Multiscale mathematical biology of cell-extracellular matrix interactions during morphogenesis

Rens, E.G.

Citation

Rens, E. G. (2018, June 27). *Multiscale mathematical biology of cell-extracellular matrix interactions during morphogenesis*. Retrieved from https://hdl.handle.net/1887/62863

Author: Rens, Lisanne
Title: Multiscale mathematical biology of cell-extracellular matrix interactions during morphogenesis
Date: 2018-06-27

Appendix: deriving forces from CPM Hamiltonian

In chapter 2, 3 and 4 we presented a hybrid cellular Potts model (CPM) which we coupled to a Finite Element Model (FEM) of the substrate. Here, the forces that a cell applies on the substrate are described by a first moment of area model. Alternatively, we can derive the forces $\vec{f}$ a single cell exerts on the matrix in the FEM from the Hamiltonian in the CPM. As an example, the Hamiltonian that describes cell-matrix force H_{cm} may describe a surface tension and line tension as:

$$H_{cm} = \lambda A + \sum_{\{(\vec{x},\vec{x}\prime)|\sigma(\vec{x})\neq\sigma(\vec{x}\prime)\}} J_{01} \tag{7.1}$$

In the CPM, cell movements are simulated by making copies from lattice sites to neighboring lattice sites, which models cell membrane deformations in a discrete fashion (see figure 7.3).

Copies only have effect on the cell membrane, as in the cell interior copies will not change the configuration and thus $\Delta H = 0$. So, forces will only be defined (non-zero) on points on the cell membrane. To couple the CPM to the FEM, the forces $\vec{f}$ need to be approximated on the *nodes* of the grid. First, we explain how we derive forces on the cell membrane using H_{cm} in the CPM and next we explain how we approximate the forces on the nodes.

Virtual work

To derive the forces a cell exerts on the matrix $\vec{f}(\vec{x})$ from the Hamiltonian H_{cm} we need to principle of virtual work. Virtual work is defined as the work of a force acting on a body if it would move along a certain virtual displacement. For instance, if a particle is subject to a constant force $\vec{f}$, then, the virtual work W this particle experiences by virtually displacing $\vec{\delta x}$ is $W = \vec{f} \cdot \vec{\delta x}$, where $\cdot$ is the dot product. In other words, virtual work is the *scalar projection* of force in the direction of the displacement $\vec{\delta x}$, multiplied with the amount of virtual displacement $|\vec{\delta x}|$.

In the CPM, a cell experiences virtual work due to potentially extending or retracting the cell membrane in a certain direction, which are dictated by the type of lattice used. When one uses the conventional 2D square lattice, forces a defined in four directions.

