BACHELOR

Kinetics of intracellular vesicle transport through narrow necks

van der Heijden, T.W.G.

Award date:
2013

Link to publication
Kinetics of intracellular vesicle transport through narrow necks

T.W.G. van der Heijden
0745667

Supervisors
R.P.T. Kusters
C. Storm

July 4, 2013
Abstract

In this report, we investigated the forces needed for molecular motors to pull intracellular vesicles through dendritic spine necks. These vesicles contain receptors, which allow the transmission of signals through the brain. At first, the bending and tension energies involved with the transport are simulated using Surface Evolver. Two limiting cases are modeled: the transport of a rigid sphere through a deformable cylinder and the transport of a deformable sphere through a rigid cylinder. The deformed cylinder shows a symmetrical energy evolution, with barriers at both the entrance and the exit of the cylinder. The sphere needs to overcome an energy barrier for both bending and tension, after which it will be able to pass through the cylinder.

After the examination of the bending and tension energies involved with the transport, a Lattice Boltzmann method is used to mimic a liquid environment of the vesicle. After all, the neurons in the brain are filled with cytoplasm. A geometry is created, consisting of two dendritic spine necks connected to each other. Then, the influences of several parameters on the motion of the capsule are investigated. The time spent by the capsule in the spine neck seems to be dependent on the reciprocal of the radius of the capsule. This possible relation could be validated by repeating the exercise for different body force densities and parameters for the capsule. The dependency of the time spent in the neck on the neck length seems to be almost linear, from a certain neck length on. The effect of the body force density on the time spent in the neck could not be determined exclusively, but there is of course an inverse relation: increasing the force means decreasing the time. The research was ended with a phase diagram, indicating whether a capsule of a certain size could pass the spine neck, given the body force density.
## Contents

1 Introduction 3

2 Theory 5
   2.1 Curvature and energy of surfaces ........................................... 5
      2.1.1 Bending and tension energy ........................................... 5
      2.1.2 Implementation in Surface Evolver ................................. 6
   2.2 Three-dimensional Lattice Boltzmann method ............................ 7
      2.2.1 Modeling the fluid .................................................... 7
      2.2.2 Geometry used in the simulations .................................. 8
      2.2.3 Parameters for the capsule used in the simulations .............. 10

3 Simulations concerning bending and tension energies 11
   3.1 Analytical approximation for bending and tension energies of a deformable cylinder 11
   3.2 Bending and tension energies of a deformable cylinder .......... 12
   3.3 Bending and tension energies of a deformable vesicle .............. 14

4 Simulations concerning capsules in liquid 16
   4.1 Vesicle radius dependency ............................................... 16
   4.2 Neck length dependency .................................................. 18
   4.3 Body force density dependency ........................................... 20
   4.4 Phase diagram ............................................................... 20

5 Discussion 23

Bibliography 26

A Surface Evolver 27

B Parametrization of the dendritic spine 30
Chapter 1

Introduction

The neuronal system uses chemical and electrical signals to transfer information through the brain [1, 2]. Electronic signals in the brain are transmitted through this network of neurons. A signal travels from the cell body of a nerve cell through the axon, the ‘sending’ nerve fiber, to the synapse. The electronic signal from the axon triggers the release of chemicals, so-called neurotransmitters, into the synaptic cleft. These neurotransmitters are captured by receptors which move on the cell membrane of dendrites, the ‘receiving’ fibers of neurons. A schematic representation of a synapse is given in figure 1.1.

Figure 1.1: Schematic representation of a synapse: neurotransmitters are released into the synaptic cleft. They are captured by AMPA receptor proteins, which move along the membrane by lateral diffusion and are actively transported to the spine head in intracellular vesicles, called endosomes. The receptors are fixed chemically on the post-synaptic density.
The most important receptor proteins are the AMPA-type glutamate receptors. The local density of these proteins is governed by three physical processes, as illustrated in figure 1.1:

- Active transport of receptor proteins in intracellular vesicles
- Passive diffusion of the proteins over the cell membrane
- Chemical fixation of the proteins on the cell membrane

In this report, we investigate the forces and energies involved with the first process: the active transport of endosomes through the ‘neck’ of the dendritic spine. The intracellular vesicle is pulled through the neck by molecular motors, attached to the underlying actin cytoskeleton [2]. What force do these motors need to exert to get the vesicle through the spine neck? To examine this, we need to track the total bending and tension energies during the motion.

In reality, both the vesicle and the spine neck are deformable, although to different degrees. Therefore, we track the energies during two limiting cases: the transport of a rigid sphere through a deformable cylinder and the transport of a deformable sphere through a rigid cylinder. In order to do this, we model the elasticity of the vesicle and the neck using the Surface Evolver software package. This software uses numerical methods to minimize the free energy of surfaces [3, 4]. In other words, it finds the equilibrium shape of a surface. The transport processes are done stepwise, and at every step, the bending and tension energies are calculated. Before the simulations with Surface Evolver, an analytical approach of the energies involved with the first limiting case is made, to be able to compare the results of the simulations to the analytical result.

Additionally, we invoke a three-dimensional Lattice Boltzmann method to mimic the presence of the surrounding liquid. After all, the neuron is filled with cytoplasm through which the vesicle needs to be transported. The method consists of a numerical approximation of the fluid with vesicle and serves as a Navier-Stokes solver [5]. In the model, spherical capsules are pushed through a constriction using a body force density on the fluid and the capsule. We investigate the influence of several parameters in the simulations. First, the influence of the capsule radius is examined. We expect to find a critical radius for which the neck cannot be passed anymore, given the geometry and body force density. Then, the neck length dependency is investigated. The effect will probably be almost linear: when the neck length is increased, the time spent in the neck also increases. Thirdly, we examine the effect of the body force density on the motion. There will, again, likely be a critical value for the force density where the vesicle gets stuck. Eventually, the results are combined into a phase diagram that shows whether the vesicle gets stuck for a certain combination of force density and size, in a given geometry. The time the capsule spends in the neck is a function of the size of the particle, the force exerted on the particle and the length of the neck.

In chapter 2, we will discuss the theoretical background concerning bending and tension of curved surfaces and the implementation in Surface Evolver of the energies involved. Moreover, we will discuss the theory behind the Lattice Boltzmann method. In chapter 3, the results of the Surface Evolver simulations are discussed. Then, in chapter 4, the results of the simulations performed with the three-dimensional Lattice Boltzmann method are covered. Finally, in chapter 5, we present the main conclusions of our work and set up a discussion about the sensitivities of the simulations and future research goals.
Chapter 2

Theory

In this chapter we discuss the theoretical background of the methods used in the project. First, in section 2.1, the theoretical background concerning bending and tension energy of surfaces is covered, which is necessary to examine the energies involved with the transport of vesicles through narrow necks. Then, we discuss how these energies are implemented in Surface Evolver. In section 2.2, we discuss the theory behind the three-dimensional Lattice Boltzmann used in the simulations. These simulations will enable us to explicitly model the cytoplasm through which the vesicle moves. We will also describe the used geometry and state the relevant parameters for the particle.

