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: September 2, 2025

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

Contents

1 DF model dynamics **3**

1.1 Area force 4

1.2 Edge force 7

1.3 Interior angle force 10

1.4 Overlap force 16

1.5 The DF SDE 29

1 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

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

where $k \in \mathbb{N}_{\geqslant 1}$ 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 $k \in \mathbb{N}_{\geqslant 1}$. 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 $\phi \in C^\infty_C(\mathbb{R})$:

$$\begin{aligned}
\int_{\mathbb{R}} |x|^k \phi'(x)\mathrm{d}x &= \int_{\mathbb{R}_{\geqslant 0}} x^k \phi'(x)\mathrm{d}x + \int_{\mathbb{R}_{<0}} (-x)^k \phi'(x)\mathrm{d}x \\
&= [x^k\phi(x)]_0^\infty - \int_{\mathbb{R}_{\geqslant 0}} kx^{k-1}\phi(x)\mathrm{d}x + [(-x)^k\phi(x)]_0^\infty - \int_{\mathbb{R}_{<0}} k(-x)^{k-1}\phi(x)\mathrm{d}x \\
&= -\int_{\mathbb{R}_{\geqslant 0}} kx^{k-1}\phi(x)\mathrm{d}x - \int_{\mathbb{R}_{<0}} k(-x)^{k-1}\phi(x)\mathrm{d}x \\
&= -\int_{\mathbb{R}} k \operatorname{sgn}(x)|x|^{k-1}\phi(x)\mathrm{d}x.
\end{aligned}$$

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

1.1 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 1.1. ***Shoelace formula for DF cells***
Let $C = (\vec{v}_1, \dots, \vec{v}_N)$ be a DF cell with $\vec{v}_j = (v_j^x, v_j^y)^T$ for $j = 1, \dots, N$. 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 1, 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 1) 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

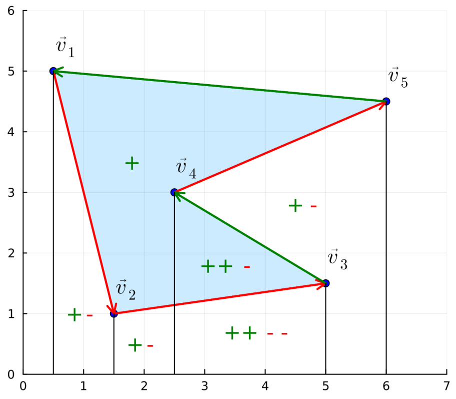


Figure 1: This figure shows a geometrical interpretation of the shoelace formula. The green arrows, which point from right to left, represent positive trapezoidal areas that contribute positively to the total area of the polygon. In contrast, the red arrows point from left to right and represent negative 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]

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

□

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 1.2. Area energy
The energy $A_k : (\mathbb{R}^2)^{N_V} \to \mathbb{R}_{\geqslant 0}$ for $k \in \mathbb{N}_{\geqslant 1}$, used to keep the cells at a constant volume, reads

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

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.

Proposition 1.3. ***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

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

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

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

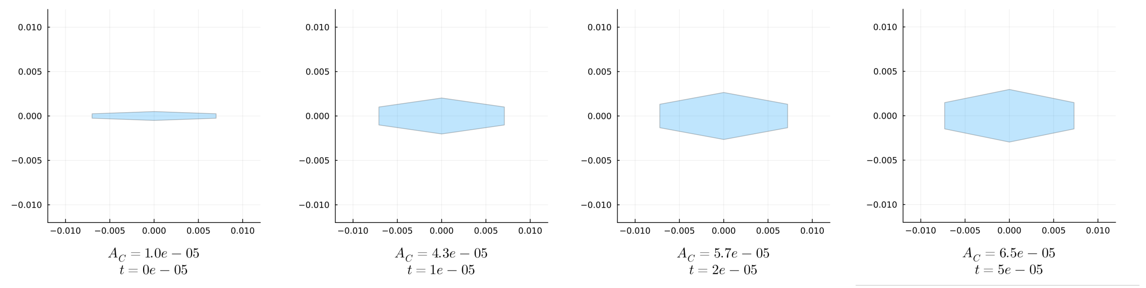
for all $1 \leqslant j \leqslant N_V$.

