Simulation of stochastic reaction-diffusion processes on lower dimensional manifolds with application in molecular biology

Siyang Wang
Abstract

Simulation of stochastic reaction-diffusion processes on lower dimensional manifolds with application in molecular biology
Siyang Wang

In this thesis, we simulate stochastically the reaction-diffusion processes in a living cell. The simulation is done in three dimension (3D) by MATLAB. The one dimensional (1D) polymers are embedded in the 3D space. The reaction and diffusion events occur both in the space and on the polymers. There is also a possibility for the molecule to jump between the 3D space and 1D polymers. Two simulation levels are used: mesoscopic and microscopic. An accurate and efficient algorithm is developed for the mesoscopic simulation. The corresponding microscopic Smoluchowski equation is solved numerically by a finite difference method in a specific coordinate system adapted to its boundary conditions. The comparison between the result of the mesoscopic simulation and the solution of the microscopic Smoluchowski equation is provided. Good agreement is observed in the experiments.
Acknowledgements

I would like to express my very deep gratitude to Professor Per Lötstedt, my master thesis supervisor, for his valuable time and constructive suggestions during the development of the project. His comments for the report are very much appreciated. Advice given by Stefan Hellander, my second supervisor, is of great help in the modeling and programming part of this thesis. The weekly meetings are important for me to keep the progress on schedule.

I would also like to thank Professor Michael Thuné for being my reviewer of the thesis project and for giving me many valuable suggestions during my master studies.
## Contents

1 Introduction 7

2 Algorithms of stochastic simulation with polymers embedded in a living cell 9
   2.1 Mesoscopic model .............................................. 9
   2.2 Propensity functions ......................................... 11
   2.3 Quotient matrix \( Q \) ........................................ 12
   2.4 Next Voxel Method ........................................... 14
   2.5 Microscopic model ........................................... 14

3 The polymer modeled as a straight line 17
   3.1 Mesoscopic simulation ....................................... 17
   3.2 Microscopical model ......................................... 18
      3.2.1 Microscopic model in space .......................... 18
      3.2.2 Microscopic model on the polymer ................. 22
   3.3 Comparison .................................................... 22
      3.3.1 Experiment 1 ............................................. 23
      3.3.2 Experiment 2 ............................................. 24
      3.3.3 Experiment 3 ............................................. 25

4 The polymer modeled as two parallel line segments 26
   4.1 Mesoscopic simulation ....................................... 27
   4.2 Microscopic model ........................................... 28
   4.3 Comparison .................................................... 32
      4.3.1 Experiment 1 ............................................. 32
      4.3.2 Experiment 2 ............................................. 34

5 An arbitrarily structured polymer embedded in the 3D space 36
   5.1 A polymer is modeled as a circle .......................... 36
      5.1.1 Mesoscopic simulation ................................. 36
      5.1.2 Microscopic model on the circle ...................... 38
      5.1.3 Experiments .............................................. 38
   5.2 A polymer is modeled as a spiral .......................... 39
      5.2.1 Mesoscopic simulation ................................. 40
      5.2.2 Experiments .............................................. 40

6 Roadblocks 42

7 Conclusion 46
1 Introduction

Mathematical modeling of biochemical kinetics has been useful to study the cellular reaction-diffusion processes and many algorithms have been developed for their simulation. Diffusion can be considered as random migration of molecules due to the thermal energy [2]. Reaction is the transformation of one set of chemical substances to another.

Generally there are two fundamental approaches to the mathematical modeling of chemical reaction-diffusion processes: deterministic and stochastic. A deterministic approach is at a macroscopic level. It is governed by the reaction rate equation, which is typically a system of nonlinear ordinary differential equations (ODEs). A crucial assumption is that the local concentration at each point in space equals to the global concentration at all times [10]. In other words, the system is well stirred or spatially homogeneous. However, this assumption does not always hold due to the complex biochemical phenomena in a living cell. Then the spatial variation has to be captured which results in a system of partial differential equations (PDEs).

The deterministic model is appropriate when there are large copy numbers of the species involved in the simulation. However, it is violated in a living cell since some molecular species appear in very small copy numbers. Thus, the stochastic modeling is necessary in order to achieve a better accuracy. In general, there are two distinct levels of stochastic simulation: microscopic level and mesoscopic level. In a microscopic model, each molecule is tracked by an exact position and is free to migrate in a continuous domain by Brownian dynamics (BD). The molecules may react and form a new compound when they are in the vicinity of each other. The positions of the molecules at time $t$ given the positions at time $t_n$ are sampled from a cumulative distribution function (CDF). The corresponding probability density function (PDF) satisfies the Smoluchowski equation. The analytical solutions are known with special initial conditions and boundary conditions. However, infinite summation and a complex integral are involved in the solution so that it is difficult to evaluate. An alternative is to solve the equations numerically using finite difference schemes or finite element methods. Different coordinate systems are used to reduce the computational complexity.

A microscopic model is accurate but computationally expensive, especially for systems with large copy numbers of some species. In addition, sometimes it is not necessary to follow the motion and reactions of single molecules. In this case, it is a waste of time and computational resources to simulate the system microscopically. A mesoscopic model is at a coarser level where the domain of simulation is partitioned into non-overlapping voxels. The trajectories of the molecules are not followed. Only the number of molecules of each species in each voxel is updated. The molecules diffuse from one voxel to its adjacent voxel by a discrete jump. Molecules in the same voxel may react with each other. The system can be considered as a Markov process. The state of the Markov process is given by the number of molecules of each species in the voxels. As a consequence, the probability of a system to be at a state only depends on the previous state. The governing equation at a mesoscopic level is the chemical master equation with the assumption that the system is well stirred. To
capture spatial effects, the size of the voxels is chosen appropriately so that the system can be considered as spatially homogeneous in each voxel. The time evolution of the system is governed by the reaction-diffusion master equation (RDME). The dimension of the domain of the RDME is the product of the number of voxels and the number of species. Due to the requirement of homogeneity in each voxel, a large number of voxels is always necessary which makes the dimension of the RDME extremely high. Thus, it is not possible to obtain the solutions directly. We are forced to use a Monte Carlo approach. There are several well-known algorithms such as Gillespie’s Stochastic Simulation Algorithm (Gillespie’s SSA) [7][8], Next Reaction Method (NRM) [6] and Next Subvolume Method (NSM) [3].

