Chapter 22

Simulating 3D Cell Shape with the Cellular Potts Model

Rabea Link and Ulrich S. Schwarz

Abstract

Computer simulations have become a widely used method for the field of mechanobiology. An important question is whether one can predict the shape and forces of cells as a function of the extracellular environment. Different types of models have been described before to simulate cell and tissue shapes in structured environments. In this chapter, we give a brief overview of commonly used models and then describe the Cellular Potts Model, a lattice-based modelling framework, in more detail. We provide a hands-on guide on how to build a model that simulates the shape of a single cell on a micropattern in three dimensions in different open source software packages using the Cellular Potts framework. A simulation is set up with an initial configuration of generalized cells that change shape and position due to an energy function that incorporates cellular volume and surface area constraints as well as interaction energies between the generalized cells.

Key words Cell adhesion, Cell shape, Micropatterns, Modelling, Cellular Potts model, Computer simulations

1 Introduction

The emergence of mechanobiology as an own scientific field was triggered by the insight that shape and forces play much more important roles for the behavior and fate of cells than formerly appreciated in the fields of cell biology and cell biophysics [1–3]. Right from its genesis in the late 1990s, advances in mechanobiology were strongly shaped by new experimental technologies, such as adhesive micropatterning [4] or traction force microscopy [5]. In addition, mathematical models and computer simulations also played an important part in understanding how cell shape and forces emerge from the interplay of biochemical signaling, cytoskeleton rearrangement and interactions with the extracellular environment [6]. Here we focus on the question how cell shape can be predicted as a function of the geometry of the extracellular environment. This central question to mechanobiology has been addressed before with many different types of models, but mainly in two-dimensional ones [7]. Here we not only address three dimensions but also provide a practical guide for the general reader how to implement such procedures in common software packages.

There are several cell-based types of models that can be used to describe the shape dynamics of single cells [8, 9] or cell clusters [10, 11]. In these models, the cell is described on a mesoscopic scale with coarse-grained parameters such as surface tension or interaction energy, which arise from integrating out microscopic degrees of freedom. An energy function that depends on the boundary of the cell is minimized in the standard models to describe cell shape and dynamics. Differences in how the cell boundary is described and the energy is minimized lead to different models. Phase field models use a continuous field ϕ that differentiates between inside ($\phi = 1$) and outside of the cell ($\phi = 0$) [12, 13]. Cells can also be described by polygons in vertex models [14, 15], where vertices are shifted during the simulation to minimize an energy function and, therefore, the cells change their shape, position, and usually also neighborhood relations during the simulation.

In this chapter, we focus on the Cellular Potts Model (CPM) [16, 17]. In contrast to the continuum models mentioned before, the CPM is a lattice model, which means that a simulated cell cannot change its shape continuously in space but occupies a number of pixels (in 2D) or voxels (in 3D) on a lattice, thus leading to discrete shapes. The discrete approach decreases computation time compared to continuum models. A simulation with the CPM is easy to set up and can be adapted to different types of experiments. The framework has been used to simulate many different biological phenomena. The first experiments that were described with a CPM were cell sorting experiments [18, 19]. Other simulations include tumor growth, invasion and progression [20], single cell motility [21, 22], migration of single cells and cell clusters [23, 24] cells on micropatterns [25, 26] and cell competition in epithelia [27]. A disadvantage of the lattice-based formalism is that the calculation of bending energies is computationally expensive, thus other simulation frameworks are better suited when bending energies are important, as, for example, during endocytosis. Here, however, we are concerned only with the shapes of whole cells, which to first order are determined by membrane and cortical tensions of the cell envelope, a situation which is well described by the CPM.

2 Materials

There are many open source software available which can be used to implement CPM-simulations. In Table 1, we give a selected overview of CPM software that is freely available, namely CompuCell3D [28], Morpheus [29], Artistoo [30], Tissue Simulation

Table 1
Open source software packages that implement the Cellular Potts Model and the corresponding websites

Software name	URL
CompuCell3D	https://compucell3d.org/
Morpheus	https://morpheus.gitlab.io/
Artistoo	https://artistoo.net/
Tissue simulation toolkit	https://sourceforge.net/projects/tst/
Chaste	https://www.cs.ox.ac.uk/chaste/
Simmune	https://www.niaid.nih.gov/research/simmune-project

Toolkit [31], CHASTE [32], and Simmune [33]. In recent years, several approaches to parallelize the Cellular Potts Model have been published [34–36], which enable large-scale simulations. We start this chapter with an overview over the CPM-concept in Subheading 3. In Subheading 4 we then explain how to use CompuCell3D and Morpheus to simulate the three-dimensional shape of a cell on a micropattern.

3 Theory

In the CPM, cell shape is defined on a lattice, *see* Fig. 1. Sites of the lattice with the same index σ belong to the same generalized cell, which can be a biological cell, a sub-cellular compartment, or a physical object such as a substrate or a pillar. Sites that belong to the surrounding medium usually have the index $\sigma = 0$. The CPM is energy-based, and the effective energies that determine the dynamics of the simulation can be assigned to quantities such as volume or surface conservation, interaction between different generalized cells or chemotaxis. This description allows the inclusion of any biological mechanism to the model as long as it can be described with an energetic formalism.

The energy function which is used in the CPM is often referred to as Hamiltonian. It assigns an energy to any given configuration of cells on a lattice. Here we discuss the CPM in 3D. In its basic version, the Hamiltonian consists of an interaction term and a term to conserve volume:

$$\begin{aligned}\mathcal{H} = &\sum_{\mathbf{i},\mathbf{j}\text{ neighbors}} \left(1-\delta_{\sigma(\mathbf{i}),\sigma(\mathbf{j})}\right) J(\tau(\sigma(\mathbf{i})),\tau(\sigma(\mathbf{j}))) \\ &+ \sum_{\sigma} \lambda_V(\tau(\sigma))\left(v(\sigma) - V_{\tau(\sigma)}\right)^2\end{aligned} \tag{1}$$