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

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

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 2 illustrates how the area force acts on a cell to either expand or contract it toward the desired target area.



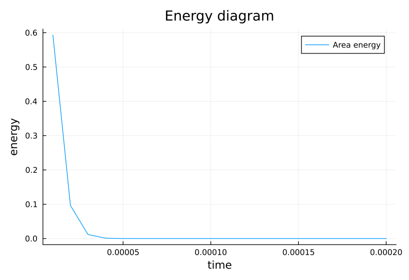


Figure 2: The top four plots show the evolution of a DF cell influenced solely by the area force, with $k = 2$ applied to the vertices and a force scaling of $4e8$, at times $t \in \{0, 1e-5, 2e-5, 5e-5\}$. Thus, we have $\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = -4e8\nabla_{\vec{v}}A_2(C)$ for all vertices. The initial cell area is $A_C = 1e-5$, while the desired cell area is set to $6.5e-5$.
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, which is also reflected in the energy diagram on the bottom left.

1.2 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 $1 \leqslant j \leqslant N_V$ as

$$e_j = \overline{\vec{v}_j\,\vec{v}_{j+1}}$$

and we 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 1.4. 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

$$E_k(C) = \sum_{j=1}^{N_V} \frac{1}{k}|E_C^j - E_d^j|^k, \tag{3}$$

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:

Proposition 1.5. ***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*

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

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*

$$\begin{aligned} \nabla_{\vec{v}_j} E_k(C) &= \operatorname{sgn}(E_C^{j-1} - E_d^{j-1}) \frac{|E_C^{j-1} - E_d^{j-1}|^{k-1}}{E_C^{j-1}} \begin{pmatrix} v_j^x - v_{j-1}^x \\ v_j^y - v_{j-1}^y \end{pmatrix} \\ &\quad + \operatorname{sgn}(E_C^j - E_d^j) \frac{|E_C^j - E_d^j|^{k-1}}{E_C^j} \begin{pmatrix} v_j^x - v_{j+1}^x \\ v_j^y - v_{j+1}^y \end{pmatrix} \end{aligned} \tag{4}$$

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

Proof.

$$\nabla_{\vec{v}_j} E_k(C) = \nabla_{\vec{v}_j} \sum_{j=1}^{N_V} \frac{1}{k} |E_C^j - E_d^j|^k$$

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

For the first summand, we can compute

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

For the second summand, we get:

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

$\square$

An isolated application of the edge force can be seen in Figure 3.

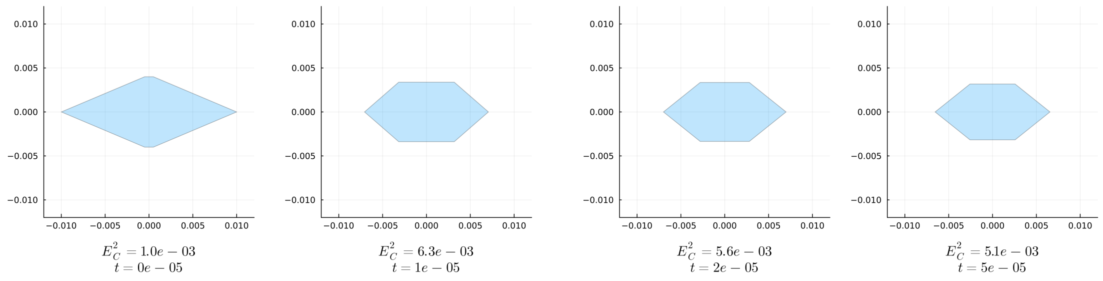
1.3 Interior angle force

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 1.6. arctan2
The function

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

is defined by:



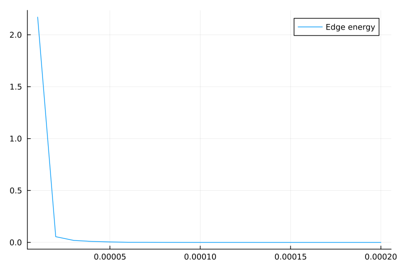


Figure 3: The top figures illustrate the evolution of a DF cell governed exclusively by the edge force, with $k = 2$ applied to the vertices and a force scaling of $3e4$, at times $t \in \{0, 1e-5, 2e-5, 5e-5\}$. Accordingly, we have $\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = -3e4\nabla_{\vec{v}}E_2(C)$ for all vertices.
The initial edge length of the top edge is $E_C^2 = 1e-3$, while the desired edge lengths are all set to $5e-3$.
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 the energy diagram on the bottom left.

