

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: October 28, 2025

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

# Contents

<b>1</b>	<b>Sanity check</b>	<b>5</b>
1.1	Reference simulations: Bruna and Chapman (2012) . . . . .	5
1.2	Reproduction of reference results . . . . .	8
1.3	Shape deformation check . . . . .	8
<b>2</b>	<b>Conclusion</b>	<b>14</b>
<b>3</b>	<b>Outlook</b>	<b>15</b>

## 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
$\frac{\partial f}{\partial v}$	Partial derivative of a scalar function $f$ with respect to a one dimensional variable $v$
$\nabla_{\vec{v}} f$	Gradient $\nabla_{\vec{v}} f = (\frac{\partial f}{\partial v_1}, \dots, \frac{\partial f}{\partial v_n})^T$ , where $f$ is a scalar function and $\vec{v} = (v_1, \dots, v_n)^T$ is a multidimensional variable
$\nabla_{\vec{v}} \cdot F$	Divergence $\nabla_{\vec{v}} \cdot F = \frac{\partial F_1}{\partial v_1} + \dots + \frac{\partial F_n}{\partial v_n}$ , where $F = (F_1, \dots, F_m)^T$ is a vector valued function
$D_{\vec{v}} F$	Jacobian matrix $D_{\vec{v}} F = \begin{pmatrix} \frac{\partial F_1}{\partial v_1} & \dots & \frac{\partial F_1}{\partial v_n} \\ \vdots & & \vdots \\ \frac{\partial F_m}{\partial v_1} & \dots & \frac{\partial F_m}{\partial v_n} \end{pmatrix}$ ,
$\Delta f$	Laplacian of a scalar function $f$
$2e - 3$	Scientific notation for $2 \times 10^{-3} = 0.002$
$d\vec{B}_i(t)$	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. $[0, \infty)$
$\mathbb{N}$	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
$P(t; \vec{x}_1, \dots, \vec{x}_{N_C})$	Joined probability density function
$\rho(t; \vec{x})$	First marginal function

# 1 Sanity check

Having introduced our cell dynamics, we now want to take a look at the simulation results. Therefore, we aim to compare our simulation results to results from an established cell model from [BC12]. In [BC12] the diffusion dynamics of first a point particle model and second a hard sphere model is studied. Thereby, the two density distributions:

- the joint probability density function  $P(\vec{X}, t)$  of the system of all cell centres  $\vec{X}$  at time  $t$ ,
- the marginal distribution function of the first particle  $p(\vec{x}_1, t)$

play an important roll.

The joint probability density function  $P(\vec{X}, t)$  is a function describing the positions of all particles in the system, while the marginal distribution function  $p(\vec{x}_1, t)$  is a function describing only the position of the first particle.

It is sufficient to consider only the marginal distribution function of first particle, because all particle act similarly.

Gaining  $p(\vec{x}_1, t)$  from  $P(\vec{X}, t)$  is a big reduction of complexity, since we reduce from a high-dimensional PDE for  $P$  to a low-dimensional PDE for  $p$ . The marginal distribution function of first particle can always be determined via

$$p(\vec{x}_1, t) = \int P(\vec{X}, t) d\vec{x}_2 \dots d\vec{x}_{N_{BC}}.$$

## 1.1 Reference simulations: Bruna and Chapman (2012)

The most simple model that gets considered for the diffusion dynamics of cell systems is the point particle model. Here the cells get modeled with sizeless points that perform a Brownian motion on the domain.

Since the cells do not have a real size, no interaction between the cells can occur, since they will never hit upon each other.

The paper [BC12] analyses these dynamics on the domain

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

on which  $N_{BC} = 400$  particles are located.

The movement of each point particle  $\vec{x}_i$  in the simulation is given by the SDE

$$d\vec{x}_i(t) = \sqrt{2} dB_i(t), \quad 1 \leq i \leq N_{BC},$$

which describes a Brownian motion in  $\Omega_{BC}$ . The reflective boundary condition on  $\partial\Omega_{BC}$  is imposed. It is known, that the joint probability density of the particle system in this setup evolves according to the diffusion equation, i.e.

$$(1) \quad \frac{\partial P}{\partial t}(\vec{X}, t) = \Delta_{\vec{X}} P = \nabla_{\vec{X}} \cdot [\nabla_{\vec{X}} P]$$

inside of the domain.