We need to notice that in a mesoscopic model, the molecules in the same voxel are indistinguishable. Thus the mesoscopic model does not represent the real trajectories of the molecules. However, it is easier to implement and preferred when the concentrations are large.

The aim of this project is to simulate the reaction-diffusion processes in a biological cell with one dimensional (1D) polymers. The stochastic modeling of the chemical reaction-diffusion processes is well established. However, a more complicated modeling is needed to represent the internal structure in a cell more precisely. In this project, the polymers inside the biological cell are involved in the reaction-diffusion processes. The reaction events and diffusion events may occur both in space and on the polymers. The molecules may also migrate between the space and the polymers. In this case, the coupling between the polymers and the three dimensional (3D) space has to be modeled. A polymer can be modeled as a straight line for simplicity, a circle or an ellipse, or an arbitrary geometry with curvature.

The remainder of this report is outlined as follows: in Section 2, we develop an algorithm at a mesoscopic level for the stochastic simulation of reaction diffusion processes with polymers involved in the model. Then we test the algorithm with different geometry of the polymers in the model. The geometry of the polymer is a straight line segment in Section 3. In Section 4, the polymer is modeled as two parallel line segments. In Section 5, curvature is involved in modeling the polymer. We test the algorithm with a polymer modeled as a circle and a spiral. The PDF from the microscopic model is used for comparison with the results from the mesoscopic simulation. In Section 6, we introduce roadblocks on the polymer and discuss two related problems. The conclusion is provided in the last section.
2 Algorithms of stochastic simulation with polymers embedded in a living cell

The intracellular structure is very complicated as shown in Figure 1(a). A molecule not only exists in the free space in a cell, but is also attached to polymers. It moves between the free space and the polymers as well. In this case, it is necessary to include the polymers in the model. The simulation domain is a cube as shown in Figure 1(b). Usually a polymer has an irregular shape which is illustrated as the red curve in the figure.

2.1 Mesoscopic model

The computational domain $\Omega$ is a cube with side length $L$ and is partitioned into non-overlapping identical voxels $v_i$ where $i = 1, 2, ..., N = n^3$. The side length of each voxel is given by $h = \frac{L}{n}$. The polymer is denoted by $EF$ with length $L_s$ and is partitioned into segments $sc_j$ with the same length where $j = 1, ..., n_s$. The number of segments is $n_s$ and the length of each segment is denoted by $h_s$. A segment $sc_j$ can be approximated as a line segment sharing the same starting point and ending point. The line segment is denoted by $sl_j$. We assume that the line segments also have the same length. Consequently, the polymer is approximated as a polygonal line. However, the polymer has its thickness (think about DNA in a cell). The cross-section is a circle with radius $\sigma$ which is a small number of magnitude $10^{-9}m$.

A molecule in the 3D space is free to migrate in the form of a discrete jump from one voxel to its adjacent voxel in at most six possible directions in a Cartesian mesh, while a
molecule on the polymer can jump from one segment to its adjacent segment in at most two possible directions [16]. For a molecule in the 3D space, it may jump from the voxel $v_i$ to the curve $sc_j$ if the distance $d_{ij}$ between the molecule and the line segment $sl_j$ is smaller than a constant $r_c(\sigma \ll r_c \ll L)$. This event is called association. Then the molecule begins to diffuse on the polymer. There is also a possibility for the molecule to jump back to one of the voxels in the 3D space. This reversible event is called dissociation.

The polymer $EF$ is considered as a polygonal line which is surrounded by a cylindrical surface with radius $r_c$. If a molecule in the 3D space is also inside the cylindrical surface, it may associate with the polymer. We illustrate part of the polymer model in Figure 2 for explanation. The illustration is in 2D for simplicity. The blue polygonal line models the polymer. The line segments shown in the figure are $sl_{j-1}$, $sl_j$ and $sl_{j+1}$. They are surrounded by the green cylinders of radius $r_c$ to compose the subvolume $sv_{j-1}$, $sv_j$ and $sv_{j+1}$, respectively. $M_j$ is the perpendicular bisector of line segment $sl_j$. We denote the intersection of $M_{j-1}$ and $M_j$ by $O_j$, and the intersection of $sl_{j-1}$ and $sl_j$ by $P_j$. In 3D, $O_jP_j$ defines a plane which is perpendicular to the plane determined by $sl_{j-1}$ and $sl_j$. The function of plane $O_jP_j$ is denoted by $f_{O_jP_j,nor_j}(x, y, z)$ where $nor_j$ is the normal of the plane determined by line segments $sl_{j-1}$ and $sl_j$. The plane $O_{j+1}P_{j+1}$ is not necessarily perpendicular to the plane determined by $M_{j-1}$, $O_j$ and $M_j$. They are bent in 3D. The interface between subvolume $sv_{j-1}$ and $sv_j$ is defined by the intersection of the cylinder and the perpendicular plane $O_jP_j$.

A point on the surface of subvolume $sv_j$ has a distance of $r_c$ to the line segment $sl_j$ assuming that the point is close to neither the interfaces $O_jP_j$ nor $O_{j+1}P_{j+1}$. The value of $r_c$ is determined by the formula

$$r_c = \frac{h}{\sqrt{\pi}}$$  \hspace{1cm} (1)
so that \( h^2 = \pi r_c^2 \). In this way, the cross sectional area of a subvolume is the same as the cross sectional area of a voxel.

Though the jumps between a voxel and a subvolume are diffusion events, they can be characterized by the following association-dissociation reaction

\[
\mathcal{A} + \mathcal{B} \xrightleftharpoons[k_d]{k_a} \mathcal{A} + \mathcal{C},
\]

where \( \mathcal{A} \) is the polymer \( EF \), \( \mathcal{B} \) is the molecule in the voxel in 3D space, \( \mathcal{C} \) is the molecule on \( EF \), \( k_a \) and \( k_d \) are the association rate and dissociation rate, respectively. Molecule \( \mathcal{B} \) and molecule \( \mathcal{C} \) can either be the same species or two different species. Here we denote them separately as they are in different regions. \( \mathcal{A} \) plays the role as an enzyme. Neither the position nor the number of \( \mathcal{A} \) is changed during the simulation. The number of \( \mathcal{A} \) in subvolume \( sv_j \) is given by \( n_A(j) = \frac{h^2}{h} \). The number of \( \mathcal{A} \) is the same in all the subvolumes. Thus, we simply write \( n_A = \frac{h^2}{h} \).