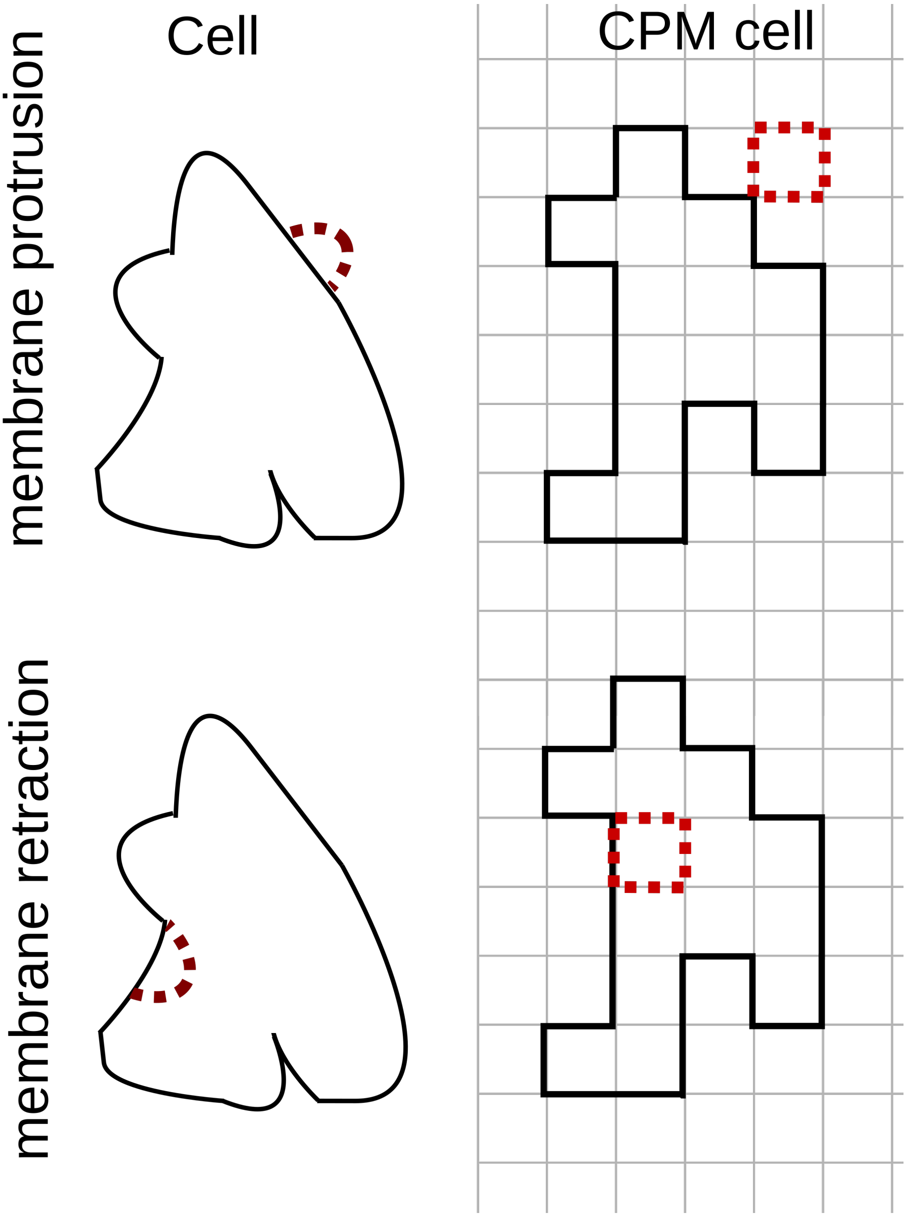

Figure 7.3: CPM discretizes cell membrane and simulates membrane protrusions and retractions

Now, let $\vec{f}(\vec{x})$ be the force on the cell membrane at $\vec{x}$ and let $-\Delta\hat{H}_{\vec{\delta x}}(\vec{x})$ be the virtual work of displacing the cell membrane at $\vec{x}$ in the direction of $\vec{\delta x}$. Then,

$$\vec{f}(\vec{x}) \cdot \vec{\delta x} = -\Delta\hat{H}_{\vec{\delta x}}(\vec{x}) \tag{7.2}$$

For a visual representation, see Figure 7.4. The possible directions of cell membrane deformations are determined by the lattice of the CPM (Figure 7.3 and 7.5). Because we are working on a 2D square grid, the CPM can calculate the virtual work in four directions, namely North, North-East, East and South-East (see Figure 7.5). For a given direction $\vec{\delta x}$, we can extend or retract the membrane, *i.e.* we can copy the spin of $\vec{x}$ onto $\sigma(\vec{x} + \vec{\delta x})$ or the other way around. Therefore, the virtual work $-\Delta\hat{H}_{\vec{\delta x}}$ in the direction $\vec{\delta x}$ is given by

$$\Delta\hat{H}_{\vec{\delta x}}(\vec{x}) = -\left(\frac{\Delta H_{\mathrm{cm}}(\sigma(\vec{x}) \to \sigma(\vec{x} + \vec{\delta x})) - \Delta H_{\mathrm{cm}}(\sigma(\vec{x} + \vec{\delta x}) \to \sigma(\vec{x}))}{2}\right), \tag{7.3}$$

Much in the same way as one would calculate a surface tension: (internal pressure - external pressure)/2, where the dividing by two arises from the membrane having two sides that contribute to the force.

Work on the nodes

Now note that finding $\vec{f}$ from virtual work (the *scalar projections* of $\vec{f}$) known/given in different directions, is an *inverse* problem. In the FEM, the forces $\vec{f}$ need to be defined on the *nodes* of the grid. So, given virtual work on the *cell membrane* (Figure 7.5), we want to derive the forces on the nodes. We approximate the virtual work in four directions on the *nodes* as follows. Let $f_N(\vec{n}), f_{NE}(\vec{n}), f_E(\vec{n})$ and $f_{SE}(\vec{n})$ be the *scalar projections* of the force $\vec{f}(\vec{n})$ on node $\vec{n}$ in the direction North, North, North-East, East and South-East, respectively. Then,