Since all particles are independent, we can compute

$$(2) \quad P(\vec{X}) = \prod_{i=1}^{N_{BC}} p(\vec{x}_i, t).$$

Therefore, we obtain the marginal distribution function as

$$(3) \quad \frac{\partial p}{\partial t}(\vec{x}_1, t) = \Delta_{\vec{x}_1} p = \nabla_{\vec{x}_1} \cdot [\nabla_{\vec{x}_1} p].$$

A next step that results in the hard sphere cell model (HSCM) is to give the cell particles a real size.

Let  $0 < \epsilon \ll 1$  be the diameter of all cells that are now two dimensional discs with the same size. This changes the dynamics of the cells immense, since they now have chance to collide into each other which is a form of interaction.

The authors of [BC12] also did a simulation with the HSCM. The setting is as similar as possible to the point particle model, because a main goal of the paper was to compare the diffusion characteristics of both models. There are still  $N_{BC} = 400$  cells located on the domain.

The initial condition of both models follows a two dimensional normal distribution with the addition that the distance of each cell centre to all others is at least  $\epsilon$ . The used distribution  $\mathcal{N}_2 \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.09^2 & 0 \\ 0 & 0.09^2 \end{pmatrix} \right)$  has an integral of one over  $\Omega_{BC}$ .

We can compute this initial condition with Algorithm 1.1.

#### **Algorithm 1.1. Computation of the initial cell system**

1. Generate a point  $\vec{x} \sim \mathcal{N}_2 \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.09^2 & 0 \\ 0 & 0.09^2 \end{pmatrix} \right)$ .
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  $N_{BC}$  cell centres are found.

Since we do not want any overlap to occur during the whole simulation with the HSCM, the feasible domain for the whole cell system is not directly  $\Omega_{BC}^{N_{BC}}$ , but instead

$$\Omega_{BC}^\epsilon = \Omega_1^\epsilon \times \dots \times \Omega_{N_{BC}}^\epsilon, \\ \Omega_i^\epsilon = \Omega_{BC} \setminus (\cup_{j \neq i} B_\epsilon(\vec{x}_j)), \quad 1 \leq i \leq N_{BC},$$

where  $B_\epsilon(\vec{x}_j)$  denotes the ball around  $\vec{x}_j$  with radius  $\epsilon$ .

This domain prevents overlaps between the cells by not allowing each cell to drift closer than  $\epsilon$  to any other cell.

HSCM cells perform the same Brownian motion as the point particles.

The next question is, how cell collisions are modelled. Unlike in our DCF model where cell interactions are modelled as forces acting inside of the domain, the cell collisions from the HSCM arise from the reflective boundary condition.

Let us assume that two cells  $i$  and  $j$  are given such that  $\|\vec{x}_i - \vec{x}_j\|_2 = \epsilon$  is true. Then, both cell centres are located at the boundary  $\partial\Omega_{BC}^\epsilon$ . Here, the reflective boundary condition is still imposed and it causes both cells to bounce off each other in the direction of the outward normal vector from the excluded area of the respectively other cell.

In [BC12] the authors managed to compute the marginal distribution function of the first particle of the HSCM. In two dimensions it is given by:

$$(4) \quad \frac{\partial p}{\partial t}(\vec{x}_1, t) = \nabla_{\vec{x}_1} \cdot \left\{ \nabla_{\vec{x}_1} \left[ p + \frac{\pi}{2} (N_{BC} - 1) \epsilon^2 p^2 \right] \right\}.$$

We can see a connection to Equation 3. Let us define a diffusion coefficient

$$D_\epsilon(p) = 1 + \pi(N_{BC} - 1) \epsilon^2 p$$

that depends on the local partial density  $p$  and the cell diameter  $\epsilon$ . For the point particles, we have  $\epsilon = 0$  and  $D_\epsilon(p) = 1$ . Thus, we can rewrite the first marginal to

$$\frac{\partial p}{\partial t}(\vec{x}_1, t) = \nabla_{\vec{x}_1} \cdot [D_\epsilon(p) \nabla_{\vec{x}_1} p].$$