2.1 Curvature and energy of surfaces

To be able to calculate the equilibrium shape of the vesicle membrane, given some constraints, and the force needed to pull the vesicle through a narrow neck, the total energy evolution needs to be calculated and tracked during the passing through the neck. The equilibrium shape of the membrane is the shape for which the total free energy, being the sum of the energies due to bending (the curvature of the surface) and surface tension, minimizes.

2.1.1 Bending and tension energy

The Canham-Helfrich bending energy $E_{\text{bend}}$ of a surface $S$ is given by the following expression [6]:

$$E_{\text{bend}} = \int_S \left( \frac{\kappa}{2} (2H)^2 + \kappa_G K \right) dA,$$

(2.1)

with $\kappa$ and $\kappa_G$ the bending rigidity and the Gaussian bending, respectively. $H$ is the mean curvature of the surface and $K$ its Gaussian curvature. They are defined by:

$$H = \frac{1}{2} \left( \frac{1}{R_1} + \frac{1}{R_2} \right),$$

(2.2a)

and:

$$K = \frac{1}{R_1 R_2},$$

(2.2b)

where $R_1$ and $R_2$ are the local, two-dimensional, radii of curvature. Due to the Gauss-Bonnet theorem, the integral $\int_S \kappa_G K dA$ is constant, so this energy can be neglected and we only need to consider the mean curvature $H$ [6, 7]. After all, we evaluate a closed surface: the vesicle. The surface $S$ can be parameterized by $s_i$ and described by a vector $\vec{R}(s_1, s_2)$. Then, two tangential vectors $\vec{R}_i$ are defined, that span the surface:

$$\vec{R}_i = \frac{\partial}{\partial s_i} \vec{R}. \quad (i = 1, 2)$$

(2.3)
CHAPTER 2. THEORY

Also a metric tensor \( g \) is defined, with elements \( g_{ij} \):

\[
g_{ij} = \vec{R}_i \cdot \vec{R}_j \quad (2.4)
\]

This metric tensor is linked to the area element \( dA \) of the surface:

\[
dA = \sqrt{\det(g)} ds_1 ds_2. \quad (2.5)
\]

For vectors \( \vec{R}_1 \) and \( \vec{R}_2 \) span the surface \( \vec{R} \), the unit normal vector of the surface, \( \vec{n} \), can be expressed in terms of the cross product of the two tangential vectors:

\[
\vec{n} = \frac{\vec{R}_1 \times \vec{R}_2}{|\vec{R}_1 \times \vec{R}_2|} \quad (2.6)
\]

Using expressions (2.3) and (2.6), a curvature tensor \( h \) can be defined, with elements \( h_{ij} \):

\[
h_{ij} = \left( \frac{\partial}{\partial s_i} \vec{R}_j \right) \cdot \vec{n}, \quad (2.7)
\]

from which the mean curvature can be calculated:

\[
H = -\frac{1}{2} g^{ij} h_{ji}, \quad (2.8)
\]

with \( g^{ij} \) elements of the inverse tensor of \( g \).

When we take equation (2.1) and use the area element \( dA \) from equation (2.5), the bending energy that needs to be minimized is given by (the integrated Gaussian curvature is a constant):

\[
E_{\text{bend}} = 2\kappa \int_S H^2 dA = 2\kappa \int_S H^2 \sqrt{\det(g)} ds_1 ds_2. \quad (2.9)
\]

The tension energy of a surface is obtained by simple integration of the surface tension \( \sigma \) over the surface \( S \):

\[
E_{\text{tens}} = \int_S \sigma dA = \int_S \sigma \sqrt{\det(g)} ds_1 ds_2. \quad (2.10)
\]

2.1.2 Implementation in Surface Evolver

(For a overview of the main features of Surface Evolver and the data files used in the simulations, see Appendix A.)

Surface Evolver is software that models surfaces by triangulation of the surface. Because the calculations are done on a discrete number of vertices, edges and faces, an approximation has to be found for the squared mean curvature \((H^2)\) in order to calculate the bending energy in equation (2.9) [4]:

Each vertex \( v \) of the surface has a number of surrounding facets with area \( A_v \). The ‘force’ \( F_v \) due to surface tension on this vertex is

\[
F_v = -\frac{\partial A_v}{\partial v}. \quad (2.11)
\]

Every facet has three vertices, so the area belonging to \( v \) is \( A_v/3 \). This gives an average mean curvature \( H_v \) at \( v \) of

\[
H_v = \frac{1}{2} \frac{F_v}{A_v/3}. \quad (2.12)
\]

The contribution of the vertex \( v \) to the total integral is then:

\[
E_{\text{bend},v} = H^2_v \cdot A = H^2_v \cdot A_v/3 = \frac{1}{4} \frac{F^2_v}{A_v/3}. \quad (2.13)
\]
The tension energy is quite easy to calculate. It is, in principle, the total area of all facets multiplied by the surface tension $\sigma$. The contribution of one facet $f$ to the total integral is thus:

$$E_{\text{tens},f} = \sigma A_f,$$  \hfill (2.14)

with $A_f$ the area of the facet $f$. The total energy is then calculated by simply summing the local bending and tension energies over all vertices $v$ and facets $f$.

**Adjusted form bending energy**

An adjusted form of the mean curvature in expression (2.12) uses the volume gradient $N_v$:

$$H_v = \frac{1}{2} F_v \cdot N_v / N_v / 3.$$

The contribution to the total integral becomes:

$$E_{bend,v} = (H_v - H_0)^2 A_v / 3 = \left[ \frac{3}{2} F_v \cdot F_v - H_0 \right)^2 \frac{A_v}{3},$$  \hfill (2.16)

with $H_0$ a prescribed mean curvature. This method is preferable because it does not need a closed surface (a complete surface around a vertex); boundary edges are taken to be on mirror symmetry planes. Additionally, the standard expression (2.13) is subject to instabilities: sharp corners grow sharper instead of averaging out. Therefore, we use this adjusted form of the bending energy (2.16) in the simulations.

### 2.2 Three-dimensional Lattice Boltzmann method

The three-dimensional Lattice Boltzmann method (LB3D) is a widely used method to model and simulate particle movement in fluids [5]. It is used to model a liquid environment of the vesicle during the transport through a narrow neck, as an approximation of the cytoplasm in human cells.

#### 2.2.1 Modeling the fluid

The Lattice Boltzmann method starts with the Lattice Boltzmann equation, which is a numerical solver for the Bhatnagar-Gross-Krook approximation of the Boltzmann equation (2.17):

$$\frac{\partial f}{\partial t} + \vec{c} \cdot \nabla f = -\frac{1}{\tau} (f - f^{eq}),$$  \hfill (2.17)

where $f$ is the local velocity distribution, $\vec{c}$ the particle velocity and $\tau$ a relaxation time.