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

The $\operatorname{arctan2}(\vec{v})$ 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:

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

With that, we can define our interior angle energy.

Definition 1.7. 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

$$I_k(C) = \sum_{j=1}^{N_V} \frac{1}{k} |I_C^j - I_d^j|^k, \tag{5}$$

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

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

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.

Proposition 1.8. ***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

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

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

$$\begin{aligned}
\nabla_{\vec{v}_j} I_k(C) = {} & \operatorname{sgn}(I_C^{j-1} - I_d^{j-1})|I_C^{j-1} - I_d^{j-1}|^{k-1} \left(-\frac{1}{\|\vec{v}_j - \vec{v}_{j-1}\|_2^2} \begin{pmatrix} v_{j-1}^y - v_j^y \\ v_j^x - v_{j-1}^x \end{pmatrix} \right) \\
& + \operatorname{sgn}(I_C^j - I_d^j)|I_C^j - I_d^j|^{k-1} \left(\frac{1}{\|\vec{v}_{j-1} - \vec{v}_j\|_2^2} \begin{pmatrix} v_{j-1}^y - v_j^y \\ v_j^x - v_{j-1}^x \end{pmatrix} \right. \\
& \qquad \left. - \frac{1}{\|\vec{v}_{j+1} - \vec{v}_j\|_2^2} \begin{pmatrix} v_{j+1}^y - v_j^y \\ v_j^x - v_{j+1}^x \end{pmatrix} \right) \\
& + \operatorname{sgn}(I_C^{j+1} - I_d^{j+1})|I_C^{j+1} - I_d^{j+1}|^{k-1} \left(\frac{1}{\|\vec{v}_j - \vec{v}_{j+1}\|_2^2} \begin{pmatrix} v_{j+1}^y - v_j^y \\ v_j^x - v_{j+1}^x \end{pmatrix} \right)
\end{aligned} \tag{6}$$

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

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

First, we will focus on the computation of $\nabla_{\vec{v}_j} \frac{1}{k} |I_d^j - I_C^j|^k$.

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

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.

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

$$\begin{aligned}\nabla_{v_j^x} g(\vec{v}_{j-1}, \vec{v}_j) &= (-1, 0)^T \\ \nabla_{v_j^y} g(\vec{v}_{j-1}, \vec{v}_j) &= (0, -1)^T\end{aligned}$$

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:

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

For the neighboring vertices, we need:

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

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

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

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

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

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

Finally, for the successor vertex, we get

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

□

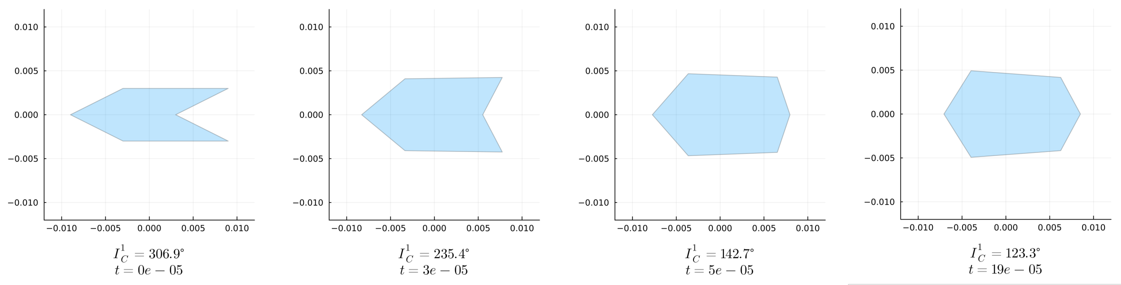
Figure 4 illustrates the isolated effect of the interior angle force.