In the case of the hard spheres, where  $\epsilon > 0$ , we can compute

$$\begin{aligned} \frac{\partial p}{\partial t}(\vec{x}_1, t) &= \nabla_{\vec{x}_1} \cdot [D_\epsilon(p) \nabla_{\vec{x}_1} p] \\ &= \nabla_{\vec{x}_1} \cdot [(1 + \pi(N_{BC} - 1) \epsilon^2 p) \nabla_{\vec{x}_1} p] \\ &= \nabla_{\vec{x}_1} \cdot [\nabla_{\vec{x}_1} p + \pi(N_{BC} - 1) \epsilon^2 p \nabla_{\vec{x}_1} p] \\ &= \nabla_{\vec{x}_1} \cdot [\nabla_{\vec{x}_1} p + \pi(N_{BC} - 1) \epsilon^2 \frac{1}{2} \nabla_{\vec{x}_1} p^2] \\ &= \nabla_{\vec{x}_1} \cdot [\nabla_{\vec{x}_1} (p + \frac{\pi}{2} (N_{BC} - 1) \epsilon^2 p^2)] \end{aligned}$$

to recover Equation 4.

When considering  $D_\epsilon(p) = 1 + \pi(N_{BC} - 1) \epsilon^2 p$  to be the diffusion coefficient, we can conclude that an increase in the number of cells  $N_{BC}$ , the cell diameter  $\epsilon$ , or the local density  $p$  leads to an increased diffusion rate of the system. Overall, we conclude that the bounce effect of the HSCM enhances the diffusion rate of the system's density.

Another evidence of this behavior is shown in Figure 2 in [BC12].

Here, we can see two Monte Carlo simulations. A Monte Carlo simulation 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 1.1. After initialization, 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 of sub squares. For each sub square, 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 a sub square. This normalisation ensures that the heatmap represents a probability density, satisfying the mass conservation condition:

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

This approach provides a smooth estimate of the empirical cell density, allowing direct comparison with the corresponding solutions of the diffusion equations. Figure ?? shows the discussed graphic from [BC12].

## 1.2 Reproduction of reference results

Before running our new dynamics that include cell flexibility, we first want to guarantee that the simulations are running in the correct setup. Therefore, we started with recreating the Monte Carlo simulation for the point particles. I always fixed the color scale to be the same as in [BC12] in order to gain comparability. The simulation parameters are the same as in [BC12].

All of our simulations run in the Julia programming language. There, we used the package ‘DifferentialEquations.jl’ with its structure ‘SDEProblem()’ and then solved it with the package inbuilt Euler Maruyama scheme that uses a constant time step size.

I employed a callback function that was triggered after each simulation step to implement a reflective boundary condition. Whenever a particle moved outside the domain, it was relocated to the position within the domain such that its distance to the domain boundary remained unchanged, effectively reflecting the particle off the boundary.

Beside of this, all particles moved according to the two dimensional Brownian motion

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

Figure ?? shows the evolution of the particle density in terms of heatmaps for different time steps. The results of our Monte Carlo simulation appear to be in good agreement with those of Bruna and Chapman, suggesting that our approach is robust and accurate.

Next, we consider the HSCM and run the Monte Carlo simulation for a cell diameter of  $\epsilon = 0.01$ . Figure 1 shows the density evolution of the HSCM.

## 1.3 Shape deformation check

In this subsection, we are going to investigate how much the cells deformed throughout our big monte carlo simulations. Therefore, we need to introduce a measure of how much a cell is deformed. There is already some theory existing on how to measure that for two dimensional figures. A basic approach is to study the ratio between cell area and cell perimeter. We call that ratio the isoperimetric quotient  $\alpha$  of that geometric figure  $F$ , i.e.

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

where  $\nu$  is a scaling factor which we will define later.

There is one 2 dimensional geometric figure that has the largest isoperimetric quotient from all: the circle. For a circle of radius  $r > 0$ , we have an area of  $area_c = \pi r^2$  and a perimeter of  $perimeter_c = 2\pi r$ . Thus, it has an isoperimetric quotient of

$$\alpha_{circle} = \nu \frac{\pi r^2}{(2\pi r)^2} = \nu \frac{1}{4\pi}.$$

We want to normalise the isoperimetric quotient to always be in the interval  $[0, 1]$ .

Therefore, we just have to choose  $\nu = 4\pi$ . Since a circle always has a maximum isoperimetric quotient, we have an upper bound of 1. It also has 0 as a lower bound since both the area and perimeter of a geometric figure are always positive.