In a Lattice Boltzmann method, particles move on a fixed lattice and collide at the lattice nodes. The evolution of populations of particles $f_i$ is given by the Lattice Boltzmann equation:

$$f_i(\vec{x} + \vec{c}_i^* \Delta t, t + \Delta t) - f_i(\vec{x}, t) = -\frac{1}{\tau} (f_i(\vec{x}, t) - f_i^{eq}(\vec{x}, t)) + F_i \Delta t.$$  \hfill (2.18)

$\tau$ is a dimensionless relaxation parameter, connected to the kinematic viscosity $\nu$ and the speed of sound in the medium $c_s$:

$$\nu = c_s^2 \left( \tau - \frac{1}{2} \right) \Delta t,$$  \hfill (2.19)

where

$$c_s = \sqrt{\frac{1}{3} \frac{\Delta x}{\Delta t}},$$  \hfill (2.20)
with $\Delta x$ the lattice constant and $\Delta t$ the length of one time step. The populations move along the velocity vectors $\vec{c}_i$ to direct neighbours. There they collide according to equation (2.18). The equilibrium populations are given by

$$f_i^{eq} = w_i \rho \left( 1 + 3 \vec{c}_i \cdot \vec{u} + \frac{9}{2} (\vec{c}_i \cdot \vec{u})^2 - \frac{3}{2} \vec{u} \cdot \vec{u} \right),$$

(2.21)

with $\vec{u}$ the velocity of the population and $w_i$ the lattice weight, depending on the lattice structure. $\rho$ is the mass density of the fluid. A body force $\vec{f}$ is implemented via $F_i$:

$$F_i = \left( 1 - \frac{1}{2 \tau} \right) w_i \left( \vec{c}_i - \vec{u} + \frac{\vec{c}_i \cdot \vec{u}}{c_s^2} \right) \cdot \vec{f}.$$

(2.22)

In this specific case, a D3Q19 lattice is used [8]. This is a three-dimensional lattice, with 19 different velocity directions. A schematic representation of the lattice is shown in figure 2.1.

![Schematic representation of the D3Q19 lattice and its lattice vectors $c_i$.](image)

**Figure 2.1: Schematic representation of the D3Q19 lattice and its lattice vectors $c_i$.**

Every time step in the simulation, the following process occur:

- Advection: Propagation of the particles from position $\vec{r}$ to $\vec{r} + \vec{c}_i$;
- Collision: At each lattice position, the particles interact; they collide. In this process, several steps are taken:
  - Calculation of Hamiltonian;
  - Calculation of forces;
  - Redistribution of particles at each site:
    - The density $\rho^\sigma$ and fluid velocity $\vec{u}^\sigma$ of each component are computed:
      $$\rho^\sigma = m^\sigma \sum_i f_i^\sigma \quad \text{and} \quad \rho^\sigma \vec{u}^\sigma = m^\sigma \sum_i \vec{c}_i f_i^\sigma,$$
      with $m^\sigma$ the molecular mass of species $\sigma$.
    - The average velocity $\vec{v}$ is calculated:
      $$\vec{v} = \sum_{\sigma} \frac{\rho^\sigma \vec{u}^\sigma}{\sum_{\sigma} \rho^\sigma},$$
      with $\tau_{\sigma}$ the dimensionless relaxation parameter of species $\sigma$ (see equation (2.18)).
    - The force term is calculated.
    - The equilibrium densities for each species and each velocity vector $f_i^{\sigma,eq}$ are found.
    - The collision takes place according to equation (2.18).

2.2.2 Geometry used in the simulations

The geometry used for the LB3D simulations basically consists of two dendritic spine necks put together, connecting the necks (for the parametrization of the spine neck, see Appendix B). The top of the dendritic spine is left open; instead of closing it, a cylinder is placed on top of the
neck. An example of this ‘double spine’ geometry is shown in figure 2.2a. In the simulations, this geometry consists of a $100 \times 100 \times 100$ grid of zeros and fives, representing the fluid and the wall, respectively. The ‘five’ stands for a solid with no wetting interaction, while a value of for example 4.5 implies a wetting interaction with parameter -0.5 [8]. In the case of figure 2.2, the neck has a length $l$ of 50, where the neck is defined as the part of the grid between the two cylinders. See figure 2.2b.

Figure 2.2: The ‘double spine’ geometry used in the LB3D simulations.

The cylinder radius $R$ and the neck radius $d$ are kept constant throughout all simulations, at $R = 20$ and $d = 9$, while the neck length $l$ and the body force $f$ vary. This body force $f$ is a force applied to the fluid on every fluid node, including the fluid nodes within the particle (see figure 2.3).

Figure 2.3: Illustration of the body force $f$ applied to the fluid.
2.2.3 Parameters for the capsule used in the simulations

The capsule put in the system to model the intracellular vesicle is defined by a triangular mesh [5]. An example of such capsule is given in figure 2.4.

The capsule is defined by several parameters:

- Radius $r$ : radius of the capsule.
- Volume constraint constant $k_v$ : quantifies the resistance to volume change of the particle due to deformations.
- Total area constraint constant $k_{at}$ : quantifies the resistance to change in total area of the capsule due to deformations.
- Local area constraint constant $k_{al}$ : quantifies the ‘tension’ of the capsule; the resistance to local area change of the capsule.
- Bending constraint constant $k_b$ : quantifies the resistance to bending of the surface.
- Shear constraint constant $k_s$ : quantifies the resistance to shear of the surface.

In figure 2.5 bending (figure 2.5b) and shear (figure 2.5c) of the surface are visualized.

![Figure 2.4: The triangulated surface of a capsule with radius $r$.](image)

![Figure 2.5: Schematic visualization of various deformations possible for the capsule.](image)
Chapter 3

Simulations concerning bending and tension energies

In this chapter, we investigate the bending and tension energies involved with the transport of a vesicle through a tube. In reality, both the vesicle and the tube (spine neck) are able to deform. Here, we analyze two limiting cases, being the transport of a rigid sphere through a deformable cylinder and the transport of a deformable sphere through a rigid cylinder. They are both analyzed with Surface Evolver and the first case is supported using a theoretical approach. For this reason, in section 3.1, we propose an analytical approximation to model the bending and tension energies of a deformed cylinder. Then, we use Surface Evolver to model the two limiting cases and to investigate their kinetics. At first, in section 3.2, a rigid sphere is transported through a deformable cylinder. Secondly, in section 3.3, an initially spherical particle (vesicle) is transported through a rigid cylinder. We monitor the bending and tension energies during both ‘transports’. For the used input data files, see Appendix A.

N.B.: The displayed energy landscapes are merely progressions as function of the position. The amplitudes of the energies are not relative to each other.