$$f_N(\vec{n}) \approx \frac{-\frac{1}{2}\Delta\hat{H}_{\vec{\delta x}_N}(\vec{n} - (\frac{1}{2}, 0)^T) - \frac{1}{2}\Delta\hat{H}_{\vec{\delta x}_N}(\vec{n} + (\frac{1}{2}, 0)^T)}{2} \tag{7.4}$$

$$f_{NE}(\vec{n}) = -\frac{\Delta\hat{H}_{\vec{\delta x}_{NE}}(\vec{n})}{\sqrt{2}} \tag{7.5}$$

$$f_E(\vec{n}) \approx \frac{-\frac{1}{2}\Delta\hat{H}_{\vec{\delta x}_E}(\vec{n} - (0, \frac{1}{2})^T) - \frac{1}{2}\Delta\hat{H}_{\vec{\delta x}_E}(\vec{n} + (0, \frac{1}{2})^T)}{2} \tag{7.6}$$

$$f_{SE}(\vec{n}) = -\frac{\Delta\hat{H}_{\vec{\delta x}_{SE}}(\vec{n})}{\sqrt{2}} \tag{7.7}$$

where $\vec{\delta x}_N = (0, 1)^T, \vec{\delta x}_{NE} = (1, 1)^T, \vec{\delta x}_E = (1, 0)^T$ and $\vec{\delta x}_{SE} = (1, -1)^T$. The $\sqrt{2}$ in the terms for $f_{NE}(\vec{n})$ and $f_{SE}(\vec{n})$ arise from $|\delta x_{NE}| = |\delta x_{SE}| = \sqrt{2}$. For an example of how the force component in direction East at a node is calculated, see figure 7.6.