The molecules may react if they are in the same voxel in space or in the same subvolume on the polymer according to the reaction equations. We explain in more detail in the experiments when this kind of reaction occurs in the simulation.

The mesoscopic stochastic simulation of reaction-diffusion processes on a 1D straight line has been analyzed in [4], and in 3D space in [16]. In our model, the coupling between the lower dimensional manifolds and the 3D space is of importance. When an association event occurs, an important issue is to determine which subvolume the molecule diffuses to.

We have claimed that if the distance between a molecule and the polymer is smaller than \( r_c \) then there is a possibility of the molecule to diffuse from the voxel to the subvolume. If a molecule \( \mathcal{B} \) is in a voxel which overlaps with at least one of the subvolumes \( sv_j \), then the molecule may diffuse from the voxel to one of those subvolumes. For a molecule \( \mathcal{C} \) in a subvolume, it may diffuse into the 3D space, and it can only diffuse to one of the voxels that overlap that subvolume. When an association event occurs, the position of the new molecule \( \mathcal{C} \) is determined with the help of propensity functions.

### 2.2 Propensity functions

The propensity function is defined as the function whose product with \( dt \) gives the probability that a particular reaction or diffusion will occur in the next infinitesimal time \( dt \) [9]. We denote the propensity functions by \( \alpha(i,t) \), where \( i = 1, 2, ..., N \) is the index of the voxel and \( t \) is the time. At time \( t \), \( \alpha(i,t)dt \) is the probability that a diffusion event or reaction event occurs in the \( i^{th} \) voxel or in one of the subvolumes overlapping voxel \( v_i \) during the time \( [t, t + dt) \) such that \( v_i \cap sv_j \neq \emptyset \). We assume that all the diffusion events in space have the same rate constant \( d \) which is given by \( d = \frac{D}{h^2} \) where \( D \) is the diffusion constant in space and \( h \) is the side length of the voxel. Similarly, the diffusion rate on the polymer \( EF \) is given by \( d_l = \frac{D_l}{h^2} \) where \( D_l \) is the diffusion constant on \( EF \). The propensity functions for
each voxel and each subvolume are computed by the formulas in [4]. However, in order to apply stochastic simulation algorithms (Gillespie’s SSA [7][8] or NSM [3]), the propensity function for a subvolume must be distributed to its surrounding voxels with the help of the overlapped volume. We denote the overlapped volume between voxel \( i \) and subvolume \( j \) by \( V_{ij} \), where \( i = 1, 2, \ldots, N \) and \( j = 1, 2, \ldots, n_s \). Let \( Q_{ij} = \frac{V_{ij}}{V_j} \) where \( V_j \) is the volume of \( sv_j \). \( V_j \) can be computed analytically by the formula \( V_j = \pi r_j^2 h_s = h^2 h_s \). Recalling the association-dissociation reaction equation (2), we can write the propensity function of the event that molecule \( B \) in voxel \( v_i \) reacts with the polymer \( A \) in subvolume \( sv_j \) as
\[
\alpha_3(i, j, t) = k_{a} n_A n_B(i, t) Q_{ij} = k_{a} n_B(i, t) V_{ij}/h^3,
\]
the propensity function of the event that the molecule \( C \) in subvolume \( sv_j \) to dissociate from the polymer into the voxel \( v_i \) is
\[
\alpha_1(i, j, t) = K_d n_A n_C(j, t) Q_{ij} = k_{d} n_C(j, t) V_{ij}/h^3.
\]
Consequently, the propensity function of a reaction event that occurs in voxel \( v_i \) is given by
\[
\alpha_r(i, t) = \alpha_{assoc}(i, t) + \alpha_{dissoc}(i, t)
= \sum_{j=1}^{n_s} \alpha_3(i, j, t) + \sum_{j=1}^{n_s} \alpha_1(i, j, t),
\]
and the propensity function of a diffusion event that occurs in voxel \( i \) is given by
\[
\alpha_d(i, t) = \alpha_B(i, t) + \alpha_C(i, t)
= 6d n_B(i, t) + 2d_l \sum_{j=1}^{n_s} n_C(j, t) Q_{ij},
\]
where \( n_B(i, t) \) and \( n_C(j, t) \) are the number of molecules \( B \) in voxel \( v_i \) at time \( t \) and the number of molecules \( C \) in subvolume \( sv_j \) at time \( t \), respectively. The total propensity function of voxel \( v_i \) at time \( t \) is
\[
\alpha(i, t) = \alpha_r(i, t) + \alpha_d(i, t).
\]
The propensity function is of importance in the stochastic simulation algorithms. We use it to sample the time to the next event occurs. Thus, the quotient matrix \( Q \) has to be computed accurately and efficiently.

### 2.3 Quotient matrix \( Q \)

\( Q_{ij} \) can be stored in a \( N \)-by-\( n_s \) matrix \( Q \) with entries \( Q(i, j) = Q_{ij} = \frac{V_{ij}}{V_j} \). Figure 2 shows that most of the overlaps are of irregular shape which makes it difficult to compute them analytically. Instead, we employ the Monte Carlo method. The domain \( \Omega \) is defined in the
Cartesian coordinate system so that one of the vertices of the mesh coincides with the origin and \( \Omega \) lies in the nonnegative octant. We generate many random points (total number: \( T \)) within \( \Omega \) and count the number of points that lie in the overlapped volume between voxel \( i \) and subvolume \( j \). This number is denoted by \( Rm(i,j) \). The element \( Q(i,j) \) of the matrix \( Q \) is computed with the help of \( Rm(i,j) \). The procedure of computing the quotient matrix \( Q \) is summarized in Algorithm 1.

**Algorithm 1** Computing the quotient matrix \( Q \)

1. **Initialization:** \( Rm = \text{zeros}(N, n_s) \), time \( t = 0 \), the number of random points \( N_r \).

2. Generate three random numbers \( r_1, r_2 \) and \( r_3 \) which are uniformly distributed between 0 and \( L \). Thus, \( G = (r_1, r_2, r_3) \) is a random point inside \( \Omega \). Set \( k = 1 \).

