = -(\nabla_{\vec{v}_1} I_k(C), \ldots, \nabla_{\vec{v}_{N_V}} I_k(C))^T,$$

where the gradient $\nabla_{\vec{v}_j} I_k(C)$ *with respect to* $\vec{v}_j = (v_j^x, v_j^y)^T$ *is given by*

$$\begin{aligned}
\nabla_{\vec{v}_j} I_k(C) = {} & \operatorname{sgn}(I_C^{j-1} - I_d^{j-1}) |I_C^{j-1} - I_d^{j-1}|^{k-1} \left(-\frac{1}{\|\vec{v}_j - \vec{v}_{j-1}\|_2^2} \begin{pmatrix} v_{j-1}^y - v_j^y \\ v_j^x - v_{j-1}^x \end{pmatrix} \right) \\
& + \operatorname{sgn}(I_C^j - I_d^j) |I_C^j - I_d^j|^{k-1} \left(\frac{1}{\|\vec{v}_{j-1} - \vec{v}_j\|_2^2} \begin{pmatrix} v_{j-1}^y - v_j^y \\ v_j^x - v_{j-1}^x \end{pmatrix} \right. \\
& \left. - \frac{1}{\|\vec{v}_{j+1} - \vec{v}_j\|_2^2} \begin{pmatrix} v_{j+1}^y - v_j^y \\ v_j^x - v_{j+1}^x \end{pmatrix} \right) \\
& + \operatorname{sgn}(I_C^{j+1} - I_d^{j+1}) |I_C^{j+1} - I_d^{j+1}|^{k-1} \left(\frac{1}{\|\vec{v}_j - \vec{v}_{j+1}\|_2^2} \begin{pmatrix} v_{j+1}^y - v_j^y \\ v_j^x - v_{j+1}^x \end{pmatrix} \right),
\end{aligned} \tag{11}$$

for all $1 \leqslant j \leqslant N_V$.

Proof.
We are looking for

$$\nabla_{\vec{v}_j} I_k(C).$$

Vertex $\vec{v}_j$ impacts the interior angles at $\vec{v}_{j-1}$, $\vec{v}_j$ and $\vec{v}_{j+1}$. Thus, we get

$$\begin{aligned}\nabla_{\vec{v}_j} I_k(C) &= \nabla_{\vec{v}_j} \sum_{j=1}^{N_V} \tfrac{1}{k} |I_d^j - I_C^j|^k \\ &= \nabla_{\vec{v}_j} \tfrac{1}{k} |I_d^{j-1} - I_C^{j-1}|^k + \nabla_{\vec{v}_j} \tfrac{1}{k} |I_d^j - I_C^j|^k + \nabla_{\vec{v}_j} \tfrac{1}{k} |I_d^{j+1} - I_C^{j+1}|^k.\end{aligned}$$

First, we will focus on the computation

$$\begin{aligned}\nabla_{\vec{v}_j} \tfrac{1}{k} |I_d^j - I_C^j|^k &= (I_d^j - I_C^j) \nabla_{\vec{v}_j} (-I_C^j) \\ &= (I_C^j - I_d^j) \nabla_{\vec{v}_j} I_C^j \\ &= (I_C^j - I_d^j) \nabla_{\vec{v}_j} \big[\operatorname{arctan2}(\vec{v}_{j-1} - \vec{v_j}) - \operatorname{arctan2}(\vec{v}_{j+1} - \vec{v_j})\big]_{[0,2\pi)}.\end{aligned}$$

At this point, the previously mentioned simplifications come into play and we use $\arctan\left(\frac{v_{j-1}^y - v_j^y}{v_{j-1}^x - v_j^x}\right)$ *instead of* $\operatorname{arctan2}(\vec{v}_{j-1} - \vec{v_j})$ *and neglect the modulo operator. In the next step, we need to compute the gradient*

$$\nabla_{\vec{v}_j} \arctan\left(\frac{v_{j-1}^y - v_j^y}{v_{j-1}^x - v_j^x}\right).$$

Therefore, we define helper functions

$$f(\vec{v}) = \arctan\left(\frac{v^y}{v^x}\right),$$

and

$$g(\vec{v}_{j-1}, \vec{v}_j) = \begin{pmatrix} v_{j-1}^x - v_j^x \\ v_{j-1}^y - v_j^y \end{pmatrix}.$$

With these helper functions, we can write

$$\arctan\left(\frac{v_{j-1}^y - v_j^y}{v_{j-1}^x - v_j^x}\right) = (f \circ g)(\vec{v}_{j-1}, \vec{v}_j),$$

and use the two-dimensional chain rule to stepwise compute the searched gradient. We have

$$\frac{\partial f(\vec{v})}{\partial v^x} = \frac{1}{1 + \left(\frac{v^y}{v^x}\right)^2} \left(-\frac{v^y}{(v^x)^2}\right) = -\frac{v^y}{(v^x)^2 + (v^y)^2},$$

$$\frac{\partial f(\vec{v})}{\partial v^y} = \frac{1}{1 + \left(\frac{v^y}{v^x}\right)^2} \frac{1}{v^x} = \frac{v^x}{(v^x)^2 + (v^y)^2},$$

and

$$\frac{\partial g(\vec{v}_{j-1}, \vec{v}_j)}{\partial v_j^x} = (-1, 0)^T, \quad \frac{\partial g(\vec{v}_{j-1}, \vec{v}_j)}{\partial v_j^y} = (0, -1)^T.$$

With that, we can compute

$$\begin{aligned}\frac{\partial(f \circ g(\vec{v}_{j-1}, \vec{v}_j))}{\partial v_j^x} &= (\nabla f \circ g(\vec{v}_{j-1}, \vec{v}_j))^T \cdot \nabla_{v_j^x} g(\vec{v}_{j-1}, \vec{v}_j) \\ &= \begin{pmatrix} -\frac{v_{j-1}^y - v_j^y}{(v_{j-1}^x - v_j^x)^2 + (v_{j-1}^y - v_j^y)^2} \\ \frac{v_{j-1}^x - v_j^x}{(v_{j-1}^x - v_j^x)^2 + (v_{j-1}^y - v_j^y)^2} \end{pmatrix}^T \cdot \begin{pmatrix} -1 \\ 0 \end{pmatrix} \\ &= \frac{v_{j-1}^y - v_j^y}{(v_{j-1}^x - v_j^x)^2 + (v_{j-1}^y - v_j^y)^2} \\ &= \frac{v_{j-1}^y - v_j^y}{\|\vec{v}_{j-1} - \vec{v}_j\|_2^2},\end{aligned}$$

and similarly

$$\begin{aligned}\frac{\partial(f \circ g(\vec{v}_{j-1}, \vec{v}_j))}{\partial v_j^y} &= (\nabla f \circ g(\vec{v}_{j-1}, \vec{v}_j))^T \cdot \nabla_{v_j^y} g(\vec{v}_{j-1}, \vec{v}_j) \\ &= \begin{pmatrix} -\frac{v_{j-1}^y - v_j^y}{(v_{j-1}^x - v_j^x)^2 + (v_{j-1}^y - v_j^y)^2} \\ \frac{v_{j-1}^x - v_j^x}{(v_{j-1}^x - v_j^x)^2 + (v_{j-1}^y - v_j^y)^2} \end{pmatrix}^T \cdot \begin{pmatrix} 0 \\ -1 \end{pmatrix} \\ &= -\frac{v_{j-1}^x - v_j^x}{(v_{j-1}^x - v_j^x)^2 + (v_{j-1}^y - v_j^y)^2} \\ &= \frac{v_j^x - v_{j-1}^x}{\|\vec{v}_{j-1} - \vec{v}_j\|_2^2}.\end{aligned}$$

Overall, we get

$$\nabla_{\vec{v}_j} \arctan\left(\frac{v_{j-1}^y - v_j^y}{v_{j-1}^x - v_j^x}\right) = \frac{1}{\|\vec{v}_{j-1} - \vec{v}_j\|_2^2} \begin{pmatrix} v_{j-1}^y - v_j^y \\ v_j^x - v_{j-1}^x \end{pmatrix}.$$

Thus, we can come back to

$$\begin{aligned}\nabla_{\vec{v}_j} \tfrac{1}{k} |I_d^j - I_C^j|^k &= (I_C^j - I_d^j) \nabla_{\vec{v}_j} \left(\arctan\left(\frac{v_{j-1}^y - v_j^y}{v_{j-1}^x - v_j^x}\right) - \arctan\left(\frac{v_{j+1}^y - v_j^y}{v_{j+1}^x - v_j^x}\right) \right) \\ &= (I_C^j - I_d^j) \left(\frac{1}{\|\vec{v}_{j-1} - \vec{v}_j\|_2^2} \begin{pmatrix} v_{j-1}^y - v_j^y \\ v_j^x - v_{j-1}^x \end{pmatrix} \right. \\ &\quad \left. - \frac{1}{\|\vec{v}_{j+1} - \vec{v}_j\|_2^2} \begin{pmatrix} v_{j+1}^y - v_j^y \\ v_j^x - v_{j+1}^x \end{pmatrix} \right).\end{aligned}$$

For the neighboring vertices, we need

$$\nabla_{\vec{v}_j} \arctan\left(\frac{v_j^y - v_{j-1}^y}{v_j^x - v_{j-1}^x}\right), \quad \nabla_{\vec{v}_j} \arctan\left(\frac{v_j^y - v_{j+1}^y}{v_j^x - v_{j+1}^x}\right).$$

Therefore, we can use the same helper function

$$f(\vec{v}) = \arctan\left(\frac{v^y}{v^x}\right),$$

but we need different functions for g, as the arrangement of the vertex coordinates differs a bit. We introduce

$$g_-(\vec{v}_{j-1}, \vec{v}_j) = \begin{pmatrix} v_j^x - v_{j-1}^x \\ v_j^y - v_{j-1}^y \end{pmatrix} = -g(\vec{v}_{j-1}, \vec{v}_j).$$

and

$$g_+(\vec{v}_j, \vec{v}_{j+1}) = \begin{pmatrix} v_j^x - v_{j+1}^x \\ v_j^y - v_{j+1}^y \end{pmatrix}.$$

The gradients are

$$\nabla_{v_j^x} g_-(\vec{v}_{j-1}, \vec{v}_j) = (1,0)^T, \quad \nabla_{v_j^y} g_-(\vec{v}_{j-1}, \vec{v}_j) = (0,1)^T,$$

$$\nabla_{v_j^x} g_+(\vec{v}_{j-1}, \vec{v}_j) = (1,0)^T, \quad \nabla_{v_j^y} g_+(\vec{v}_{j-1}, \vec{v}_j) = (0,1)^T.$$

Thus, the Jacobian of both g_- and g_+ is the identity matrix and we can just neglect it in the following chain rules. For the previous vertex, we get

$$\begin{aligned}
&\nabla_{\vec{v}_j} \tfrac{1}{k} |I_C^{j-1} - I_d^{j-1}|^k = \\
&\quad = \operatorname{sgn}(I_C^{j-1} - I_d^{j-1}) |I_C^{j-1} - I_d^{j-1}|^{k-1} \\
&\qquad \nabla_{\vec{v}_j} \left(\arctan\left(\frac{v_{j-2}^y - v_{j-1}^y}{v_{j-2}^x - v_{j-1}^x}\right) - \arctan\left(\frac{v_j^y - v_{j-1}^y}{v_j^x - v_{j-1}^x}\right)\right) \\
&\quad = \operatorname{sgn}(I_C^{j-1} - I_d^{j-1}) |I_C^{j-1} - I_d^{j-1}|^{k-1} \nabla_{\vec{v}_j} \left(- \arctan\left(\frac{v_j^y - v_{j-1}^y}{v_j^x - v_{j-1}^x}\right)\right) \\
&\quad = \operatorname{sgn}(I_C^{j-1} - I_d^{j-1}) |I_C^{j-1} - I_d^{j-1}|^{k-1} (-\nabla_{\vec{v}_j} (f \circ g_-(\vec{v}_{j-1}, \vec{v}_{j-1}))) \\
&\quad = \operatorname{sgn}(I_C^{j-1} - I_d^{j-1}) |I_C^{j-1} - I_d^{j-1}|^{k-1} (-(\nabla f) \circ g_-(\vec{v}_{j-1}, \vec{v}_{j-1})) \\
&\quad = \operatorname{sgn}(I_C^{j-1} - I_d^{j-1}) |I_C^{j-1} - I_d^{j-1}|^{k-1} \left(-\frac{1}{\|\vec{v}_j - \vec{v}_{j-1}\|_2^2} (v_{j-1}^y - v_j^y, v_j^x - v_{j-1}^x)^T \right).
\end{aligned}$$

Finally, for the successor vertex, we get

$$
\begin{aligned}
&\nabla_{\vec{v}_j} \tfrac{1}{k} |I_C^{j+1} - I_d^{j+1}|^k = \\
&\quad = \operatorname{sgn}(I_C^{j+1} - I_d^{j+1}) |I_C^{j+1} - I_d^{j+1}|^{k-1} \\
&\qquad \nabla_{\vec{v}_j} \left(\arctan\left(\frac{v_j^y - v_{j+1}^y}{v_j^x - v_{j+1}^x} \right) - \arctan\left(\frac{v_{j+2}^y - v_{j+1}^y}{v_{j+2}^x - v_{j+1}^x} \right) \right) \\
&\quad = \operatorname{sgn}(I_C^{j+1} - I_d^{j+1}) |I_C^{j+1} - I_d^{j+1}|^{k-1} \nabla_{\vec{v}_j} \left(\arctan\left(\frac{v_j^y - v_{j+1}^y}{v_j^x - v_{j+1}^x} \right) \right) \\
&\quad = \operatorname{sgn}(I_C^{j+1} - I_d^{j+1}) |I_C^{j+1} - I_d^{j+1}|^{k-1} \nabla_{\vec{v}_j} \left(f \circ g_+(\vec{v}_{j-1}, \vec{v}_{j-1}) \right) \\
&\quad = \operatorname{sgn}(I_C^{j+1} - I_d^{j+1}) |I_C^{j+1} - I_d^{j+1}|^{k-1} (\nabla f) \circ g_+(\vec{v}_{j-1}, \vec{v}_{j-1}) \\
&\quad = \operatorname{sgn}(I_C^{j+1} - I_d^{j+1}) |I_C^{j+1} - I_d^{j+1}|^{k-1} \left(\frac{1}{\|\vec{v}_j - \vec{v}_{j+1}\|_2^2} (v_{j+1}^y - v_j^y, v_j^x - v_{j+1}^x)^T \right).
\end{aligned}
$$

Equation (11) *now follows by summing up the three parts.*

□

Figure 12 illustrates the isolated effect of the interior angle force.

4.4 Overlap force

Unlike the previous energies, which act independently on each cell, the overlap force is the first to account for interactions between multiple cells, thereby introducing cell-to-cell interaction into the simulation.

Deforming overlap force

The first overlap force, that we want to introduce, is an adapted form of the overlap force introduced in [Vog23]. It degenerates a cell overlap by influencing that cell shapes of the affected cells.
The challenging aspect of computing the overlap force lies in detecting overlaps within the cell system.

Definition 4.9. Overlap cell
An overlap cell between two DF cells C_i and C_m is a DF cell in the sence of Definition 3.2, composed of all vertices of C_i that lie inside C_m, all vertices of C_m that lie inside C_i and the intersection points of the cell walls of C_i and C_m.

Each pair of cells can have more than one overlap cell in unfavourable configurations. To be able to catch the correct dynamic in all cases, we need to introduce the set of all overlaps between a pair of DF cells.

Definition 4.10. Set of overlaps Ω_{C_i,C_m}
Let $C_i, C_m \in \vec{C}$ be two DF cells. Then, the set of overlaps Ω_{C_i,C_m} is defined as

$$\Omega_{C_i,C_m} = \{\, D \mid D \text{ is an overlap cell formed between } C_i \text{ and } C_m \,\}.$$

Once all overlaps have been identified, we apply a dynamic similar to that of the area force, but with a desired area of zero. This generates a force that acts to eliminate the overlap by reducing its area to zero. The resulting force is then applied to the vertices of the original cells that define the overlapping region.
The first step in detecting overlaps is identifying the intersection points between cell boundaries. Intersections can be identified by representing the cell edges as line segments and computing the intersection points between segments belonging to different cells.
Having found all intersections, we can apply the following algorithm, that can be used to compute all overlaps between two cells.

Algorithm 4.11. ***Computation of a discrete overlap***

INPUT:

- *Discrete cells C and ζ*
- *List I of unused intersections of C and ζ*

```
function constructOverlap(C, ζ, I)
    usedIntersections = List{Intersection}(I[1])
    newOverlap = List{Vertices}(I[1])
    currentIntersection = I[1]
    for counter = 1 : length(I) do
        if counter is even then
            newPath, newIntersection = findPath(currentIntersection, C, I)
        else
            newPath, newIntersection = findPath(currentIntersection, ζ, I)
        end if
        append!(newOverlap, newPath)
        if newIntersection == I[1] then
            return newOverlap, usedIntersections
        else
            append!(newOverlap, newIntersection)
            append!(usedIntersections, newIntersection)
            currentIntersection = newIntersection
        end if
    end for
end function
```

OUTPUT:

- *A single intersection 'newOverlap' which occurs between C and ζ and which uses vertices from C and ζ as well as only intersections from I*
- *A list 'usedIntersections' of all intersection that are used in 'newOverlap'*

The algorithm begins by selecting the first intersection point $I[1]$ from the list I as the initial vertex of the overlap cell 'newOverlap'. This point is also added to the list 'usedIntersections'

Next, the function 'getOverlap' calls another function, 'findPath', which determines the path along the discrete cell ζ from the current intersection point to the next intersection in I encountered while traversing the edges of ζ. This next intersection is also returned by the function. The identified path is a list of vertices in ζ that lie strictly between the two intersections. It may be empty if the next intersection occurs on the same edge as the current one. Both the path and the newly found intersection are appended to 'newOverlap', and the intersection is also added to the list usedIntersections.
Since each intersection implies changing the cell from which the overlapping cell uses the edges, 'findPath' is now applied to the other cell. Again, it will deliver the next intersection as well as a list of the in between laying vertices. The vertex list always gets appended to 'newOverlap'.
If the newly found intersection is equal to the initial intersection $I[1]$, then the construction of the discrete overlap cell 'newOverlap' is complete. At this point, both 'newOverlap' and 'usedIntersections' can be returned by the function 'constructOverlap'.
Otherwise, the newly found intersection is appended to both 'newOverlap' and 'usedIntersections', and the process continues by calling 'findPath' on the other discrete cell. This step is repeated until the starting intersection is reached, completing the overlap cell construction.
Once an overlap between C and ζ has been successfully extracted, all intersections used in its construction can be removed from the list I, since each intersection point belongs to exactly one overlap. As long as I is not empty, the function 'constructOverlap' can be called again with the updated list to extract the next overlap. When I is empty, we can be certain that all intersections between C and ζ have been processed, and thus all overlaps between the two cells have been identified.

Each time 'findPath' is called, it is not immediately clear in which direction the function should traverse the vertices of the given cell. However, the correct direction can be determined using the following approach.
Starting from the current intersection passed into the function, move a small distance in one direction along the edge of the given cell where the intersection is located. Next, check whether this new point lies within the boundaries of the other cell as well. If the point is found in both cells, the chosen direction is correct. If not, then the opposite direction must be used.
A simple method to determine whether a point lies inside a polygon is to draw a ray from the point to the outside of the polygon. The number of intersections between the ray and the polygon's edges determines the point's position. If the number of intersections is odd, the point is inside the polygon. If it is even, the point is outside the polygon.

After introducing the method for detecting overlaps, we can now define the overlap force, which acts on the cell vertices involved in an overlap. This force is first computed based on the geometry of the overlap and then distributed to the corresponding vertices of the original cells.

Definition 4.12. Overlap energy
Let C_i and C_m be two cells from the system $\vec{C}$ and Ω_{C_i,C_m} be the set of all overlaps that appear between C_i and C_m, like explained above. Then, the total overlap energy $O_k : (\mathbb{R}^{2N_V})^{N_C} \to \mathbb{R}$ of the cell system is given by the formula

$$O_k(\vec{C}) = \sum_{i=1}^{N_C} \sum_{m=i+1}^{N_C} \sum_{D \in \Omega_{C_i,C_m}} \tfrac{1}{k}|A_D|^k, \tag{12}$$

where A_D is the area of the overlap D.
We define the inner bracket to be the overlap energy of the cell pair C_i and C_m

$$O_k^{i,m} : (\mathbb{R}^{2N_V})^2 \longrightarrow \mathbb{R},$$

$$(C_i, C_m) \longmapsto O_k^{i,m}(C_i, C_m) = \sum_{D \in \Omega_{C_i,C_m}} \tfrac{1}{k}|A_D|^k.$$

To decrease the overlap areas during the simulation, we evaluate the gradient flow of the area energy with a desired area of zero which indicates the direction of motion for each vertex for reducing the overlap areas.