3.1 Analytical approximation for bending and tension energies of a deformable cylinder

In order to validate the simulations we will preset in section 3.2, an analytical approximate model is made, aiming at calculating the bending and tension energies involved in the transportation of a rigid sphere through a deformable tube. Therefore, a cylinder is modeled on which we superimpose a ‘bump’ representing the presence of the solid sphere. The parametrization of this bumped cylinder is as follows:

\[
\begin{align*}
  x(u, v) &= \left(1 + \frac{2}{2\pi(h-v)^2}\right) \cos(u) \\
  y(u, v) &= \left(1 + \frac{2}{2\pi(h-v)^2}\right) \sin(u) \\
  z(u, v) &= v
\end{align*}
\]  

(3.1)

with \(0 < u < 2\pi\), \(-5 < v < 5\) and \(h\) a parameter for the location of the ‘bump’. A schematic view of the system is given in figure 3.1. The location of the ‘bump’ is varied by changing \(h\) and at each step the bending and tension energies are calculated, using equations (2.9) and (2.10). Strictly speaking, the given graphs of \(E_{\text{bend}}\) and \(E_{\text{tens}}\) are divided by \(2\kappa\) and \(\sigma\), respectively. The result is shown in figure 3.2. Firstly, the graph is completely symmetric in the middle of the cylinder (in this case: \(h = 0\)), as one may expect considering the symmetries involved. Secondly, when the ‘bump’ reaches the cylinder, the tension energy increases, which makes sense, as the total area of the cylinder increases. A more notable phenomenon is the decrease of the bending energy...
when the perturbation reaches the cylinder. This seems peculiar, because the equilibrium system is disturbed. However, at first, the disturbance of the cylinder can be seen as just an increase in radius. For the mean curvature $H$ of the cylinder is dependent on $\frac{1}{R}$ (equation (2.2a)), and its area on $R$, the bending energy $E_{\text{bend}}$ would be dependent on the reciprocal of $R$ ($E_{\text{bend}} = \int_S H^2 \, dA \sim \frac{1}{R} \cdot R \sim \frac{1}{R}$). This would mean that, at first, the bending energy indeed decreases. When the bump proceeds through the tube, its curvature causes, as expected, a rise in the bending energy of the cylinder.

In physical systems it is required that the total energy increases when deforming the cylinder, as the created equilibrium shape (the cylinder) has the lowest free energy. In other words, the sum of the bending and tension energies should be larger than or equal to the total energy of the undisturbed cylinder.

After this theoretical approximation of the energies involved in the transport of a rigid sphere through a deformable cylinder, we will next, in section 3.2, examine (approximately) the same situation. However, we now perform numerical simulations, using Surface Evolver.

### 3.2 Bending and tension energies of a deformable cylinder

First, a system is initialized consisting of a cuboid. Its length is fixed and the constraint for the system is the minimum radius of what will become the cylinder. The system is refined and equilibrated until the cuboid has become a cylinder, with its symmetry axis in the $z$-direction. Subsequently, a hard constraint is made around a certain value of $h$, which acts as a rigid sphere being forced through the cylinder, where $h$ is the position of the center of the sphere. During the simulation, the value of $h$ is varied gradually and the bending and tension energies are monitored. To make sure the cylinder follows the contour of the sphere fairly closely, the tension energy $E_{\text{tens}}$ is taken to be much more important than the bending energy $E_{\text{bend}}$. The equilibrium shape of the cylinder with sphere depends, of course, on the ratio between $|E_{\text{tens}}|$ and $|E_{\text{bend}}|$ (thus the values of $\sigma$ and $\kappa$), but we have chosen for $|E_{\text{tens}}| \gg |E_{\text{bend}}|$ to check if the Surface Evolver simulations correspond to the analytical model in section 3.1. The results of the simulations are shown in figure 3.3a.

This model should produce roughly the same results as the analytical approach in section 3.1. As can be seen from figure 3.3a, this is the case for the tension energy. It also has a maximum at the middle of the cylinder and has approximately the same shape. The bending energy, however,
CHAPTER 3. SIMULATIONS CONCERNING BENDING AND TENSION ENERGIES

Figure 3.3: Graph of the bending energy $E_{\text{bend}}$ and tension energy $E_{\text{tens}}$ as function of the position of the rigid sphere $h$, obtained by 1000 iterations per step, for $|E_{\text{tens}}| \gg |E_{\text{bend}}|$.

shows some minor deviations with the analytical model. Still, some justification can be done by looking at the model more closely.

The energy evolution of the transportation is highly similar to the analytical approach. The largest difference is the sharpness of the peaks: the peaks in the energy landscape of the Surface Evolver model are less sharp than the peaks in the analytical model. This could be caused by the parametrization of the bumped cylinder. The bump is, of course, not spherical, but only an approximation; the actual form has a curvature slightly different from the curvature of a sphere. The second peak in the energy is slightly smaller than the first; the figure is not symmetric. This is caused by numerical artifacts in the data, see figure 3.3b. The fluctuations in the first peak are much larger than in the second peak, which is probably an artifact from the forced constraint on the surface.

Another reason for the deviation could be the number of iterations performed before measuring the energies. To examine this factor, the same process is repeated with a varying number of iterations. The results are shown in figure 3.4a.

From figure 3.4a can be seen that the energy curve converges to a (nearly) symmetrical curve, highly similar to the curve in figure 3.2. To justify the number of iterations performed, the difference in peak height $\Delta E_{\text{peak}}$ relative to the final peak height $E_{\text{eq}}$ is evaluated as function of the number of iterations. The result is shown in figure 3.4b. From figure 3.4b can be derived that the relative peak energy deviation declines to approximately 10% within 200 iterations. This means a minimum of 200 iterations needs to be used to obtain a reliable result. Furthermore, the deviation barely decreases any further when increasing the number of iterations to more than 200. This is probably because of the (already discussed) numerical fluctuations in the first peak, shown in figure 3.3b.

After having investigated the deformations of a cylinder extensively and having obtained a reliable result, we now make the transition to the other limiting case concerning vesicle transport: the transport of a deformable sphere through a rigid cylinder.
3.3 Bending and tension energies of a deformable vesicle

In this section, the cylinder is kept rigid and the vesicle is made deformable. Initially, the system consists of a cube, with a fixed target volume. The system is refined and equilibrated until the cube has become a sphere, with the given target volume. Then a constraint is imposed: the local radius of the sphere is limited by the radius of a cylinder with a certain length. Analogous to section 3.2, the cylinder constraint is moved gradually over the sphere. Due to the constraint, the sphere deforms and its bending and tension energies are tracked as function of the location of the top of the cylinder $h$. The results are displayed in figure 3.5.

Figure 3.5: The bending energy $E_{\text{bend}}$ and tension energy $E_{\text{tens}}$ as function of the position of the rigid cylinder $h$. 
In figure 3.5, both the bending energy and the tension energy show approximately the same evolution along the transportation through the cylinder. This is to be explained by the 'mushroom-like' shape of the vesicle when entering the cylinder. Both the curvature and the total area are at its maximum value, because of the border between the cylinder and the free space above it. This border causes a sharply curved surface of the vesicle, rising the bending energy of the surface. The vesicle is also stretched strongly, causing the tension energy to rise. When the vesicle exits the cylinder, the system is left to relax freely, so both energies decrease gradually. Physically, the shape of the energy evolution means the vesicle needs to pass a barrier when entering a rigid cylinder, but it is allowed to exit the cylinder freely. This was to be expected; once the vesicle has been able to enter the cylinder, it will also be able to leave it.
Chapter 4

Simulations concerning capsules in liquid

In this chapter, three-dimensional Lattice Boltzmann simulations are performed to mimic a liquid environment of the vesicle, in addition to its bending and tension energies. Human cells, including neurons, are filled with cytoplasm, through which the vesicle moves; we model the cytoplasm with a liquid. The geometry used, consisting of two spine necks connected to each other, has already been discussed in section 2.2.