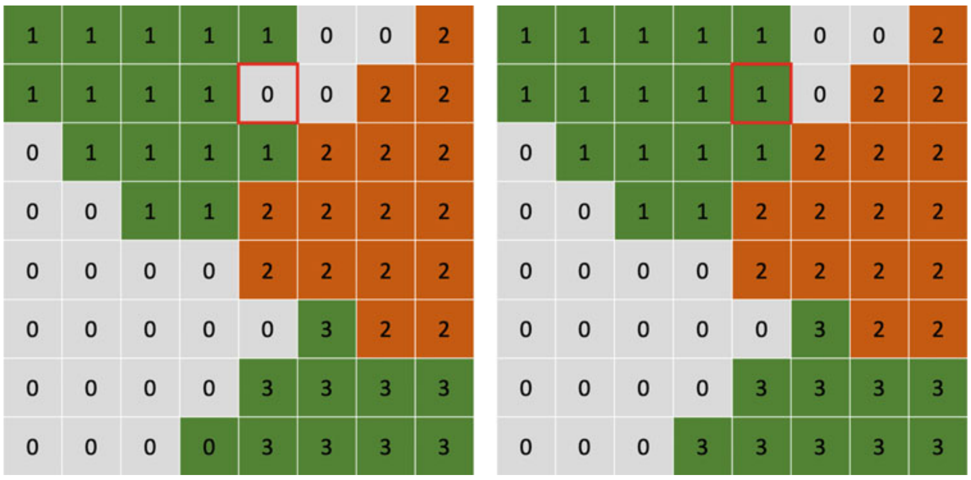


Fig. 1 2D Cellular Potts Model with three cells $\sigma = 1, 2, 3$ and medium $\sigma = 0$. Two cells have the same cell type τ and are thus depicted with the same color. A lattice site is picked at random (red) and its index is changed to that of the neighboring cell. Then the energy difference $\Delta\mathcal{H}$ due to this index change is calculated and the Metropolis algorithm determines if the spin flip is accepted

The first term sums over all neighboring voxels **i**, **j**. The Kronecker Delta $\delta_{\sigma(\mathbf{i}),\sigma(\mathbf{j})}$ is defined as 1, if the cells at **i** and **j** are the same, and 0 if they are different. This ensures that only if **i** and **j** belong to different cells the interaction term is different from 0. The interaction energy J then depends on the cell types of the cells at positions **i** and **j**. The second term of the Hamiltonian sums over all cells in the simulation. It has the form of an elastic energy, which means that the term is 0 if the cell has the target volume $V_{\tau(\sigma)}$ and the energy increases quadratically for differences between the actual volume $v(\sigma)$ and the target volume $V_{\tau(\sigma)}$. The strength with which this constraint is enforced is given by the rigidity $\lambda_V(\tau(\sigma))$, which depends on the cell type. The higher the value of λ_V, the more energy it costs the cell to have a volume that differs from the target value. The volume energy is needed to avoid that the cells shrink and disappear when the total energy is minimized during the simulation.

The Hamiltonian in Eq. 1 can be extended to include other important aspects like surface area constraints and chemotaxis. The surface area of a biological cell is fixed by the amount of cell membrane. This can be described with an elastic area constraint, similar to the volume constraint in Eq. 1:

$$\mathcal{H}_{\text{area}} = \sum_{\sigma} \lambda_A(\tau(\sigma))(a(\sigma) - A_{\tau(\sigma)})^2. \tag{2}$$

Adding the area constraint Eq. 2 to the Hamiltonian allows us to use negative interaction energies J. Using negative interaction energies J without constraining the area would lead to cells that maximize their surface area to reduce the total energy of the system, possibly even causing the cells to break into smaller parts.

A simple form of chemotaxis has the form of a potential energy

$$\mathcal{H}_{\text{chem}} = \mu(\tau(\sigma))C(\mathbf{i}), \tag{3}$$

where $C(\mathbf{i})$ is the chemical field at the target site $\mathbf{i}$ and $\mu(\tau(\sigma))$ describes the strength of the cellular response to the chemical field.

Depending on the biological system, other forms of the Hamiltonian can be better suited. For example, Albert and Schwarz [26] developed a 2D CPM based on the geometrical Hamiltonian introduced by Vianay et al. [25] to describe cell shape and forces on micropatterns. The Hamiltonian was chosen to be

$$\mathcal{H} = \sigma A + \lambda_s l + \sum_{\text{arc } i} \frac{EA}{2L_{0,i}}(L_i - L_{0,i})^2 - \frac{E_0}{A_{\text{ref}} + A_{\text{ad}}} A_{\text{Ad}}. \tag{4}$$

In contrast to the original Hamiltonian in Eqs. 1 and 2, the cell area A and perimeter l of the cell are not constrained by an elastic energy term but can vary in response to the adhesive environment. The main model parameters are now surface tension σ and line tension λ_s. An additional elastic line tension $\lambda_e = EA(L - L_0)/L_0$ is introduced to describe reinforced actin edge fibers that form between adhesive parts of the micropattern. EA is the one-dimensional elastic modulus of the fiber and L and L_0 are the contour length and the rest length, respectively. The last term describes the energy gain from cell adhesion, which is linear during initial cell spreading and then plateaus. A_{ad} is the adhesive area covered by the cell and E_0 and A_{ref} are parameters to control the strength and saturation of the adhesive energy. This model predicts the dynamics of cell shapes on micropatterned substrates but does not work in an unstructured environment.

A CPM simulation is set up with an initial configuration of the generalized cells, which is shown in Fig. 1 for illustrative purposes in 2D. A modified version of the Metropolis algorithm [16, 38] is used to evolve the cells and find the minimal energy configuration. First a site that is at the boundary of a generalized cell is chosen at random. The index of this site is then changed to the index of a randomly chosen neighboring site. Using the Hamiltonian from Eq. 1, the energy difference $\Delta\mathcal{H} = \mathcal{H}_{\text{after}} - \mathcal{H}_{\text{before}}$ between the old and the new configuration is calculated. If the new position has the same or lower energy, $\Delta\mathcal{H} \leq 0$ the index change is accepted. In case the new configuration has a higher energy than the old one, $\Delta\mathcal{H} > 0$, the index change is accepted with the probability

$$p = e^{-\frac{\Delta\mathcal{H}}{T}}. \tag{5}$$

T is called the simulation temperature, because it is the physical temperature when using the Metropolis algorithm to simulate the Boltzmann ensemble with microscopic degrees of freedom. In the CPM, the simulation temperature T does not correspond to the physical temperature, but a higher T value increases the probability to accept a spin flip and thus leads to more fluctuations in the simulation. At $T = 0$, only spin flips that lower the energy in the system are accepted, which prevents the simulation from overcoming energy barriers. Therefore, all simulations use an intermediate simulation temperature as compromise to generate dynamic shape changes while still keeping clear cell identities. One Monte Carlo step consists of n spin flip attempts, where n is the number of lattice sites. Note that only the indices of neighboring cells can be chosen for the spin flip. This modification of the Metropolis algorithm breaks detailed balance, which would be required if one simulated on a microscopic level [39].