Definition 4.13. Intersection point and adjacent vertices
The vertices of an overlap cell D can be divided into the vertices that are either in C_i in C_m, we call that set

$$V(D) = \{\vec{v} \in D \mid \vec{v} \in C_i \cup C_m\},$$

and into the vertices that are neither in C_i nor C_m, named

$$W(D) = D \backslash V(D).$$

All overlap vertices in $W(D)$ are intersections between the cells C_i and C_m.
Each intersection $\vec{w}$ is dependent on two vertices of each cell that limit the edges that intersect, where the intersection point $\vec{w}$ arises.
We call those four vertices **adjacent** to the intersection $\vec{w}$. All intersection adjacent vertices will get an extra deforming overlap dynamic applied, since they influence the overlap area, and thus the overlap energy, via the intersection point they create. From each cell we get one vertex that is part of the overlap cell, called the inside vertex, and one vertex that is not part of the overlap, called the outside vertex. For each intersection $\vec{w} \in W(D)$, we call its adjacent vertices

$$\mathrm{adj}(\vec{w}) = \{\vec{v}^{\,i}_{\mathrm{in}}, \vec{v}^{\,i}_{\mathrm{out}}, \vec{v}^{\,m}_{\mathrm{in}}, \vec{v}^{\,m}_{\mathrm{out}}\}.$$

In order to refer to the inside or outside vertices, we define the sets

$$\mathrm{in}(\vec{w}) = \{\vec{v}^{\,i}_{\mathrm{in}}, \vec{v}^{\,m}_{\mathrm{in}}\}, \quad \mathrm{out}(\vec{w}) = \{\vec{v}^{\,i}_{\mathrm{out}}, \vec{v}^{\,m}_{\mathrm{out}}\}.$$

Figure 14 illustrates what the in- and outside vertices of an intersection are.
Given the four adjacent vertices, we can always compute the intersection with the function:

$$w : (\mathbb{R}^2)^4 \longrightarrow \mathbb{R}^2,$$

$$(\vec{v}^{\,i}_{\mathrm{out}}, \vec{v}^{\,i}_{\mathrm{in}}, \vec{v}^{\,m}_{\mathrm{out}}, \vec{v}^{\,m}_{\mathrm{in}}) \longmapsto \vec{v}^{\,i}_{\mathrm{out}} + \frac{(\vec{v}^{\,m}_{\mathrm{out}} - \vec{v}^{\,i}_{\mathrm{out}}) \times (\vec{v}^{\,m}_{\mathrm{in}} - \vec{v}^{\,m}_{\mathrm{out}})}{(\vec{v}^{\,i}_{\mathrm{in}} - \vec{v}^{\,i}_{\mathrm{out}}) \times (\vec{v}^{\,m}_{\mathrm{in}} - \vec{v}^{\,m}_{\mathrm{out}})} (\vec{v}^{\,i}_{\mathrm{in}} - \vec{v}^{\,i}_{\mathrm{out}}),$$

where $(a^x, a^y)^T \times (b^x, b^y)^T = a^x b^y - a^y b^x$ denotes the two-dimensional cross product.

Figure 14: Here, we can see a DF cell setup with two cells having an overlap. The intersection points $\vec{w}_1$ and $\vec{w}_2$ are marked green. We can also see all inside vertices colored in black and the outside vertices colored in white. In this example, we have $\text{adj}(\vec{w}_1) = \{\vec{v}_1^1, \vec{v}_2^1, \vec{v}_2^2, \vec{v}_3^2\}$, $\text{in}(\vec{w}_1) = \{\vec{v}_1^1, \vec{v}_3^2\}$ and $\text{out}(\vec{w}_1) = \{\vec{v}_2^1, \vec{v}_2^2\}$ for the first intersection and $\text{adj}(\vec{w}_2) = \{\vec{v}_1^1, \vec{v}_6^1, \vec{v}_4^2, \vec{v}_5^2\}$, $\text{in}(\vec{w}_2) = \{\vec{v}_1^1, \vec{v}_4^2\}$ and $\text{out}(\vec{w}_2) = \{\vec{v}_6^1, \vec{v}_5^2\}$ for the second intersection.

Proposition 4.14. ***Partial derivatives of intersection points***
We use $w(\vec{v}_{out}^i, \vec{v}_{in}^i, \vec{v}_{out}^m, \vec{v}_{in}^m)$ to compute the influence of the adjacent vertices to the intersection point via their partial derivatives. In this proposition, we compute the needed partial derivatives.
We define the following auxiliary terms:

$$f = (\vec{v}_{out}^m - \vec{v}_{out}^i) \times (\vec{v}_{in}^m - \vec{v}_{out}^m), \quad g = (\vec{v}_{in}^i - \vec{v}_{out}^i) \times (\vec{v}_{in}^m - \vec{v}_{out}^m),$$

$$t = \frac{f}{g}, \quad w = \vec{v}_{out}^i + t(\vec{v}_{in}^i - \vec{v}_{out}^i).$$

It is sufficient to just compute the partial derivatives with respect to $\vec{v}_{in}^i$ and $\vec{v}_{out}^i$. If we want the dynamic for the vertices of the other cell, we just switch the arrangement of the arguments (switch $\vec{v}_{in}^i$ with $\vec{v}_{in}^j$ and $\vec{v}_{out}^i$ with $\vec{v}_{out}^j$) and then use the same partial derivatives as for the first cell.

The partial derivatives are:

$$D_{\vec{v}^i_{out}} w(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in}) : (\mathbb{R}^2)^4 \longrightarrow \mathbb{R}^{2\times 2},$$

$$D_{\vec{v}^i_{out}} w(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in}) = (1-t)I_2 + \frac{g-f}{g^2}(\vec{v}^i_{in} - \vec{v}^i_{out}) \begin{pmatrix} -(v^{m,y}_{in} - v^{m,y}_{out}) \\ v^{m,x}_{in} - v^{m,x}_{out} \end{pmatrix}^T, \tag{13}$$

$$D_{\vec{v}^i_{in}} w(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in}) : (\mathbb{R}^2)^4 \longrightarrow \mathbb{R}^{2\times 2},$$

$$D_{\vec{v}^i_{in}} w(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in}) = tI_2 \; + \; \frac{f}{g^2}(\vec{v}^i_{in} - \vec{v}^i_{out}) \begin{pmatrix} -(v^{m,y}_{in} - v^{m,y}_{out}) \\ v^{m,x}_{in} - v^{m,x}_{out} \end{pmatrix}^T. \tag{14}$$

Proof.

First of all, we compute the gradients of f and g

$$\begin{aligned} \nabla_{\vec{v}^i_{out}} f &= \nabla_{\vec{v}^i_{out}} [(v^{m,x}_{out} - v^{i,x}_{out})(v^{m,y}_{in} - v^{m,y}_{out}) - (v^{m,y}_{out} - v^{i,y}_{out})(v^{m,x}_{in} - v^{m,x}_{out})] \\ &= \begin{pmatrix} -(v^{m,y}_{in} - v^{m,y}_{out}) \\ v^{m,x}_{in} - v^{m,x}_{out} \end{pmatrix}, \end{aligned}$$

$$\begin{aligned} \nabla_{\vec{v}^i_{in}} f &= \nabla_{\vec{v}^i_{in}} [(v^{m,x}_{out} - v^{i,x}_{out})(v^{m,y}_{in} - v^{m,y}_{out}) - (v^{m,y}_{out} - v^{i,y}_{out})(v^{m,x}_{in} - v^{m,x}_{out})] \\ &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \end{aligned}$$

$$\begin{aligned} \nabla_{\vec{v}^i_{out}} g &= \nabla_{\vec{v}^i_{out}} [(v^{i,x}_{in} - v^{i,x}_{out})(v^{m,y}_{in} - v^{m,y}_{out}) - (v^{i,y}_{in} - v^{i,y}_{out})(v^{m,x}_{in} - v^{m,x}_{out})] \\ &= \begin{pmatrix} -(v^{m,y}_{in} - v^{m,y}_{out}) \\ v^{m,x}_{in} - v^{m,x}_{out} \end{pmatrix}, \end{aligned}$$

$$\begin{aligned} \nabla_{\vec{v}^i_{in}} g &= \nabla_{\vec{v}^i_{in}} [(v^{i,x}_{in} - v^{i,x}_{out})(v^{m,y}_{in} - v^{m,y}_{out}) - (v^{i,y}_{in} - v^{i,y}_{out})(v^{m,x}_{in} - v^{m,x}_{out})] \\ &= \begin{pmatrix} v^{m,y}_{in} - v^{m,y}_{out} \\ -(v^{m,x}_{in} - v^{m,x}_{out}) \end{pmatrix}. \end{aligned}$$

Now, we can succeed with the gradients of t

$$\begin{aligned}
\nabla_{\vec{v}_{out}^{\,i}} t &= \nabla_{\vec{v}_{out}^{\,i}} \frac{f}{g} \\
&= \frac{(\nabla_{\vec{v}_{out}^{\,i}} f)g - (\nabla_{\vec{v}_{out}^{\,i}} g)f}{g^2} \\
&= \frac{1}{g^2}\left(\begin{pmatrix} -(v_{in}^{m,y} - v_{out}^{m,y}) \\ v_{in}^{m,x} - v_{out}^{m,x} \end{pmatrix} g - \begin{pmatrix} -(v_{in}^{m,y} - v_{out}^{m,y}) \\ v_{in}^{m,x} - v_{out}^{m,x} \end{pmatrix} f\right) \\
&= \frac{g-f}{g^2}\begin{pmatrix} -(v_{in}^{m,y} - v_{out}^{m,y}) \\ v_{in}^{m,x} - v_{out}^{m,x} \end{pmatrix},
\end{aligned}$$

$$\begin{aligned}
\nabla_{\vec{v}_{in}^{\,i}} t &= \nabla_{\vec{v}_{in}^{\,i}} \frac{f}{g} \\
&= \frac{(\nabla_{\vec{v}_{in}^{\,i}} f)g - (\nabla_{\vec{v}_{in}^{\,i}} g)f}{g^2} \\
&= \frac{1}{g^2}\left(-\begin{pmatrix} v_{in}^{m,y} - v_{out}^{m,y} \\ -(v_{in}^{m,x} - v_{out}^{m,x}) \end{pmatrix} f\right) \\
&= \frac{f}{g^2}\begin{pmatrix} -(v_{in}^{m,y} - v_{out}^{m,y}) \\ v_{in}^{m,x} - v_{out}^{m,x} \end{pmatrix}.
\end{aligned}$$

And finally, we can compute the partial derivatives of $w = (w_1, w_2)^T$

$$\frac{\partial w_1}{\partial v_{out}^{i,x}} = 1 + \frac{\partial t}{\partial v_{out}^{i,x}}(v_{in}^{i,x} - v_{out}^{i,x}) - t, \quad \frac{\partial w_1}{\partial v_{out}^{i,y}} = 0 + \frac{\partial t}{\partial v_{out}^{i,y}}(v_{in}^{i,x} - v_{out}^{i,x}) + 0,$$

$$\frac{\partial w_2}{\partial v_{out}^{i,x}} = 0 + \frac{\partial t}{\partial v_{out}^{i,x}}(v_{in}^{i,y} - v_{out}^{i,y}) + 0, \quad \frac{\partial w_2}{\partial v_{out}^{i,y}} = 1 + \frac{\partial t}{\partial v_{out}^{i,y}}(v_{in}^{i,y} - v_{out}^{i,y}) - t,$$

$$\begin{aligned}
\implies D_{\vec{v}_{out}^{\,i}} w &= (1-t)I_2 + (\vec{v}_{in}^{\,i} - \vec{v}_{out}^{\,i})(\nabla_{\vec{v}_{out}^{\,i}} t)^T \\
&= (1-t)I_2 + (\vec{v}_{in}^{\,i} - \vec{v}_{out}^{\,i})\left(\frac{g-f}{g^2}\begin{pmatrix} -(v_{in}^{m,y} - v_{out}^{m,y}) \\ v_{in}^{m,x} - v_{out}^{m,x} \end{pmatrix}\right)^T \\
&= (1-t)I_2 + \frac{g-f}{g^2}(\vec{v}_{in}^{\,i} - \vec{v}_{out}^{\,i})\begin{pmatrix} -(v_{in}^{m,y} - v_{out}^{m,y}) \\ v_{in}^{m,x} - v_{out}^{m,x} \end{pmatrix}^T,
\end{aligned}$$

$$\frac{\partial w_1}{\partial v_{in}^{i,x}} = \frac{\partial t}{\partial v_{in}^{i,x}}(v_{in}^{i,x} - v_{out}^{i,x}) + t, \quad \frac{\partial w_1}{\partial v_{in}^{i,y}} = \frac{\partial t}{\partial v_{in}^{i,y}}(v_{in}^{i,x} - v_{out}^{i,x}) + 0,$$
$$\frac{\partial w_2}{\partial v_{in}^{i,x}} = \frac{\partial t}{\partial v_{in}^{i,x}}(v_{in}^{i,y} - v_{out}^{i,y}) + 0, \quad \frac{\partial w_2}{\partial v_{in}^{i,y}} = \frac{\partial t}{\partial v_{in}^{i,y}}(v_{in}^{i,y} - v_{out}^{i,y}) + t,$$

$$\begin{aligned}
\implies D_{\vec{v}_{in}^{\,i}} w &= tI_2 + (\vec{v}_{in}^{\,i} - \vec{v}_{out}^{\,i})(\nabla_{\vec{v}_{in}^{\,i}} t)^T \\
&= tI_2 + (\vec{v}_{in}^{\,i} - \vec{v}_{out}^{\,i}) \left(\frac{f}{g^2} \begin{pmatrix} -(v_{in}^{m,y} - v_{out}^{m,y}) \\ v_{in}^{m,x} - v_{out}^{m,x} \end{pmatrix} \right)^T \\
&= \; tI_2 \; + \; \frac{f}{g^2}(\vec{v}_{in}^{\,i} - \vec{v}_{out}^{\,i}) \begin{pmatrix} -(v_{in}^{m,y} - v_{out}^{m,y}) \\ v_{in}^{m,x} - v_{out}^{m,x} \end{pmatrix}^T .
\end{aligned}$$

Proposition 4.15. ***Deforming overlap force***
Each overlap cell $D \in \Omega_{C_i,C_m}$ is a list of vertices, that form the overlap, just like a DF cell.
The deforming overlap gradient is then given by

$$\begin{aligned}
\nabla_{\vec{v}_j^{\,i}} O_k(\vec{C}) = \sum_{D \in \Omega_{C_i,C_m}} |A_D|^{k-1} \Bigg(& \mathbf{1}_{V(D)}(\vec{v}_j^{\,i}) \nabla_{\vec{v}_j^{\,i}} A(D) + \\
& + \sum_{\vec{w} \in W(D)} \left(\mathbf{1}_{out(w)}(\vec{v}_j^{\,i}) D_{\vec{v}_{out}^{\,i}} \vec{w} + \mathbf{1}_{in(w)}(\vec{v}_j^{\,i}) D_{\vec{v}_{in}^{\,i}} \vec{w} \right) \nabla_{\vec{w}} A(D) \Bigg),
\end{aligned} \tag{15}$$

for all $1 \leqslant j \leqslant N_V$ and $1 \leqslant i \leqslant N_C$, where $out(w)$, $in(w) \subset adj(w)$ denote the sets of outside and inside overlap-adjacent vertices, respectively.
Note, that the Formulas 13 and 14 define $\frac{\partial \vec{w}}{\partial \vec{v}_{j,out}^{\,i}}$ and $\frac{\partial \vec{w}}{\partial \vec{v}_{j,in}^{\,i}}$, respectively.
The difference between $\nabla_{\vec{v}_j^{\,i}} A(D)$ and $\nabla_{\vec{w}} A(D)$ is that $\nabla_{\vec{v}_j^{\,i}} A(D)$ uses the neighboring overlap vertices of $\vec{v}_j^{\,i}$ itself in the Area Gradient Formula 7, whereas $\nabla_{\vec{w}} A(D)$ uses the neighbors of the corresponding intersection point in D, which is not the same overlap vertex as $\vec{v}_j^{\,i}$. A_D is the area of the overlap D.
The indicator function $\mathbf{1}_A(\vec{v})$ equals one, if $\vec{v} \in A$ and is zero otherwise.
The deforming overlap force $F_k^{(\hat{O})} : (\mathbb{R}^{2N_V})^{N_C} \to (\mathbb{R}^{2N_V})^{N_C}$ that gets applied on the whole cell system $\vec{C} = (C_1, \ldots, C_{N_C})$ is then given by

$$F_k^{(\hat{O})}(\vec{C}) = -(\nabla_{\vec{v}_1^{\,1}} O_k(\vec{C}), \ldots, \nabla_{\vec{v}_{N_V}^{\,1}} O_k(\vec{C}), \; \cdots, \vec{v}_1^{\,N_C} O_k(\vec{C}), \ldots, \nabla_{\vec{v}_{N_V}^{\,N_C}} O_k(\vec{C}))^T.$$

For addressing the overlap force that acts on cell $1 \leqslant i \leqslant N_C$, we define

$$F_{k,i}^{(\hat{O})} : (\mathbb{R}^{2N_V})^{N_C} \to (\mathbb{R}^{2N_V}),$$
$$F_{k,i}^{(\hat{O})}(\vec{C}) = -(\nabla_{\vec{v}_1^i} O_k(\vec{C}), \ldots, \nabla_{\vec{v}_{N_V}^i} O_k(\vec{C}))^T.$$

figures//forces/defOverlap1/t0.png figures//forces/defOverlap1/t1.png

figures//forces/defOverlap1/t2.png figures//forces/defOverlap1/t3.png

Figure 15: The top four plots show the evolution of a DF cell influenced solely by the deforming overlap force, with $k = 1$ applied to the vertices and a force scaling of 6×10^4, at times $t \in \{0, 1 \times 10^{-5}, 2 \times 10^{-5}, 3 \times 10^{-5}\}$.
In this case, we have $\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = -6 \times 10^4 \nabla_{\vec{v}} \hat{O}_1(C_1, C_2)$ for all vertices. The overlap area D is indicated below each plot. Click *here* to view the corresponding animation (GIF). Initially, the overlap area is 3×10^{-5}, which is relatively large compared to the cell area of 6.5×10^{-5}. However, it is completely resolved after just two time steps, as also illustrated in the energy diagram in Figure 16.

Figure 16: The deforming overlap force resolves the overlap in two time steps.

Proof of Proposition 4.15.
Although we did not noted it like this before, we must be aware that A_D is actually dependent on all overlap vertices, e.g. $A_D = A(D)$. We will also use that notation in the coming computation. Since the area $A(D)$ is always positive, we can drop the absolute value.
We aim to compute

$$\begin{aligned}
\nabla_{\vec{v}_j^i} O_k(\vec{C}) &= \nabla_{\vec{v}_j^i} \sum_{i=1}^{N_C} \left(\sum_{m=i+1}^{N_C} \left(\sum_{D \in \Omega_{C_i,C_m}} \tfrac{1}{k} A_D^k \right) \right) \\
&= \nabla_{\vec{v}_j^i} \sum_{m \neq i} O_k^{i,m}(C_i, C_m) \\
&= \nabla_{\vec{v}_j^i} \sum_{m \neq i} \sum_{D \in \Omega_{C_i,C_m}} \tfrac{1}{k} A(D)^k \\
&= \sum_{m \neq i} \sum_{D \in \Omega_{C_i,C_m}} \nabla_{\vec{v}_j^i} \tfrac{1}{k} A(D)^k.
\end{aligned}$$

Now, there are different cases.
Case 1: *$\vec{v}_j^i \notin D$ and $\vec{v}_j^i \notin adj(\vec{w})\ \forall \vec{w} \in W(D)$*
In the first case, the considered vertex is neither a vertex from the overlap cell D, nor adjacent to any intersection point.
Hence, this vertex has zero impact on the overlap and its gradient is zero, i.e.

$$\nabla_{\vec{v}_j^i} \tfrac{1}{k} A_D^k = 0.$$

Case 2: *$\vec{v}_j^i \notin D$ and $\exists \vec{w} \in W(D) :\ \vec{v}_j^i \in adj(\vec{w})$*
For case 2, we consider a vertex that is not directly an overlap vertex, but it influences the overlap cell by influencing an intersection point $\vec{w}$. This means that $\vec{v}_j^i$ is

an outside adjacent vertex of the intersection $\vec{w}$. We compute

$$\begin{aligned}\nabla_{\vec{v}_j^i}\tfrac{1}{k}A(D)^k &= A_D^{k-1}\nabla_{\vec{v}_j^i}A(D)\\ &= A_D^{k-1}\sum_{\vec{w}\in W(D)}\mathbf{1}_{out(w)}(\vec{v}_j^i)D_{\vec{v}_{out}^i}\vec{w}\nabla_{\vec{w}}A(D),\end{aligned}$$

where, according to Equation (13)

$$D_{\vec{v}_{out}^i}\vec{w} = (1-t)I_2 + \frac{g-f}{g^2}(\vec{v}_{in}^i - \vec{v}_j^i)\begin{pmatrix}-(v_{in}^{i,y}-v_j^{i,y})\\ v_{in}^{i,x}-v_j^{i,x}\end{pmatrix}^T,$$

because $\vec{v}_j^i$ is an outside vertex in this case and $\vec{v}_{in}^i$ is the vertex adjacent to $\vec{v}_j^i$ in cell i, such that these vertices build the edge causing the intersection.
The gradient $\nabla_{\vec{w}}A(D)$ can easily be computed via the Shoelace Formula 4.1, as in the area gradient from Formula 7:

$$\nabla_{\vec{w}}A(D) = \frac{1}{2}\begin{pmatrix}d_{j+1}^y - d_{j-1}^y\\ d_{j-1}^x - d_{j+1}^x\end{pmatrix},$$

where $\vec{d}_{j-1}^D = (d_{j-1}^x, d_{j-1}^y)^T$ and $\vec{d}_{j+1}^D = (d_{j+1}^x, d_{j+1}^y)^T$ are the neighboring vertices of the intersection $\vec{w}$ in the overlap D.