1.4 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 similar to 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. An overlap is treated as a DF cell in its own right, composed of the vertices from each of the two overlapping cells that lie inside the other, along with the two intersection points where the cell boundaries intersect.
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



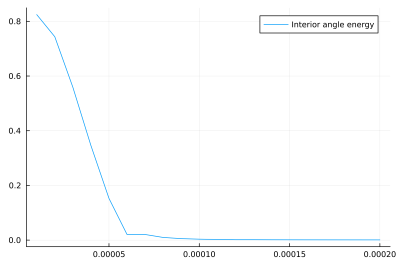


Figure 4: The top four plots depict the evolution of a DF cell subject only to the interior angle force with $k = 2$ applied to the vertices and a force scaling of $1e-1$, at times $t \in \{0, 3e-5, 5e-5, 19e-5\}$. In this case, we have $\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = -1e - 1\nabla_{\vec{v}} I_2(C)$ 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 the energy diagram on the bottom left, 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.

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 1.9. ***Computation of a discrete overlap***

INPUT:

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

```
function CONSTRUCTOVERLAP($C$, $\zeta$, $I$)
    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, $\zeta$, $I$)
        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 1.10. Overlap energy
Let C_i and C_m be two cells from the system $\vec{C}$ and $\Omega_{i,m}$ 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

$$O_k(\vec{C}) = \sum_{i=1}^{N_C} \left(\sum_{m=i+1}^{N_C} \left(\sum_{D \in \Omega_{i,m}} \frac{1}{k} |A_D|^k \right) \right), \tag{7}$$

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

$$O_k^{i,m} : (\mathbb{R}^{2N_V})^2 \to \mathbb{R}$$
$$O_k^{i,m}(C_i, C_m) = \sum_{D \in \Omega_{i,m}} \tfrac{1}{k} |A_D|^k.$$

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 1.11. 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

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

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

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

Given the four adjacent vertices, we can always compute the intersection with the function:

$$w : (\mathbb{R}^2)^4 \to \mathbb{R}^2,$$
$$w(\vec{v}^{\,i}_{\text{out}}, \vec{v}^{\,i}_{\text{in}}, \vec{v}^{\,m}_{\text{out}}, \vec{v}^{\,m}_{\text{in}}) = \vec{v}^{\,i}_{\text{out}} + \frac{(\vec{v}^{\,m}_{\text{out}} - \vec{v}^{\,i}_{\text{out}}) \times (\vec{v}^{\,m}_{\text{in}} - \vec{v}^{\,m}_{\text{out}})}{(\vec{v}^{\,i}_{\text{in}} - \vec{v}^{\,i}_{\text{out}}) \times (\vec{v}^{\,m}_{\text{in}} - \vec{v}^{\,m}_{\text{out}})} (\vec{v}^{\,i}_{\text{in}} - \vec{v}^{\,i}_{\text{out}}),$$

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.

Proposition 1.12. ***Partial derivatives of intersection points***