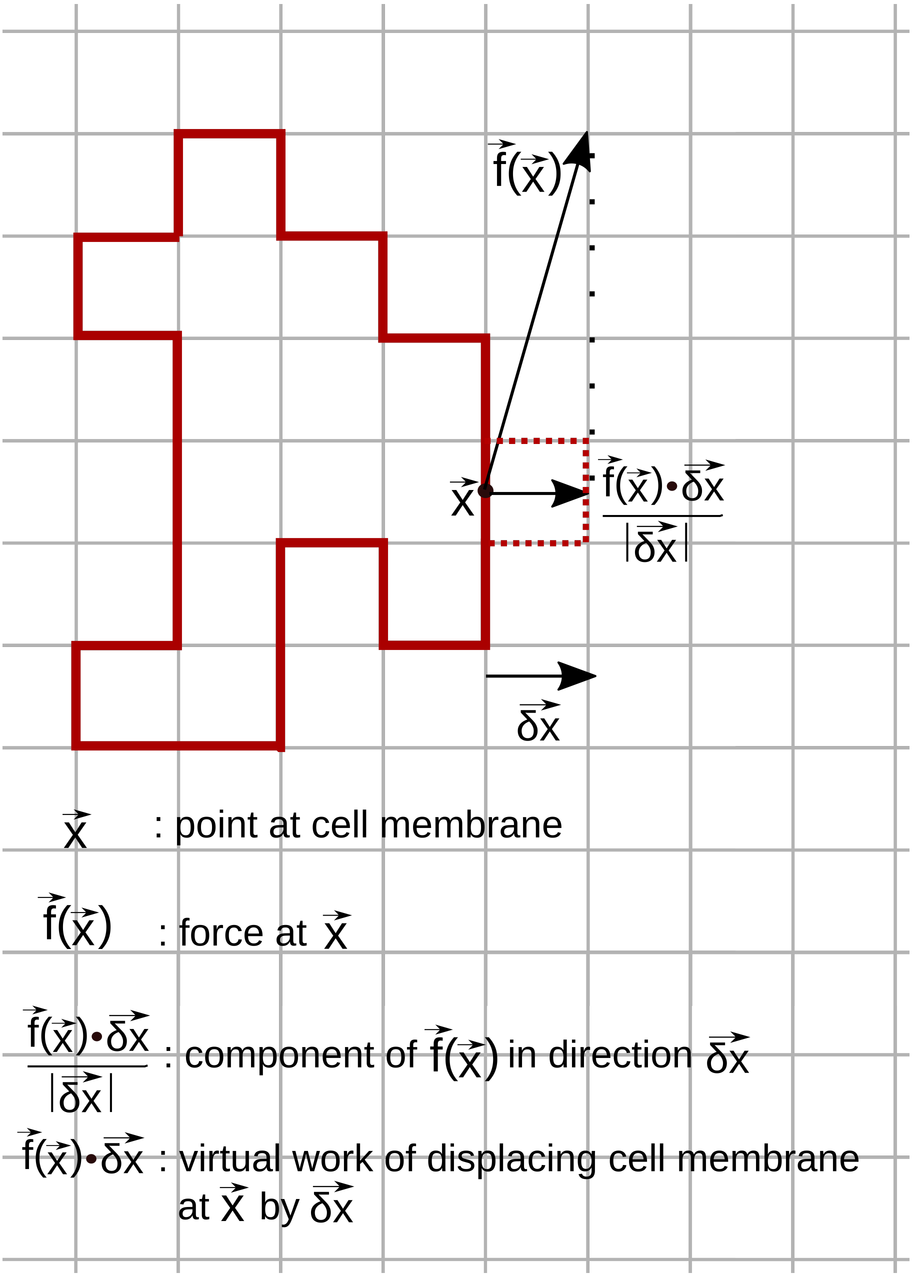

Figure 7.4: Force and virtual work in the CPM

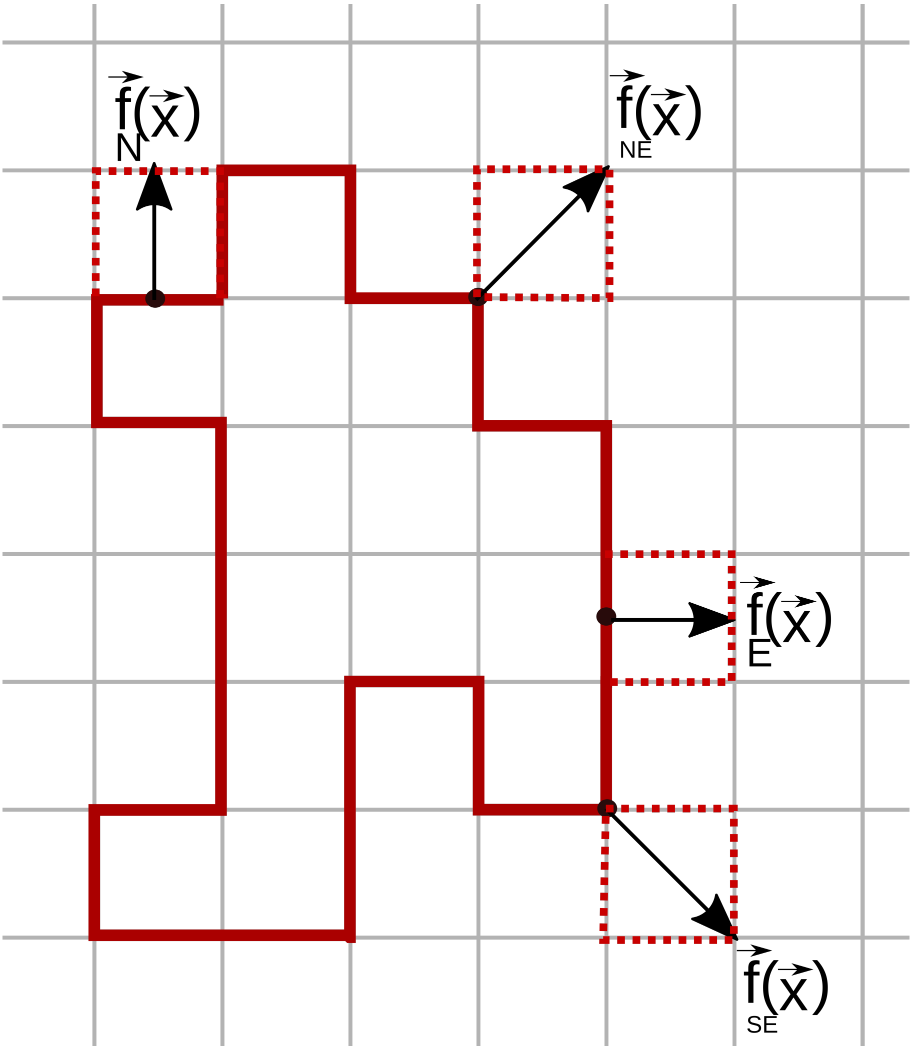

Figure 7.5: Possible direction of forces in a 2D square grid CPM

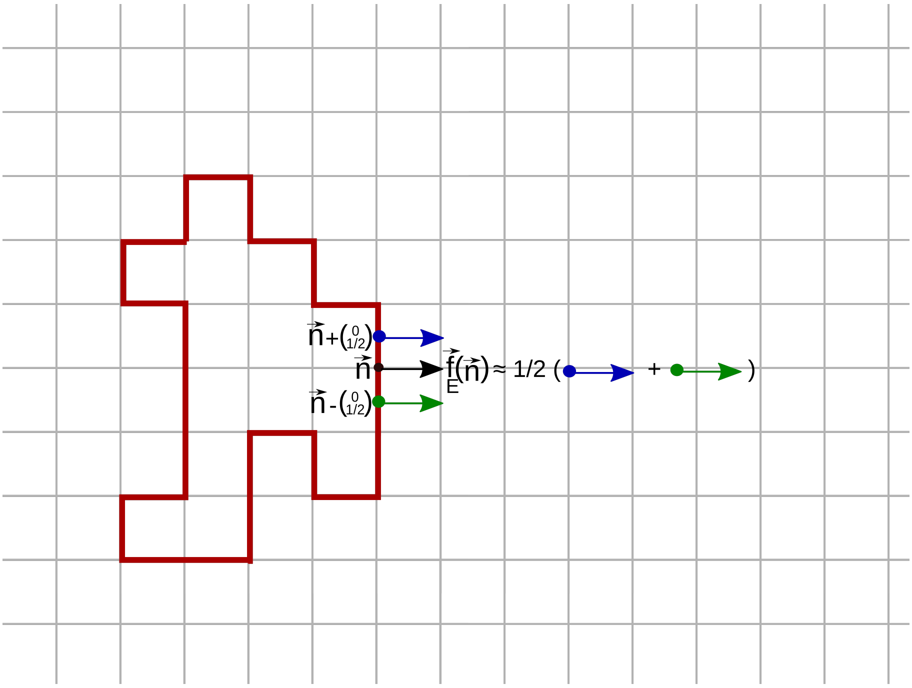

Figure 7.6: Example of approximating force component in direction East at a node

Deriving force from virtual work

Now, to find $\vec{f}(\vec{n})$ we can use equation 7.2 which gives us

$$\frac{\vec{f}(\vec{n})\cdot\delta\vec{x}_N}{|\delta\vec{x}_N|} = f_N(\vec{n}) \tag{7.8}$$

$$\frac{\vec{f}(\vec{n})\cdot\delta\vec{x}_{NE}}{|\delta\vec{x}_{NE}|} = f_{NE}(\vec{n}) \tag{7.9}$$

$$\frac{\vec{f}(\vec{n})\cdot\delta\vec{x}_E}{|\delta\vec{x}_E|} = f_E(\vec{n}) \tag{7.10}$$

$$\frac{\vec{f}(\vec{n})\cdot\delta\vec{x}_{SE}}{|\delta\vec{x}_{SE}|} = f_{SE}(\vec{n}) \tag{7.11}$$