These properties are also easy to compute for our DF cells. We can use the Shoelace Formula ?? for the cell area, and for getting the perimeter, we just have to add up the lengths of all cells.

We can use this approach to analyse how much our cell shapes changed in our big

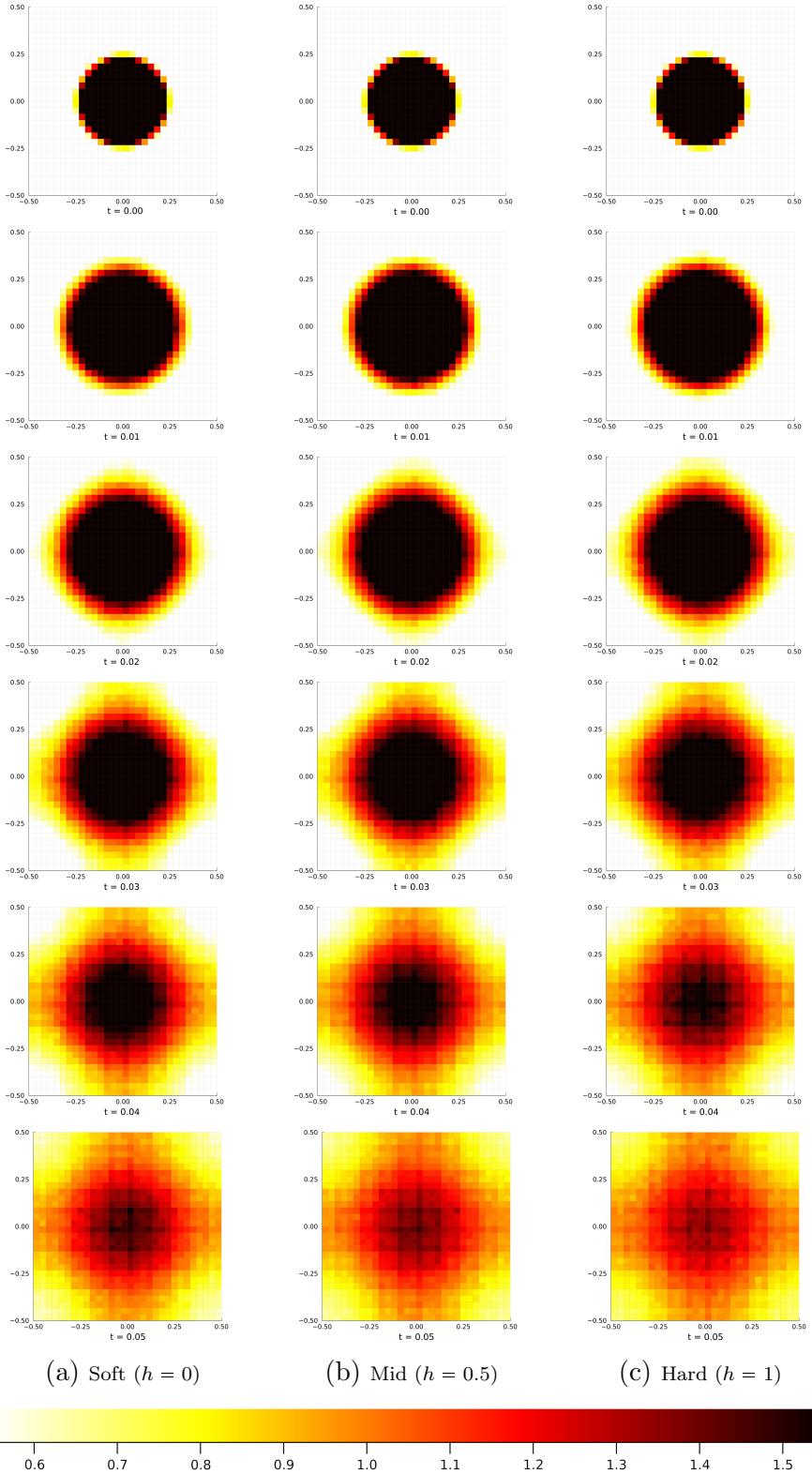


Figure 1: Heatmaps of a Monte Carlo simulation of the DF cell model with different hardness values at the times  $t \in \{0.00, 0.01, \dots, 0.05\}$ . Left column (a) shows hardness 0, we can see hardness 0.5 in the middle (b) and hardness 1 on the right (c). We can observe that the diffusion rate increases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