We use $w(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in})$ 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 helper functions:

$$f : (\mathbb{R}^2)^3 \to \mathbb{R} \quad f(\vec{v}^i_{out}, \vec{v}^m_{out}, \vec{v}^m_{in}) = (\vec{v}^m_{out} - \vec{v}^i_{out}) \times (\vec{v}^m_{in} - \vec{v}^m_{out})$$

$$g : (\mathbb{R}^2)^4 \to \mathbb{R} \quad g(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in}) = (\vec{v}^i_{in} - \vec{v}^i_{out}) \times (\vec{v}^m_{in} - \vec{v}^m_{out})$$

$$t : (\mathbb{R}^2)^4 \to \mathbb{R} \quad t(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in}) = \frac{f(\vec{v}^i_{out}, \vec{v}^m_{out}, \vec{v}^m_{in})}{g(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in})}$$

$$w : (\mathbb{R}^2)^4 \to \mathbb{R}^2 \quad w(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in}) = \vec{v}^i_{out} + t(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in})(\vec{v}^i_{in} - \vec{v}^i_{out})$$

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

(8)
$$\frac{\partial w(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in})}{\partial \vec{v}^i_{out}} : (\mathbb{R}^2)^4 \to \mathbb{R}^{2\times 2},$$
$$\frac{\partial w(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in})}{\partial \vec{v}^i_{out}} = (1-t)I_2 + \frac{g-f}{g^2}(\vec{v}^i_{in} - \vec{v}^i_{out}) \begin{pmatrix} -(v^{m,y}_{in} - v^{m,y}_{out}) \\ v^{m,x}_{in} - v^{m,x}_{out} \end{pmatrix}^T,$$

(9)
$$\frac{\partial w(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in})}{\partial \vec{v}^i_{in}} : (\mathbb{R}^2)^4 \to \mathbb{R}^{2\times 2},$$
$$\frac{\partial w(\vec{v}^i_{out}, \vec{v}^i_{in}, \vec{v}^m_{out}, \vec{v}^m_{in})}{\partial \vec{v}^i_{in}} = tI_2 + \frac{f}{g^2}(\vec{v}^i_{in} - \vec{v}^i_{out}) \begin{pmatrix} -(v^{m,y}_{in} - v^{m,y}_{out}) \\ v^{m,x}_{in} - v^{m,x}_{out} \end{pmatrix}^T.$$

Proof.

For the ease of notation, we will drop the arguments, but keep in mind that f, g and t are dependent on the vertices.
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:

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

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

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

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

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

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

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

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

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

Proposition 1.13. ***Deforming overlap force***
Let $\Omega_{i,m}$ be the set of all overlaps between the cells C_i and C_m. Each overlap cell $D \in \Omega_{i,m}$ is a list of vertices, that form the overlap, just like a DF cell.