4 Methods

In the following, we will discuss how to simulate a cell on a micropattern in 3D with different CPM software. We will simulate a single cell with volume and surface constraints on an H-shaped micropattern, which often is used to immobilize a single cell or a cell doublet without polarization [40], in contrast to the crossbow shape, which leads to polarization [41], or the square, which leads to rotation [40]. In the simulations, the H-pattern is implemented with fixed generalized cell types, which we call `Adhesive` and `NonAdhesive`. To build a simulation, many parameters must be set. We will suggest parameters throughout the instructions. To set up a simulation, parameter scans are a helpful tool to determine the parameter range that best corresponds to the experiment.

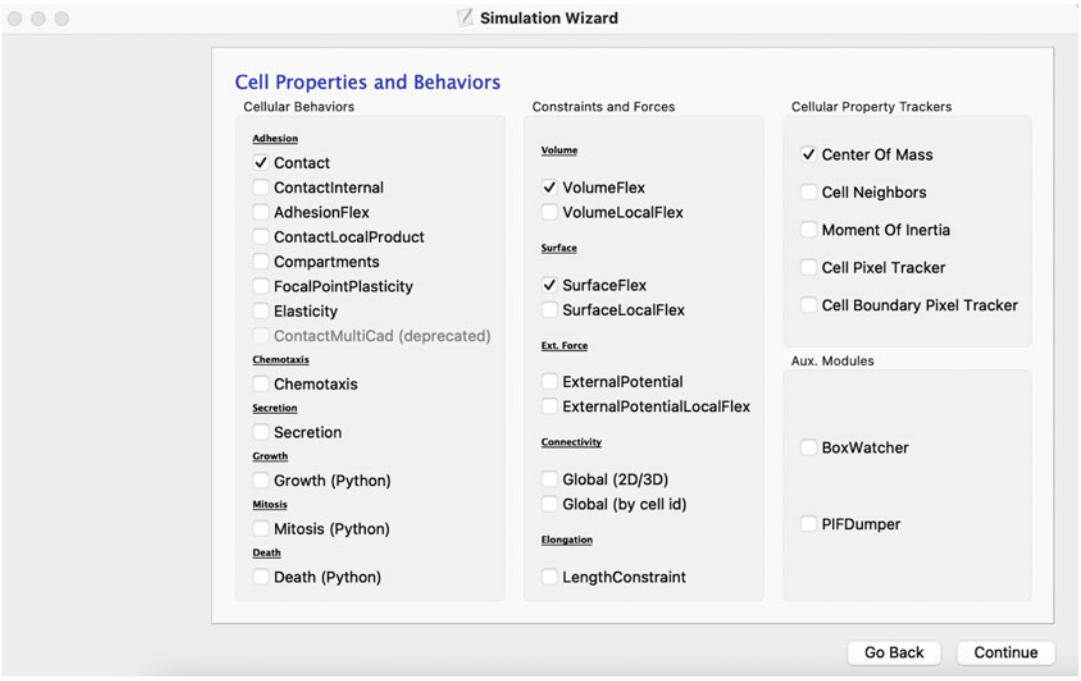
To simulate the cell shape on a micropattern, a good lattice size is $100 \times 100 \times 50$. The micropattern is in the x-y plane at $z = 0$. We choose the thickness of the adhesive part to be 4 voxels wide and 70 voxels long. Then, the cell is initialized as a half sphere with radius 28, sitting at the center of the micropattern. The target volume of the cell is $V = 50,000$ voxels, which is close to the initial cell volume.

4.1 CompuCell3D

The CompuCell3D software is provided by the group of James Glazier at Indiana University, USA. To download the software, follow the instructions at https://compucell3d.org/SrcBin#MostRecent. After installation, double-click the `twedit++.command` file to open the editor. The editor is customized to simplify the definition of new CompuCell3D simulations.

1. You can initialize a new simulation by choosing `CC3D Project` → `New CC3D Project...` in the menu bar.
2. The simulation wizard appears. You can enter the name and directory of your simulation. For the simulation type, choose `Python+XML`.

3. Next, you have to decide the dimensions of your simulation. A good choice for the micropattern simulation is 100×100×50. For the boundary condition, choose `NoFlux`. Change the `Average Membrane Fluctuations`, which corresponds to the temperature T in Eq. 5 to 700. In addition, change the `Pixel Copy Range (NeighborOrder)` to 3 and the number of steps to 1000. Because you want to start with a very specific initial cell layout, you will customize this after the simulation wizard is complete. Select the `Continue` button.
4. Add `Cell`, `Adhesive`, and `NonAdhesive` as generalized cell types. Because you only want the `Cell` to change shape during the simulation, freeze the other two cell types by selecting the corresponding check boxes.
5. In the next part, you are asked to add chemical fields, which you do not need for your simulation.
6. Then, you can specify cell properties and behaviors, *see* Fig. 2. The cellular behaviors and the constraints and forces add terms to the Hamiltonian. For the simulation you want to create, adhesion, volume, and surface energies are needed. These terms are added to the Hamiltonian when you select the check boxes for `Contact`, `VolumeFlex`, and `SurfaceFlex`.
7. After this, the simulation wizard is complete and code corresponding to your selected settings is automatically