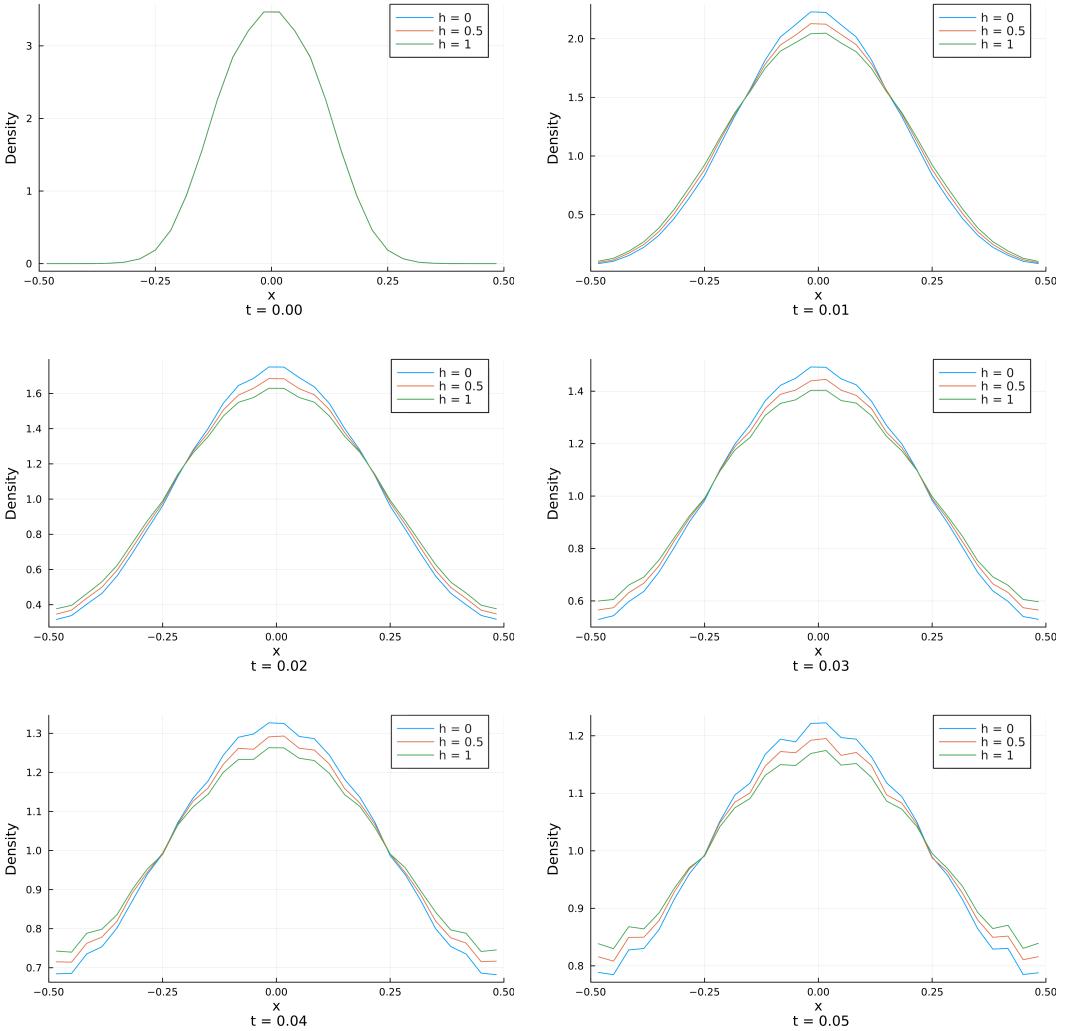


Figure 2: This figure shows the evolution of the cross section density for our three Monte Carlo simulations at the sample times  $t \in \{0, 0.01, \dots, 0.05\}$ . Initially, each simulation starts with the same distribution. Note that the scaling of the  $y$ -axis changes from  $[0, 3.5]$  at  $t = 0$  to  $[0.7, 1.3]$  at  $t = 0.05$ , indicating diffusion in the density distribution for all hardnesses. As the plots show, the higher the hardness, the faster the diffusion for  $t > 0$ , since we observe a lower density in the middle ( $x = 0$ ) and a higher density near the interval borders for higher hardness values. Thus, the density distribution is already more even for higher hardnesses at  $t > 0$ .