Case 3: *$\vec{v}_j^i \in D$ and $\vec{v}_j^i \notin adj(\vec{w})\ \forall \vec{w} \in W(D)$*
Now, $\vec{v}_j^i$ is a pure inside overlap vertex, in the sence that it does not have an intersection point as a neighbor in the overlap. This dynamic is quite easy, since we just have to use the Shoelace Formular 4.1 for once more:

$$\begin{aligned}\nabla_{\vec{v}_j^i}\tfrac{1}{k}A(D)^k &= A_D^{k-1}\nabla_{\vec{v}_j^i}A(D)\\ &= A_D^{k-1}\mathbf{1}_{V(D)}(\vec{v}_j^i)\frac{1}{2}\begin{pmatrix}d_{j+1}^y - d_{j-1}^y\\ d_{j-1}^x - d_{j+1}^x\end{pmatrix},\end{aligned}$$

where $\vec{d}_{j-1}^D = (d_{j-1}^x, d_{j-1}^y)^T$ and $\vec{d}_{j+1}^D = (d_{j+1}^x, d_{j+1}^y)^T$ are the neighboring vertices of $\vec{v}_j^i$ in the overlap D.

Case 4: *$\vec{v}_j^i \in D$ and $\exists \vec{w} \in W(D) :\ \vec{v}_j^i \in adj(\vec{w})$*
In the last case, the considered vertex is an overlap vertex and also adjacent to at least one intersection point. Thus, we have to add both dynamics from the last two

cases and then use the partial derivative for inside vertices.

$$
\begin{aligned}
\nabla_{\vec{v}_j^{\,i}} \tfrac{1}{k} A(D)^k &= A_D^{k-1} \nabla_{\vec{v}_j^{\,i}} A(D) \\
&= A_D^{k-1} \left(\mathbf{1}_{V(D)}(\vec{v}_j^{\,i}) \nabla_{\vec{v}_j^{\,i}} A(D) + \sum_{\vec{w} \in W(D)} \mathbf{1}_{adj(w)}(\vec{v}_j^{\,i}) D_{\vec{v}_j^{\,i}} \vec{w} \, \nabla_{\vec{w}} A(D) \right) \\
&= A_D^{k-1} \Bigg(\mathbf{1}_{V(D)}(\vec{v}_j^{\,i}) \nabla_{\vec{v}_j^{\,i}} A(D) \\
&\quad + \sum_{\vec{w} \in W(D)} \left(\mathbf{1}_{out(w)}(\vec{v}_j^{\,i}) D_{\vec{v}_{out}^{\,i}} \vec{w} \, \nabla_{\vec{w}} A(D) + \mathbf{1}_{in(w)}(\vec{v}_j^{\,i}) D_{\vec{v}_{in}^{\,i}} \vec{w} \, \nabla_{\vec{w}} A(D) \right) \Bigg) \\
&= A_D^{k-1} \Bigg(\mathbf{1}_{V(D)}(\vec{v}_j^{\,i}) \nabla_{\vec{v}_j^{\,i}} A(D) \\
&\quad + \sum_{\vec{w} \in W(D)} \left(\mathbf{1}_{out(w)}(\vec{v}_j^{\,i}) D_{\vec{v}_{out}^{\,i}} \vec{w} + \mathbf{1}_{in(w)}(\vec{v}_j^{\,i}) D_{\vec{v}_{in}^{\,i}} \vec{w} \right) \nabla_{\vec{w}} A(D) \Bigg),
\end{aligned}
$$

where $out(w)$, $in(w) \subset adj(w)$ denote the sets of outside and inside overlap-adjacent vertices, respectively.
The difference between $\nabla_{\vec{v}_j^{\,i}} A(D)$ and $\nabla_{\vec{w}} A(D)$ is, that $\nabla_{\vec{v}_j^{\,i}} A(D)$ uses the neighboring overlap vertices of $\vec{v}_j^{\,i}$ itself in the Area Gradient Formula 7, whereas $\nabla_{\vec{w}} A(D)$ uses the neighbors of the according intersection point in D which is not the same overlap vertex as $\vec{v}_j^{\,i}$.
Overall:
Actually, Case 4 already provides the final formulation, since in the other cases the additional terms vanish due to the indicator functions.

□

Figure 15 illustrates the interaction between two overlapping cells, highlighting the effect of the overlap force on their vertices.

Bounce overlap force

While the previously introduced overlap force effectively reduces cell overlap by deforming the cells' shapes, it does not directly separate them spatially—leaving cells temporarily stuck together, relying on random Brownian motion to diffuse apart. With just that force, it is hard to compare the DF cell model to the hard sphere cell model, where overlaps are solved by reflecting them away from each other, resulting in a real distance that both cells have after a really small amount of time.
To address this limitation and ensure a smoother conceptual and mechanical transition from the hard disc cell model, where non-deformable cells simply bounce off one another, we introduce a second overlap degeneration force. This additional force, which we refer to as the bounce overlap force, acts not by changing cell shape but by actively transporting overlapping cells away from each other. This mechanism

captures the spatial repulsion characteristic of rigid body interactions while complementing the shape based degeneration of overlaps in deformable cells. In the end, we will use a combination of both overlap forces to get a nice transition from the HCSM to the DF cell model.
In [BC12] overlapping cells with a radius of r that have a centre-to-centre distance of $2r - a$ will be reflective apart in one time step (that is 10^{-5}), resulting in a distance of $2r + a$ between the two cell centres afterwards.
The following force does the exact same for our DF cells. But we need the following assumptions:

1. Our DF cells model circular discs.
2. The cells have a radius of $r \in \mathbb{R}_{>0}$.
3. We can compute the cell centre $\vec{x} = \frac{1}{N_V} \sum_{j=1}^{N_V} \vec{v}_j$ which will be used to determine the distance between two cells.

Definition 4.16. Bounce overlap force
Let us consider two DF cells C_i and C_l, with centres at $\vec{x}_i$ and $\vec{x}_l$, respectively. We assume that each cell has a fixed radius $r > 0$. We define the vector $\mathrm{d}\bar{o}_{i,l} \in \mathbb{R}^2$, which is applied equally to all vertices of cell C_i, representing the repulsive overlap force caused by cell C_l. It is given by

$$\mathrm{d}\bar{o}_{i,l} = \mathbf{1}_{\|\vec{x}_i - \vec{x}_l\|_2 < 2r}(2r - \|\vec{x}_i - \vec{x}_l\|_2)\frac{\vec{x}_i - \vec{x}_l}{\|\vec{x}_i - \vec{x}_l\|_2}.$$

This force is zero if the distance between the cell centres satisfies $\|\vec{x}_i - \vec{x}_l\|_2 \geqslant 2r$. Otherwise, if the cells overlap (i.e., $\|\vec{x}_i - \vec{x}_l\|_2 < 2r$), the vector $\mathrm{d}\bar{o}_{i,l}$ points from $\vec{x}_l$ to $\vec{x}_i$ and has magnitude equal to the overlap depth $a = 2r - \|\vec{x}_i - \vec{x}_l\|_2$. At the same time in the simulation, the same magnitude of displacement is applied to all vertices of C_l in the opposite direction, i.e., along $\vec{x}_l - \vec{x}_i$. This means that all vertices of cell C_i are displaced away from C_l in the direction $\vec{x}_i - \vec{x}_l$ such that the resulting displacement is sufficient to separate the two cells' centres by exactly $2r + a$.
The force $F_{i,l}^{(\bar{O})}$ that acts on cell C_i due to its interaction with cell C_l is given by

$$F_{i,l}^{(\bar{O})}(C_i, C_l) = (\mathrm{d}\bar{o}_{i,l}, \ldots, \mathrm{d}\bar{o}_{i,l})^T \in \mathbb{R}^{2N_V},$$

where the vector $\mathrm{d}\bar{o}_{i,l}$ is repeated N_V times, once for each vertex of cell C_i.
The total bounce overlap force $F_i^{(\bar{O})} : (\mathbb{R}^{2N_V})^{N_C} \to \mathbb{R}^{2N_V}$ acting on cell C_i due to all other cells in the system is then defined as

$$F_i^{(\bar{O})}(\vec{C}) = \sum_{l \neq i} F_{i,l}^{(\bar{O})}(C_i, C_l).$$

In order to achieve a similarly fast degeneration of the overlap as in [BC12], we need to scale the force with the scaling factor $\alpha^{(\bar{O})} = 10^5$ as the time needed to resolve such an overlap in [BC12] was always one time step which is 10^{-5}. The according dynamic is shown in Figure 17.

To account for varying cell stiffness, we introduce a new parameter $h \in [0, 1]$ that controls how 'hard' the cells are. The total overlap force is then defined as a weighted combination of two overlap force types:

$$F^{(\mathbf{O})} = h \cdot F^{(\bar{O})} + (1 - h) \cdot F^{(\hat{O})},$$

where $F^{(\bar{O})}$ denotes the bounce off overlap force and $F^{(\hat{O})}$ the shape deforming overlap force.
When $h = 1$, the cells are maximally stiff. In this case, shape deformation is entirely

Figure 17: The top four plots present the evolution of a DF cell governed solely by the bounce overlap force, applied to the vertices with a force scaling of 1×10^5, at times $t \in \{0, 1 \times 10^{-5}, 2 \times 10^{-5}, 3 \times 10^{-5}\}$.
In this case, we have $\frac{\partial C_i}{\partial t} = 1 \times 10^5 \, F_i^{\bar{O}}(C_1, C_2)$ for both cells. Click *here* to view the corresponding animation (GIF).
The overlap vanishes within the very first time step, leaving a visible gap between the two cells. The corresponding energy development is illustrated in Figure 18.

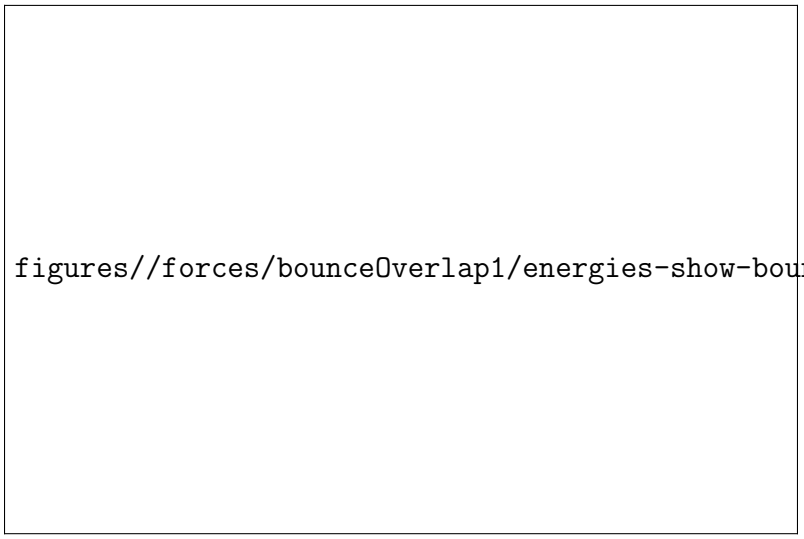


Figure 18: The overlap energy reaches zero after one iteration.

suppressed: all overlaps are resolved solely through the bounce off mechanism, and the shape preserving forces become redundant since the cells always retain their desired configurations.
As h decreases, the cells become progressively softer. The influence of the bounce off force diminishes, while the shape deforming overlap force gains dominance, allowing cells to deform more in response to contact with neighbors.
With the introduction of the hardness parameter h, we have established a mechanism that enables a smooth transition from the HSCM dynamics introduced in [BC12] to our new DF cell model.
For $h = 1$, the dynamics are identical to the original HSCM model, as we will show in the following chapter. By gradually decreasing h, we can continuously adapt the system behavior toward the pure deformable (DF) cell model. In this way, we can systematically investigate how the dynamics evolve between the two regimes. In the limiting case $h = 0$, we recover the DF dynamics without the bounce overlap force term.

4.5 The DF SDE

Having defined all individual force contributions acting on the cell vertices, we now combine them to formulate the full dynamics of the system in terms of a stochastic differential equation.
One challenging aspect that remains is the appropriate scaling of all forces. It has become evident that the system is highly sensitive to these scaling parameters. If the shape recovering forces are too small, the recovery process is excessively slow. Conversely, if they are too large, the numerical integration scheme tends to become unstable.
To ensure numerical stability while preserving the intended dynamics, we systematically tested the appropriate scaling for each force type, focusing in particular on the shape preserving forces and the deforming overlap force.
Our method was to isolate each of these forces in simulation and determine the

threshold at which the system becomes unstable. Specifically, we ran simulations with only one force active at a time and gradually increased its scaling factor until numerical instabilities emerged. We then selected the maximum stable scaling as the operative value for that force.
These tests were conducted using a fixed time step of 10^{-5}, with cell configurations composed of six vertices. To rigorously challenge the model, we initialized the cells in deliberately distorted and uncomfortable shapes, which are most prone to triggering instabilities. The configurations used in these tests are the same as those shown in the figures throughout this chapter, where the isolated effects of each force were illustrated following their respective introductions. This allowed us to verify that the chosen scalings are robust even under unfavorable conditions.
For the bounce overlap force, a scaling factor of 10^5 is necessary to reproduce the original bounce off dynamics used in [BC12].
Summarizing all these efforts, we arrived at the following force scalings used in the simulations:

Force type	**Scaling parameter**
Area force	$\alpha_A = 4.0 \times 10^8$
Edge force	$\alpha_E = 3.0 \times 10^4$
Interior angle force	$\alpha_I = 1.0 \times 10^{-1}$
Deforming overlap force	$\alpha_{\hat{O}} = 6.0 \times 10^4$
Bounce overlap force	$\alpha_{\bar{O}} = 1.0 \times 10^5$

Table 2: Scaling parameters for different force types

These scaling par ameters serve as the foundation for the complete DF SDE model, which we now introduce.

Definition 4.17. The DF SDE
We define the vectors

$$e^x_{N_V} = (1,0,1,0,\ldots,1,0)^T, \quad e^y_{N_V} = (0,1,0,1,\ldots,0,1)^T \in \mathbb{R}^{2N_V},$$

which allow us to distribute a two-dimensional Brownian motion

$$\mathrm{d}\vec{B}_i(t) = (\mathrm{d}B^x_i(t), \mathrm{d}B^y_i(t))^T,$$

to the x and y coordinates of the vertices of a cell, respectively. The cell hardness parameter $h \in [0,1]$ is assumed to be given. The deterministic part $F^i(\vec{C})$ of the **DF SDE** is given by

$$\mathbf{F}^i : (\mathbb{R}^{2N_V})^{N_C} \longrightarrow \mathbb{R}^{2N_V},$$

$$\begin{aligned} \mathbf{F}^i(\vec{C}) = {} & \alpha_A F^{(A)}_2(C_i) + \alpha_E F^{(E)}_2(C_i) + \alpha_I F^{(I)}_2(C_i) + \\ & (1-h)\alpha_{\hat{O}} F^{(\hat{O})}_{1,i}(\vec{C}) + h\alpha_{\bar{O}} F^{(\bar{O})}_i(\vec{C}), \end{aligned} \tag{16}$$

for each cell $1 \leqslant i \leqslant N_C$.
Each scaling parameter $\alpha \geqslant 0$. Each F represents one of our forces that got defined

in this chapter. For the shape preserving forces, we choose $k = 2$ as then the difference between current state and desired state influences the intensity of the force which is nice. But for the deforming overlap force, we choose $k = 1$ we want it to have a strong impact whenever a small overlap arises. This configuration seems to be the best to model the physics we would like to achieve. With that we can write down the cell wise formulated DF SDE:

$$\mathrm{d}C_i(t) = \mathbf{F}^i(\vec{C}(t))\mathrm{d}t + \mathrm{d}B_i^x(t)\, e_{N_V}^x + \mathrm{d}B_i^y(t)\, e_{N_V}^y. \tag{17}$$

Figure 19: This simulation shows two DF cells evolving according to the dynamics for $i = 1, 2$ $\frac{\partial C_i}{\partial t} = \mathbf{F}^i(C_1, C_2)$ with hardness $h = 1$, force scalings as listed in Table 2 and without Brownian motion. Click *here* to view the corresponding animation (GIF). The cells are initially generated with overlap. Then, the dynamics from $\mathbf{F}^i$ alone resolve the overlap. Since hardness $h = 1$ was chosen, no deforming overlap force is active and the cell shape remains unchanged. Consequently, the shape preserving forces are inactive, as the cells stay in their desired states.
This is also reflected in the energy diagram in Figure 22.

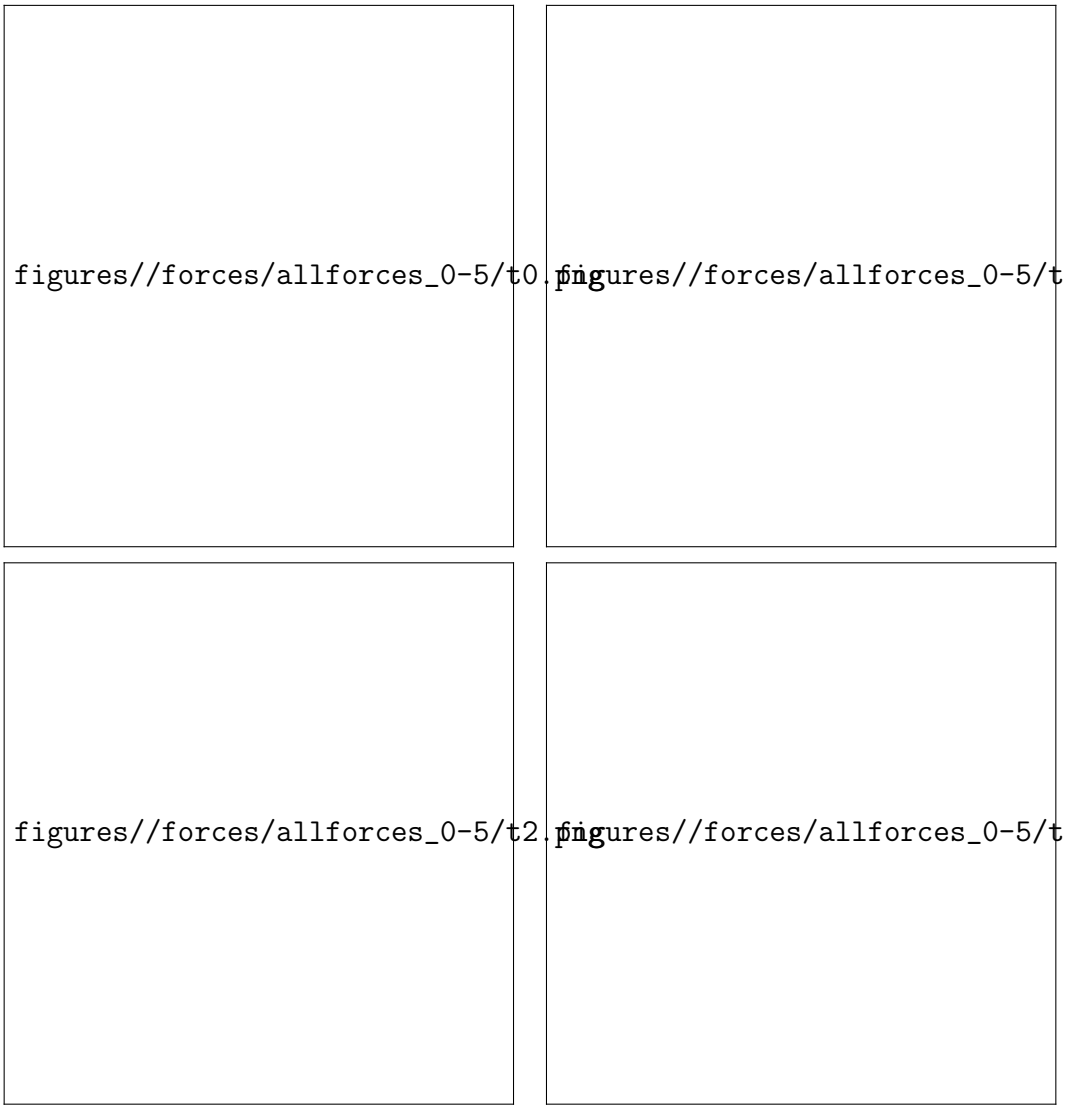


Figure 20: This simulation shows two DF cells again evolving according to the dynamics $i = 1, 2$ $\frac{\partial C_i}{\partial t} = \mathbf{F}^i(C_1, C_2)$, with hardness $h = 0.5$, force scalings as listed in Table 2, and without Brownian motion. Click *here* to view the corresponding animation (GIF). The cells are initially generated with overlap. Afterwards, the dynamics from $\mathbf{F}^i$ alone resolve the overlap.
In contrast to the previous simulation, the cell shapes now change because the deforming overlap force is active. The overlap is still removed within a single time step. By time step 10, the cell shapes are nearly restored, as also illustrated in Figure 23.

Figure 21: This simulation shows two DF cells evolving according to the dynamics for $i = 1, 2$ $\frac{\partial C_i}{\partial t} = \mathbf{F}^i(C_1, C_2)$, with hardness $h = 0$, force scalings as listed in Table 2 and without Brownian motion. Click *here* to view the corresponding animation (GIF). The cells are initially generated with overlap. Afterwards, the dynamics from $\mathbf{F}^i$ alone attempt to resolve the overlap.
In this case, only the deforming overlap force is active. This leads to a repeating interplay: the overlap is reduced, the cell shape is restored, and this restoration again induces overlap. Under this setup, neither the overlap nor the desired cell shapes are fully resolved within 20 time steps, although all energy levels remain comparatively low. We can also see this in the energy diagram in Figure 24.