Simulation Wizard

Cell Properties and Behaviors

Cellular Behaviors
- Adhesion: [✓ Contact], ContactInternal, AdhesionFlex, ContactLocalProduct, Compartments, FocalPointPlasticity, Elasticity, ContactMultiCad (deprecated)
- Chemotaxis: Chemotaxis
- Secretion: Secretion
- Growth: Growth (Python)
- Mitosis: Mitosis (Python)
- Death: Death (Python)

Constraints and Forces
- Volume: [✓ VolumeFlex], VolumeLocalFlex
- Surface: [✓ SurfaceFlex], SurfaceLocalFlex
- Ext. Force: ExternalPotential, ExternalPotentialLocalFlex
- Connectivity: Global (2D/3D), Global (by cell id)
- Elongation: LengthConstraint

Cellular Property Trackers
- [✓ Center Of Mass], Cell Neighbors, Moment Of Inertia, Cell Pixel Tracker, Cell Boundary Pixel Tracker

Aux. Modules
- BoxWatcher, PIFDumper

Fig. 2 Graphical user interface of the `Simulation Wizard` in CompuCell3D

generated. The code is distributed in three files, the `Main Python Script` which you do not need to change, the `XML Script`, where properties that stay the same throughout the simulation can be set, and the `Python Steppables` file, where you can use the Python programming language (Python Software Foundation, https://www.python.org/) to describe more complicated starting conditions, changes during the simulation and finishing steps.

8. Open the `XML Script`. In the first section of the `XML Script`, you can find the `Metadata` of the simulation, which you do not need to change. Then, the `Potts` section specifies the parameters that are important for the Cellular Potts Model algorithm, which you already specified in the simulation wizard. Then, several sections follow that specify which plugins are included in the simulation. In the `<Plugin Name="CellType">`, you can find the cell types, as defined in the wizard.
9. The `VolumeFlex` energy is included in the `<Plugin Name="Volume">` section and corresponds to the elastic volume constraint in Eq. 1. Here, you need to change the parameters assigned by the wizard: For the generalized cell type which you called `Cell` the target volume V must be changed to 50,000, and the parameter describing the strength of the elastic volume energy λ_V must be set to 2.0. It is called `LambdaVolume` in the simulation. Since the other two cell types are fixed you do not need volume energies for these cell types and can optionally delete the respective lines in the `XML Script`.
10. You can do the same in the next section for the elastic surface energy, which adds Eq. 2 to the Hamiltonian used in the simulation. You should change the parameters in the section `<Plugin Name="Surface">`. The target surface A of the cell must be changed to 13,000 and the parameter λ_A describing the strength to 1.0. The other lines for `Adhesive` and `NonAdhesive` cell types can be deleted.
11. The `<Plugin Name="CenterOfMass">` allows the user to easily access the center of mass of the cells in the simulation, which is not needed for your simulation but can be helpful in many cases.
12. The last plugin `<Plugin Name="Contact">` specifies the interaction energies between cell types. Here you define how much energy is gained or lost when two different cell types are close to each other. The interaction energies you should use are the following:

```
<Energy Type1="Medium" Type2="Medium">0.0</Energy>
<Energy Type1="Medium" Type2="Cell">20.0</Energy>
<Energy Type1="Medium" Type2="Adhesive">0.0</Energy>
<Energy Type1="Medium" Type2="NonAdhesive">0.0</Energy>
```

```
<Energy Type1="Cell" Type2="Cell">0.0</Energy>
<Energy Type1="Cell" Type2="Adhesive">-150.0</Energy>
<Energy Type1="Cell" Type2="NonAdhesive">-1.0</Energy>
<Energy Type1="Adhesive" Type2="Adhesive">0.0</Energy>
<Energy Type1="Adhesive" Type2="NonAdhesive">0.0</Energy>
<Energy Type1="NonAdhesive" Type2="NonAdhesive">0.0</Energy>
```

These interaction energies ensure that increasing the interface between cell and medium increases the energy and is thus unfavorable. What is more important in our context, increasing the interface between cell and adhesive decreases the overall energy of the system and ensures that the cell attaches to the micropattern. Because the `Cell` has a large volume, the `NeighborOrder` in the `<Plugin Name="Contact">` can be increased to 60.

13. Finally, you need to delete the `Steppable` that details the initial configuration of the simulation. We will write our own initial cell configuration in the `Python Steppables` file. Save the changes in the `XML Script`.
14. Open the `Python Steppables` file. In this file, you find a Python class with methods objects. These methods will be called when the simulation is initialized (`__init__`), before the first Monte Carlo step (`start`), at every Monte Carlo step (`step`), after the last Monte Carlo step (`finish`) or when the user stops the simulation (`on_stop`).
15. You should use the `start` method to describe the initial configuration of our simulation. In the code snippet below you create three cells, one of each cell type. Please make sure that you assign the correct `TypeId` to the correct cell. The `TypeId` of a cell type is defined in the `CellType` plugin in the `XML Script`.

```
cell = self.potts.createCell()
cell.type = 1
adhesive = self.potts.createCell()
adhesive.type = 2
nonadhesive = self.potts.createCell()
nonadhesive.type = 3
```

Then, we assign lattice voxels to the different cell types. We create here an initial configuration with a `NonAdhesive` surface of height 1 with an `Adhesive` “H” shaped micropattern. We place the cell as a hemisphere with radius 28 centered on the surface.

```
self.cell_field[0:99,0:99,0:1] = nonadhesive
self.cell_field[20:25,15:85,0:1] = adhesive
self.cell_field[20:80,48:52,0:1] = adhesive
self.cell_field[75:80,15:85,0:1] = adhesive
x_0=50
y_0=50
z_0=1
radius=28
for x in range(22,79):
   for y in range(22,79):
      for z in range(2,30):
        if (x-x_0)**2+(y-y_0)**2+(z-z_0)**2<
        radius**2:
           self.cell_field[x,y,z] = cell
```

16. To gain some insight on the simulation you can print the cell surface and volume to the terminal at every 10 Monte Carlo steps by adding the following lines to `step`:

```
for cell in self.cell_list:
   if cell.type==1 and mcs%10==0:
      print("area=", cell.surface, "volume=",
      cell.volume)
```

    Save the `Python Steppables` file.