The deforming overlap gradient is then given by

$$\begin{aligned}\nabla_{\vec{v}_j^i} O_k(\vec{C}) = \sum_{D\in\Omega_{i,m}} |A_D|^{k-1} \Bigg(&\mathbf{1}_{V(D)}(\vec{v}_j^i)\nabla_{\vec{v}_j^i} A(D) + \\ &+ \sum_{\vec{w}\in W(D)} \left(\mathbf{1}_{out(w)}(\vec{v}_j^i)\frac{\partial \vec{w}}{\partial \vec{v}_{j,out}^i} + \mathbf{1}_{in(w)}(\vec{v}_j^i)\frac{\partial \vec{w}}{\partial \vec{v}_{j,in}^i}\right)\nabla_{\vec{w}} A(D)\Bigg),\end{aligned} \tag{10}$$

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 8 and 9 define $\frac{\partial \vec{w}}{\partial \vec{v}^i_{j,out}}$ and $\frac{\partial \vec{w}}{\partial \vec{v}^i_{j,in}}$, respectively.
The difference between $\nabla_{\vec{v}^i_j} A(D)$ and $\nabla_{\vec{w}} A(D)$ is that $\nabla_{\vec{v}^i_j} A(D)$ uses the neighboring overlap vertices of $\vec{v}^i_j$ itself in the Area Gradient Formula 2, whereas $\nabla_{\vec{w}} A(D)$ uses the neighbors of the corresponding intersection point in D, which is not the same overlap vertex as $\vec{v}^i_j$. 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

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

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

$$F_{k,i}^{(\hat{O})} : (\mathbb{R}^{2N_V})^{N_C} \to (\mathbb{R}^{2N_V}),$$
$$F_{k,i}^{(\hat{O})}(\vec{C}) = -(\nabla_{\vec{v}^i_1} O_k(\vec{C}), \ldots, \nabla_{\vec{v}^i_{N_V}} O_k(\vec{C}))^T.$$

Proof.
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

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

Now, there are different cases.
Case 1: *$\vec{v}^i_j \notin D$ and $\vec{v}^i_j \notin$ 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:

$$\nabla_{\vec{v}^i_j} \frac{1}{k} A_D^k = 0.$$

Case 2: *$\vec{v}^i_j \notin D$ and $\exists \vec{w} \in W(D) : \ \vec{v}^i_j \in$ 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 $\vec{v}^i_j$ is

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

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

where, according to Equation 8

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

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 1.1, as in the area gradient from Formula 2:

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

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 1.1 for once more:

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

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.

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

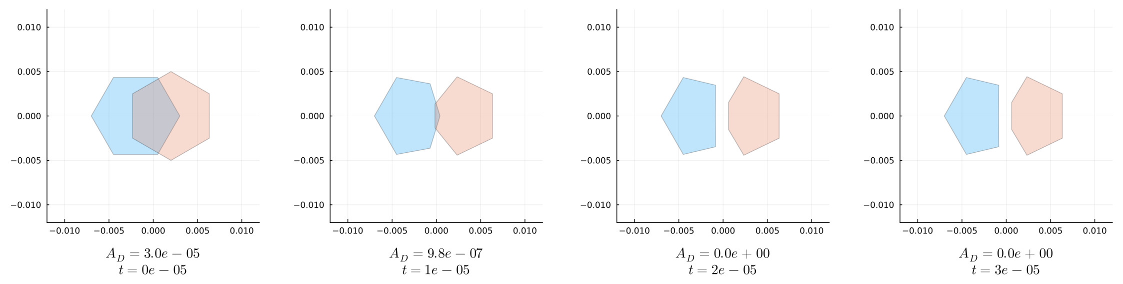
where $out(w)$, $in(w) \subset adj(w)$ denote the sets of outside and inside overlap-adjacent vertices, respectively.