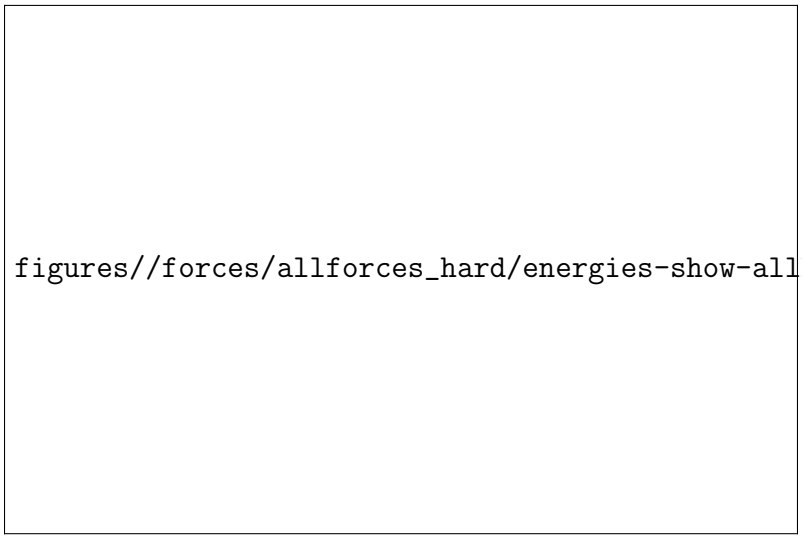


Figure 22: The bounce overlap force eliminates the overlap within a single time step. The energy diagram shows that the overlap energy drops to zero immediately, while the other energies remain constant at zero as the cell shapes do not change.

Figure 23: The energy diagram shows that the overlap energy drops to zero immediately, while the shape preserving energies increase initially as the cells deform to resolve the overlap. By time step 10, the cell shapes are nearly restored, which is reflected in the decrease of all energies.

Figure 24: The energy diagram shows that the overlap energy drops initially but then increases again as the cells deform to restore their shapes. This deformation again induces overlap, leading to a repeating cycle. But overall, the total energy converges to a low level, indicating that the system stabilizes over time.

5 DF model validation and simulation analysis

In this chapter, we continue by performing and analysing cell simulations based on the DF model. The chapter is structured into four subsections: we first introduce the reference model from [BC12], followed by a reproduction of their published results to validate our implementation. We then extend the analysis to simulations involving deformable cells and finally conclude with a quantitative assessment of cell shape deformation.
In order to position our simulations within the context of established modelling approaches, we first aim to compare our DF model results with those of the well-studied cell interaction model introduced by Bruna and Chapman (2012) [BC12]. In the following subsection, we briefly summarise the theoretical formulation of that reference model before presenting our numerical reproduction of their findings.

5.1 Reference simulations: Bruna and Chapman (2012)

The point particle and hard sphere models from [BC12] were already introduced in Section 2. In the following, we restate the explicit cell dynamics and corresponding first marginals, using specified parameters from the reference paper [BC12], to set the stage for our subsequent numerical reproduction.
The paper [BC12] investigates its problem on the square domain

$$\Omega_{BC} = [-0.5, 0.5]^2,$$

in which $N_C = 400$ particles are located.
The point particle dynamics are defined by

$$\mathrm{d}\vec{x}_i(t) = \sqrt{2}\,\mathrm{d}\vec{B}_i(t), \quad \vec{x}_i \in \Omega_{BC}^{\circ},$$

for $1 \leqslant i \leqslant N_C$, where

$$\Omega_{BC}^{\circ} = (-0.5, 0.5)^2,$$

denotes the interior of the domain Ω_{BC}. Reflective boundary conditions are imposed on the boundary $\partial\Omega_{BC}$ to prevent particles from leaving the domain.
Equation (1) provides the general form of the first marginal of the point particle model. In this case, we set the diffusion coefficient to $D = 1$ and omit any external force field, i.e. $f(\vec{x}) = 0$. Consequently, the first marginal simplifies to

$$\frac{\partial \rho(t;\vec{x})}{\partial t} = \nabla_{\vec{x}} \cdot \nabla_{\vec{x}} \rho(t;\vec{x}), \quad \vec{x} \in \Omega_{BC},\ t > 0, \tag{18}$$

which is expressed in divergence form and represents the diffusion of independent Brownian particles within the confined domain.

In order to transition from the point particle system to the hard sphere cell model, we introduce the cell diameter $0 < \epsilon \ll 1$. As recalled from the introduction, Section 2, the hard sphere dynamics are defined by

$$\mathrm{d}\vec{x}_i(t) = \sqrt{2}\,\mathrm{d}\vec{B}_i(t), \quad \vec{x}_i \in \Omega_{BC,\epsilon}^{\circ},$$

where we again choose $D = 1$ and $f(\vec{x}) = 0$.
Here, $\Omega_{BC,\epsilon}$ denotes the excluded volume domain, as defined in Equation (2), which excludes regions where two cells would overlap. This domain possesses a more complex boundary $\partial\Omega_{BC,\epsilon}$, comprising both the external boundary of Ω_{BC} and additional excluded areas arising from the finite size of the cells. Consequently, the reflective boundary condition leads to a significantly higher number of reflections compared to the point particle model.
When computing the first marginal of the hard sphere model, we obtain Equation (3). In our specific setting, this expression simplifies to

$$\frac{\partial \rho}{\partial t} = \nabla_{\vec{x}} \cdot \nabla_{\vec{x}} \left[\rho + \tfrac{\pi}{2}(N_C - 1)\epsilon^2 \rho^2\right], \quad \vec{x} \in \Omega_{BC},\ t > 0, \tag{19}$$

which accounts for the additional diffusive correction caused by volume exclusion between cells.
The initial condition for both the point particle and hard sphere models follows a two-dimensional normal distribution, subject to the constraint that the distance between any two cell centres is at least ϵ. The employed distribution $\mathcal{N}_2(0,\, 0.09^2 \cdot I_2)$ is normalised to have an integral of one over Ω_{BC}. This initial condition is computed using Algorithm 5.1, which iteratively samples non-overlapping cell positions. The resulting spatial configuration, illustrating how cell overlaps are avoided, is shown in Figure 3.

Algorithm 5.1. ***Computation of the initial cell system***

1. *Generate a point* $\vec{x} \sim \mathcal{N}_2(0,\, 0.09^2 \cdot I_2)$.
2. *If for all already generated centres* $\vec{x}_j : \|\vec{x} - \vec{x}_j\|_2 > \epsilon$ *is true, use* $\vec{x}$ *as the next cell centre, otherwise discard the point and restart with step 1 until* N_C *cell centres are found.*

The marginal equation (19) exhibits an enhanced diffusion rate because of the additional positive non-linear term $\frac{\pi}{2}(N_C - 1)\epsilon^2\rho^2$ inside the diffusion operator. This term represents excluded volume interactions between finite sized particles, which especially bias motion at regions of high density and thereby accelerate the overall spread.
As a result, the effective collective diffusion increases with particle number N_C, particle size ϵ, and local density ρ. We can rewrite Equation (19) to explicitly show this enhanced diffusion effect as follows:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} &= \nabla_{\vec{x}} \cdot \nabla_{\vec{x}} [\rho + \frac{\pi}{2}(N_C - 1)\epsilon^2\rho^2] \\
&= \nabla_{\vec{x}} \cdot [\nabla_{\vec{x}}\rho + \frac{\pi}{2}(N_C - 1)\epsilon^2 \nabla_{\vec{x}}\rho^2] \\
&= \nabla_{\vec{x}} \cdot [\nabla_{\vec{x}}\rho + \frac{\pi}{2}(N_C - 1)\epsilon^2 2\rho\nabla_{\vec{x}}\rho] \\
&= \nabla_{\vec{x}} \cdot [\underbrace{(1 + \pi(N_C - 1)\epsilon^2\rho)}_{= D(\epsilon, N_C, \rho)} \nabla_{\vec{x}}\rho],
\end{aligned}$$

where we used the product rule for gradients in the third line. Here,

$$D(\epsilon, N_C, \rho) = 1 + \pi(N_C - 1)\epsilon^2\rho,$$

acts as an effective diffusion coefficient that depends on the particle size, total particle number, and local concentration. For $\epsilon = 0$, we recover Equation (18), corresponding to point particles with $D = 1$ and for $\epsilon > 0$, we obtain Equation (19) as shown in the computation. This reformulation clearly demonstrates how excluded volume interactions in the hard sphere cell model lead to an enhanced diffusion rate compared to the point particle case.

Figure 25: This figure from [BC12] contains the following four plots, all of them are shown at time $t = 0.05$. For all plots, the initial condition is normally distributed with mean $(0, 0)^T$ and standard deviation 0.09. (a) shows the solution of the linear diffusion Equation (1) for point particles. (b) shows the histogram of a Monte Carlo simulation of the point particle model. (c) shows the solution of the non-linear diffusion Equation (3) for finite sized particles. (d) shows the histogram of a Monte Carlo simulation of the hard sphere model. The Monte Carlo simulations used 10^4 simulation runs each with a time step size of 10^{-5}. We can see that the hard sphere model in (c) and (d) shows a quicker diffusion rate as the cell concentration in the centrum of the domain has already diffused more compared to the point particle model in (a) and (b).

Another evidence of this behaviour is shown in Figure 25 which is originally from the considered paper [BC12]. Here, we can see the result of the application of the first marginal dynamics from Equation (18) and Equation (19) alongside heatmaps

generated from two Monte Carlo simulations.
A Monte Carlo simulation, as for example described in [MU49] is a computational technique that uses random sampling to model and analyse complex systems or processes that are difficult to solve analytically. It repeatedly generates random inputs according to specified probability distributions and computes the resulting outcomes to estimate quantities like averages, variances, or distributions.
In our case, the Monte Carlo simulations are used to track the positions of cell centres over time. Each simulation begins from an initial configuration of cells, which is consistently generated using Algorithm 5.1. After initialisation, the prescribed dynamics - either the point particle model or the hard sphere model - are applied, and the positions of the cell centres are recorded at a fixed time point, $t = 0.05$.
To visualise the results, we construct heatmaps representing the spatial distribution of cells at the final time. This is done by discretizing the domain into a uniform grid ofsubsquares. For eachsubsquare, we count how many cells fall within it across all simulations. The resulting counts are normalised by dividing by the total number of cells N_C, the number of simulations, and the area of asubsquare. This normalisation ensures that the heatmap represents a probability density, satisfying the mass conservation condition:

$$\sum_{i \in \text{subsquares}} \text{value}_i \cdot \text{area}_i = 1.$$

By the Law of Large Numbers, these heatmaps converge to the first marginal distribution as both the number of independent simulations increases and the spatial discretisation of Ω is refined. This procedure, detailed in Section 6, yields a smooth and consistent estimate of the empirical cell density, enabling a direct comparison with the corresponding solutions of the continuum diffusion equations.

5.2 Reproduction of reference results

We now proceed to apply our own implementation of the DF model to reproduce the reference results, thereby validating the correctness and reliability of our simulation code. All simulations were conducted in the Julia programming language [BEKS17], using the package 'DifferentialEquations.jl' [RN17]. The stochastic differential equation system was defined through the 'SDEProblem()' structure and solved via the built in Euler-Maruyama integration scheme with a constant time step size. This approach provides both flexibility and numerical reliability for large scale stochastic simulations. Julia offers significant advantages for our purpose due to its high computational performance, native support for parallelisation, and clear syntax that allows for a concise yet expressive implementation of mathematical models. Moreover, the 'DifferentialEquations.jl' ecosystem is highly optimised for stochastic and deterministic solvers alike, ensuring efficient and reproducible numerical experiments. Many of the visualisations throughout this thesis, for instance Figure 2, were created using the Julia package 'Plots.jl' [CSR+23].

We began with the implementation of the simplest case: the point particle model. All simulation parameters, such as the domain, number of particles, and diffusion coefficients, were chosen as described in the previous subsection to ensure

full comparability with the reference setup from [BC12]. Specifically, we simulated $N_C = 400$ particles within the domain $\Omega = [-0.5, 0.5]^2$, initialised according to Algorithm 5.1. The initial positions were drawn from a two-dimensional normal distribution $\mathcal{N}_2(0,\, 0.09^2 \cdot I_2)$ using the function 'MvNormal()' provided by the 'Distributions.jl' package [BPA+21]. Each particle then evolved according to the point particle dynamics with diffusion coefficient $D = 1$ and reflective boundary conditions on $\partial\Omega$. The stochastic differential equations were solved with 'DifferentialEquations.jl' [RN17] using its inbuilt Euler-Maruyama scheme with a constant time step.
We performed 10^4 independent Monte Carlo simulations to obtain statistically significant results. Parallelisation of the Monte Carlo runs was achieved through Julia's built in 'Distributed' package [Wri23], ensuring efficient use of computational resources.
Using our self developed simulation framework, we generated two-dimensional heatmaps representing the cell density distributions over time. These results are directly comparable to Figure 25(b) from [BC12]. As shown in Figure 26, the outcomes of our point particle simulations closely replicate the reference results, confirming the correctness and reliability of our numerical implementation. To ensure direct comparability, we fixed the colour scale in our visualisations to match that used in [BC12].

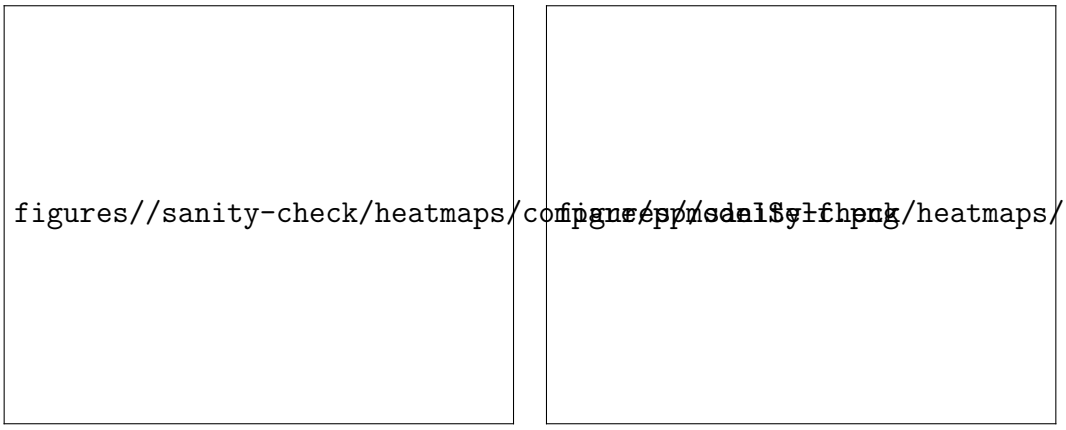


Figure 26: The left panel shows the heatmap obtained from our new point particle simulation, while the right panel presents the corresponding result from [BC12]. We applied the same color scale as in Bruna (2012) to both panels to facilitate direct comparison. Despite minor differences, the two heatmaps exhibit a very similar overall structure and intensity pattern.

Having successfully validated our implementation with the point particle model, we proceeded to mimic the hard sphere cell model (HSCM) as described in [BC12]. To this end, we employed our DF model framework. In this thesis, cells are represented as regular hexagons with $N_V = 6$ vertices. This choice allows for potential shape deformation at each vertex while maintaining computational efficiency, which is essential given the tens of thousands of Monte Carlo simulations performed.
Consistent with the hard sphere model, the hexagonal cells have a diameter of $\epsilon = 0.01$, defined as the distance between two opposite vertices. To replicate the

dynamics of hard spheres, shape deformation is disabled by setting the cell hardness to $h = 1$ (see Section 4). Consequently, all shape preserving forces from the DF model can be neglected, and only the bounce overlap force is applied.

Simulation 5.2. Hard DF model – Monte Carlo Simulation Setup
In this simulation, each cell $C_i = (\vec{v}_1^{\,i}, \ldots, \vec{v}_6^{\,i})$ represents a list of vertices. The desired cell shape is given as a regular hexagon with a diameter of $\epsilon = 0.01$, defined as the distance between two opposite vertices. The dynamic of the cells, significant for hardness $h = 1$, is governed by the SDE

$$\mathrm{d}C_i(t) = 10^5 \cdot F_i^{(\bar{O})}(\vec{C})\mathrm{d}t + \mathrm{d}B_i^x(t)\, e_{N_V}^x + \mathrm{d}B_i^y(t)\, e_{N_V}^y, \quad 1 \leqslant i \leqslant 400,$$

where the vectors

$$e_{N_V}^x = (1, 0, 1, 0, \ldots, 1, 0)^T, \quad e_{N_V}^y = (0, 1, 0, 1, \ldots, 0, 1)^T \in \mathbb{R}^{2N_V},$$

distribute a two-dimensional Brownian motion

$$\mathrm{d}\vec{B}_i(t) = (\mathrm{d}B_i^x(t), \mathrm{d}B_i^y(t))^T,$$

to the x and y coordinates of the cell vertices.
Reflective boundary conditions are applied collectively to each cell: if the centre

$$\vec{c}_i = \frac{1}{6}\sum_{k=1}^{6} \vec{v}_k^{\,i},$$

of a cell crosses the domain boundary $\partial\Omega$, the entire cell is pushed back into the interior Ω°.
The SDEs are solved using the Euler–Maruyama scheme with constant time step $\Delta t = 10^{-5}$. A total of 10^4 independent Monte Carlo simulations were performed to ensure statistical significance. The simulations were carried out on the time interval $[0, 0.05]$.

Using this setup, we generated heatmaps representing the evolution of cell density over time. Figure 27 compares our simulation results with the corresponding results from [BC12].
As evident from Figure 27, our simulation reproduces the general spatial patterns and density evolution of the hard sphere model. Minor differences, such as a slightly higher density in the domain centre, are expected due to variations in implementation, interpolation methods, and colour map scaling. Overall, the results confirm that our DF model accurately captures the behaviour of hard sphere cells when shape deformation is disabled, providing a solid foundation for subsequent simulations with deformable cells.

5.3 DF model simulations with deformable cells

After confirming that our implementation correctly reproduces the macroscopic dynamics of non-overlapping (hard) particles described in [BC12], we now generalize the model to account for cell deformability.

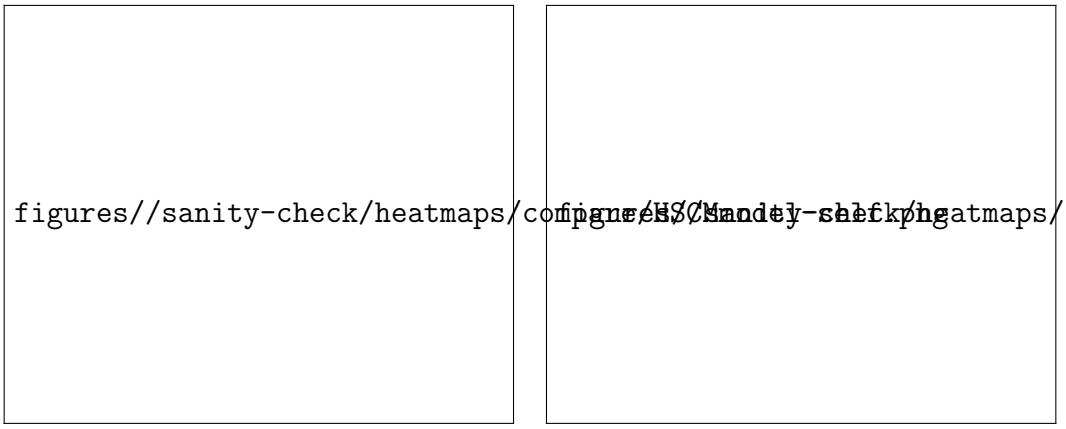


Figure 27: Comparison of hard cell simulations: the left panel shows the heatmap obtained from our DF model implementation, and the right panel shows the corresponding result from [BC12]. The colour scale was matched to that used in the reference for a meaningful visual comparison. Overall, the two heatmaps display closely matching spatial patterns, with only minor variations.

In biological systems, cell-cell interactions are rarely perfectly hard; instead, cells can locally deform upon contact, redistributing internal stresses and slightly modifying their effective motion. To capture this behaviour, we relax the hard sphere constraint by introducing a soft deformable force that acts when two cells are sufficiently close to each other.
This modification leads to an extended stochastic description in which the overlap force $F_i^{(\hat{O})}(\vec{C})$, defined in 4.15, supplements the purely repulsive (bounce) force $F_i^{(\bar{O})}(\vec{C})$ (4.16), whenever we use a hardness $h < 1$. The resulting system of SDEs describes the coupled motion of deformable cells under diffusion and interaction forces, thereby forming the soft DF model. In the following, we derive two SDE formulation, called soft and mid DF model, and discuss their relation to the non-deformable case, the hard DF model.
The soft DF model corresponds to the case of fully deformable cells without any hard repulsion, i.e. with hardness $h = 0$. It is governed by the consecutive SDE.

Simulation 5.3. Soft DF model - Monte Carlo simulation setup
In this simulation, each cell $C_i = (\vec{v}_1^{\,i}, \ldots, \vec{v}_6^{\,i})$ represents a list of vertices. The desired cell shape is given as a regular hexagon with a diameter of $\epsilon = 0.01$, defined as the distance between two opposite vertices. The dynamic of the cells, significant for hardness $h = 0$, is governed by the SDE

$$\mathrm{d}C_i(t) = \mathbf{F}^i(\vec{C}(t))\mathrm{d}t + \mathrm{d}B_i^x(t)\, e_{N_V}^x + \mathrm{d}B_i^y(t)\, e_{N_V}^y, \quad 1 \leqslant i \leqslant 400,$$

where the deterministic force $\mathbf{F}^i(\vec{C}(t))$ is defined as

$$\mathbf{F}^i(\vec{C}) = \alpha_A F_2^{(A)}(C_i) + \alpha_E F_2^{(E)}(C_i) + \alpha_I F_2^{(I)}(C_i) + \alpha_{\hat{O}} F_{1,i}^{(\hat{O})}(\vec{C}).$$

The force scalings are chosen as in Table 2.