17. You can start the simulation now: right-click on the `.cc3d` file on the left panel and select `Open In Player`. You should see the initial configuration of the simulation as in Fig. 3. In the `Player`, a graphics window appears where you can change between a 2D and a 3D view. If you do not see the simulation correctly, you can use the `Visualization` menu to zoom or reset your view. You can use the `Model Editor` to change parameter values during a simulation to see the effects directly. The volume and area of the cell is printed to the terminal. After less than 100 Monte Carlo steps, the final cell shape is reached, *see* Fig. 4, and the shape only fluctuates slightly due to the finite simulation temperature *T*.

4.2 Morpheus

The Morpheus software is provided by the group of Andreas Deutsch at Technical University Dresden, Germany. You can obtain the software from https://morpheus.gitlab.io/. After the installation, you can open the graphical user interface, *see* Fig. 6. In the `Documents` section on the upper left part of the user interface, you can see the structure of a Morpheus simulation. We will go through these points and discuss how to set up the simulation of a 3D cell on an H-shaped micropattern in the following.

1. Start by saving the empty simulation. Select the `Save` button and choose a name and a directory for the simulation.

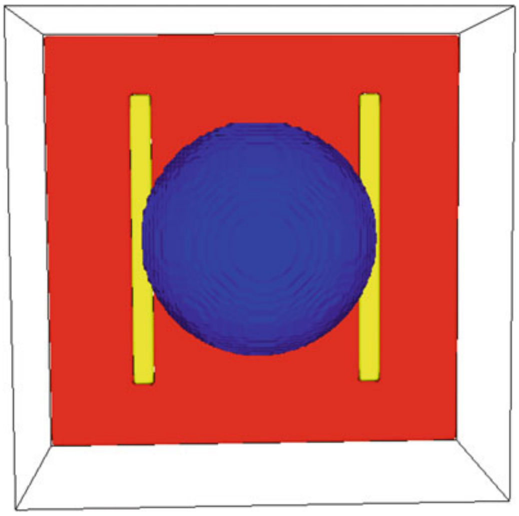


Fig. 3 After starting the simulation in CompuCell3D, the initial configuration of a single cell on an H-shaped micropattern can be seen in the graphics window

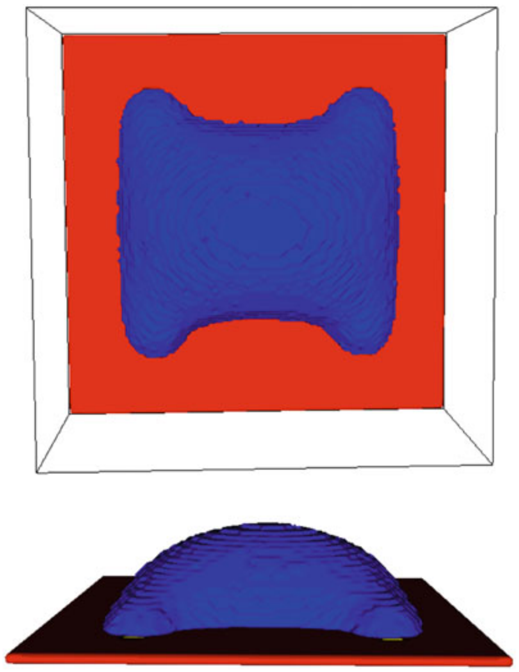


Fig. 4 Simulation result of a single cell on an H-shaped micropattern with CompuCell3D

Optionally, you can add a title and a description to the empty simulation.

2. Click on `Space` to move on to the next part. Select `Lattice` to change the lattice configuration of the simulation. In the `Attributes` window, change the lattice class from `linear` to `cubic` to change the space to three dimensions. Change the `Neighborhood` to 3. Change the size of the lattice to `100,100,50`.
3. Move on to `Time` and change the `StopTime` to 1000.
4. Next, double-click on `CellTypes`. You need four different cell types, *see* Fig. 6. In the Morpheus environment, the order in which we introduce the cell types is important. Start with the medium: click on `CellType`, change the class from `biological` to `medium` and add the name `Medium`. Unselect `CellType` and add another element by clicking on the + symbol in the top left corner and then choose `CellType`. Change the name of the new cell type to `Adhesive`. This cell type is frozen in our simulation. Select the adhesive cell type and click again on +. Select `FreezeMotion` and change the `Condition` to 1.0 to freeze the adhesive cell type. Repeat the same process to add the `NonAdhesive` cell type and freeze it as well. Finally, add the `Cell` as the last cell type, and do not freeze it but instead add a `VolumeConstraint` and a `Surfa-ceConstraint`. Change the parameters of the `VolumeCon-straint` to `strength 2` and `target 50000` and for the `SurfaceConstraint` change the `mode` to `surface` and choose the `strength 1` and `target 8000`.
5. Double-click on `CPM`. Select `Interaction` and click on the + to add `Contact`. This defines the interaction energies *J*. Add a total of three `Contact` elements and define the interaction energies between cell medium, cell adhesive, and cell nonadhesive to 20.0, − 15000.0, and − 1.0, respectively. The interaction energy between cell and adhesive is much lower in this simulation because we cannot choose a higher neighborhood order. Change the `Neighborhood Order` of `ShapeSurface` to 7, which is much lower because Morpheus does not allow the use of large neighborhoods. For a small neighborhood on a three-dimensional cubic lattice the 7th level neighborhood is a good choice [42]. Then, click on + and add the `MonteCarlo-Sampler` to define the temperature and the neighborhood size for choosing updates in the modified Metropolis algorithm. In `MetropolisKinetics`, change the `temperature` to `700.0`.
6. Now you can move on to the `CellPopulations`, where you define the initial configuration for the simulation. Start with the `Adhesive` cell type. Change the type of the `Population` to `Adhesive`. Then add `InitCellObjects` and change the `Arrangement` from `Sphere` to `Box`. To initialize the H-pattern, set the `origin` of the `Box` to `20.0,15.0,0.0`