In this chapter, we examine the influence of several parameters on the kinetics of the capsule in the geometry: in section 4.1, the dependency of the particle radius on its kinetics is examined. Then, in section 4.2, we investigate the influence of the length of the constriction and in section 4.3, we examine the dependency on the applied body force density. In section 4.4, we combine the previous results to make a phase diagram to visualize whether the capsule passes the spine neck given a certain combination of body force density and size of the particle. In these sections, all quantities are expressed in lattice units.

In order to let a large capsule pass the neck, the tension \( k_{al} \) needs to be sufficiently low. Otherwise, the particle is not able to deform enough to pass through the neck. In this chapter, the constraint parameters for the capsule in the simulations are (as discussed in section 2.2):

\[
\begin{align*}
  k_v &= 1, \\
  k_{at} &= 0, \\
  k_{al} &= 0.5, \\
  k_b &= 0.05 \	ext{ and } k_s = 0.018.
\end{align*}
\]

4.1 Vesicle radius dependency

At first, the geometry as in figure 2.2 in chapter 2 is used to investigate the influence of the radius of the particle on its kinetics in this specific geometry. The neck length is \( l = 50 \) and the neck radius is \( d = 9 \). Also, \( z_{\text{start}} = 25 \), to make the geometry symmetric. The body force density applied to the fluid is \( f = 90 \cdot 10^{-6} \). An example of the progress of the capsule through the tube is shown in figure 4.1.

![Figure 4.1: The progress of a capsule with \( r = 15 \) through the geometry.](image-url)
We track the position \( z \) of a particle with radius \( r \) as function of the number of time steps \( t \) and investigate the evolution of the graph when the relative radius \( r/d \) is varied. The results are shown in figure 4.2a. The shapes of the graphs are quite as expected. Capsules smaller than or about<br><br>(a) The position \( z \) of the capsule with radius \( r \) as function of the number of time steps \( t \), for varying \( r/d \). \( r/d \) varies from 0.22 to 2.00.<br><br>(b) The time spent in the neck \( t_{\text{neck}} \) as function of the relative radius \( r/d \). There is a critical value for the radius: \( r_C/d \approx 1.736 \). The log-log plot of \( t_{\text{neck}} \) as function of \( (r_C - r)/d \) indicates a reciprocal dependency; its slope is -1.<br><br>Figure 4.2: LB3D simulations with varying radius of the capsule \( r \) to examine the influence of \( r \) on the kinetics of the capsule.<br><br>the same size as the spine neck are accelerated when passing the neck, as they do not experience any friction with the walls and the fluid accelerates in the neck. For larger particles, the capsule decelerates, because it touches the walls and needs to deform to be able to pass the neck. For even larger particles, the capsule gets stuck and does not pass the neck at all. From figure 4.2a, there seems to be a critical value of \( r/d \) for which the capsule gets stuck. To find that critical value \( r_C/d \), the time spent in the neck \( t_{\text{neck}} \) is displayed as function of \( r/d \) in figure 4.2b. From figure 4.2b, \( t_{\text{neck}} \) seems to be dependent on \( r/d \) as \( \sim \frac{1}{(r_C - r)/d} \). The found value of \( r_C/d \) is \( r_C/d \approx 1.736 \). This value is used to test the reciprocal dependency of \( t_{\text{neck}} \) on \( r/d \). Therefore, a log-log plot is made of \( t_{\text{neck}} \) as function of \( (r_C - r)/d \). The result is also shown in figure 4.2b. The straight line in the figure has slope -1. In other words, \( t_{\text{neck}} \sim \left( \frac{r_C - r}{d} \right)^{-1} \), so the dependency of \( t_{\text{neck}} \) on \( (r_C - r)/d \) indeed seems to be reciprocal for this specific geometry. A particle with this critical radius would be at the top of an energy barrier, as in figure 3.5 in chapter 3. One would need only a small force to either get the particle through the neck or to pull the capsule from it. To check conclusively whether this dependency is truly reciprocal, one should repeat this exercise for different body force densities and parameters of the capsule.<br><br>Justification for simulations<br><br>When the system is initialized, both the fluid and the particle are at rest. Then a body force density is applied to the fluid, making the whole system move. The question rises whether the capsule suffers from any start-up deviations in both shape and velocity and if it would be better to let the system equilibrate before ‘measuring’. Therefore, the contour of a large vesicle (in this case, \( r = 14 \)) is examined, for these particles deform the strongest. In figure 4.3 the contour of the vesicle is displayed at \( z = 20 \) (well before touching the neck) and \( z = 120 \) (the same position, after one cycle).<br><br>As can be seen in figure 4.3, no difference can be distinguished between the two shapes, so apparently the deviations concerning shapes are negligible.<br><br>To examine the deviation in start-up velocity \( v_{\text{start}} \) compared to the velocity after one cycle \( v_{\text{end}} \), a linear fit is made through the data from \( z = 8 \) (the starting position of the capsule) to \( z = 25 \).
(the start of the neck) and from \(z = 108\) to \(z = 125\) (the same positions, after one cycle). An example is given in figure 4.4a.

![Figure 4.3: Contours of a particle with \(r = 14\), at positions \(z = 20\) and \(z = 120\).](image)

The deviation of the first slope compared to the second is then calculated and repeated for different radii. The result is shown in figure 4.4b. Figure 4.4b shows that the deviation between the two velocities is always well below 3%. This means the start-up effects are negligible, so the system does not have to equilibrate before tracking the position. The capsule has already almost reached its equilibrium motion before entering the spine neck.

### 4.2 Neck length dependency

In this section, the influence of the length of the spine neck on the kinetics of the vesicle is examined. Therefore, the geometry of section 4.1 needs to be slightly adjusted. The radii of both the spine head \((R)\) and the neck \((d)\) have the same values; \(R = 20\) and \(d = 9\). The applied body force density is also still \(f = 90 \cdot 10^{-6}\). The position of the start of the neck \(z_{\text{start}}\) is now fixed at \(z_{\text{start}} = 20\), to make sure that the particle always has the same distance to travel before entering the neck. The neck length is varied from \(l = 50\) to \(l = 80\). We investigate the process using two different radii of the particle: \(r = 13\) and \(r = 14\). At first, the position of the center of mass \(z\) is tracked as function of the number of time steps \(t\). The results are shown in figure 4.5.