3. If \( k \leq n_s \), compute the distance \( r_{Gk} \) between the point \( G \) and line segment \( sl_k \); otherwise update \( t := t + 1 \) and go back to step (2).

4. If \( r_{Gk} > r_c \), the point \( G \) is outside the subvolume \( sv_k \). Update \( k := k + 1 \) and go back to step (3). Otherwise, the point \( G \) may be inside the subvolume \( sv_k \). Go to step (5).

5. Compute the function \( f_{O_j,P_j,nor_j}(x,y,z) \) and \( f_{O_{j+1},P_{j+1},nor_{j+1}}(x,y,z) \) of the interface \( O_jP_j \) and \( O_{j+1}P_{j+1} \), respectively. Point \( G \) is inside subvolume \( sv_k \) if \( f_{O_j,P_j,nor_j}(r_1,r_2,r_3) \cdot f_{O_{j+1},P_{j+1},nor_{j+1}}(r_1,r_2,r_3) < 0 \), go to step (6); otherwise update \( k := k + 1 \) and go back to step (3).

6. Determine the index of the voxel \( i \) where the point \( G \) is, with the help of its coordinate.

7. Update \( Rm(i,k) := Rm(i,k) + 1 \), \( t := t + 1 \). Go back to (2) if \( t \leq N_r \).

8. Compute \( Q(i,j) = \frac{V_i}{V_j} = \frac{Rm(i,j)}{\sum_{k=1}^{N_s} Rm(i,k)} \) for \( i = 1, 2, ..., N \) and \( j = 1, 2, ..., n_s \).

By the central limit theorem, Monte Carlo method has a low convergence rate \( \frac{1}{\sqrt{N_r}} \) where \( N_r \) is the number of sample points. A large number of iterations in the algorithm is required to obtain an accurate result. In order to propose an appropriate value of \( N_r \), we make a simple experiment. We use the Monte Carlo method to compute the volume of a ball with radius 1 and compare the relative error with respect to the number of random points in the simulation. The result is shown in Table 1. The result obtained with \( 10^7 \) random points is satisfactory so that \( N_r = 10^7 \) is used in all the simulations in this thesis. Though Monte Carlo method is not computationally efficient, for a certain model we only compute the quotient matrix \( Q \) once and save the data. The data is loaded from its path when running the simulation. \( r_c \) is small compared with \( L \). Thus, most elements of matrix \( Q \) are zeros which makes it sparse. We can take advantage of it to achieve an efficient implementation.
Table 1: Relative error of computing the volume of the ball with respect to the random points used in the simulation

<table>
<thead>
<tr>
<th>$N_r$</th>
<th>Analytical volume</th>
<th>Volume by Monte Carlo Method</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^3$</td>
<td>4.1888</td>
<td>4.3040</td>
<td>2.75%</td>
</tr>
<tr>
<td>$10^4$</td>
<td>4.1888</td>
<td>4.1744</td>
<td>-0.34%</td>
</tr>
<tr>
<td>$10^5$</td>
<td>4.1888</td>
<td>4.1788</td>
<td>-0.24%</td>
</tr>
<tr>
<td>$10^6$</td>
<td>4.1888</td>
<td>4.1917</td>
<td>0.07%</td>
</tr>
<tr>
<td>$10^7$</td>
<td>4.1888</td>
<td>4.1894</td>
<td>0.01%</td>
</tr>
</tbody>
</table>

In step (5), the functions for the interface between two subvolumes are needed to compute the index of the subvolume where the random point lies. We can compute the functions for all the interfaces and generate a look-up-table before applying Algorithm 1. It helps us to achieve a more computational efficient implementation.

2.4 Next Voxel Method

Gillespie’s SSA has been widely used to simulate the mesoscopic kinetics in spatially homogeneous systems. However, the direct use of Gillespie’s SSA is too slow for simulation in a spatial inhomogeneous system. Elf, Dončić and Ehrenberg have developed NSM [3] to speed up Gillespie’s SSA. NSM takes advantage of the fact that reaction intensities and diffusion intensities only change in the voxels which are involved in the reaction event or diffusion event. Thus, it is sufficient to only update the reaction intensities and diffusion intensities for those particular voxels.

However, NSM is an algorithm suitable for the simulations in a free space. In order to use it in our model with lower dimensional manifolds, changes are needed. The propensities of reaction and diffusion exist both in the 3D space and the lower dimensional manifolds. A feasible approach to be compatible with NSM is to distribute the propensities from the lower dimensional manifolds to the 3D space. The procedure is summarized in Algorithm 2.

Algorithm 2 is tested by different structures of the polymers in the following sections. The results are compared with the ones obtained from the microscopic model.

2.5 Microscopic model

The diffusion process is governed by Smoluchowski equation

$$\frac{\partial \phi}{\partial t} = D \Delta \phi,$$

where $D$ is the diffusion constant and $\phi := \phi(r, t)$ is the PDF of finding a molecule at position $r$ at time $t$. Thus, the governing equation for the diffusion in the 3D space in the Cartesian
Algorithm 2 Next Voxel Method (NVM)

(1) Initialize the number of molecules in the voxels and subvolumes. Set \( t = 0 \).

(2) Compute the reaction propensity function \( \alpha_r(i,t) \), diffusion propensity function \( \alpha_d(i,t) \) and total propensity function \( \alpha(i,t) \) for each voxel using equation (5), (6) and (7), respectively.

(3) For voxel \( i \), generate a random number \( r_i \) which is uniformly distributed in \([0,1]\). Then compute the time to the first reaction or diffusion event occurs in voxel \( i \) using the formula \( t_i = \frac{1}{\alpha_i} \ln(\frac{1}{r_i}) \).

(4) Find the smallest value among \( t_1, t_2, ..., t_N \) and denote it by \( t_\lambda \). Then the next event occurs in voxel \( \lambda \) at time \( t := t + t_\lambda \).

(5) Generate a random number \( r_a \) which is uniformly distributed in \([0,1]\). The event will be a chemical reaction if \( r_a < \alpha_r(\lambda,t)/\alpha(\lambda,t) \), otherwise a diffusion event occurs.