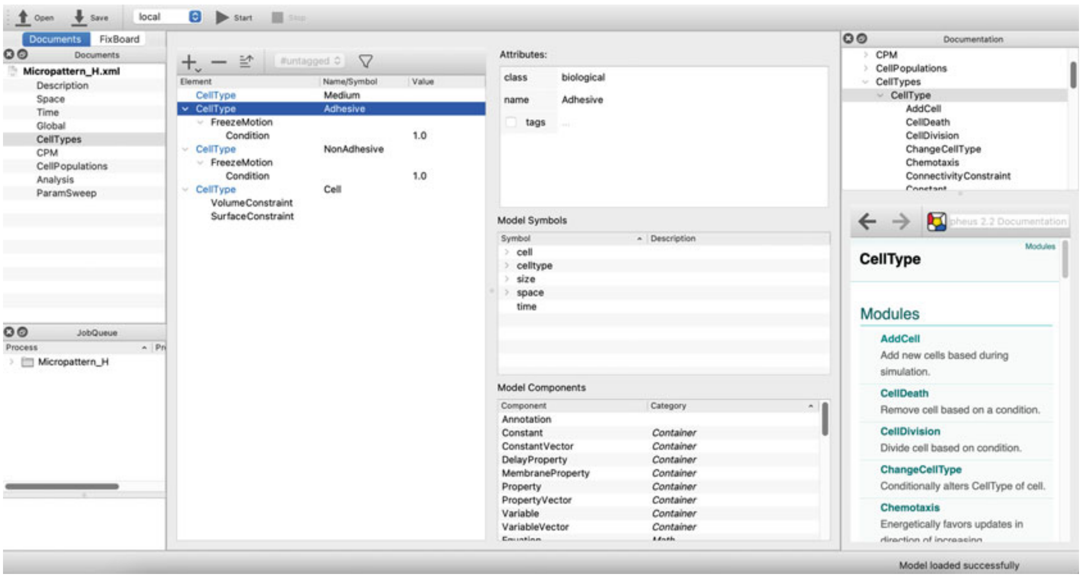
and the `size` to `5.0,70.0,1.0`. This is the first bar of the H-pattern. For the other bars, add two more `Arrangements` to `InitCellObjects`. The `origin` of the second `Box` is `20.0,48.0,0.0` with a `size` of `60.0,4.0,1.0` and for the third `Box` the `origin` is `75.0,15.0,0.0` with a `size` vector given by `5.0,70.0,1.0`. To include the `NonAdhesive` cell type to the initial configuration, add another `Population` and change the `type` to `NonAdhesive`. Add again the `InitCellObjects`. Change the `Sphere` to a `Box`, and change the `size` of the `Box` to `100.0,100.0,1.0`. This `Box` will not overwrite the previously defined `Adhesive` boxes, which is why the order of defining the cell types is important here. Finally, we can include the `Cell` in the initial configuration with a half sphere: Add the third `Population` and change the `type` to `Cell`. Add again `InitCellObjects`, but this time use the predefined `Sphere`. Change the `center` of the sphere to `50.0,50.0,1.0` and the `radius` to `28.0`.

7. In the `Analysis` part, add the `VtkPlotter` and change the `Channel` to `cell.type`, which allows you to visualize the results, e.g., in ParaView [43].
8. Then, you can start the simulation. 3D simulations are not visualized in Morpheus, but because the `VtkPlotter` was added, the results can be visualized in other programs like ParaView, *see* Fig. 5.

4.3 Conclusions

In this chapter, we have described in detail how one can simulate 3D cell shape in structured environments using the CPM. We have chosen the H-pattern as a standard pattern for cell immobilization without polarization [40], but it is easy to extend this approach now to other patterns of interest. To learn more about the importance of the parameter choice, we recommend to use different values than the ones suggested above and to see what changes in the simulations. For example, it is instructive to change the volume and surface area rigidities λ_V and λ_A as well as the target volume V and target surface area A.

It is interesting to note that the two software packages gave different results: while the shape predicted by CompuCell3D as shown in Fig. 4 has the invaginated shape often seen in experiments, the shape predicted by Morpheus as shown in Fig. 5 is more rounded. Technically, this difference arises from the fact that Morpheus uses a smaller neighborhood area to evaluate the spin Hamiltonian [42]. In fact the same result as obtained by Morpheus can be obtained by decreasing neighborhood order in CompuCell3D. Due to its larger variability in this aspect, CompuCell3D might be better suited to simulate single cell shape, while Morpheus is well suited to simulate ensembles of cells for which individual shape is less important. For a detailed comparison with experiments, the



Documents

- Micropattern_H.xml
  - Description
  - Space
  - Time
  - Global
  - CellTypes
  - CPM
  - CellPopulations
  - Analysis
  - ParamSweep

JobQueue

Process
Micropattern_H

Element	Name/Symbol	Value
CellType	Medium	
CellType	Adhesive	
FreezeMotion		
Condition		1.0
CellType	NonAdhesive	
FreezeMotion		
Condition		1.0
CellType	Cell	
VolumeConstraint		
SurfaceConstraint		

Attributes:

class	biological
name	Adhesive
tags	…

Model Symbols

Symbol	Description
cell	
celltype	
size	
space	
time	

Model Components

Component	Category
Annotation	
Constant	*Container*
ConstantVector	*Container*
DelayProperty	*Container*
MembraneProperty	*Container*
Property	*Container*
PropertyVector	*Container*
Variable	*Container*
VariableVector	*Container*

- CPM
- CellPopulations
- CellTypes
  - CellType
    - AddCell
    - CellDeath
    - CellDivision
    - ChangeCellType
    - Chemotaxis
    - ConnectivityConstraint

CellType

Modules

AddCell
Add new cells based during simulation.

CellDeath
Remove cell based on a condition.

CellDivision
Divide cell based on condition.

ChangeCellType
Conditionally alters CellType of cell.