Reflective boundary conditions are applied collectively to each cell: if the centre

$$\vec{c}_i = \frac{1}{6}\sum_{k=1}^{6} \vec{v}_k^i,$$

of a cell crosses the domain boundary $\partial\Omega$, the entire cell is pushed back into the interior Ω°.
The SDEs are solved using the Euler–Maruyama scheme with constant time step $\Delta t = 10^{-5}$. A total of 10^4 independent Monte Carlo simulations were performed to ensure statistical significance. The simulations were carried out on the time interval $[0, 0.05]$.

While the soft DF model introduces deformability by allowing overlapping interactions, it is instructive to consider an intermediate configuration that gradually transitions between the purely hard and fully deformable regimes. To this end, we define a mid DF model, which retains the fundamental features of the hard sphere dynamics, but introduces a limited degree of deformation.
In this setup, the deforming overlap force is partially active, providing a tunable stiffness that bridges the discontinuous transition between entirely rigid and fully soft cell interactions. This intermediate formulation allows us to quantify the influence of deformability in a controlled manner by smoothly varying the stiffness parameter.
This not only aids in verifying the robustness of our implementation but also clarifies how emergent macroscopic properties evolve with cell stiffness. The corresponding stochastic representation, referred to as the mid DF model SDE, incorporates both the bounce and deforming overlap forces with an intermediate weighting. The formulation reads as follows:

Simulation 5.4. Mid DF model - Monte Carlo simulation setup
In this simulation, each cell $C_i = (\vec{v}_1^i, \ldots, \vec{v}_6^i)$ represents a list of vertices. The desired cell shape is given as a regular hexagon with a diameter of $\epsilon = 0.01$, defined as the distance between two opposite vertices. The dynamic of the cells, significant for hardness $h = 0.5$, is governed by the SDE

$$\mathrm{d}C_i(t) = \mathbf{F}^i(\vec{C}(t))\mathrm{d}t + \mathrm{d}B_i^x(t)\, e_{N_V}^x + \mathrm{d}B_i^y(t)\, e_{N_V}^y, \quad 1 \leqslant i \leqslant 400,$$

where the deterministic force $\mathbf{F}^i(\vec{C}(t))$ is defined as

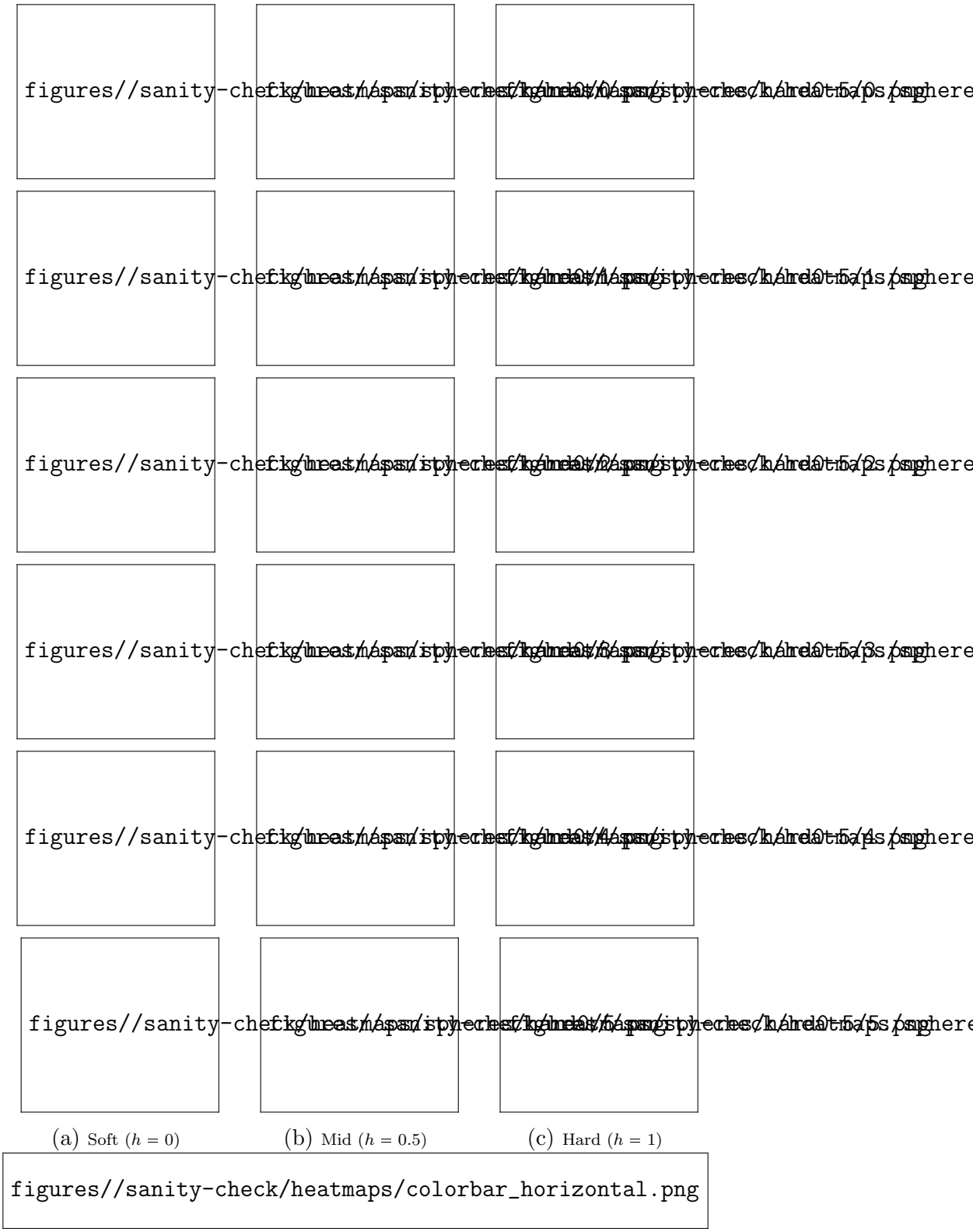
$$\mathbf{F}^i(\vec{C}) = \alpha_A F_2^{(A)}(C_i) + \alpha_E F_2^{(E)}(C_i) + \alpha_I F_2^{(I)}(C_i) + 0.5\alpha_{\hat{O}} F_{1,i}^{(\hat{O})}(\vec{C}) + 0.5\alpha_{\bar{O}} F_i^{(\bar{O})}(\vec{C}).$$

The force scalings are chosen as in Table 2.
Reflective boundary conditions are applied collectively to each cell: if the centre

$$\vec{c}_i = \frac{1}{6}\sum_{k=1}^{6} \vec{v}_k^i,$$

of a cell crosses the domain boundary $\partial\Omega$, the entire cell is pushed back into the interior Ω°.
The SDEs are solved using the Euler–Maruyama scheme with constant time step $\Delta t = 10^{-5}$. A total of 10^4 independent Monte Carlo simulations were performed to ensure statistical significance. The simulations were carried out on the time interval $[0, 0.05]$.

Having established the three DF model formulations — the hard, mid, and soft variants — we now turn to a direct visual comparison of their collective dynamics. Figure 28 presents one of the central results of this work, illustrating the temporal evolution of the cell density for each model side by side. Each panel shows the corresponding heatmap representation of the simulated particle distribution, normalised to unit mass, thus enabling a direct comparison of the spatial spreading behaviour under the different interaction assumptions.
This figure highlights how progressively relaxing the rigidity constraint from the hard to the soft DF model alters the emergent cell dynamics. The hard DF system, governed by strictly repulsive collisions, shows the fastest diffusion rate of our three models. In contrast, the soft DF model, that only uses cell deformation for overlap degeneration, exhibits a narrower spatial distributions due to the missing push dynamic from the bounce overlap force in case of cell overlap. The mid DF configuration naturally occupies an intermediate regime, demonstrating a gradual transition in diffusion rate.
Together, these visualisations provide an intuitive understanding of how cell deformability influences the macroscopic evolution of the system and serve as a qualitative validation of the modelling framework. We therefore proceed by analysing these results in greater detail in the following discussion.



(a) Soft ($h = 0$) (b) Mid ($h = 0.5$) (c) Hard ($h = 1$)

Figure 28: Heatmaps of a Monte Carlo simulation of the DF cell model with different hardness values at the times $t \in \{0.00, 0.01, \ldots, 0.05\}$. Left column (a) shows hardness 0, we can see hardness 0.5 in the middle (b) and hardness 1 on the right (c). We can oberserve that the diffusion rate increases with increasing hardness.

Having compared the full spatial density evolution through heatmaps, we next consider a reduced one-dimensional representation of the cell density to facilitate quantitative comparison between models. To this end, we introduce cross-section plots, which provide averaged density profiles extracted from the two-dimensional simulation domain Ω.
The procedure is as follows. Starting from the discrete density field obtained from a DF simulation, the quadratic domain Ω is subdivided into $N_H \times N_H$subsquares, where $N_H \in \mathbb{N}$ defines the spatial resolution with a corresponding step size $\Delta x = 1/N_H$, as Ω has a side length of 1.
For each vertical line across the domain, we compute the integrated density by summing the density values of all subsquares along that line, multiplied by the step size Δx. This yields a vector of length N_H representing the average cell density per vertical line.
The same operation is then repeated along all horizontal lines to obtain a second vector. The final one-dimensional density profile is obtained as the arithmetic mean of the two vectors, thereby combining information from both orientations to reduce sampling bias.

The resulting averaged density vector ρ_{cross} is then plotted against the spatial coordinate in $[-0.5, 0.5]$, providing a concise visualisation of the overall distribution of cells along the domain.
By construction, the cross-section profile is normalised to unit mass,

$$\sum_{i=1}^{N_H} \rho_{\text{cross}}(x_i)\, \Delta x = 1,$$

ensuring direct comparability between simulations with different model parameters or deformability assumptions.
This representation offers a clear and compact way to quantify the spread and peak structure of the cell population, complementing the qualitative insights from the heatmaps. The cross-section plots for our three DF model variants at different time points are shown in Figure 29.

The cross-section plots allow for a more direct quantitative comparison of the density evolution across the three DF models. Initially, all simulations start from an identical spatial distribution, ensuring that any subsequent differences arise solely from the respective interaction rules. Note that the scaling of the y-axis changes from $[0, 3.5]$ at $t = 0$ to $[0.7, 1.3]$ at $t = 0.05$, reflecting the diffusive spreading of the density field in all cases.
As the plots reveal, increasing the hardness of the model leads to a faster redistribution of the density over time. For $t > 0$, higher hardness values correspond to a lower density at the centre ($x = 0$) and an increased density towards the boundaries of the interval. Consequently, the density profile becomes more uniform more rapidly for harder interactions, indicating enhanced effective diffusion within the system.
This behaviour aligns with the expected dynamics, as stronger repulsive constraints promote faster homogenisation of the cell population. The same trend is evident in the heatmap visualisations from Figure 28, where the harder DF models display a

Figure 29: Temporal evolution of the cross-section density for the three DF models at sample times $t \in \{0, 0.01, \ldots, 0.05\}$. Each curve represents the spatially averaged one-dimensional density, normalised to unit mass. The progressive flattening of the profiles illustrates the diffusive spreading of the cell population, with harder models exhibiting a faster redistribution of density over time.

visibly faster outward spread of density from the centre.
In summary, the consistent diffusion trends across hardness regimes validate our DF model implementations and motivate the subsequent extension to deformable cell shapes.

5.4 Shape deformation check

In this final subsection, we focus on analysing the deformation aspect of our model, which represents the key novel feature introduced in the DF model. While the previous sections examined how cell hardness influences collective diffusion behaviour, we now turn our attention to the question of how individual cells change their shape during our simulations.

To quantify this, we require a consistent measure of cell deformation that can be evaluated throughout our large scale Monte Carlo simulations. Several approaches have been proposed in the literature for characterising the shape of two-dimensional objects, such as the method used by [MOK10] in the context of analysing animal colour patterns.
A simple yet effective measure is based on the ratio between a cell's area and its perimeter, which reflects how close its shape is to an ideal circle. We denote this quantity as the asphericity α of a geometric figure F, defined as

$$\alpha_F = 4\pi \frac{\text{area}_F}{\text{perimeter}_F^2},$$

where the scaling factor 4π ensures that a perfect circle yields $\alpha_\text{c} = 1$. Lower values of α_F therefore indicate stronger deviations from circularity, corresponding to more pronounced cell deformation.

The circle is the only two-dimensional geometric figure that attains the maximal asphericity value of 1. For a circle of radius $r > 0$ we have $\text{area}_\text{c} = \pi r^2$ and $\text{perimeter}_\text{c} = 2\pi r$. Hence

$$\alpha_\text{c} \;=\; 4\pi \frac{\pi r^2}{(2\pi r)^2} \;=\; 1.$$

The classical isoperimetric inequality states that for any (sufficiently regular) planar region F with area A and perimeter P one has

$$P^2 \;\geqslant\; 4\pi A,$$

with equality if and only if F is a circle. Dividing by P^2 and multiplying by the factor 4π gives

$$4\pi \, \frac{A}{P^2} \;\leqslant\; 1,$$

and therefore $\alpha_F \leqslant 1$ for every region F, with $\alpha_F = 1$ only for the circle.
We can conclude that the asphericity of a two-dimensional figure is always bounded between 0 and 1. The lower bound 0 is trivial since both area and perimeter are positive for any non-degenerate region, and hence $\alpha_F \in [0, 1]$.
In our simulations, the asphericity is straightforward to compute from the DF cell geometry. The cell area is obtained using the Shoelace formula (Proposition 4.1), while the perimeter is calculated by summing the Euclidean lengths of the successive edges of the polygonal cell boundary. The resulting value is normalised by the constant factor 4π, ensuring that perfectly circular cells have $\alpha = 1$, and deviations below this value quantify the degree of deformation.
In our simulations, the desired cell shape is a regular hexagon with a diameter of $\epsilon = 0.01$. Since a perfectly circular cell cannot be represented with a finite number of vertices $N_V < \infty$, the theoretical maximum asphericity of $\alpha = 1$ cannot be attained numerically.
For any DF cell configuration with $N_V = 6$ vertices, the highest possible asphericity is achieved by this regular hexagonal shape. This is a classical geometric result, analogous to the proof of the isoperimetric inequality, and it can be generalised to

polygons with any number of vertices $N_V \geqslant 3$. Geometrically, this means that, for a fixed number of vertices, the regular polygon is the shape that most closely approximates a circle.

For any regular polygon P_{N_V} with N_V vertices, we have a cell perimeter of

$$P_{P_{N_V}} = N_V l,$$

where $l > 0$ denotes its edge length.
The polygons area is given by

$$A_{P_{N_V}} = \tfrac{1}{4} N_V l^2 \cot\left(\frac{\pi}{N_V}\right).$$

This yields an asphericity of

$$\begin{aligned} \alpha_{P_{N_V}} &= 4\pi \frac{\frac{1}{4} N_V l^2 \cot\left(\frac{\pi}{N_V}\right)}{(N_V l)^2} \\ &= \frac{\pi}{N_V} \cot\left(\frac{\pi}{N_V}\right). \end{aligned}$$

Thus, for our desired regular hexagon with $N_V = 6$ vertices, we have an asphericity of

$$\alpha_{P_6} = \frac{\pi}{6} \cot\left(\frac{\pi}{6}\right) = \frac{\pi}{6}\sqrt{3} \approx 0.907.$$

Figure 30 and Table 3 illustrate the asphericity values for different DF cell shapes, including our target regular hexagon configuration, alongside a perfect circle for reference. There, we can also see an indictation that as the number of vertices increases, the asphericity approaches 1, consistent with the geometric interpretation that polygons with more vertices better approximate a circular shape.

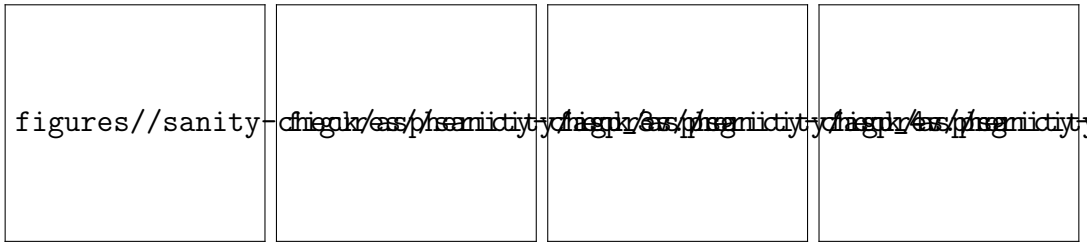


Figure 30: Examples of DF cells and a circle illustrating their geometric properties. The DF cells are represented as regular polygons with $N_V \in \{3, 4, 6\}$ vertices. For each case, the cell shape, area, perimeter and asphericity are shown in Table 3. The third DF cell with $N_V = 6$ vertices corresponds to the target DF cell configuration used in our Monte Carlo simulations, exhibiting an asphericity of approximately $\alpha \approx 0.907$. As the number of vertices increases, the asphericity rises, reaching its maximum value of $\alpha = 1.0$ for the circular case.

We apply the asphericity measure to monitor cell shape changes during our Monte Carlo simulations of the DF model with varying hardness values. For each of the soft,

Cell shape	Area	Perimeter	Asphericity
Triangle	3.25×10^{-5}	0.0260	0.605
Square	5.00×10^{-5}	0.0283	0.785
Hexagon	6.50×10^{-5}	0.0300	0.907
Circle	7.85×10^{-5}	0.0314	1.000

Table 3: Geometric properties of different DF cell shapes and a circle. The asphericity values increase with the number of vertices, approaching 1 as the shape becomes more circular. The target DF cell shape used in our Monte Carlo simulations has an asphericity of approximately $\alpha \approx 0.907$.

mid, and hard configurations, we tracked the asphericity of all 400 cells throughout the simulation, recording their values at the sample times $t \in \{0, 0.01, \ldots, 0.05\}$. Each simulation setup was repeated for 50 independent runs to obtain statistically robust averages of the asphericity evolution.

Figure 31 presents a multi-bar plot showing the distributions of cell asphericity at these time points for the three different hardness settings. This visualisation allows for a direct comparison of how cell deformability evolves under varying interaction stiffness, highlighting the dynamic effects of increasing hardness on the preservation or loss of cell circularity over time.

The asphericity analysis clearly demonstrates the influence of the hardness parameter on the degree of cell deformation during the simulations. For the hard configuration ($h = 1$), all cells retain their initial asphericity of $\alpha \approx 0.907$ throughout, confirming that no shape deformation occurs when only the non-deforming bounce overlap force is active.

In contrast, the soft and mid-hardness simulations display a broader distribution of asphericity values, indicating that cells experience measurable deformation during interactions. The soft model ($h = 0$) shows the widest spread of values, reaching down to $\alpha \approx 0.3$. The apshericity values below 0.7 where reached in the 50 simulations of the soft DF model, and can not be seen in Figure 31, since it was limited to the range $[0.7, 0.92]$ for better visual clarity. However, the mid-hardness model ($h = 0.5$) remains closer to the undeformed state with $\alpha \in [0.84, 0.92]$.

The most pronounced deformations occur early in the simulation ($t > 0$), when the cell density is highest in the central region and collisions are most frequent. Over time, the number of cells returning towards their original asphericity increases, as reflected by the growing proportion of cells in the last interval $[0.9, 0.92]$. This relaxation behaviour suggests that deformation is largely transient and driven by local crowding effects rather than permanent structural changes.

In summary, the observed deformation behaviour aligns well with the theoretical expectations of the DF model. The results confirm that cell deformability decreases with increasing hardness, and that deformation primarily occurs during periods of high local cell density. The consistency between the expected and simulated trends indicates that the implementation of the deformable cell interactions and the corresponding Monte Carlo setup are both correct and robust. Consequently, the DF model faithfully captures the transition from rigid to deformable cell dynamics across

the investigated hardness range.