Both graphs are very similar to each other. We see, however, a small deviation in the curves, even
before entering the neck, while all curves should coincide there. After all, the body force density $f$ and the capsule radius $r$ are the same. These deviations are caused by the change in geometry. The same box of $100 \times 100 \times 100$ is used, but the neck length $l$ is increased. This effectively fills the box with more ‘wall’, so increasing $l$ decreases the present fluid volume, and thus the flow changes. From figure 4.5 can be seen that for a longer neck, the time to reach the end of the geometry increases, as was to be expected. To investigate this dependency, the number of time steps in the neck $t_{\text{neck}}$ is plotted as function of the neck length $l$ in figure 4.6.

From figure 4.6, it seems that the dependency of $t_{\text{neck}}$ on $l$ is almost linear. This is probably not entirely true. For one thing, the line should intersect both axes at the origin: when the neck length is zero, there is no neck, thus no time needed to pass it. For another, when the neck length is about the same size as the capsule, the dependency would likely be disturbed. This has to do with the capsule ‘feeling’ the presence of the exit of the neck. It is allowed to relax to its spherical shape before its other end has even entered the neck. The relaxation causes an extra force, pulling the capsule through the neck. The possible linearity will, probably, only be valid from a certain neck length. However, the geometry unfortunately does not allow further investigation of this limit.

---

**Figure 4.5:** The position $z$ as function of the number of time steps $t$, for two different radii of the capsule.

**Figure 4.6:** The number of time steps spent in the neck $t_{\text{neck}}$ as function of the neck length $l$, for two different radii of the capsule.
4.3 Body force density dependency

In this section, we investigate the influence of the body force density on the kinetics of the capsule in the neck. The same geometry as in section 4.1 is used: \( R = 20, d = 9, l = 50 \) and \( z_{\text{start}} = 25 \). The body force density is varied from \( f = 0.5 \cdot 10^{-6} \) to \( f = 90 \cdot 10^{-6} \) and a particle with radius \( r = 11 \) is used. We track again the position of the capsule \( z \) as function of the number of time steps \( t \). The result is shown in figure 4.7a.

\[
\text{(a) The position } z \text{ of the capsule with radius } r = 11 \text{ as function of the number of time steps } t, \text{ for varying body force density } f.
\]

\[
\text{(b) The time spent in the neck } t_{\text{neck}} \text{ as function of the body force density } f.
\]

Figure 4.7: LB3D simulations with varying body force density \( f \) to examine the influence of \( f \) on the kinetics of the capsule with radius \( r = 11 \).

The curves in figure 4.7a follow the expectations. When we decrease the body force density \( f \), the capsule will take longer before it deforms, so the time spent in the neck \( t_{\text{neck}} \) increases. The transition between different curves goes smoothly. Around a certain critical body force density \( f_C \), the capsule makes a transition from passing the neck to getting stuck. To find the value of \( f_C \), we plot \( t_{\text{neck}} \) as function of \( f \). The result is shown in figure 4.7b.

From figure 4.7b, no exclusive relation can be found between \( t_{\text{neck}} \) and \( f \), other than that \( t_{\text{neck}} \) decreases as \( f \) increases. There seems, however, to be some kind of inverse dependency, with a critical body force density \( f_C \) of about \( f_C \approx 10 \cdot 10^{-6} \). Especially around this critical force density \( f_C \), the curves deviate more from each other. The resolution cannot be increased much further, for the simulations get less reliable at this limit: as soon as the position does not change, a pressure buildup occurs, which is able to ‘choppily’ push the capsule through the neck. These abrupt steps differ per simulation, so it is possible for smaller forces to get the capsule through faster than larger forces.

4.4 Phase diagram

In order to make a phase diagram which indicates whether the capsule will pass through the neck or not, we need to combine the results of sections 4.1 and 4.3 and perform extra simulations in this range of radius of the capsule \( r \) and body force density \( f \). The geometry is also the same as in these sections, for the dependency on the neck length \( l \) seems to be linear when we increase \( l \) above 50, according to section 4.2. We vary the radius \( r \) between \( r = 9 \) and \( r = 15 \) and the body force density \( f \) between \( f = 0 \) and \( f = 90 \cdot 10^{-6} \). We do not investigate particles with \( r < 9 \), because \( d = 9 \) and particles will always pass the neck when smaller than the neck itself, for \( f > 0 \). The results of the simulations are shown in figure 4.8.

We see from figure 4.8 that, of course, when we increase the size of the particle \( r/d \), the force density \( f \) needed to pass the neck also increases. The horizontal dashed line represents the data from section 4.1, where the size \( r/d \) is varied while keeping the body force density \( f \) constant.
CHAPTER 4. SIMULATIONS CONCERNING CAPSULES IN LIQUID

Figure 4.8: Results of the simulations with varying \( r \) and \( f \). For every combination of \( r/d \) and \( f \) is monitored whether the capsule passes the neck or not.

The vertical dashed line represents the data from section 4.3, where the size of the particle is kept constant, but the body force density \( f \) is varied. For every measured size \( r/d \), the critical force density \( f_C \) is calculated by averaging the smallest ‘pass-force’ and the largest ‘stuck-force’. The result is a phase diagram, which is shown in figure 4.9.

Figure 4.9: A phase diagram that indicates whether the capsule passes the neck (in this specific geometry), for a certain combination of the size \( r/d \) and the applied body force density \( f \). The black line represents the critical value of the body force density \( f_C \), as function of \( r/d \).

In the range of the small particles (\( r/d = 0.0 - 1.5 \)) the critical force seems to rise increasingly. When the particles get larger, this effect decreases, probably because the larger particles will be able to feel the exit of the spine neck. They do not need to be deformed completely, for they are allowed to relax before entering the neck entirely. Less deformation means less energy loss, so a relatively smaller force needed to push the capsule through the neck.
The obtained phase diagram provides us with a boundary between passing the neck and getting stuck. When biological systems regulate the transportation process, it is useful to be able to make transitions while changing as little as possible. Therefore, in this case, the particle would be of about 1.5 times the neck radius: only a little variation in both the applied force and the particle size can cause a transition between passing and getting stuck.

To be able to relate the obtained phase diagram to the biological system, we need to convert the lattice units to SI units, using equations (2.19) and (2.20) in chapter 2. However, this has not yet been done and is left as a subject for future research.
Chapter 5

Discussion

In this chapter, we will summarize the conclusions of this project. The general goal of this project was to investigate the forces needed for the molecular motors to pull an intracellular vesicle, containing AMPA receptors, through a dendritic spine neck. We did this by modeling the bending and tension energies during the transport of the vesicle. Because, in reality, both the vesicle and the spine neck are able to deform, we started by modeling two limiting cases of this problem, being the transport of a rigid sphere through a deformable cylinder and the transport of a deformable sphere through a rigid cylinder.