(6) If a chemical reaction event occurs:
   
   (a) Generate a random number \( r_b \) which is uniformly distributed in \([0,1]\). It will be an association event if \( \alpha_{assoc}(\lambda,t)/\alpha_r(\lambda,t) \), otherwise it will be a dissociation event.
   
   (b) Determine the index of subvolume \( \lambda' \) which is involved in the reaction event with the help of a newly generated random number \( r_c \) uniformly distributed in \([0,1]\). \( \lambda' := \min(\lambda) \) such that \( \sum_{j=1}^{\lambda'} Q(\lambda,j) \geq r_c \).
   
   (c) Update the number of molecules accordingly.
   
   (d) Recalculate the propensity functions and the time to the next event for the voxels that overlap subvolume \( \lambda' \).

(7) If a diffusion event occurs:
   
   (a) Generate a random number \( r_d \) which is uniformly distributed in \([0,1]\). Molecule \( B \) diffuses away if \( r_d < \alpha_B(\lambda,t)/\alpha_d(\lambda,t) \), otherwise molecule \( C \) diffuses away.
   
   (b) If molecule \( B \) diffuses away:
      
      (i) Randomly choose one of the six directions and compute the index of the voxel \( \lambda_b \) that the molecule diffuses to.
      
      (ii) Update the number of molecules accordingly.
      
      (iii) Recalculate the propensity functions and the time to the next event for subvolume \( \lambda \) and \( \lambda_b \).
   
   (c) If molecule \( C \) diffuses away:
      
      (i) Generate a random number \( r_e \) which is uniformly distributed in \([0,1]\) to determine the index of the voxel \( \lambda' \) which the molecule diffuses from. \( \lambda' := \min(\lambda) \) such that \( \sum_{j=1}^{\lambda'} Q(\lambda,j) \geq r_e \).
      
      (ii) Randomly choose one of the two directions to determine the index of the subvolume \( \hat{\lambda} \) that the molecule diffuses to.
      
      (iii) Update the number of molecules accordingly.
      
      (iv) Recalculate the propensity functions and the time to the next event for voxels which are overlapped with subvolume \( \lambda' \) and \( \hat{\lambda} \).

(8) Find the smallest value among \( t_1, t_2, ..., t_N \) and denote it by \( t_\lambda \). Then the next event occurs in voxel \( \lambda \) at time \( t := t + t_\lambda \). If \( t \leq T \) go back to step (5), otherwise stop.
Figure 3: A 1D polymer is embedded in the 3D space. The boundary conditions are posed in a locally defined coordinate system near the polymer.

The governing equation for the diffusion on a 1D line segment is

\[
\frac{\partial \phi}{\partial t} = D_l \frac{\partial^2 \phi}{\partial s^2},
\]

where \( D_l \) is the diffusion constant on the 1D line segment and \( \phi(s, t) \) is the PDF of finding a molecule at position \( s \) at time \( t \).

The reaction process is characterized by the boundary conditions of the governing PDEs. For the model with 1D polymer embedded in the 3D space, the boundary condition is

\[
2\pi \sigma D \frac{\partial \phi}{\partial r} \bigg|_{r=\sigma} = k_a \phi(r = \sigma, t| r_0, t_0) - k_d \phi(*, t| r_0, t_0),
\]

where \( \sigma \) is the radius of the polymer (much smaller than \( r_c \)), \( D \) is the diffusion constant, \( k_a \) is the association rate, \( k_d \) is the dissociation rate and \( \phi(r, t) \) is the PDF of finding a molecule at position \( r \) at time \( t \). In equation (11), \( r \) is defined locally in the coordinates as shown in Figure 3, where \( r \) is perpendicular to \( s \) and \( s \) is the tangent line to the dash curve which models the polymer. When solving the equation, we rewrite the PDE and boundary conditions in a specific coordinate system according to the structure of the polymer in order to reduce the complexity of the problem. We explain in more detail in the following experiments.
3 The polymer modeled as a straight line

In this section, we simulate the reaction-diffusion processes in a living cell with a polymer embedded in the 3D space. The polymer is modeled as a straight line. We simulate at a mesoscopic level as well as a microscopic level. The comparison between the results from these two approaches is provided.

3.1 Mesoscopic simulation

The mesoscopic model is shown is Figure 4(a). The simulation domain $\Omega$ is discretized into identical voxels and the line segment $EF$ is discretized into identical subsegments which are surrounded by a cylindrical surface (red). We employ Algorithm 2 to simulate the reaction-diffusion processes. For computing the quotient matrix $Q$, we take advantage of the special geometry of the polymer in this model. The polymer is modeled as a line segment $EF$ so that it has no curvature. By discretizing $EF$ into identical subsegments, the subvolumes in the interior are identical cylinders with radius $r_c$ and height $h_s = \frac{L_s}{n_s}$ where $L_s$ is the length of $EF$ and $n_s$ is the number of subsegments. An algorithm for computing the quotient matrix $Q$ explicitly for the model in this section is summarized in Algorithm 3.

This is a simplified version of Algorithm 1 and is more computationally efficient. In step (5), the index of voxel $i$ is computed with the help of the coordinates of the grid points. In step (6), the index of subvolume $j$ is more difficult to compute. The approach is to first compute the function of the plane $P$ which is perpendicular to line $EF$ and passes through the point $E$ (or $F$), and then determine $j$ with the help of the distance between the random point $G$ and plane $P$. The volume of the subvolumes can be computed either analytically by $V_j = h^2 h_s$ or numerically as in step (8). In the simulation, we compute it numerically to
Algorithm 3 Computing the quotient matrix $Q$ for the polymer modeled as a line segment

(1) Initialization: $R_m = \text{zeros}(N, n_s), t = 0, N_r = 10^7$.

(2) Generate three random numbers $r_1, r_2$ and $r_3$ which are uniformly distributed between 0 and $L$. Thus, $G = (r_1, r_2, r_3)$ is a ‘random point’ inside $\Omega$.

(3) Calculate the distance $r_G$ between the point $G$ and the line $EF$.

(4) If $r_G > r_c$, the point $G$ is outside the cylinder. Update $t := t + 1$ and go back to (2). Otherwise, the point $G$ is inside the cylinder.

(5) Determine the index of the voxel where the point $G$ is. The index is denoted by $i$.

(6) Determine the index of the subvolume where the point $G$ is. The index is denoted by $j$.