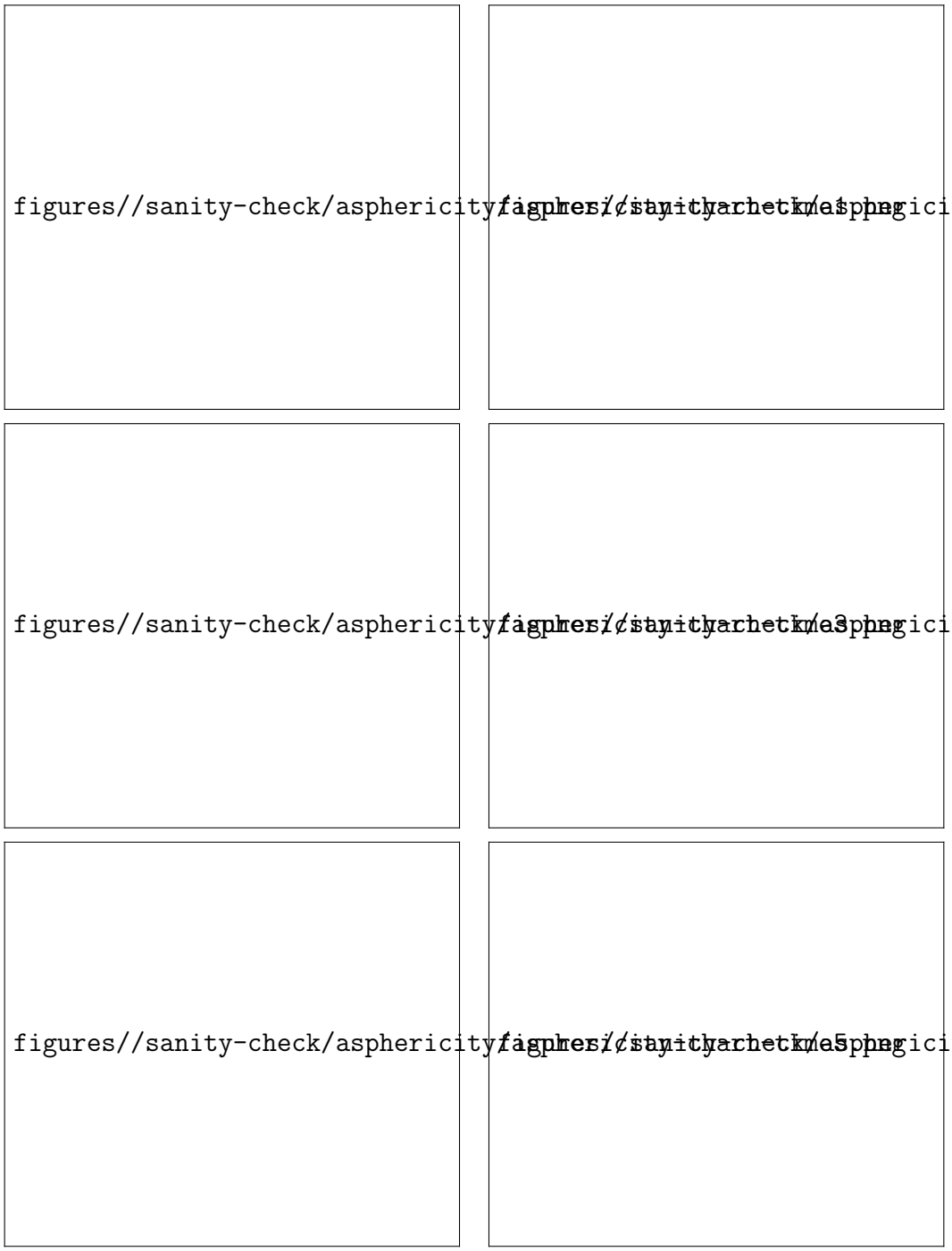


Figure 31: Distributions of cell asphericity over time for the DF simulations with hardness values $h \in \{0, 0.5, 1\}$. Each setup consists of 400 cells averaged over 50 independent runs. Initially, all cells have $\alpha \approx 0.907$, corresponding to the regular hexagonal shape. While the hard model maintains this value throughout, the soft and mid-hardness simulations exhibit a broader range of asphericities, reflecting increasing shape deformation at lower hardness.

6 Mean field limit

In the previous chapter, we employed Monte Carlo simulations to validate the microscopic cell model and to illustrate the qualitative behaviour emerging from the underlying gradient flow dynamics. While such simulations provide valuable empirical insight, they do not yet reveal the macroscopic, continuum level structure of the system.
In this chapter, we therefore develop a systematic framework to pass from individual cell dynamics to a description in terms of probability measures and densities. We introduce the empirical measure associated with large ensembles of simulated cells, explain its convergence to the first marginal of the N_C cell system, and analyse the subsequent mean field limit as $N_C \to \infty$.
This leads naturally to a deterministic transport equation for the limiting cell density, which provides a mathematically transparent interpretation of the collective dynamics. We illustrate this approach with an explicit low-dimensional needle cell example. Afterwards, we compute the mean field PDEs for our DF model with sorely the area, edge or interior angle force applied, respectively. Unfortunately, we did not manage to do the same for our interaction term in this thesis, as there was not enough time.
In order to analyse the macroscopic system dynamic, we start by studying the transition from the microscopic point particle model with $N_C \in \mathbb{N}$ particles to a macroscopic density description where $N_C \to \infty$. We want to consider the point particle Monte Carlo simulations from Chapter 5. Therefore, we define the empirical measure

$$\begin{aligned} \mu_t^{(N_C,N_S)} : \mathcal{B}(\mathbb{R}^2) &\longrightarrow [0,1], \\ A \longmapsto \mu_t^{(N_C,N_S)}(A) &= \frac{1}{N_C N_S} \sum_{i=1}^{N_C} \sum_{s=1}^{N_S} \delta_{\vec{x}_i^{(s)}(t)}(A), \end{aligned}$$

where N_C is again the number of cells in each simulation, N_S denotes the number of simulations in the Monte Carlo simulation and $\vec{x}_i^{(s)}(t) \in \Omega \subset \mathbb{R}^2$ is the location of point particle $1 \leqslant i \leqslant N_C$ in simulation $1 \leqslant s \leqslant N_S$ and at time $t \in [0,T]$.
$\mathcal{B}(\mathbb{R}^2)$ is the Borel σ-algebra on $\mathbb{R}^2$ and $\delta_{\vec{x}_i(t)}$ denotes the Dirac measure:

$$\begin{aligned} \delta_{\vec{x}_i(t)} : \mathcal{B}(\mathbb{R}^2) &\longrightarrow \{0,1\}, \\ A \longmapsto \delta_{\vec{x}_i(t)}(A) &= \begin{cases} 1 & \text{if } \vec{x}_i(t) \in A, \\ 0 & \text{if } \vec{x}_i(t) \notin A. \end{cases} \end{aligned}$$

For any test function $\phi \in C_c^\infty(\mathbb{R}^2)$, the Dirac measure satisfies

$$\int_{\mathbb{R}^2} \phi(x) \, \mathrm{d}\delta_{\vec{x}_i(t)}(x) = \phi(\vec{x}_i(t)).$$

For a set $A \in \mathcal{B}(\mathbb{R}^2)$, the quantity $\mu_t^{(N_C,N_S)}(A)$ denotes the relative proportion of the N_C particles that lie in A at time t, averaged over all N_S simulations.
The heatmaps in Chapter 5 are obtained by evaluating this empirical measure on a family of subsets $\{A_{ij}\}_{i,j=1}^{N_H}$, where N_H^2 is the number ofsubsquares into which we

partition the domain $\Omega = [-0.5, 0.5]^2$. Each subsquare A_{ij} has side length $\frac{1}{N_H}$, and increasing N_H yields an increasingly fine spatial resolution of Ω.
If, in addition, we let $N_S \to \infty$, then by the Central Limit Theorem the empirical heatmaps converge to their expectation, which coincides with the first marginal distribution of the system, since the simulations are independent and identically distributed. In our computations we used 10 000 simulations to ensure a reliable numerical approximation of this first marginal.

The focus of this chapter lays in another limit, namely the mean field limit. Here, we let the number of cells $N_C \to \infty$. In the mean field limit, we can study the transition from the microscopic model view, that uses a finite number of cells, $N_C < \infty$, to a macroscopic model view where we consider the whole system's density without individual cells.
Figure 32 illustrates how empirical particle distributions converge towards their underlying mean field density as the number of particles increases. The first three panels depict empirical measures obtained from samples of size $N_C \in \{20, 200, 20\,000\}$, drawn independently from the same Gaussian distribution $\mathcal{N}_2((0,0), 0.09^2 \cdot I_2)$. To visualise these empirical measures, the domain $\Omega = [-0.5, 0.5]^2$ is partitioned into 50×50subsquares, on which the empirical measure

$$\mu^{N_C}(A) = \frac{1}{N_C} \sum_{i=1}^{N_C} \delta_{\vec{x}_i}(A),$$

is evaluated and displayed as a heatmap.
As the particle number grows, the empirical distribution becomes progressively smoother and more faithful to the true underlying density. For small N_C, random fluctuations dominate the visual appearance, while for larger N_C these fluctuations average out, and the heatmap increasingly resembles the continuous Gaussian density shown in the fourth panel.

In the preceding chapters, we wrote ρ for the first marginal, as the mean field limit was not under consideration. To emphasise the dependence on the number of cells N_C, we adopt the notation ρ^{N_C} throughout this chapter.
We now want to derive a temporal PDE for the time evolution of a point particles system's density. At time $t = 0$, the first marginal $\rho_0^{N_C}$ converges to the distribution where we sample the particles from, as $N_C \to \infty$, like illustrated in Figure 32. Thus, the initial mean field density $\bar{\rho}_0$ is given by the distribution where we sample the particles from.
As the first marginal distribution changes over time due to the particle dynamics, so does the density in the mean field limit. Let the particle dynamic be given by the gradient flow of a certain energy. For N_C point particles $\vec{x}_i(t) \in \mathbb{R}^2$, we use a cell wise energy

$$\begin{aligned} E : \mathbb{R}^2 &\longrightarrow \mathbb{R}, \\ \vec{x} &\longmapsto E(\vec{x}). \end{aligned}$$

We define the particle dynamic via

$$\frac{\mathrm{d}\vec{x}_i(t)}{\mathrm{d}t} = -\nabla E(\vec{x}_i(t)) \in \mathbb{R}^2, \quad 1 \leqslant i \leqslant N_C.$$

We consider the empirical measure

$$\mu_t^{N_C}(A) = \tfrac{1}{N_C} \sum_{i=1}^{N_C} \delta_{\vec{x}_i(t)}(A).$$

Let $\phi \in C_c^\infty(\mathbb{R}^2, \mathbb{R})$ be a test function with gradient field $\nabla\phi : \mathbb{R}^2 \to \mathbb{R}^2$.
We assume that $\mu_t^{N_C}$ has density $\rho_t^{N_C}$ ($\mathrm{d}\mu_t^{N_C} = \rho_t^{N_C}(x)\,\mathrm{d}x$) and we have convergence

$$\rho_t^{N_C} \xrightarrow{N_C\to\infty} \bar{\rho}_t.$$

First, we consider

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\int \phi(x)\,\mathrm{d}\mu_t^{N_C} &= \frac{\mathrm{d}}{\mathrm{d}t}\int \phi(x)\,\rho_t^{N_C}(x)\,\mathrm{d}x \\ &\xrightarrow{N_C\to\infty} \frac{\mathrm{d}}{\mathrm{d}t}\int \phi(x)\,\bar{\rho}_t(x)\,\mathrm{d}x \\ &= \int \phi(x)\,\frac{\partial\bar{\rho}_t(x)}{\partial t}\,\mathrm{d}x, \end{aligned} \tag{20}$$

using first the density $\rho_t^{N_C}$ of $\mu_t^{N_C}$ and then convergence $\rho_t^{N_C} \xrightarrow{N_C\to\infty} \bar{\rho}_t$.

figures//density/muplot_combined.png

Figure 32: Empirical measures for three increasing particle numbers: 20, 200, and 20 000, compared with the limiting density $\mathcal{N}_2((0,0), 0.09^2 \cdot I_2)$ shown in the fourth panel. Particles are sampled i.i.d. from the same distribution, and the empirical measure μ^{N_C} is visualised on a 50×50 grid ofsubsquares. As N_C increases, the empirical heatmaps become smoother and converge to the Gaussian density, demonstrating the transition towards the mean field limit. A consistent colour scale allows direct comparison across all subplots.

Then, we use the definition of the empirical measure to obtain

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\int \phi(x)\,\mathrm{d}\mu_t^{N_C} &= \frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{1}{N_C}\sum_{i=1}^{N_C}\phi(\vec{x}_i^{(s)}(t))\right] \\
&= -\frac{1}{N_C}\sum_{i=1}^{N_C}\nabla\phi(\vec{x}_i^{(s)}(t))\cdot\nabla E(\vec{x}_i^{(s)}(t)) \\
&= -\int \nabla\phi(x)\cdot\nabla E(x)\,\mathrm{d}\mu_t^{N_C} \\
&= -\int \nabla\phi(x)\cdot\nabla E(x)\,\rho_t^{N_C}(x)\,\mathrm{d}x \\
&\xrightarrow{N_C\to\infty} -\int \nabla\phi(x)\cdot\nabla E(x)\,\bar{\rho}_t(x)\,\mathrm{d}x \\
&= \int \phi(x)\nabla\cdot(\bar{\rho}_t(x)\nabla E(x))\,\mathrm{d}x.
\end{aligned} \tag{21}
$$

We consider the difference from Equations (20) and (21) for $N_C \to \infty$

$$
\begin{aligned}
0 &= \int \phi(x)\,\frac{\partial\bar{\rho}_t(x)}{\partial t}\,\mathrm{d}x - \int \phi(x)\nabla\cdot(\bar{\rho}_t(x)\nabla E(x))\,\mathrm{d}x \\
&= \int \phi(x)\left(\frac{\partial\bar{\rho}_t(x)}{\partial t} - \nabla\cdot(\bar{\rho}_t(x)\nabla E(x))\right)\,\mathrm{d}x.
\end{aligned}
$$

As this holds true for any $\phi \in C_c^\infty(\mathbb{R}^2,\mathbb{R})$, we obtain the mean field PDE

$$
\frac{\partial\bar{\rho}_t(x)}{\partial t} - \nabla\cdot(\bar{\rho}_t(x)\nabla E(x)) = 0. \tag{22}
$$

For a force function $F(C) = -\nabla_C E(C)$, the we get the formulation

$$
\frac{\partial\bar{\rho}_t(x)}{\partial t} + \nabla\cdot(\bar{\rho}_t(x)F(x)) = 0. \tag{23}
$$

This mean field PDE reveals how the macroscopic density changes over time, as the collective effect of the microscopic particle dynamics driven by the energy E.

6.1 1d needles example

In order to demonstrate how the preceding computations can be applied in practice, we begin with a lower-dimensional example. In this setting, each cell consists of two vertices situated in one spatial dimension,

$$
C = \{v_1, v_2\}, \quad v_1, v_2 \in \mathbb{R}.
$$

We use the cell wise energy

$$
E(C) = \tfrac{1}{2}\left||v_1 - v_2| - E_d\right|^2,
$$

where E_d is the desired edge length of each cell. This energy lets each needle cell to recover a length of E_d. This is the one-dimensional analogue to the edge energy

from Equation (8) with $k = 2$ for needle cells with just one edge.
Applying gradient flow dynamics yields

$$\begin{aligned}\frac{\mathrm{d}v_1}{\mathrm{d}t} &= -\frac{\partial E(C)}{\partial v_1} \\ &= -\frac{\partial}{\partial v_1}\left[tfrac12||v_1 - v_2| - E_d|^2\right] \\ &= -(|v_1 - v_2| - E_d)\frac{\partial}{\partial v_1}\left[|v_1 - v_2|\right] \\ &= -\operatorname{sgn}(v_1 - v_2)(|v_1 - v_2| - E_d),\end{aligned}$$

and

$$\frac{\mathrm{d}v_2}{\mathrm{d}t} = \operatorname{sgn}(v_1 - v_2)(|v_1 - v_2| - E_d).$$

The resulting cell dynamic takes the compact form

$$\frac{\partial C}{\partial t} = -\nabla_C E(C) = F(C) = \underbrace{\operatorname{sgn}(v_1 - v_2)(|v_1 - v_2| - E_d)}_{=\alpha}\begin{pmatrix}-1\\1\end{pmatrix}.$$

For the subsequent derivation we require the computations

$$\begin{aligned}\frac{\partial \alpha}{\partial v_1} &= \frac{\partial}{\partial v_1}\left[\operatorname{sgn}(v_1 - v_2)(|v_1 - v_2| - E_d)\right] \\ &= \frac{\partial}{\partial v_1}\left[\operatorname{sgn}(v_1 - v_2)\right](|v_1 - v_2| - E_d) + \operatorname{sgn}(v_1 - v_2)\frac{\partial}{\partial v_1}\left[|v_1 - v_2| - E_d\right] \\ &= 0 + \operatorname{sgn}(v_1 - v_2)\frac{\partial}{\partial v_1}\left[|v_1 - v_2|\right] \\ &= \operatorname{sgn}(v_1 - v_2)\operatorname{sgn}(v_1 - v_2) \\ &= 1,\end{aligned}$$

and

$$\begin{aligned}\frac{\partial \alpha}{\partial v_2} &= \frac{\partial}{\partial v_2}\left[\operatorname{sgn}(v_1 - v_2)(|v_1 - v_2| - E_d)\right] \\ &= \frac{\partial}{\partial v_2}\left[\operatorname{sgn}(v_1 - v_2)\right](|v_1 - v_2| - E_d) + \operatorname{sgn}(v_1 - v_2)\frac{\partial}{\partial v_2}\left[|v_1 - v_2| - E_d\right] \\ &= 0 + \operatorname{sgn}(v_1 - v_2)\frac{\partial}{\partial v_2}\left[|v_1 - v_2|\right] \\ &= \operatorname{sgn}(v_1 - v_2)(-\operatorname{sgn}(v_1 - v_2)) \\ &= -1.\end{aligned}$$

With these identities at hand, we compute the divergence

$$\begin{aligned}
\nabla_C \cdot (\bar{\rho} F) &= \frac{\partial}{\partial v_1}\left[\bar{\rho} F_1\right] + \frac{\partial}{\partial v_2}\left[\bar{\rho} F_2\right] \\
&= \frac{\partial}{\partial v_1}\left[\bar{\rho}\right] F_1 + \bar{\rho}\frac{\partial}{\partial v_1}\left[F_1\right] + \frac{\partial}{\partial v_2}\left[\bar{\rho}\right] F_2 + \bar{\rho}\frac{\partial}{\partial v_2}\left[F_2\right] \\
&= -\frac{\partial}{\partial v_1}\left[\bar{\rho}\right]\alpha - \bar{\rho}\frac{\partial}{\partial v_1}\left[\alpha\right] + \frac{\partial}{\partial v_2}\left[\bar{\rho}\right]\alpha + \bar{\rho}\frac{\partial}{\partial v_2}\left[\alpha\right] \\
&= -\frac{\partial \bar{\rho}}{\partial v_1}\alpha - \bar{\rho} + \frac{\partial \bar{\rho}}{\partial v_2}\alpha - \bar{\rho} \\
&= -2\bar{\rho} + \alpha\left(-\frac{\partial \bar{\rho}}{\partial v_1} + \frac{\partial \bar{\rho}}{\partial v_2}\right),
\end{aligned}$$

where $F = (F_1, F_2)^T$ and $\bar{\rho}$ is the density in the mean field limit.
Consequently, in the mean field limit $N_C \to \infty$, the density $\bar{\rho}$ satisfies

$$\frac{\partial \bar{\rho}}{\partial t} - 2\bar{\rho} + \alpha\left(-\frac{\partial \bar{\rho}}{\partial v_1} + \frac{\partial \bar{\rho}}{\partial v_2}\right) = 0,$$

according to Equation (23).
To illustrate this model, we present two corresponding simulations. First, we consider a finite system of $N_C = 400$ needle cells. Each needle is initially sampled from

$$C_i = (v_1^i, v_2^i) \sim \mathcal{N}_2((0.5, 0.5)^T, 0.09^2 \cdot I_2), \quad 1 \leqslant i \leqslant 400.$$

We then evolve the system under the dynamics

$$\frac{\partial C}{\partial t} = F(C) = \operatorname{sgn}(v_1 - v_2)(|v_1 - v_2| - E_d)(-1, 1)^T,$$

using a desired edge length of $E_d = 0.2$. The resulting ODE system is integrated with an explicit Euler method using a time step of $\Delta t = 10^{-3}$ over the time interval $[0, 1]$, giving 100 time steps.
Figure 33 visualises the evolution of the one-dimensional needle cell system. Each cell is represented by a blue point in the (v_1, v_2) plane, where the horizontal axis corresponds to the position of the first vertex and the vertical axis to that of the second vertex. Such a representation is only possible in this lower-dimensional setting, since the full vertex configuration of a cell can be embedded in $\mathbb{R}^2$.
At initial time, the $N_C = 400$ cells are sampled from $\mathcal{N}_2((0.5, 0.5), 0.09^2 \cdot I_2)$. As time progresses, the dynamics drive each cell towards its desired edge length $E_d = 0.2$. In the scatter plot, this manifests as a gradual migration of points towards the two diagonal lines defined by $|v_1 - v_2| = 0.2$, corresponding to cells whose vertex separation has achieved the desired value. By the final time, all cells lie precisely on these two diagonals, confirming that the system converges to a configuration in which every cell has relaxed to the target length.

Subsequently, we examine the associated mean field dynamics. We choose an initial condition given by $\mathcal{N}_2((0.5, 0.5)^T, 0.09^2 \cdot I_2)$ on the domain $\Omega = [0.0, 1.0]^2$ and evolve it using the PDE

$$\frac{\partial \bar{\rho}}{\partial t} - \frac{\partial}{\partial v_1}[\bar{\rho}\alpha] + \frac{\partial}{\partial v_2}[\bar{\rho}\alpha] = 0,$$

which arises from the earlier derivation of the density evolution equation. The discretisation is given by

$$\begin{aligned}
\Omega &\longrightarrow \{A_{ij}\}_{i,j=1}^{500}\text{subsquares}, \\
\bar{\rho} &\longrightarrow \bar{\rho}_{ij}^k \text{ density value on } A_{ij} \text{ at time step } k \in \mathbb{N}, \\
\partial_t \bar{\rho} &\longrightarrow \frac{\bar{\rho}_{ij}^{k+1} - \bar{\rho}_{ij}^k}{\Delta t}, \\
\alpha &\longrightarrow \alpha_{ij}^k \text{ value on } A_{ij} \text{ at time step } k \in \mathbb{N}, \\
-\frac{\partial}{\partial v_1}[\bar{\rho}\alpha] &\longrightarrow \frac{-\bar{\rho}_{i,j+1}^k \alpha_{i,j+1}^k + \bar{\rho}_{i,j-1}^k \alpha_{i,j-1}^k}{2\Delta x}, \\
\frac{\partial}{\partial v_2}[\bar{\rho}\alpha] &\longrightarrow \frac{\bar{\rho}_{i+1,j}^k \alpha_{i+1,j}^k - \bar{\rho}_{i-1,j}^k \alpha_{i-1,j}^k}{2\Delta x},
\end{aligned}$$

with grid spacing $\Delta x = \frac{1}{500}$.