The difference between $\nabla_{\vec{v}_j^i}A(D)$ and $\nabla_{\vec{w}}A(D)$ is, that $\nabla_{\vec{v}_j^i}A(D)$ uses the neighboring overlap vertices of $\vec{v}_j^i$ itself in the Area Gradient Formula 2, 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 **??** 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



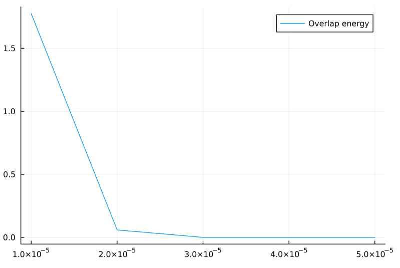


Figure 5: The top four plots show the evolution of a DF cell influenced solely by the deforming overlap force, with $k = 1$ applied to the vertices and a force scaling of $6e4$, at times $t \in \{0, 1e-5, 2e-5, 3e-5\}$.
In this case, we have $\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = -6e4\nabla_{\vec{v}}\hat{O}_1(C_1, C_2)$ 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 $3e-5$, which is relatively large compared to the cell area of $6.5e-5$. However, it is completely resolved after just two time steps, as also illustrated in the energy diagram on the bottom left.

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 1.14. 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

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

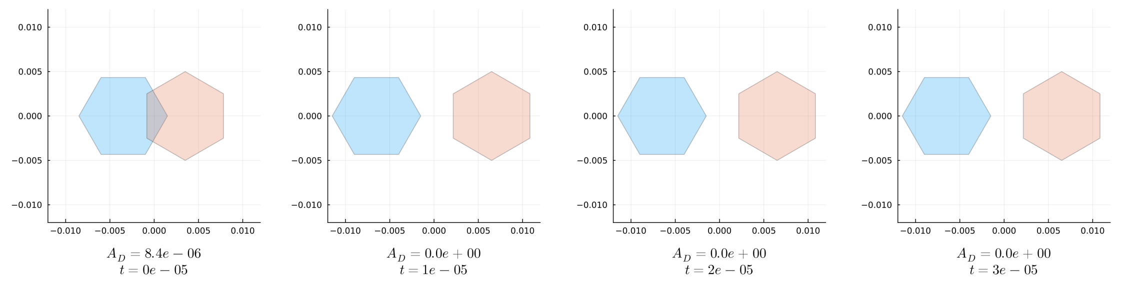
where the vector $\mathrm{d}\bar{o}_{i,l}$ is repeated N_V times, once for each vertex of cell C_i.
The total bounce overlap force $F_i^{(\bar{O})} : (\mathbb{R}^{2N_V})^{N_C} \to \mathbb{R}^{2N_V}$ acting on cell C_i due to all other cells in the system is then defined as

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

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}.
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 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.



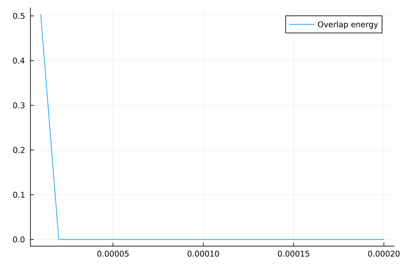


Figure 6: The top four plots present the evolution of a DF cell governed solely by the bounce overlap force, applied to the vertices with a force scaling of $1e5$, at times $t \in \{0, 1e-5, 2e-5, 3e-5\}$.
In this case, we have $\frac{\mathrm{d}C_i}{\mathrm{d}t} = 1e5 F_i^{\bar{O}}(C_1, C_2)$ for both 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 the diagram on the left.

1.5 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	$\alpha_A = 4.0 \cdot 10^8$
Edge force	$\alpha_E = 3.0 \cdot 10^4$
Interior angle force	$\alpha_I = 1.0 \cdot 10^{-1}$
Deforming overlap force	$\alpha_{\hat{O}} = 6.0 \cdot 10^4$
Bounce overlap force	$\alpha_{\bar{O}} = 1.0 \cdot 10^5$

Table 1: 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 1.15. The DF SDE
We define the vectors

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

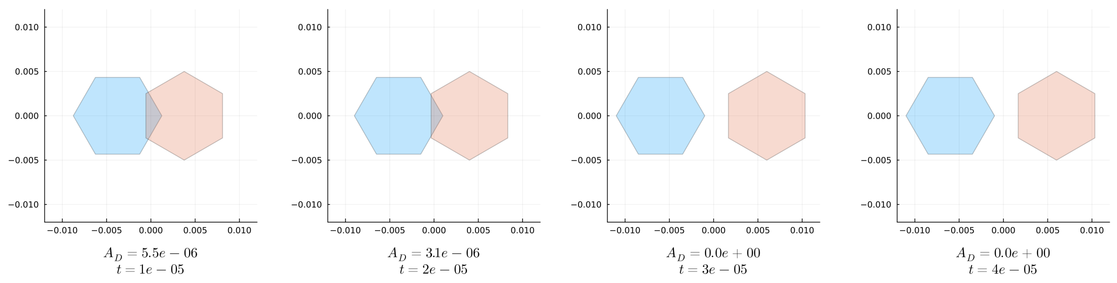
which allow us to distribute a two dimensional Brownian motion $\mathrm{d}\vec{B}^{\,i} = (\mathrm{d}B^{i,x}, \mathrm{d}B^{i,y})^T$ to the x and y coordinates of a cell's vertices, respectively. The cell hardness parameter $h \in [0, 1]$ is assumed to be given. The deterministic part $F^i(\vec{C})$ of the **DF SDE** is given by