(7) Update $R_m(i, j) = R_m(i, j) + 1, t := t + 1$. Go to (1) if $t \leq N_r$.

(8) Compute $Q(i, j) = \frac{V_i}{V_j} = \frac{R_m(i, j)}{\sum_{i=1}^{N_r} R_m(i, j)}$ for $i = 1, 2, ..., N$ and $j = 1, 2, ..., 2n_s$.

achieve a purely numerical algorithm.

There are two ways to examine the result of Algorithm 3:

1. We plot all the points that are inside the cylinder, as described in the fourth step in Algorithm 3. We should observe the shape of the cylinder if the total number of interactions $N_r$ is large. In addition, there should not be any points outside the cylinder ($N_r = 10^5$ is enough to observe the shape, we use $N_r = 10^7$ for a better accuracy.).

2. The volume of each subvolume in space can be computed analytically which can be compared with $\sum_{i=1}^{n^3} V(i, j)$.

In Figure 4(b), we show the illustration of random points for computing the quotient matrix $Q$ of the line segment $EF$ which is the diagonal of the simulation domain $\Omega$. All the random points form a geometry of a cylinder which is exactly what we expect from the algorithm. There is no point outside the cylinder which demonstrates that the algorithm works fine for selecting the random points.

3.2 Microscopical model

3.2.1 Microscopic model in space

In a microscopic model, diffusions and reactions are treated in a continuous domain. The molecules are spherical with a certain radius. They diffuse by Brownian motion with the
diffusion coefficient $D$.

Fick’s second law [19] predicts how the concentration of molecules changes with respect to time:

$$\frac{\partial \phi}{\partial t} = D \nabla^2 \phi,$$

where $\phi$ is the concentration and $D$ is the diffusion coefficient. It is analogous to the heat equation. The boundary conditions explain the biological phenomenon on the surface of the cylinder of radius $\sigma \ll r_c$. If we apply a finite difference method in the Cartesian coordinate system, it would be difficult to discretize the space near the boundary, i.e. the surface of the cylinder. Instead we take advantage of the spherical coordinate system. In this model, we are interested in the distance between the molecule and the polymer, but not the exact coordinate of the molecule. Thus, we can place the cylinder along the $z$-axis as shown in Figure 5. Then the PDF of finding a molecule at a radial position $r = \sqrt{x^2 + y^2}$ is reduced to the solution of a 1D problem. The PDF $p(r,t|r_0,t_0)$ of finding a molecule at position $r$ at time $t$, given that the molecule is at position $r_0$ at time $t_0$ satisfies the Smoluchowski equation

$$\frac{\partial p}{\partial t} = D \left( \frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \frac{\partial p}{\partial r} \right),$$

with initial condition

$$p(r,t_0|r_0,t_0) = \delta(r - r_0)$$

and boundary conditions

$$\lim_{r \to \infty} p(r,t|r_0,t_0) = 0,$$

$$2\pi \sigma D \frac{\partial p}{\partial r} |_{r=\sigma} = k_d p(r = \sigma, t|r_0,t_0) - k_a p(\ast, t|r_0,t_0).$$
The initial condition (14) tells that at time \( t_0 \) the molecule starts at a point which has a distance \( r_0 \) to the line. The probability of the molecule being infinitely far away from the line is 0 as stated in (15). The boundary condition (16) explains the phenomenon when the molecule attaches the line, where \( \sigma \) is the radius of the polymer. \( k_a \) and \( k_d \) are the association rate and dissociation rate, respectively. \( p(\cdot, t|r_0, t_0) \) is the probability that the molecule is on the polymer at time \( t \) with the condition that the distance between the molecule and the polymer is \( r_0 \) at time \( t_0 \) and is computed by the formula

\[
p(\cdot, t|r_0, t_0) = 1 - \int_{r}^{\infty} 4\pi r^2 p(r, t|r_0, t_0) dr.
\]  

(17)

In Bani-Hashemian’s master thesis [1], a similar PDE has been solved numerically by Crank-Nicolson method. We use the same method but explain it in more detail.

We use the uniform grid in time. Thus, the time interval \([0, T]\) is discretized into \( M \) identical subintervals with time step \( \Delta t = \frac{T}{M} \). It is more complicated to discretize in space since \( r_0 \) has to be one of the grid points and there is no upper bound for \( r \). Firstly, we discretize the interval \([0, r_0]\) into \( N_1 \) identical subintervals with \( \Delta r = \frac{r_0}{N_1} \). Secondly, we set \( r_{\text{max}} = r_0 + 4\sqrt{2DT} \) [1] then discretize the interval \([r_0, r_{\text{max}}]\) into identical subintervals with length \( \Delta r \). If \( r_{\text{max}} - r_0 \) is not divisible by \( \Delta r \), \( r_{\text{max}} \) is increased in order to obtain a uniform grid. We denote the number of subintervals in \([r_0, r_{\text{max}}]\) by \( N_2 := \lceil \frac{r_{\text{max}}-r_0}{\Delta r} \rceil \). Thus, the total number of subintervals in space is given by \( N = N_1 + N_2 \).

We apply the Crank-Nicolson method to the derivatives in equation (13).

\[
\frac{\partial p}{\partial t}(r_i, t_{n+1/2}) = \frac{p_{i}^{n+1} - p_{i}^{n}}{\Delta t},
\]

(18)

\[
\frac{\partial p}{\partial r}(r_i, t_{n+1/2}) = \frac{1}{2} \left( \frac{p_{i+1}^{n+1} - p_{i}^{n+1}}{2\Delta r} + \frac{p_{i}^{n+1} - p_{i-1}^{n+1}}{2\Delta r} \right),
\]

(19)

\[
\frac{\partial^2 p}{\partial r^2}(r_i, t_{n+1/2}) = \frac{1}{2(\Delta r)^2} ((p_{i+1}^{n+1} - 2p_{i}^{n+1} + p_{i-1}^{n+1}) + (p_{i+1}^{n+1} - 2p_{i}^{n+1} + p_{i-1}^{n+1}) + (p_{i+1}^{n+1} - 2p_{i}^{n+1} + p_{i-1}^{n+1})).
\]

(20)