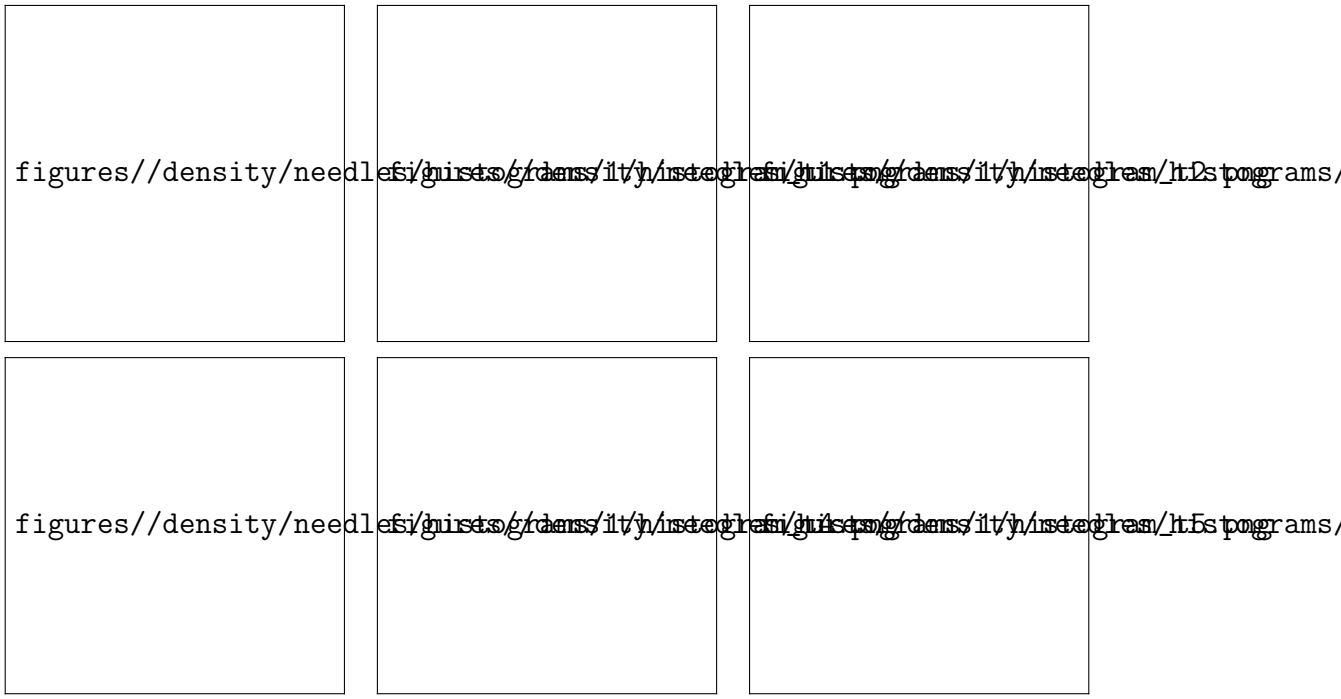


Figure 33: Scatter plots showing the evolution of $N_C = 400$ one-dimensional needle cells at times $t \in \{0.0, 0.2, \ldots, 1.0\}$. Each blue point represents a single cell, with the horizontal axis indicating the location of the first vertex v_1 and the vertical axis the location of the second vertex v_2. The initial conditions are drawn from $\mathcal{N}_2((0.5, 0.5), 0.09^2 \cdot I_2)$. The dynamics aim to achieve a desired edge length of $E_d = 0.2$, corresponding to the two diagonal lines defined by $|v_1 - v_2| = 0.2$.

Figure 34: Evolution of the mean field density starting from the initial distribution $\mathcal{N}_2((0.5, 0.5), 0.09^2 \cdot I_2)$ on the domain $[0, 1]^2$. The PDE dynamics transport the density towards the two diagonal lines defined by $|v_1 - v_2| = 0.2$, along which the mass becomes concentrated over time. The plots display the density at successive time instances and include small oscillations between the diagonals arising from the central difference discretisation; these could be removed through an upwind treatment of the spatial gradients.

Figure 34 illustrates the evolution of the cell density in the mean field limit. The initial distribution is given by $\mathcal{N}_2((0.5, 0.5), 0.09^2 \cdot I_2)$ on the domain $[0, 1]^2$, forming a concentrated region in the centre of the domain. Under the action of the mean field PDE, the density is gradually transported towards the two diagonal lines characterised by $|v_1 - v_2| = 0.2$, mirroring the behaviour observed in the finite particle simulation from Figure 33. As time evolves, the solution develops sharp ridges along these diagonals, and by the final time almost all mass is concentrated on there, indicating convergence towards the desired edge length in the continuum description. This example is a very nice depiction of how a first marginal density in a N_C cell system (microscopic view) corresponds to its mean field density (macroscopic view). It also validates our derivation of the mean field PDE from Equation (22).
During the evolution, small oscillations appear in the region between the two diagonals. These artefacts are purely numerical and originate from the central difference discretisation used for the spatial derivatives. They can be removed by employing an upwind scheme, which would provide the correct directional bias in the discretisation of the fluxes and thereby suppress non-physical oscillations.
Both simulations exhibit exclusion effects that remind of the behaviour observed in the earlier hard disc model shown in Figure 3. In the hard disc example, the diagonal region of the domain was inaccessible because any placement of the second disc in that area would necessarily lead to overlap with the first disc. This exclusion was enforced explicitly: overlap was prohibited both by the initial configuration and

by the dynamics, which strictly prevented discs from entering the forbidden region. The geometric volume exclusion in the hard disc system corresponds to the desired state constraint in the needle model: whereas discs must avoid overlaps, needles are driven toward configurations in which their length equals E_d, and both mechanisms create analogous forbidden regions in the state space. The needle model does not impose such exclusions a priori. Initially, needle cell configurations with lengths unequal to E_d are allowed. Nevertheless, the dynamics, driven to reduce the energy

$$E(C) = \tfrac{1}{2}||v_1 - v_2| - E_d|^2,$$

naturally steer the system toward states where the needle length matches E_d. As the evolution proceeds, the density gradually depletes along the diagonal region of the (v_1, v_2) space, reproducing the same 'forbidden' diagonal line that appeared in the hard disc case.
Thus, while the hard disc model exhibits a hard exclusion (geometric non-overlap), the needle system develops a soft exclusion: the dynamics energetically penalise undesirable configurations until the system self organises into a feasible state.

6.2 DF model mean field PDE

Now, we want to transfer the computation for PDE (22) to our DF model. Therefore, we adapt our empirical measure. In the DF model, we model cells as polygons with $N_V \in \mathbb{N}$ vertices. Thus, we need to use δ_{C_i} that uses high-dimensional subsets $A \subset \mathbb{R}^{2N_V}$, i.e.

$$\begin{aligned} \mu_t^{(N_C,N_V)} &: \mathcal{B}(\mathbb{R}^{2N_V}) \longrightarrow [0,1], \\ A \longmapsto \mu_t^{(N_C,N_V)}(A) &= \frac{1}{N_C}\sum_{i=1}^{N_C} \delta_{C_i(t)}(A), \end{aligned}$$

where $C_i(t) = (\vec{v}_1^{\,i}, \ldots, \vec{v}_{N_V}^{\,i})$ is cell i at time t.
We use a cell wise energy

$$E : \mathbb{R}^{2N_V} \to \mathbb{R},$$

with gradient

$$\nabla E : \mathbb{R}^{2N_V} \to \mathbb{R}^{2N_V}.$$

Later on, we want to insert the three shape preserving energies, i.e. the area, edge and interior angle energy, that all have the above attributes. We can derive the mean field PDE for this setup as follows. Let $\phi \in C_c^\infty(\mathbb{R}^{2N_V}, \mathbb{R})$ be a test function on the higher-dimensional space $\mathbb{R}^{2N_V}$. We assume that the empirical measure $\mu_t^{(N_C,N_V)}$ has density $\rho_t^{(N_C,N_V)}$ that converges to the mean field density $\bar{\rho}_t^{N_V}$, as $N_C \to \infty$. A rigorous analysis establishing this convergence remains an interesting direction for

future work. Similarly to the calculation at the beginning of this chapter, we observe

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int \phi(C)\,\mathrm{d}\mu_t^{(N_C,N_V)} &= \frac{\mathrm{d}}{\mathrm{d}t}\int \phi(C)\,\rho_t^{(N_C,N_V)}(C)\,\mathrm{d}C \\ &\xrightarrow{N_C\to\infty} \frac{\mathrm{d}}{\mathrm{d}t}\int \phi(C)\,\bar{\rho}_t^{N_V}(C)\,\mathrm{d}C \\ &= \int \phi(C)\,\frac{\partial}{\partial t}\left[\bar{\rho}_t^{N_V}(C)\right]\,\mathrm{d}C,\end{aligned}$$

and

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int \phi(C)\,\mathrm{d}\mu_t^{(N_C,N_V)} &= \frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{1}{N_C}\sum_{i=1}^{N_C}\phi(C_i(t))\right] \\ &= -\frac{1}{N_C}\sum_{i=1}^{N_C}\nabla\phi(C_i(t))\cdot\nabla E(C_i(t)) \\ &= -\int \nabla\phi(C)\cdot\nabla E(C)\,\mathrm{d}\mu_t^{(N_C,N_V)} \\ &= -\int \nabla\phi(C)\cdot\nabla E(C)\,\rho_t^{(N_C,N_V)}(C)\,\mathrm{d}C \\ &\xrightarrow{N_C\to\infty} -\int \nabla\phi(C)\cdot\nabla E(C)\,\bar{\rho}_t^{N_V}(C)\,\mathrm{d}C \\ &= \int \phi(C)\nabla\cdot(\bar{\rho}_t^{N_V}(C)\nabla E(C))\,\mathrm{d}C.\end{aligned}$$

We can conclude, with the same derivation as before, that

$$\frac{\partial\bar{\rho}_t^{N_V}(C)}{\partial t} - \nabla\cdot(\bar{\rho}_t^{N_V}(C)\nabla E(C)) = 0. \tag{24}$$

For each energy E, we need a representation of the cell wise gradient $\nabla_C E(C)$ that we can plug into the mean field PDE from Equation (24). Therefore, we define the vector of all vertex coordinates of cell C as

$$V_C = (v_1^x, v_1^y, \ldots, v_{N_V}^x, v_{N_V}^y)^T \in \mathbb{R}^{2N_V}.$$

We compute the according matrices that enable as a gradient representation with matrix multiplication for whole cells, and not vertex wise, like already computed in the previous chapters.

Area energy

First, we consider pure scaled area energy, i.e.

$$E_A(C) = \alpha_A A_2(C) = \frac{\alpha_A}{2}|A_C - A_d|^2,$$

with $A_2(C)$ defined at Equation (6) with $k = 2$ and a constant scaling factor $\alpha_A > 0$. This energy has the gradient

$$\nabla_C E_A(C) = \begin{pmatrix} \nabla_{\vec{v}_1} \\ \vdots \\ \nabla_{\vec{v}_{N_V}} \end{pmatrix} E_A(C),$$

with

$$\begin{aligned} \nabla_{\vec{v}_j} E_A(C) &= \alpha_A \nabla_{\vec{v}_j} A_2(C) \\ &= \frac{\alpha_A}{2}(A_C - A_d) \begin{pmatrix} v^y_{j+1} - v^y_{j-1} \\ v^x_{j-1} - v^x_{j+1} \end{pmatrix}, \end{aligned}$$

where we used

$$\nabla_{\vec{v}_j} A_2(C) = \frac{1}{2}(A_C - A_d) \begin{pmatrix} v^y_{j+1} - v^y_{j-1} \\ v^x_{j-1} - v^x_{j+1} \end{pmatrix},$$

from Equation (7).
We conclude the matrix $M_A \in \mathbb{R}^{2N_V \times 2N_V}$ as

$$M_A = \begin{pmatrix} 0 & 0 & \boxed{\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}} & 0 & 0 & \cdots & 0 & 0 & \boxed{\begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix}} \\ 0 & 0 & & 0 & 0 & & 0 & 0 & \\ \boxed{\begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix}} & & 0 & 0 & \boxed{\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}} & 0 & 0 & \cdots & 0 & 0 \\ & & 0 & 0 & & 0 & 0 & & 0 & 0 \\ & & \ddots & & & \ddots & & & \\ 0 & 0 & & \cdots & 0 & 0 & \boxed{\begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix}} & 0 & 0 & \boxed{\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}} \\ 0 & 0 & & & 0 & 0 & & 0 & 0 & \\ \boxed{\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}} & & 0 & 0 & & 0 & 0 & \boxed{\begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix}} & 0 & 0 \\ & & 0 & 0 & \cdots & 0 & 0 & & 0 & 0 \end{pmatrix},$$

such that, we can write down the cell wise gradient as

$$\nabla_C E_A(C) = \frac{\alpha_A}{2}(A_C - A_d) \begin{pmatrix} v^y_2 - v^y_{N_V} \\ v^x_{N_V} - v^x_2 \\ \vdots \\ v^y_1 - v^y_{N_V - 1} \\ v^x_{N_V - 1} - v^x_1 \end{pmatrix} = \frac{\alpha_A}{2}(A_C - A_d) M_A V_C. \tag{25}$$

Edge energy

Now, we want to consider the exclusive scaled edge energy, i.e.

$$E_E(C) = \alpha_E E_2(C) = \frac{\alpha_E}{2} \sum_{j=1}^{N_V} |E_C^j - E_d^j|^2,$$

with actual edge length $E_C^j = \|\vec{v}_j - \vec{v}_{j+1}\|_2$ and desired length E_d^j at edge j. We took $E_2(C)$ from Equation (8) with $k = 2$. In this case we can compute the gradient as

$$\nabla_C E_E(C) = \begin{pmatrix} \nabla_{\vec{v}_1} \\ \vdots \\ \nabla_{\vec{v}_{N_V}} \end{pmatrix} E_E(C),$$

with

$$\begin{aligned} \nabla_{\vec{v}_j} E_E(C) &= \alpha_E \nabla_{\vec{v}_j} E_2(C) \\ &= \alpha_E \left(\frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} \begin{pmatrix} v_j^x - v_{j-1}^x \\ v_j^y - v_{j-1}^y \end{pmatrix} + \frac{E_C^j - E_d^j}{E_C^j} \begin{pmatrix} v_j^x - v_{j+1}^x \\ v_j^y - v_{j+1}^y \end{pmatrix} \right), \end{aligned}$$

where we used

$$\nabla_{\vec{v}_j} E_2(C) = \frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} \begin{pmatrix} v_j^x - v_{j-1}^x \\ v_j^y - v_{j-1}^y \end{pmatrix} + \frac{E_C^j - E_d^j}{E_C^j} \begin{pmatrix} v_j^x - v_{j+1}^x \\ v_j^y - v_{j+1}^y \end{pmatrix},$$

from Equation (9).
If we define matrices for the prefactors: $D_E, D_{E-} \in \mathbb{R}^{2N_V \times 2N_V}$ as

$$D_E = \operatorname{diag}\left(\frac{E_C^1 - E_d^1}{E_C^1}, \frac{E_C^1 - E_d^1}{E_C^1}, \ldots, \frac{E_C^{N_V} - E_d^{N_V}}{E_C^{N_V}}, \frac{E_C^{N_V} - E_d^{N_V}}{E_C^{N_V}} \right),$$

and

$$D_{E-} = \operatorname{diag}\left(\frac{E_C^{N_V} - E_d^{N_V}}{E_C^{N_V}}, \frac{E_C^{N_V} - E_d^{N_V}}{E_C^{N_V}}, \frac{E_C^1 - E_d^1}{E_C^1}, \frac{E_C^1 - E_d^1}{E_C^1}, \ldots, \frac{E_C^{N_V-1} - E_d^{N_V-1}}{E_C^{N_V-1}} \right).$$

And rewrite the following parts as

$$M_E = \begin{pmatrix} \boxed{\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}} & \boxed{\begin{matrix} -1 & 0 \\ 0 & -1 \end{matrix}} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \cdots & & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ & \ddots & \ddots & & & \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & & & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \boxed{\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}} & \boxed{\begin{matrix} -1 & 0 \\ 0 & -1 \end{matrix}} \\ \boxed{\begin{matrix} -1 & 0 \\ 0 & -1 \end{matrix}} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \cdots & & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \boxed{\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}} \end{pmatrix} \in \mathbb{R}^{2N_V \times 2N_V},$$

and

$$M_{E-} = \begin{pmatrix} \boxed{\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \cdots & & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \boxed{\begin{matrix} -1 & 0 \\ 0 & -1 \end{matrix}} \\ \boxed{\begin{matrix} -1 & 0 \\ 0 & -1 \end{matrix}} & \boxed{\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \cdots & & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ & & \ddots & \ddots & & \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & & \cdots & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \boxed{\begin{matrix} -1 & 0 \\ 0 & -1 \end{matrix}} & \boxed{\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}} \end{pmatrix} \in \mathbb{R}^{2N_V \times 2N_V}.$$

We can conclude a cell wise edge energy gradient as

$$\nabla_C E_E(C) = \alpha_E \left(D_{E-} M_{E-} + D_E M_E \right) V_C. \tag{26}$$

Interior angle energy

Next, we consider a system with pure scaled interior angle energy, i.e.

$$E_I(C) = \alpha_I I_2(C) = \frac{\alpha_I}{2} \sum_{j=1}^{N_V} |I_C^j - I_d^j|^2,$$

where we used

$$I_2(C) = \sum_{j=1}^{N_V} \frac{1}{2} |I_C^j - I_d^j|^2,$$

from Equation (10) with $k = 2$.
In Chapter 4, we already computed the first derivative as

$$\begin{aligned} \nabla_{\vec{v}_j} I_2(C) = (I_C^{j-1} - I_d^{j-1}) & \left(-\frac{1}{(E_C^j)^2} \begin{pmatrix} v_{j-1}^y - v_j^y \\ v_j^x - v_{j-1}^x \end{pmatrix} \right) \\ & + (I_C^j - I_d^j) \left(\frac{1}{(E_C^j)^2} \begin{pmatrix} v_{j-1}^y - v_j^y \\ v_j^x - v_{j-1}^x \end{pmatrix} - \frac{1}{(E_C^{j+1})^2} \begin{pmatrix} v_{j+1}^y - v_j^y \\ v_j^x - v_{j+1}^x \end{pmatrix} \right) \\ & + (I_C^{j+1} - I_d^{j+1}) \left(\frac{1}{(E_C^{j+1})^2} \begin{pmatrix} v_{j+1}^y - v_j^y \\ v_j^x - v_{j+1}^x \end{pmatrix} \right), \end{aligned}$$

in Equation (11).
We define the matrices $D_I, D_{I-}, D_{I+} \in \mathbb{R}^{2N_V \times 2N_V}$ as

$$D_I = \operatorname{diag}\big(I_C^1 - I_d^1, I_C^1 - I_d^1, I_C^2 - I_d^2, I_C^2 - I_d^2, \dots, I_C^{N_V} - I_d^{N_V}, I_C^{N_V} - I_d^{N_V}\big),$$
$$D_{I-} = \operatorname{diag}\big(I_C^{N_V} - I_d^{N_V}, I_C^{N_V} - I_d^{N_V}, I_C^1 - I_d^1, I_C^1 - I_d^1, \dots, I_C^{N_V-1} - I_d^{N_V-1}, I_C^{N_V-1} - I_d^{N_V-1}\big),$$
$$D_{I+} = \operatorname{diag}\big(I_C^2 - I_d^2, I_C^2 - I_d^2, \dots, I_C^{N_V} - I_d^{N_V}, I_C^{N_V} - I_d^{N_V}, I_C^1 - I_d^1, I_C^1 - I_d^1\big).$$

The terms of form $\frac{1}{(E_C^j)^2}$ and $\frac{1}{(E_C^{j+1})^2}$ are summerised in $B_I, B_{I+} \in \mathbb{R}^{2N_V \times 2N_V}$ as,

$$B_I = \operatorname{diag}\big((E_C^1)^{-2}, (E_C^1)^{-2}, \dots, (E_C^{N_V})^{-2}, (E_C^{N_V})^{-2}\big),$$
$$B_{I+} = \operatorname{diag}\big((E_C^2)^{-2}, (E_C^2)^{-2}, \dots, (E_C^{N_V})^{-2}, (E_C^{N_V})^{-2}, (E_C^1)^{-2}, (E_C^1)^{-2}\big).$$

Finally, we put together missing vertex terms with matrices $M_{I-}, M_{I+} \in \mathbb{R}^{2N_V \times 2N_V}$ as

$$M_{I-} = \begin{pmatrix}
\boxed{\begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix}} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \cdots & & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \boxed{\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}} \\
\boxed{\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}} & \boxed{\begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix}} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \cdots & & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\
 & & \ddots & \ddots & & \\
\begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & & & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \boxed{\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}} & \boxed{\begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix}}
\end{pmatrix},$$

$$M_{I+} = \begin{pmatrix}
\boxed{\begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix}} & \boxed{\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \cdots & & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\
 & & \ddots & \ddots & & \\
\begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & & & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \boxed{\begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix}} & \boxed{\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}} \\
\boxed{\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \cdots & & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \boxed{\begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix}}
\end{pmatrix}.$$

We can write down the cell wise interior angle gradient as

$$\nabla_C E_I(C) = \alpha_I \left(-D_{I-} B_I M_{I-} + D_I \left(B_I M_{I-} - B_{I+} M_{I+}\right) + D_{I+} B_{I+} M_{I+}\right) V_C. \tag{27}$$

Shape preserving energies

If we now consider a combination of the three shape preserving energies, i.e.

$$E_{\text{shape}}(C) = E_A(C) + E_E(C) + E_I(C),$$

we can use the linearity of the gradient to easily write down

$$\nabla_C E_{\text{shape}}(C) = \nabla_C E_A(C) + \nabla_C E_E(C) + \nabla_C E_I(C). \tag{28}$$

The three gradients are given in the Equations (25), (26) and (27). The mean field PDE for the DF model with shape preserving energies is then given by inserting the above gradient into Equation (24), i.e.

$$\frac{\partial \bar{\rho}_t^{N_V}(C)}{\partial t} - \nabla \cdot (\bar{\rho}_t^{N_V}(C) (\nabla_C E_A(C) + \nabla_C E_E(C) + \nabla_C E_I(C))) = 0.$$

7 Conclusion

In this work, we develop a vertex based discrete form (DF) model that captures the behavior of deformable cells with realistic mechanical interactions, including shape preservation, cell-cell overlap resolution, and stochastic motion.