Chemotaxis
Energetically favors updates in

Model loaded successfully

Fig. 5 Graphical user interface in Morpheus. The structure of the simulation can be seen in the `Documents` section. Parameters can be changed in the `Attributes` section

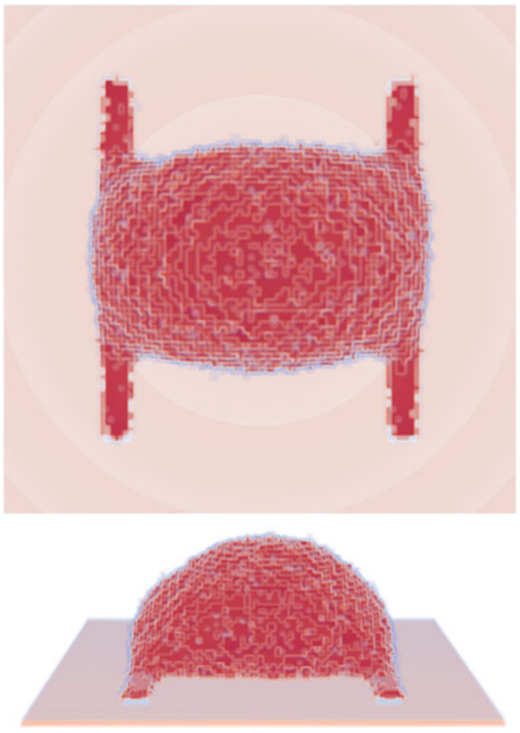


Fig. 6 Simulation result of a single cell on an H-shaped micropattern with Morpheus, visualized in ParaView

shapes obtained by the simulations can be compared with a variety of methods [6]. These include volume, surface area, and sphericity of the 3D cell shapes. In addition, one can compare the so-called landmark points, such as the height of the cell. More evolved methods to compare shapes would include Fourier descriptors and moment invariants, as implemented, for example, in the Konstanz information miner (KNIME) [44].

Acknowledgments

This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy via the Excellence Cluster 3D Matter Made to Order (EXC-2082/1 – 390761711). USS is a member of the Interdisciplinary Center for Scientific Computing (IWR) at Heidelberg.

References

1. Discher DE, Janmey P, Wang Y-L (2005) Tissue cells feel and respond to the stiffness of their substrate. Science 310:1139–1143. ISSN: 0036-8075, 1095–9203. http://www.sciencemag.org/content/310/5751/1139
2. Iskratsch T, Wolfenson H, Sheetz MP (2014) Appreciating force and shape—the rise of mechanotransduction in cell biology. Nat Rev Mol Cell Biol 15:825–833. ISSN: 1471–0072. http://www.nature.com/nrm/journal/v15/n12/abs/nrm3903.html
3. Hannezo E, Heisenberg C-P (2019) Mechanochemical feedback loops in development and disease. Cell 178:12–25. ISSN: 0092–8674. https://www.sciencedirect.com/science/article/pii/S0092867419306233
4. Ruprecht V et al. (2017) How cells respond to environmental cues – insights from bio-functionalized substrates. J Cell Sci 130: 51–61. ISSN: 0021-9533, 1477–9137. http://jcs.biologists.org/content/130/1/51
5. Schwarz US, Soine JRD (2015) Traction force microscopy on soft elastic substrates: a guide to recent computational advances. Biochim Biophys Acta Mol Cell Res Mechanobiol 1853: 3095–3104. ISSN: 0167-4889. http://www.sciencedirect.com/science/article/pii/S0167488915001822
6. Bodor DL, Pönisch W, Endres RG, Paluch EK (2020) Of cell shapes and motion: the physical basis of animal cell migration. Dev Cell 52: 550–562. ISSN: 1534-5807. https://doi.org/10.1016/j.devcel.2020.02.013
7. Albert PJ, Schwarz US (2016) Modeling cell shape and dynamics on micropatterns. Cell Adhes Migr 10:516–528. https://doi.org/10.1080/19336918.2016.1148864
8. Holmes WR, Edelstein-Keshet LA (2012) Comparison of computational models for eukaryotic cell shape and motility. PLoS Comput Biol 8:e1002793. ISSN: 1553-7358. https://doi.org/10.1371/journal.pcbi.1002793
9. Danuser G, Allard J, Mogilner A (2013) Mathematical modeling of eukaryotic cell migration: insights beyond experiments. Annu Rev Cell Dev Biol 29:501–528. https://doi.org/10.1146/annurev-cellbio-101512-122308
10. Liedekerke PV, Palm MM, Jagiella N, Drasdo D (2015) Simulating tissue mechanics with agent-based models: concepts, perspectives and some novel results. Comput Particle Mech 2:401–444. https://doi.org/10.1007/s40571-015-0082-3
11. Metzcar J, Wang Y, Heiland R, Macklin P (2019) A review of cell-based computational modeling in cancer biology. JCO Clin Cancer Inform 3:1–13. ISSN: 2473-4276. https://pubmed.ncbi.nlm.nih.gov/30715927/
12. Ziebert F, Swaminathan S, Aranson IS (2012) Model for self-polarization and motility of keratocyte fragments. J R Soc Interface 9:1084–1092. https://royalsocietypublishing.org/doi/abs/10.1098/rsif.2011.0433
13. Moure A, Gomez H (2021) Phase-field modeling of individual and collective cell migration. Arch Comput Methods Eng 28:311–