For both cases we calculated the bending and tension energies involved, using Surface Evolver. In addition, we validated the case of a rigid sphere through a deformable cylinder with an analytical approximation. Therefore, we calculated the energies of a cylinder, on which we superimposed a symmetrical ‘bump’. The position of the bump was determined by a parameter $h$. Then, we varied $h$ and tracked the bending and tension energies as function of $h$. The result was a symmetrical energy landscape. The tension energy increased when the bump entered the cylinder, for the total area of the bumped cylinder increased. The bending, however, showed a small drop in energy when the bump entered the cylinder. This could be explained by the mean curvature of the cylinder, which depends on the reciprocal of its radius $R$. When the bump enters, at first only the local radius increases, so the curvature (and with it, the energy) decreases. When the bump enters further, its curvature causes the energy to rise again.

When the same situation, the transport of a rigid sphere through a deformable cylinder, was modeled with Surface Evolver, we found a similar curve. The deviations with the analytical model were due to the difference in shape (the analytical ‘bump’ did not have the same shape as the sphere in Surface Evolver) and numerical artifacts. Unfortunately, even when increasing the number of iterations (or equilibration steps), these artifacts did not seem to disappear completely. This was probably due to the forcing of the sphere through the cylinder. It is likely that the result will be less sensitive when the system (cylinder and sphere) is initialized again for every value of $h$, which could be a topic for future research.

Afterwards, the other limit case was investigated: the transport of a deformable sphere through a rigid cylinder. We found that the vesicle needs to overcome an energy barrier when entering the tube, for both bending and tension energy. When it has completely deformed and is traveling through the tube, the energy remains constant. When the vesicle then exits the cylinder, it is left to equilibrate gradually, so the energies decrease, also, gradually. Physically, this means that the vesicle needs to overcome an energy barrier at the entrance of the neck, and when it has completely entered, it can also pass the neck without further barriers. Possibly, in the future, both the deformable sphere and deformable cylinder can be combined into one model to simulate the more realistic situation. After all, both the vesicle and the spine neck are deformable, be it to different degrees.

In addition to evaluating the tension and bending energies related to the transport of a sphere through a cylinder, we modeled the deformable sphere with a Lattice Boltzmann simulation to include the liquid environment in which the vesicle has to be transported. The reason for this
is that all human cells, including neurons, are filled with cytoplasm. For these simulations, a ‘double spine’ geometry was used as a model for the transport of a vesicle through a rigid spine neck. At the boundaries of the geometry, the conditions were periodic. The force used to move the vesicle through the neck was implemented as a body force density: a force density applied to the whole fluid, inside and outside of the capsule. The way in which these forces on the capsule were implemented, was therefore not physically accurate. In reality, the vesicles in the dendritic spines are pulled forward by molecular motors and not pushed by a body force on the whole fluid. Future investigation has to clarify whether the found relations concerning the transport of capsules through the geometry are comparable to the ‘real’ situation. With this knowledge, the results may eventually be related to experimental results of biophysicists.

We started the Lattice Boltzmann simulations by examining the influence of the radius of the capsule on its kinetics. The result was as expected: the larger the particle, the longer it took before it passed the constriction in the geometry. When the capsule with radius \( r \) became larger than a critical size \( r_C/d \), with \( d \) the radius of the neck, it got stuck. For this specific ‘double spine’ geometry and a fixed body force density \( f = 90 \cdot 10^{-6} \), the time spent in the neck \( t_{\text{neck}} \) seemed dependent on the reciprocal of the size of the particle \( r/d \), with a critical size \( r_C/d \approx 1.736 \).

Subsequently, we investigated the effect of the spine neck length \( l \) on the motion of the capsule, which showed an almost linear behavior: the time the capsule spent in the spine neck \( t_{\text{neck}} \) was proportional to the length of the neck \( l \). This linear dependency would probably come in effect from a certain neck length. The first argument supporting this is that when \( l \) is equal to zero, there is no neck, so the capsule does not need any time to pass it. The curve should therefore intersect at the origin. Secondly, when the neck length \( l \) becomes about the same dimension as the radius of the capsule \( r \), the capsule starts to ‘feel’ the exit of the neck. It is able to relax before it has completely entered, creating an extra force that pulls the capsule out, which decreases the time spent in the neck. This real dependency could not be checked: the geometry did not allow further examination. This could be solved in the future by slightly changing the geometry and decreasing the neck length.

Afterwards, we examined the influence of the applied body force density \( f \). Of course, the time spent in the neck \( t_{\text{neck}} \) increased when the body force density \( f \) decreased, but no conclusive relation could be found between \( t_{\text{neck}} \) and \( f \). There seems, however, to be a critical force density \( f_C \), for which the particle gets stuck in the geometry. For this specific geometry and a capsule with radius \( r = 11 \), the found critical force is approximately \( f_C \approx 10 \cdot 10^{-6} \). The resolution could not be increased much further, for when the particle gets stuck, a pressure buildup occurs. This buildup causes the capsule to pass the neck in abrupt steps, making the simulation results less reliable. The steps turn out to differ per simulation, so it was possible for smaller forces to get the capsule through faster than larger forces.

The final result concerning the Lattice Boltzmann simulations was a phase diagram, that showed whether the capsule would pass the neck (in this specific geometry), given a certain body force density \( f \) and the particle size \( r/d \). In the range of small particles, the critical force \( f_C \) seemed to rise almost exponentially. This effect decreased when the capsules became bigger. This is because the larger particles are, as mentioned before, able to feel the exit of the neck, decreasing the force needed to push them through.
Acknowledgments

I would like to thank Remy Kusters and Kees Storm from group TPS, who were really involved and interested in the project, and whom I could consult whenever I needed help or advise. I would also like to thank Badr Kaoui and Jens Harting from group MTP, who made the Lattice Boltzmann simulations possible in the first place, helped map out specific goals for the research and were available to help solve any problems during the project. Besides, my thanks to Sebastian Schmieschek, who recompiled the code right away when I had the need for it.

Furthermore, thanks to Cyril Vrusch, Mathijs Vermeulen, Bart van Lith, Marijke Valk and Stefan Paquay, for their company during the internship and the many fruitful (and less fruitful) discussions.
Bibliography


Appendix A

Surface Evolver

The Surface Evolver package

Surface Evolver is a computer program that minimizes the energy of surfaces [3]. These surfaces can be subject to constraints. The minimization is done by evolving the surface down the energy gradient. Surface Evolver is used in this report to determine equilibrium shapes of vesicles being pulled through a tube (as an analogy for the spine neck) and their bending free energy. We will preset the main features in the proceeding of this section. Then, the data files for the simulations are given. For more information, see [3] and [4].

Initializing problems

In the data file, the geometry, constraints, parameters and energy contributions are defined. The geometry is initialized by defining vertices, defined by coordinates \((x, y, z)\) and connected by edges. These edges form faces and multiple faces can form a body. All vertices, edges and faces can be subject to constraints, such as fixed coordinates or contact angles. These constraints may or may not contribute to the total energy. When the total geometry with constraints is defined in the data file, the software can be started and the defined system can be evolved.