8 Outlook

An interesting extension of the current model would involve assigning individual desired states to each cell, in contrast to the uniform desired state used throughout this study. This modification would naturally lead to cell-specific energies and corresponding forces, as both would depend on the unique desired configuration of each cell. Incorporating such heterogeneity could allow the model to capture more complex biological behaviors, such as differentiation, cell-type-specific migration, or adaptive responses to environmental cues. - incorporate external force f,
- curved surfaces - 3d - cell division - more parameter studies - use more vertices for more accurate shape and dynamics - limit $N_V \to \infty$ - overdamping - new shapes - Additionally, many vertex models incorporate rules that govern changes in connection among vertices, and therefore allow for changes in cell neighbor relationships. -¿ These approximations are suitable in the case of tightly packed cell sheets, where the intercellular space is negligible, and is based on experimental observations that cells in epithelial tissues are often arranged in polygonal or polyhedral structures (19) and can move around relative to other cells (20) - mention that doing the cross-section plot with a finer spatial resolution shows oscillations that are numerical artifacts - mean field limit for our overlap forces - existence of mean field limit density


figures//sanity-check/crosssections/hdx30/cross_section_00.png
figures//sanity-check/crosssections/hdx30/cross_section_01.png
figures//sanity-check/crosssections/hdx30/cross_section_02.png
figures//sanity-check/crosssections/hdx30/cross_section_03.png
figures//sanity-check/crosssections/hdx30/cross_section_04.png

Figure 35

Appendix

This appendix provides additional computations and intermediate analyses that were carried out during the course of this study but did not ultimately form part of the main body of the thesis. Their inclusion here aims to offer a more comprehensive account of the exploratory steps and supplementary considerations that informed the final results. Moreover, the author acknowledges the use of AI-based language tools, in particular chatbots [Ope25, DDNMK24], which were employed exclusively to improve wording and clarity. All scientific content, including mathematical derivations, modelling arguments, and analytical reasoning, was produced and rigorously validated by the author.

Computation of $\nabla_{\vec{v}_j} \cdot \nabla_{\vec{v}_j} A_2(C)$

We consider pure area energy

$$A_2(C) = \frac{1}{2}|A_C - A_d|^2,$$

defined at Equation (6) with $k = 2$.
The area gradient is given by

$$\nabla_{\vec{v}_j} A_2(C) = \frac{1}{2}(A_C - A_d) \begin{pmatrix} v_{j+1}^y - v_{j-1}^y \\ v_{j-1}^x - v_{j+1}^x \end{pmatrix},$$

in Equation (7).

We compute the second derivative

$$\begin{aligned}
\nabla_{\vec{v}_j} \cdot \nabla_{\vec{v}_j} A_2(C) &= \nabla_{\vec{v}_j} \cdot \left[\frac{1}{2}(A_C - A_d) \begin{pmatrix} v^y_{j+1} - v^y_{j-1} \\ v^x_{j-1} - v^x_{j+1} \end{pmatrix} \right] \\
&= \frac{1}{2} \left(\frac{\partial}{\partial v^x_j} \left[(A_C - A_d)(v^y_{j+1} - v^y_{j-1}) \right] + \frac{\partial}{\partial v^y_j} \left[(A_C - A_d)(v^x_{j-1} - v^x_{j+1}) \right] \right) \\
&= \frac{1}{2} \left((v^y_{j+1} - v^y_{j-1}) \frac{\partial}{\partial v^x_j} \left[A_C \right] + (v^x_{j-1} - v^x_{j+1}) \frac{\partial}{\partial v^y_j} \left[A_C \right] \right) \\
&= \frac{1}{2} \Bigg((v^y_{j+1} - v^y_{j-1}) \frac{\partial}{\partial v^x_j} \left[\frac{1}{2} \sum_{l=1}^{N} (v^x_l v^y_{l+1} - v^x_{l+1} v^y_l) \right] + \\
&\quad + (v^x_{j-1} - v^x_{j+1}) \frac{\partial}{\partial v^y_j} \left[\frac{1}{2} \sum_{l=1}^{N} (v^x_l v^y_{l+1} - v^x_{l+1} v^y_l) \right] \Bigg) \\
&= \frac{1}{2} \Bigg((v^y_{j+1} - v^y_{j-1}) \frac{1}{2} \frac{\partial}{\partial v^x_j} \left[(v^x_j v^y_{j+1} - v^x_{j+1} v^y_j) + (v^x_{j-1} v^y_j - v^x_j v^y_{j-1}) \right] + \\
&\quad + (v^x_{j-1} - v^x_{j+1}) \frac{1}{2} \frac{\partial}{\partial v^y_j} \left[(v^x_j v^y_{j+1} - v^x_{j+1} v^y_j) + (v^x_{j-1} v^y_j - v^x_j v^y_{j-1}) \right] \Bigg) \\
&= \frac{1}{4} \left((v^y_{j+1} - v^y_{j-1}) \left(v^y_{j+1} - v^y_{j-1} \right) + (v^x_{j-1} - v^x_{j+1})(v^x_{j-1} - v^x_{j+1}) \right) \\
&= \frac{1}{4} \left((v^y_{j+1} - v^y_{j-1})^2 + (v^x_{j-1} - v^x_{j+1})^2 \right) \\
&= \frac{1}{4} \| \vec{v}_{j+1} - \vec{v}_{j-1} \|_2^2.
\end{aligned}$$

Computation of $\nabla_{\vec{v}_j} \cdot \nabla_{\vec{v}_j} E_2(C)$

We consider pure edge energy

$$E_2(C) = \sum_{j=1}^{N_V} \frac{1}{2} |E^j_C - E^j_d|^2,$$

defined at Equation (8) with $k = 2$. The gradient is given in Equation (9) as

$$\nabla_{\vec{v}_j} E_2(C) = \frac{E^{j-1}_C - E^{j-1}_d}{E^{j-1}_C} \begin{pmatrix} v^x_j - v^x_{j-1} \\ v^y_j - v^y_{j-1} \end{pmatrix} + \frac{E^j_C - E^j_d}{E^j_C} \begin{pmatrix} v^x_j - v^x_{j+1} \\ v^y_j - v^y_{j+1} \end{pmatrix}.$$

For the following computation, we use the derivatives

$$\begin{aligned}
\frac{\partial}{\partial v_j^x} E_C^{j-1} &= \frac{\partial}{\partial v_j^x} \|\vec{v}_{j-1} - \vec{v}_j\|_2 \\
&= \frac{\partial}{\partial v_j^x} \left[\left((v_{j-1}^x - v_j^x)^2 + (v_{j-1}^y - v_j^y)^2 \right)^{\frac{1}{2}} \right] \\
&= \frac{1}{2\|\vec{v}_{j-1} - \vec{v}_j\|_2} \frac{\partial}{\partial v_j^x} \left[(v_{j-1}^x - v_j^x)^2 + (v_{j-1}^y - v_j^y)^2 \right] \\
&= -\frac{v_{j-1}^x - v_j^x}{\|\vec{v}_{j-1} - \vec{v}_j\|_2} \\
&= \frac{v_j^x - v_{j-1}^x}{E_C^{j-1}},
\end{aligned}$$

and similarly

$$\frac{\partial}{\partial v_j^x} E_C^j = \frac{v_j^x - v_{j+1}^x}{E_C^j}, \quad \frac{\partial}{\partial v_j^y} E_C^{j-1} = \frac{v_j^y - v_{j-1}^y}{E_C^{j-1}}, \quad \frac{\partial}{\partial v_j^y} E_C^j = \frac{v_j^y - v_{j+1}^y}{E_C^j}.$$

We also need

$$\begin{aligned}
\frac{\partial}{\partial v_j^x} \left[\frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} \right] &= \frac{\partial}{\partial v_j^x} \left[1 - \frac{E_d^{j-1}}{E_C^{j-1}} \right] \\
&= -E_d^{j-1} \frac{\partial}{\partial v_j^x} \left[(E_C^{j-1})^{-1} \right] \\
&= -E_d^{j-1} (-(E_C^{j-1})^{-2}) \frac{\partial}{\partial v_j^x} \left[E_C^{j-1} \right] \\
&= \frac{E_d^{j-1}}{(E_C^{j-1})^2} \frac{v_j^x - v_{j-1}^x}{E_C^{j-1}} \\
&= \frac{E_d^{j-1}}{(E_C^{j-1})^3} (v_j^x - v_{j-1}^x),
\end{aligned}$$

and, respectively

$$\frac{\partial}{\partial v_j^x} \left[\frac{E_C^j - E_d^j}{E_C^j} \right] = \frac{E_d^j}{(E_C^j)^3} (v_j^x - v_{j+1}^x), \quad \frac{\partial}{\partial v_j^y} \left[\frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} \right] = \frac{E_d^{j-1}}{(E_C^{j-1})^3} (v_j^y - v_{j-1}^y),$$

$$\frac{\partial}{\partial v_j^y} \left[\frac{E_C^j - E_d^j}{E_C^j} \right] = \frac{E_d^j}{(E_C^j)^3} (v_j^y - v_{j+1}^y).$$

We continue with the computation of the second derivative of the first summand in

the first derivative of the edge energy

$$
\begin{aligned}
\nabla_{\vec{v}_j} \cdot \left[\frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} \begin{pmatrix} v_j^x - v_{j-1}^x \\ v_j^y - v_{j-1}^y \end{pmatrix} \right] &= \\
&= \frac{\partial}{\partial v_j^x} \left[\frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} (v_j^x - v_{j-1}^x) \right] + \frac{\partial}{\partial v_j^y} \left[\frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} (v_j^y - v_{j-1}^y) \right] \\
&= \frac{\partial}{\partial v_j^x} \left[\frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} \right] (v_j^x - v_{j-1}^x) + \frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} \frac{\partial}{\partial v_j^x} \left[(v_j^x - v_{j-1}^x) \right] \\
&\quad + \frac{\partial}{\partial v_j^y} \left[\frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} \right] (v_j^y - v_{j-1}^y) + \frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} \frac{\partial}{\partial v_j^y} \left[(v_j^y - v_{j-1}^y) \right] \\
&= \frac{E_d^{j-1}}{(E_C^{j-1})^3} (v_j^x - v_{j-1}^x)(v_j^x - v_{j-1}^x) + \frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} \\
&\quad + \frac{E_d^{j-1}}{(E_C^{j-1})^3} (v_j^y - v_{j-1}^y)(v_j^y - v_{j-1}^y) + \frac{E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} \\
&= \frac{E_d^{j-1}}{(E_C^{j-1})^3} (\underbrace{(v_j^x - v_{j-1}^x)^2 + (v_j^y - v_{j-1}^y)^2}_{= (E_C^{j-1})^2}) + \frac{2(E_C^{j-1} - E_d^{j-1})}{E_C^{j-1}} \\
&= \frac{E_d^{j-1}}{E_C^{j-1}} + \frac{2(E_C^{j-1} - E_d^{j-1})}{E_C^{j-1}} \\
&= \frac{2E_C^{j-1} - E_d^{j-1}}{E_C^{j-1}} \\
&= 2 - \frac{E_d^{j-1}}{E_C^{j-1}}.
\end{aligned}
$$

The computation for the second summand is analogous and it yields

$$
\nabla_{\vec{v}_j} \cdot \left[\frac{E_C^{j} - E_d^{j}}{E_C^{j}} \begin{pmatrix} v_j^x - v_{j+1}^x \\ v_j^y - v_{j+1}^y \end{pmatrix} \right] = 2 - \frac{E_d^{j}}{E_C^{j}}.
$$

Overall, we get

$$
\nabla_{\vec{v}_j} \cdot \nabla_{\vec{v}_j} E_2(C) = 4 - \frac{E_d^{j-1}}{E_C^{j-1}} - \frac{E_d^{j}}{E_C^{j}}.
$$

With that result, we can finally compute the searched divergence

$$\begin{aligned}
\nabla_C \cdot (\bar{\rho}\, \nabla_C E_2(C)) &= \sum_{j=1}^{N_V} \nabla_{\vec{v}_j} \cdot (\bar{\rho}\, \nabla_{\vec{v}_j} E_2(C)) \\
&= \sum_{j=1}^{N_V} \left((\nabla_{\vec{v}_j} \cdot \bar{\rho}) \cdot \nabla_{\vec{v}_j} E_2(C) + \bar{\rho}\, (\nabla_{\vec{v}_j} \cdot \nabla_{\vec{v}_j} E_2(C)) \right) \\
&= \sum_{j=1}^{N_V} \left((\nabla_{\vec{v}_j} \cdot \bar{\rho}) \cdot \nabla_{\vec{v}_j} E_2(C) + \bar{\rho}\, (4 - \frac{E_d^{j-1}}{E_C^{j-1}} - \frac{E_d^j}{E_C^j}) \right) \\
&= \sum_{j=1}^{N_V} \left((\nabla_{\vec{v}_j} \cdot \bar{\rho}) \cdot \nabla_{\vec{v}_j} E_2(C) \right) + \bar{\rho}\, (4N_V - 2 \sum_{j=1}^{N_V} \frac{E_d^j}{E_C^j}),
\end{aligned}$$

because

$$\sum_{j=1}^{N_V} (4 - \frac{E_d^{j-1}}{E_C^{j-1}} - \frac{E_d^j}{E_C^j}) = 4N_V - 2 \sum_{j=1}^{N_V} \frac{E_d^j}{E_C^j},$$

as we still use circular indexing, e.g. $E_C^0 = E_C^{N_V}$.

References

[AT20] Ricard Alert and Xavier Trepat. Physical models of collective cell migration. *Annual Review of Condensed Matter Physics*, 11(1):77–101, 2020.

[BC12] Maria Bruna and S. Jonathan Chapman. Excluded-volume effects in the diffusion of hard spheres. *Phys. Rev. E*, 85:011103, Jan 2012.

[BCR17] Maria Bruna, S. Jonathan Chapman, and Martin Robinson. Diffusion of particles with short-range interactions. *SIAM Journal on Applied Mathematics*, 77(6):2294–2316, 2017.

[BEKS17] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

[BPA+21] Mathieu Besançon, Theodore Papamarkou, David Anthoff, Alex Arslan, Simon Byrne, Dahua Lin, and John Pearson. Distributions.jl: Definition and modeling of probability distributions in the juliastats ecosystem. *Journal of Statistical Software*, 98(16):1–30, 2021.

[BSY+18] Arman Boromand, Alexandra Signoriello, Fangfu Ye, Corey S. O'Hern, and Mark D. Shattuck. Jamming of deformable polygons. *Phys. Rev. Lett.*, 121:248003, Dec 2018.

[CCP+14] Danfeng Cai, Shann-Ching Chen, Mohit Prasad, Li He, Xiaobo Wang, Valerie Choesmel-Cadamuro, Jessica K Sawyer, Gaudenz Danuser, and Denise J Montell. Mechanical feedback through e-cadherin promotes direction sensing during collective cell migration. *Cell*, 157(5):1146–1159, 2014.

[CSR+23] Simon Christ, Daniel Schwabeneder, Christopher Rackauckas, Michael Krabbe Borregaard, and Thomas Breloff. Plots.jl – a user extendable plotting api for the julia programming language. 2023.

[CV15] Andrew G Clark and Danijela Matic Vignjevic. Modes of cancer cell invasion and the role of the microenvironment. *Curr Opin Cell Biol*, 36(13-22):10–1016, 2015.

[DDNMK24] Ali Doosthosseini, Jonathan Decker, Hendrik Nolte, and Julian M. Kunkel. SAIA, February 2024.

[DRSB+05] Olivia Du Roure, Alexandre Saez, Axel Buguin, Robert H Austin, Philippe Chavrier, Pascal Siberzan, and Benoit Ladoux. Force mapping in epithelial cell migration. *Proceedings of the National Academy of Sciences*, 102(7):2390–2395, 2005.

[FG09] Peter Friedl and Darren Gilmour. Collective cell migration in morphogenesis, regeneration and cancer. *Nature reviews Molecular cell biology*, 10(7):445–457, 2009.

[FNW+95] Peter Friedl, Peter B Noble, Paul A Walton, Dale W Laird, Peter J Chauvin, Roger J Tabah, Martin Black, and Kurt S Zänker. Migration of coordinated cell clusters in mesenchymal and epithelial cancer explants in vitro. *Cancer research*, 55(20):4557–4560, 1995.

[FOBS14] Alexander G. Fletcher, Miriam Osterfield, Ruth E. Baker, and Stanislav Y. Shvartsman. Vertex models of epithelial morphogenesis. *Biophysical Journal*, 106(11):2291–2304, 2014.

[GT18] Matthew Good and Xavier Trepat. Cell parts to complex processes, from the bottom up, 2018.

[Her32] Earl H Herrick. Mechanism of movement of epidermis, especially its melanophores, in wound healing, and behavior of skin grafts in frog tadpoles. *The Biological Bulletin*, 63(2):271–286, 1932.

[Hol14] SJ Holmes. The behavior of the epidermis of amphibians when cultivated outside the body. *J. exp. Zool*, 17(2):281–295, 1914.

[HV23] Lea Happel and Axel Voigt. Coordinated motion of epithelial layers on curved surfaces. *Cluster of Excellence, Physics of Life, TU Dresden*, 2023.

[JGS18] Frank Jülicher, Stephan W Grill, and Guillaume Salbreux. Hydrodynamic theory of active matter. *Reports on Progress in Physics*, 81(7):076601, 2018.

[MJR+13] M Cristina Marchetti, Jean-François Joanny, Sriram Ramaswamy, Tanniemola B Liverpool, Jacques Prost, Madan Rao, and R Aditi Simha. Hydrodynamics of soft active matter. *Reviews of modern physics*, 85(3):1143–1189, 2013.

[MOK10] Seita Miyazawa, Michitoshi Okamoto, and Shigeru Kondo. Blending of animal colour patterns by hybridization. *Nature communications*, 1(1):66, 2010.

[MU49] Nicholas Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949. PMID: 18139350.

[Ope25] OpenAI. Chatgpt (version gpt-5.1). `https://chat.openai.com`, 2025. Large language model.

[PJJ15] Jacques Prost, Frank Jülicher, and Jean-François Joanny. Active gel physics. *Nature physics*, 11(2):111–117, 2015.

[RCCT17] Pere Roca-Cusachs, Vito Conte, and Xavier Trepat. Quantifying forces in cell biology. *Nature cell biology*, 19(7):742–751, 2017.

[RN17] Christopher Rackauckas and Qing Nie. Differentialequations.jl – a performant and feature-rich ecosystem for solving differential equations in julia. *The Journal of Open Research Software*, 5(1), 2017. Exported from https://app.dimensions.ai on 2019/05/05.

[Sho14] ShoelaceFormula. Green's theorem and area of polygons. blogoverflow, June 2014. Published by: apnorton. URL: https://math.blogoverflow.com/2014/06/04/greens-theorem-and-area-of-polygons/. LAst accessed on 23.11.2023.

[Sho22] ShoelaceIlustration. Deriving the trapezoid formula. URL: https://commons.wikimedia.org/wiki/File:Trapez-formel-prinz.svg, January 2022. Published by user 'Ag2gaeh'. Last accessed on 23.11.2023.

[TWA+09] Xavier Trepat, Michael R Wasserman, Thomas E Angelini, Emil Millet, David A Weitz, James P Butler, and Jeffrey J Fredberg. Physical forces during collective cell migration. *Nature physics*, 5(6):426–430, 2009.

[Vog23] Tim Vogel. Modelling of cells and their dynamics. 2023.

[VT66] RB Vaughan and JP Trinkaus. Movements of epithelial cell sheets in vitro. *Journal of cell science*, 1(4):407–413, 1966.

[Wri23] James Wrigley. Distributed.jl. 2023.

[WV21] Dennis Wenzel and Axel Voigt. Multiphase field models for collective cell migration. *Physical Review E*, 104(5):054410, 2021.