Then equation (13) can be written as

\[
\frac{p_{i}^{n+1} - p_{i}^{n}}{\Delta t} = \frac{D}{2(\Delta r)^2} ((p_{i+1}^{n+1} - 2p_{i}^{n+1} + p_{i-1}^{n+1}) + (p_{i+1}^{n+1} - 2p_{i}^{n+1} + p_{i-1}^{n+1}) + \frac{D}{4r_i \Delta r} (p_{i+1}^{n+1} - p_{i}^{n+1} + p_{i+1}^{n+1} - p_{i-1}^{n+1}),
\]

(21)

where \( i = 2, 3, ..., N - 1 \) and \( n = 1, 2, ..., M - 1 \). After rearranging the terms by putting the value of \( p \) at time step \( n \) on the right hand side and the value of \( p \) at time step \( n+1 \) on the left hand side, we get:

\[
LHS(n+1) = p_{i-1}^{n+1} \left( \frac{D \Delta t}{2(\Delta r)^2} - \frac{D \Delta t}{4r_i \Delta r} \right) + p_{i}^{n+1} \left( -1 - \frac{D \Delta t}{(\Delta r)^2} \right) + p_{i+1}^{n+1} \left( \frac{D \Delta t}{2(\Delta r)^2} + \frac{D \Delta t}{4r_i \Delta r} \right)
\]

(22)
\[ RHS(n) = p_{n-1}^{n} \left( -\frac{D\Delta t}{2(\Delta r)^2} + \frac{D\Delta t}{4r_i\Delta r} \right) + p_{n}^{n} \left( -1 + \frac{D\Delta t}{(\Delta r)^2} \right) + p_{n+1}^{n} \left( -\frac{D\Delta t}{2(\Delta r)^2} - \frac{D\Delta t}{4r_i\Delta r} \right), \]

where \( LHS(n+1) \equiv RHS(n) \). Let

\[ \begin{align*}
\alpha_i &= \frac{D\Delta t}{2(\Delta r)^2} - \frac{D\Delta t}{4r_i\Delta r} \\
\beta_1 &= -1 - \frac{D\Delta t}{(\Delta r)^2} \\
\gamma_i &= \frac{D\Delta t}{2(\Delta r)^2} + \frac{D\Delta t}{4r_i\Delta r} \\
A_i &= -\frac{D\Delta t}{2(\Delta r)^2} + \frac{D\Delta t}{4r_i\Delta r} \\
\beta_2 &= -1 + \frac{D\Delta t}{(\Delta r)^2} \\
\Gamma_i &= -\frac{D\Delta t}{2(\Delta r)^2} - \frac{D\Delta t}{4r_i\Delta r}.
\end{align*} \]

Then equation (21) can be written as

\[ \alpha_i p_{i-1}^{n+1} + \beta_1 p_i^{n+1} + \gamma_i p_{i+1}^{n+1} = A_i p_{i-1}^{n} + \beta_2 p_i^{n} + \Gamma_i p_{i+1}^{n}. \]

Equation (25) can be expressed as

\[ \begin{align*}
i &= 2 & \alpha_2 p_1^{n+1} + \beta_1 p_2^{n+1} + \gamma_2 p_3^{n+1} &= A_2 p_1^{n} + \beta_2 p_2^{n} + \Gamma_2 p_3^{n} \\
i &= 3 & \alpha_3 p_2^{n+1} + \beta_1 p_3^{n+1} + \gamma_3 p_4^{n+1} &= A_3 p_2^{n} + \beta_2 p_3^{n} + \Gamma_3 p_4^{n} \\
& \vdots & \vdots & \vdots & \vdots \\
i &= N - 1 & \alpha_{N-1} p_{N-2}^{n+1} + \beta_1 p_{N-1}^{n+1} + \gamma_{N-1} p_N^{n+1} &= A_{N-1} p_{N-2}^{n} + \beta_2 p_{N-1}^{n} + \Gamma_{N-1} p_N^{n}.
\end{align*} \]

Now we can rewrite (26) in the form of \( Ap^{n+1} + b_1^{n+1} = Bp^{n} + b_2^{n} \) where \( b_1^{n+1} \) and \( b_2^{n} \) are in terms of \( p_1^{n+1} \) and \( p_2^{n} \), respectively. It is difficult to solve the system since the left hand side of the equation can not easily be written as a product of a matrix and a vector. Thus we make an approximation by using the value of \( p \) at time step \( n \) to compute \( b_1^{n+1} \). Then we have \( Ap^{n+1} = Bp^{n} + b_2^{n} \) where

\[ A = \begin{bmatrix}
\beta_1 & \gamma_2 \\
\alpha_3 & \beta_1 & \gamma_3 \\
\alpha_4 & \beta_1 & \gamma_4 \\
\vdots & \ddots & \ddots \\
\alpha_{N-2} & \beta_1 & \gamma_{N-2} \\
\alpha_{N-1} & \beta_1 & \gamma_{N-1}
\end{bmatrix}, \quad p^{n+1} = \begin{bmatrix}
p_2^{n+1} \\
p_3^{n+1} \\
\vdots \\
p_N^{n+1}
\end{bmatrix}, \quad b_1^{n} = \begin{bmatrix}
\alpha_2 \ast p_1^{n} \\
0 \\
\vdots \\
0
\end{bmatrix}, \]

and

\[ B = \begin{bmatrix}
\beta_2 & \Gamma_2 \\
A_3 & \beta_2 & \Gamma_3 \\
A_4 & \beta_2 & \Gamma_4 \\
\vdots & \ddots & \ddots \\
A_{N-2} & \beta_2 & \Gamma_{N-2} \\
A_{N-1} & \beta_2 & \gamma_{N-1}
\end{bmatrix}, \quad p^{n} = \begin{bmatrix}
p_2^{n} \\
p_3^{n} \\
\vdots \\
p_{N-1}^{n}
\end{bmatrix}, \quad b_2^{n} = \begin{bmatrix}
\alpha_2 \ast p_1^{n} \\
0 \\
\vdots \\
0
\end{bmatrix}. \]
A is a tridiagonal matrix so that we can take advantage of its special structure to solve the system. We use backward substitution instead of computing the inverse of the matrix. The vector $p^{n+1}$ contains all the values of $p$ at time step $n+1$ on the interior points, but not on the boundary. We compute $p_1^{n+1}$ and $p_N^{n+1}$ from the boundary conditions. Equation (16) can be discretized as

$$
2\pi\sigma D \frac{p_2^{n+1} - p_1^{n+1}}{2\Delta r} = k_ap_1^{n+1} - k_d \left( 1 - \int_{\sigma}^{\infty} 2\pi rp(r|r_0,t_0)dr \right). \tag{29}
$$