Evolving surfaces

The Surface Evolver software works with a command prompt, where commands have to be given and the evolution of the surface can be watched real-time. When a data file is loaded, the defined faces are automatically triangulated. The most performed actions are:

- **refinement** \((r)\) of the surface: each triangle on the surface is divided in smaller triangles;
- **evolution** \((g)\) of the surface: every vertex is moved so, that the total energy decreases;
- **vertex averaging** \((V)\): every vertex is moved to the average position of its neighbors.

Using these commands recursively, the surface reaches its equilibrium shape. The detail in which it is displayed depends on its refinement.

Extracting properties from the simulations

In order to extract properties from the simulations, like the bending energies or the position of the center of mass of the bodies, so-called quantities can be defined. Quantities are integrals over vertices, edges or faces, that represent the needed properties. The values of the quantities can be displayed any time during the simulation.
APPENDIX A. SURFACE EVOLVER

Data file for the transport of a rigid sphere through a deformable cylinder

```
parameter r_tube = 1
parameter r_sph = 2
parameter start_t = -11
parameter z_sph = 3
parameter height = 8
parameter tens = 1

gravity_constant 0

quantity smc energy method star_perp_sq_mean_curvature global
quantity cyl_tens energy modulus 1e6 method density_facet_area global

constraint zmax
formula: z=start_t+height

constraint zmin
formula: z=start_t

constraint tube nonnegative
formula:
  (z<(z_sph+r_sph) && z>(z_sph-r_sph) && ((r_sph)^2-(z-z_sph)^2)>(r_tube)^2)
    ? x^2+y^2+(z-z_sph)^2-r_sph^2 : x^2+y^2-r_tube^2

vertices
1  r_tube  0  start_t  constraint tube zmin
2   0  r_tube  start_t  constraint tube zmin
3  -r_tube  0  start_t  constraint tube zmin
4   0  -r_tube  start_t  constraint tube zmin
5   0   r_tube  start_t+height  constraint tube zmax
6   0   r_tube  start_t+height  constraint tube zmax
7  -r_tube  0  start_t+height  constraint tube zmax
8   0  -r_tube  start_t+height  constraint tube zmax

edges
1  1  2  constraint tube zmin
2  2  3  constraint tube zmin
3  3  4  constraint tube zmin
4  4  1  constraint tube zmin
5  5  6  constraint tube zmax
6  6  7  constraint tube zmax
7  7  8  constraint tube zmax
8  8  5  constraint tube zmax
9  1  5  constraint tube
10 2  6  constraint tube
11 3  7  constraint tube
12 4  8  constraint tube

faces
1  1  10  -5  -9  tension tens constraint tube
2  2  11  -6  -10  tension tens constraint tube
3  3  12  -7  -11  tension tens constraint tube
4  4  9  -8  -12  tension tens constraint tube

bodies
1  1  2  3  4  density 0

read
ini:={quiet;{r;g100;V 50}4};
adv:={quiet;logfile "tubebend_data.txt";{(z_sph:=z_sph-0.1);recalc;(g;V)1000;
  printf "%g %g %g\n",-z_sph,smc.value,cyl_tens.value)750;logfile off};
```
Data file for the transport of a deformable sphere through a rigid cylinder

```
parameter vol = 8
parameter tens = 1
parameter z_cyl = -9
parameter height = 8
parameter cyl_rad = 0.75

SUPPRESS_WARNING 1055
SUPPRESS_WARNING 1792

quantity smc energy modulus 1 method star_perp_sq_mean_curvature global
quantity sph_tens energy modulus 10 method density_facet_area global
quantity z_cm conserved method vertex_scalar_integral global
scalar_integrand: z/vertex_count

constraint cyl nonpositive
formula: (z<z_cyl+height && z>z_cyl) ? x^2+y^2-(cyl_rad)^2 : -1

vertices
1 1 0 0 constraint cyl
2 0 1 0 constraint cyl
3 -1 0 0 constraint cyl
4 0 -1 0 constraint cyl
5 1 0 1 constraint cyl
6 0 1 1 constraint cyl
7 -1 0 1 constraint cyl
8 0 -1 1 constraint cyl

edges
1 1 2 constraint cyl
2 2 3 constraint cyl
3 3 4 constraint cyl
4 4 1 constraint cyl
5 5 6 constraint cyl
6 6 7 constraint cyl
7 7 8 constraint cyl
8 8 5 constraint cyl
9 1 5 constraint cyl
10 2 6 constraint cyl
11 3 7 constraint cyl
12 4 8 constraint cyl

faces
1 -4 -3 -2 -1 constraint cyl tension tens
2 5 6 7 8 constraint cyl tension tens
3 1 10 -5 -9 constraint cyl tension tens
4 2 11 -6 -10 constraint cyl tension tens
5 3 12 -7 -11 constraint cyl tension tens
6 4 9 -8 -12 constraint cyl tension tens

bodies
1 1 2 3 4 5 6 volume vol density 0
```

read
ini:={r;g200;V 150}3;
adv:={quiet;logfile "spherebend_data.txt";{z_cyl:=z_cyl+0.1};recalc;{g;V}100;
    printf "%g %g %g %g\n",z_cyl+height,z_cm.value,smc.value,sph_tens.value)750;logfile off};
Appendix B

Parametrization of the dendritic spine

As parametrisation of the spine, roughly the same parametrization is used as in [1], with some minor adjustments. The basis is:

\[
\begin{align*}
  x(u,v) &= R \sin(u) \cos(v) \\
  y(u,v) &= R \sin(u) \sin(v) \\
  z(u,v) &= B - \frac{R \cos(u)}{\cos(u)}
\end{align*}
\]  

(B.1)

with $A$ and $B$ positive parameters, related to the spine neck width and length, respectively. $R$ is the maximum radius of the spine, $u$ is the angle between the position vector and the negative $z$-axis ($\frac{\pi}{8} < u < \pi$) and $v$ is the rotational angle around the $z$-axis ($0 < v < 2\pi$) (see figure B.1).

![Schematic representation of the parametrization of the spine.](image)

Figure B.1: Schematic representation of the parametrization of the spine.

Parametrization spine as surface equation

To be able to implement the geometry of the spine in three-dimensional Lattice Boltzmann, the parametrization in equations (B.1) had to be rewritten. For convenience, the parameter $B$ is set to zero. The parametrization in (B.1) can then be rewritten as a surface equation:

\[ x^2 + y^2 + (Au)^2 = R^2. \]  

(B.2)
Then, the angle $u$ had to be rewritten in Carthesian coordinates as follows:

$$ u = -\arctan \left( \frac{\sqrt{x^2 + y^2}}{z} \right). $$

(B.3)

This expression for $u$ suffices, for the spine is cut off at the $x,y$-plane. The rest of the spine is replaced by a cylinder. Combining (B.2) and (B.3), the expression for the surface equation becomes:

$$ x^2 + y^2 + A^2 z^2 \arctan^2 \left( \frac{\sqrt{x^2 + y^2}}{z} \right) - R^2 = 0. $$

(B.4)

Equation (B.4) can then be used (in adjusted form) to create the ‘double spine’ geometry used in the Lattice Boltzmann simulations.