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

$$\begin{aligned} \mathbf{F}^i(\vec{C}) = \alpha_A F_2^{(A)}(C_i) + \alpha_E F_2^{(E)}(C_i) + \alpha_I F_2^{(I)}(C_i) + \\ (1-h)\alpha_{\hat{O}} F_{1,i}^{(\hat{O})}(\vec{C}) + h\alpha_{\bar{O}} F_i^{(\bar{O})}(\vec{C}) \end{aligned} \tag{11}$$

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:

$$\mathrm{d}C_i = \mathbf{F}^i(\vec{C})\mathrm{d}t + \mathrm{d}B^{i,x}\, e^x_{N_V} + \mathrm{d}B^{i,y}\, e^y_{N_V}.$$



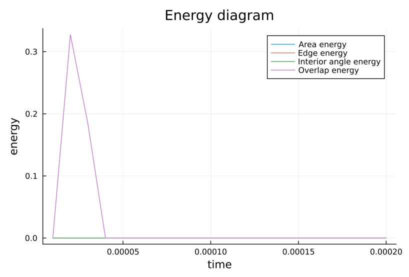
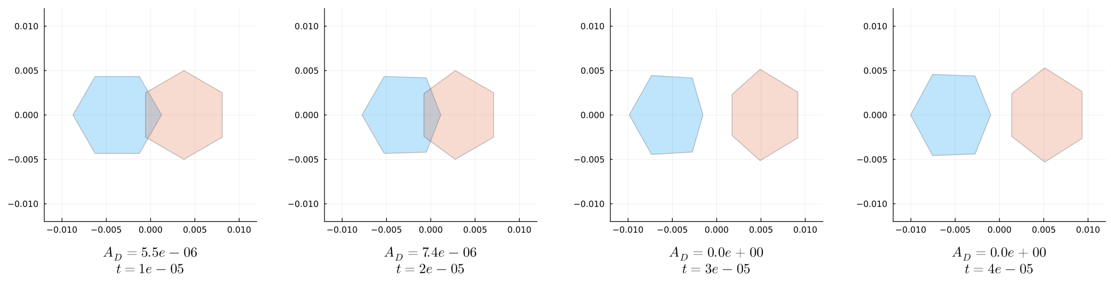


Figure 7: This simulation shows two DF cells evolving according to the dynamics for $i = 1, 2$ $\frac{\mathrm{d}C_i}{\mathrm{d}t} = \mathbf{F}^i(C_1, C_2)$ with hardness $h = 1$, force scalings as listed in Table 1 and without Brownian motion. Click *here* to view the corresponding animation (GIF). The cells are initially generated without overlap and are pushed together during the first two time steps. Afterwards, 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, which shows only an initial spike in the overlap energy. The bounce overlap force then eliminates the overlap within a single time step after the push.



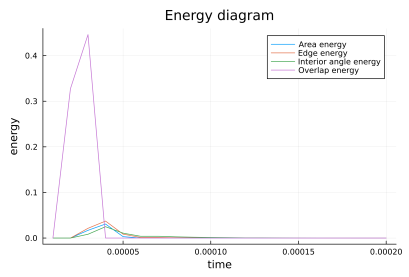
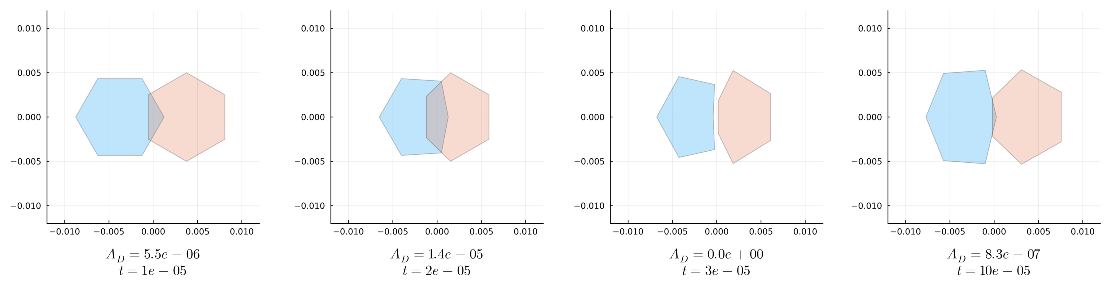


Figure 8: This simulation shows two DF cells again evolving according to the dynamics $i = 1, 2$ $\frac{\mathrm{d}C_i}{\mathrm{d}t} = \mathbf{F}^i(C_1, C_2)$, with hardness $h = 0.5$, force scalings as listed in Table 1, and without Brownian motion. Click *here* to view the corresponding animation (GIF). The cells are initially generated without overlap and are pushed together during the first two time steps. 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 after the push. By time step 10, the cell shapes are nearly restored, as illustrated in the energy diagram.



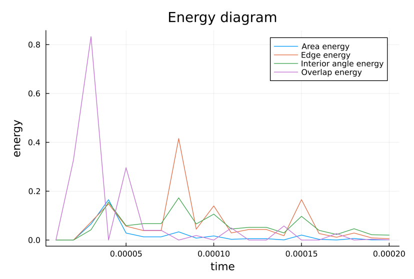


Figure 9: This simulation shows two DF cells evolving according to the dynamics for $i = 1, 2$ $\frac{\mathrm{d}C_i}{\mathrm{d}t} = \mathbf{F}^i(C_1, C_2)$, with hardness $h = 0$, force scalings as listed in Table 1, and without Brownian motion. Click *here* to view the corresponding animation (GIF). The cells are initially generated without overlap and are pushed together during the first two time steps. 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.

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, September 2, 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.

[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.

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