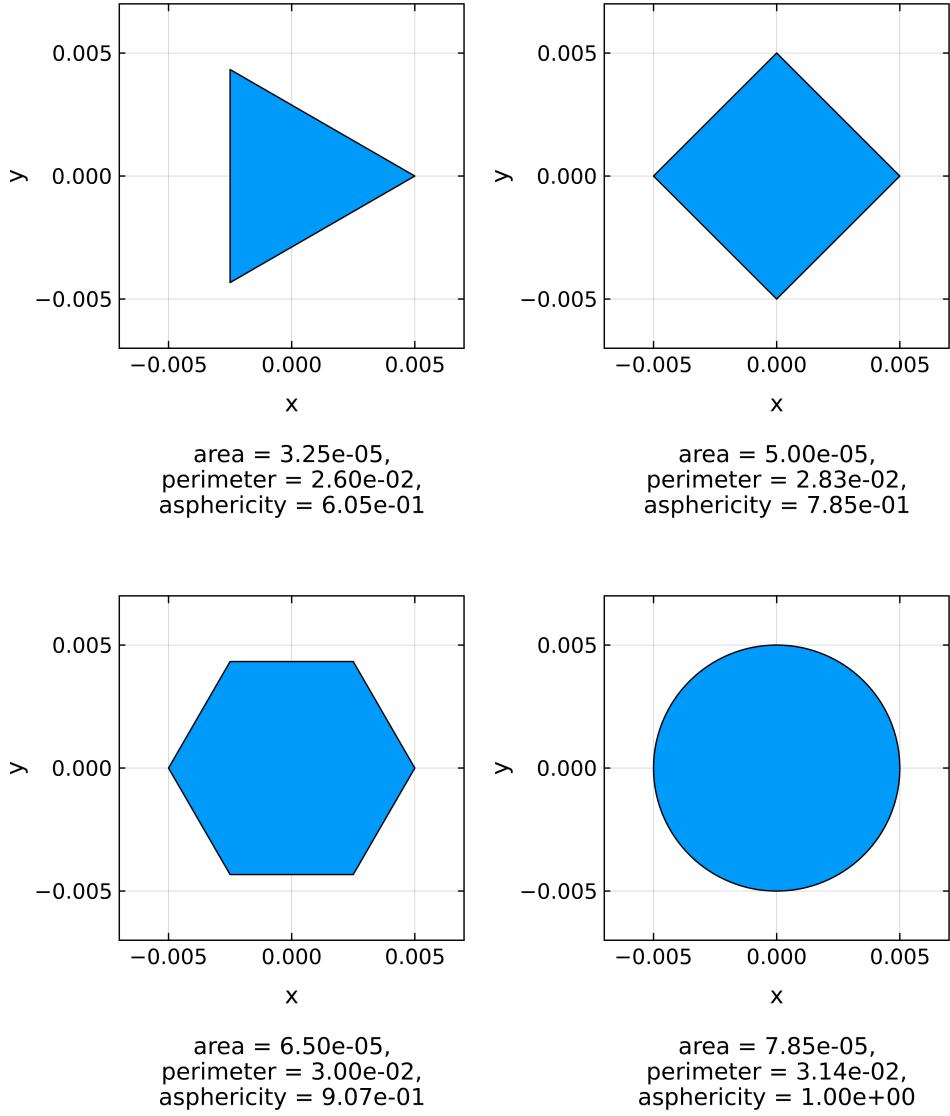


Figure 3: Examples of DF cells and a circle illustrating their geometric properties. The DF cells are represented as regular polygons with  $N_V \in \{3, 4, 6\}$  vertices. For each case, the cell shape, area, perimeter and asphericity are shown. The third DF cell with  $N_V = 6$  vertices corresponds 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.