This is an inverse problem. We can approximate $\vec{f}(\vec{n})$ by minimizing the sum of squares

$$\begin{gathered}\left(\frac{\vec{f}(\vec{n})\cdot\delta\vec{x}_N}{|\delta\vec{x}_N|} - f_N(\vec{n})\right)^2 + \left(\frac{\vec{f}(\vec{n})\cdot\delta\vec{x}_{NE}}{|\delta\vec{x}_{NE}|} - f_{NE}(\vec{n})\right)^2 \quad +\ldots \\ \left(\frac{\vec{f}(\vec{n})\cdot\delta\vec{x}_E}{|\delta\vec{x}_E|} - f_E(\vec{n})\right)^2 + \left(\frac{\vec{f}(\vec{n})\cdot\delta\vec{x}_{SE}}{|\delta\vec{x}_{SE}|} - f_{SE}(\vec{n})\right)^2\end{gathered} \tag{7.12}$$

We use a Levenberg-Marquardt algorithm that numerically minimizes equation 7.12 for $\vec{f}(\vec{n}) = (V cos(\alpha), V sin(\alpha))^T$ with $\alpha \in [0, 2\pi]$ and $V \in [0, \inf]$ with initial condition $\vec{f}(\vec{n}) = \max\{f_N(\vec{n}), f_{NE}(\vec{n}), f_E(\vec{n}), f_{SE}(\vec{n})\}$.