344. ISSN: 18861784. https://link.springer.com/article/10.1007/s11831-019-09377-1

14. Alt S, Ganguly P, Salbreux G (2017) Vertex models: from cell mechanics to tissue morphogenesis. Philos Trans R Soc B Biol Sci 372. ISSN: 14712970. https://doi.org/10.1098/rstb.2015.0520
15. Fletcher AG, Osterfield M, Baker RE, Shvartsman SY (2014) Vertex models of epithelial morphogenesis. Biophys J 106:2291–2304. ISSN: 0006-3495. https://doi.org/10.1016/j.bpj.2013.11.4498
16. Anderson A, Rejniak K (eds) (2007) Single cell based models in biology and medicine. ISBN: 978-3-7643-8101-1 and 3-7643-8101-9. Birkhäuser, Basel; Berlin [u.a.]
17. Scianna M, Preziosi L (2013) A Cellular Potts Model simulating cell migration on and in matrix environments. Math Biosci Eng 10: 235–261. https://doi.org/10.3934/mbe.2013.10.235
18. Graner F, Glazier JA (1992) Simulation of biological cell sorting using a two-dimensional extended Potts model. Phys Rev Lett 69:2013–2016. https://doi.org/10.1103/PhysRevLett.69.2013
19. Graner F, Glazier JA (1992) Simulation of the differential adhesion driven rearrangement of biological cells. Phys Rev E: Stat Nonlinear Soft Matter Phys 47:2128. https://doi.org/10.1103/PhysRevE.47.2128
20. Szabó A, Merks RMH (2013) Cellular Potts modeling of tumor growth, tumor invasion, and tumor evolution. Front Oncol 3: 87. ISSN: 2234-943X. http://www.ncbi.nlm.nih.gov/pubmed/23596570http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3627127
21. Marée AFM, Jilkine A, Dawes A, Grieneisen VA, Edelstein-Keshet L (2006) Polarization and movement of keratocytes: a multiscale modelling approach Bull Math Biol 5:1169–1211. ISBN: 1153800691. https://doi.org/10.1007/s11538-006-9131-7
22. Marée AFM, Grieneisen VA, Edelstein-Keshet L (2012) How cells integrate complex stimuli: the effect of feedback from phosphoinositides and cell shape on cell polarization and motility. PLoS Comput Biol 8:e1002402. ISSN: 1553-7358. https://doi.org/10.1371/journal.pcbi.1002402
23. Albert PJ, Schwarz US (2016) Dynamics of cell ensembles on adhesive micropatterns: bridging the gap between single cell spreading and collective cell migration. PLOS Comput Biol 12: e1004863. ISSN: 1553-7358. https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004863
24. Thüroff F, Goychuk A, Reiter M, Frey E (2019) Bridging the gap between single-cell migration and collective dynamics. eLife 8: e46842. ISSN: 2050084X. https://doi.org/10.7554/eLife.46842
25. Vianay B et al. (2009) Single cells spreading on a protein lattice adopt an energy minimizing shape. Phys Rev Lett 105:128101. https://doi.org/10.1103/PhysRevLett.105.128101
26. Albert PJ, Schwarz US (2014) Dynamics of cell shape and forces on micropatterned substrates predicted by a cellular Potts model. Biophys J 106:2340–2352. ISSN: 15420086. https://doi.org/10.1016/j.bpj.2014.04.036
27. Gradeci D et al. (2021) Cell-scale biophysical determinants of cell competition in epithelia. eLife 10. ISSN: 2050-084X. https://elifesciences.org/articles/61011
28. Swat MH et al. (2012) Multi-scale modeling of tissues using CompuCell3D. Methods Cell Biol 110:325–366. ISSN: 0091-679X. https://doi.org/10.1016/B978-0-12-388403-9.00013-8
29. Starruß J, De Back W, Brusch L, Deutsch A (2014) Morpheus: a user-friendly modeling environment for multiscale and multicellular systems biology. Bioinformatics 30:1331–1332. ISSN: 14602059. https://doi.org/10.1093/bioinformatics/btt772
30. Wortel IM, Textor J (2021) Artistoo, a library to build, share, and explore simulations of cells and tissues in the web browser. eLife 10. ISSN: 2050084X. https://doi.org/10.7554/eLife.61288
31. Daub JT, Merks RM (2015) Cell-based computational modeling of vascular morphogenesis using tissue simulation toolkit. Methods Mol Biol 1214:67–127. ISSN: 10643745. https://link.springer.com/protocol/10.1007/978-1-4939-1462-3_6
32. Cooper FR et al. (2020) Chaste: cancer, heart and soft tissue environment. J Open Source Softw 5:1848. ISSN: 2475-9066. https://joss.theoj.org/papers/10.21105/joss.01848
33. Meier-Schellersheim M, Mack G (1999) SIMMUNE, a tool for simulating and analyzing immune system behavior. arXiv: 9903017v1 [cs]. http://arxiv.org/abs/cs/9903017
34. Chen N, Glazier JA, Izaguirre JA, Alber MS (2007) A parallel implementation of the Cellular Potts Model for simulation of cell-based morphogenesis. Comput Phys Commun 176: 670–681. https://www.sciencedirect.com/science/article/pii/S0010465507002044

35. Gusatto É, Mombach JCM, Cercato FP, Cavalheiro GH (2005) An efficient parallel algorithm to evolve simulations of the cellular Potts model. Parallel Process Lett 15:199–208. https://doi.org/10.1142/S0129626405002155
36. Berghoff M, Rosenbauer J, Hoffmann F, Schug A (2020) Cells in Silico-introducing a high-performance framework for large-scale tissue modeling. BMC Bioinform 21:1–21. ISSN: 14712105. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-03728-7
37. Tomeu AJ, Salguero AG (2020) A lock free approach to parallelize the Cellular Potts Model: application to ductal carcinoma in situ. J Integr Bioinform 17:20190070. https://doi.org/10.1515/jib-2019-0070
38. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. J Chem Phys 21:1087. https://doi.org/10.1063/1.1699114
39. Voss-Böhme A (2012) Multi-scale modeling in morphogenesis: a critical analysis of the Cellular Potts Model. PLoS ONE 7:42852. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0042852
40. Tseng Q et al. (2012) Spatial organization of the extracellular matrix regulates cell-cell junction positioning. Proc Natl Acad Sci 109:1506–1511. https://www.pnas.org/doi/abs/10.1073/pnas.1106377109
41. Thery M et al. (2022) Anisotropy of cell adhesive microenvironment governs cell internal organization and orientation of polarity. Proc Natl Acad Sci 103:19771–19776. https://www.pnas.org/doi/abs/10.1073/pnas.0609267103
42. Magno R, Grieneisen VA, Marée AFM (2015) The biophysical nature of cells: potential cell behaviours revealed by analytical and computational studies of cell surface mechanics. BMC Biophys 8:1–37. http://www.biomedcentral.com/2046-1682/8/8
43. Ahrens J, Geveci B, Law C (2005) ParaView: an end-user tool for large-data visualization. Visualization handbook, pp 717–731. https://doi.org/10.1016/B978-012387582-2%2F50038-1
44. Dietz C, Berthold MR (2022) Focus on bio-image informatics. In: De Vos WH, Munck S, Timmermans J-P (eds). Springer, Cham, pp 179–197. ISBN: 978-3-319-28549-8. https://doi.org/10.1007/978-3-319-28549-8_7