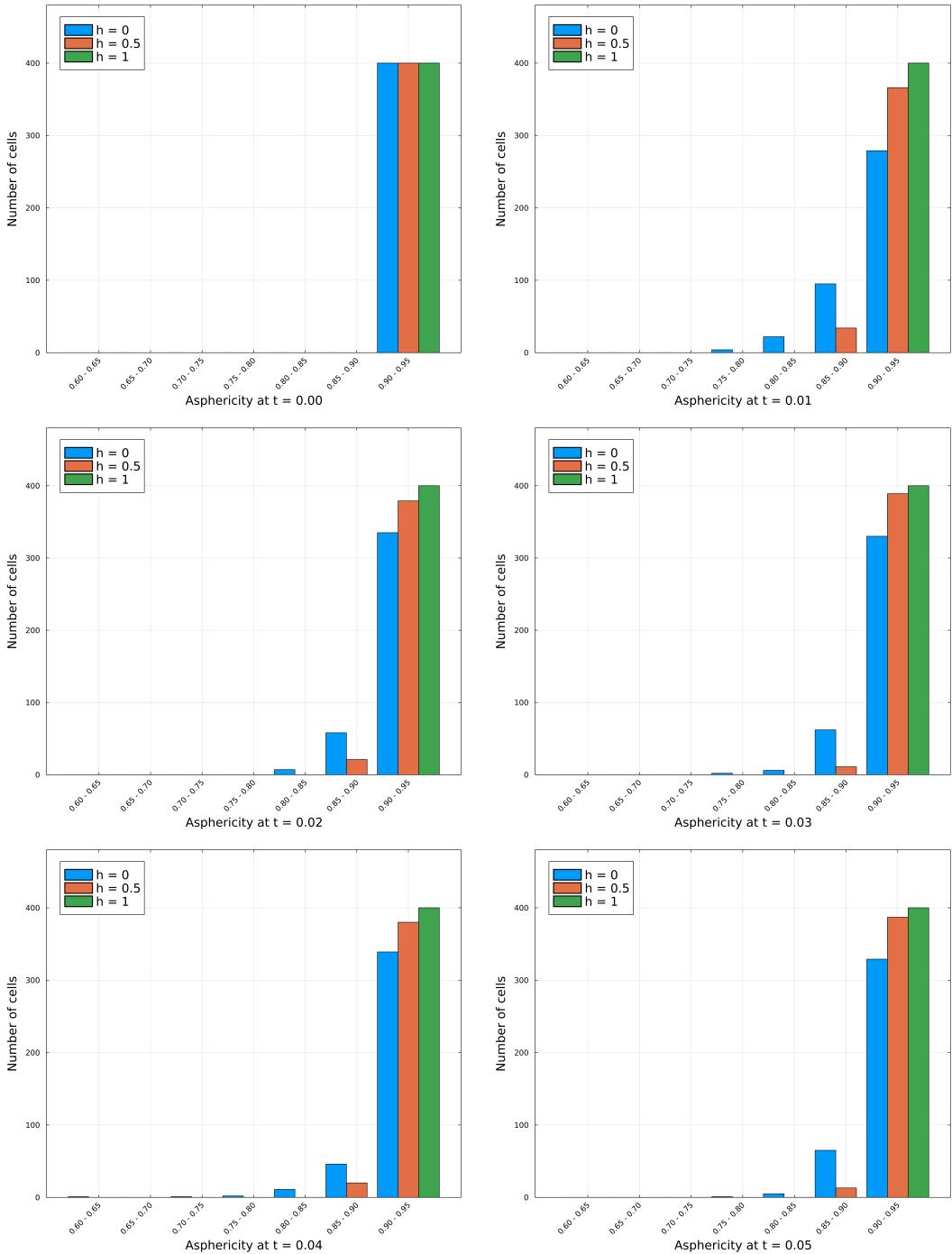


Figure 4: This figure illustrates how the cell asphericities change in the Monte Carlo simulations for hardnesses  $h \in \{0, 0.5, 1\}$ . Initially, all cells are in their desired states, as shown in the third picture of Figure 3, having an asphericity of  $\alpha \approx 0.907$ . For  $h = 1$ , we can see that all 400 cells keep their standard asphericity of  $\alpha \approx 0.907$ , as no cell shape deformation is done in this setting. For the other two simulations, we can see that the cells actually change their shapes, resulting in an asphericity spread in the interval  $\alpha \in [0.6, 0.95]$  for the soft model ( $h = 0$ ) and  $\alpha \in [0.85, 0.95]$  for the mid hardness model ( $h = 0.5$ ).

## 2 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.

### 3 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
- limit  $N_V \rightarrow \infty$
- 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)

## **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, October 28, 2025

Technische Universität Dresden

Matriculation Number: 4930487

## References

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