The integral in equation (29) is computed numerically by trapezoidal rule. After rearranging the terms we get

$$
p_1^{n+1} = \frac{\pi\sigma D}{k_a\Delta r + \pi\sigma D} p_2^{n+1} + \frac{k_d\Delta r}{k_a\Delta r + \pi\sigma D} \left( 1 - \int_{\sigma}^{\infty} 2\pi rp(r|r_0,t_0)dr \right). \tag{30}
$$

Equation (15) gives

$$
p_i^j = 0, \tag{31}
$$

where $i = 1, 2, ..., M$. Now we have obtained all the values of $p$ at time step $n+1$. Then the time is advanced for one time step successively until the end of the simulation.

### 3.2.2 Microscopic model on the polymer

For the simulation of 1D diffusion process, the probability $\phi$ of finding a molecule at position $x$ and time $t$ is given by [16]

$$
\phi(x,t) = \frac{1}{l} + \sum_{n=1}^{\infty} \frac{2}{l} \cos(kn\pi) \cos \frac{n\pi x}{l} e^{-D\left(\frac{\pi^2}{n^2}\right)^2 t}, \tag{32}
$$

where $l$ is the length of the simulation domain, $k$ is the initial position and $D$ is the diffusion constant. If the molecule is placed at the point $x_s$ in space initially, the corresponding initial position $k$ on $EF$ is the one which has the shortest distance to $x_s$.

### 3.3 Comparison

In this section, we compare the results obtained from the mesoscopic simulation and the solution of the microscopic Smoluchowski equation. We aim at using the same parameters in both models. It has been shown that in the mesoscopic model, as the side length $h$ of the voxels decreases, all the association and dissociation reactions are eventually lost [13]. Thus, the result is far from the one obtained in the microscopic model. In order to make a comparison, the reaction rates in the mesoscopic simulation must be scaled. Two different formulas have been developed in [5] and [12]. The formula in [12] is

$$
k_{meso} = \frac{(L/h)^2}{\tau_{micro} - \left[ \frac{L^2}{2\pi D} \log\left( \frac{L}{h} \right) + \frac{0.1951L^2}{4D} \right]}, \tag{33}
$$

22
Table 2: Computing $k_{meso}$

<table>
<thead>
<tr>
<th>$k_{micro}$</th>
<th>$k_{meso}$ in [12]</th>
<th>$k_{meso}$ in [5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5e-13</td>
<td>17.5155</td>
<td>16.4352</td>
</tr>
<tr>
<td>5e-11</td>
<td>79.7298</td>
<td>61.3464</td>
</tr>
<tr>
<td>5e-9</td>
<td>82.6660</td>
<td>63.0693</td>
</tr>
<tr>
<td>5e-7</td>
<td>82.6965</td>
<td>63.0870</td>
</tr>
<tr>
<td>5e-5</td>
<td>82.6968</td>
<td>63.0872</td>
</tr>
</tbody>
</table>

where $\tau_{micro} \approx [1 + \alpha F(\lambda)]L^2/k_{micro}$, $\alpha = k_{micro} \frac{2\sqrt{2}}{\pi D}$ and $F(\lambda) = [4\log(1/\lambda) - (1 - \lambda^2)(3 - \lambda^2)]/(4(1 - \lambda^2)^2)$, while the formula in [5] is

$$k_{meso} = \frac{k_{micro}}{1 + \alpha \log[1 + 0.544(1 - \beta)/\beta]} \cdot \frac{1}{h^2 - \pi \sigma^2},$$

(34)

where $\alpha = \frac{k_{micro}}{2\pi D}$ and $\beta = \frac{\sigma}{\sigma + h}$. When the reaction rate is small in the microscopic model, these two formulas give two results which are close to each other. As the reaction rate in the microscopic model increases, the reaction rate in the mesoscopic model reaches its limit. However, the formulas give two different limits which are shown in Table 2 with the parameters $L = 3 \cdot 10^{-6}$, $\sigma = 1.005 \cdot 10^{-9}$ and $D = 10^{-12}$. We do not go into the detail of the derivation of the formulas. Our experiments have shown that the formula in [12] gives a better convergence with the results in the microscopic model. We use the formula in [12] in the whole thesis when it is needed.

The parameters used in all the experiments in this section are $L$, $\sigma$, $D$ as above, $E = (0, 0, 0)$, $F = (L, L, L)$, $n = 20$, $n_s = 35$ and $T = 0.1$.

3.3.1 Experiment 1

At time $t = 0$ in the mesoscopic simulation, 10000 molecules $B$ are placed in the $4210^{th}$ voxel in space. The coordinate of the midpoint of the $4210^{th}$ voxel is $(0.525L, 0.525L, 0.525L)$. Equivalently in the microscopic model, 10000 molecules $B$ are placed at a distance $r_0 = 1.2247 \cdot 10^{-7}$ to $EF$. In this case, $r_0 \approx 0.8h$. The association rate in the microscopic model is $k_{a,\text{micro}} = 5 \cdot 10^{-13}$ and in the mesoscopic simulation is $k_{a,\text{meso}} = 17.5115$ by the formula (33). The same rate is used for the association event and dissociation event. Thus, the dissociation rate in the microscopic model is given by $k_{d,\text{micro}} = \frac{k_{a,\text{micro}}}{h^2} = 22.2222$ and in the mesoscopic simulation by $k_{d,\text{meso}} = k_{a,\text{meso}} = 17.5115$. We simulate up to 0.1s in both levels of approximation. The method of solving the Smoluchowski equation is deterministic so we solve it once. However, more simulations are needed in the mesoscopic model in order to get an accurate and convergent result. Here we take 100 identical simulations and compute the average values at the end. Then we plot the distribution of molecules in Figure 6.
In Figure 6(a), the histogram shows the mesoscopic number of molecules at a distance $r$ to $EF$ at time $t = 0.1$. The number has been scaled in order to compare with the red curve which is the solution of the Smoluchowski equation. Very good agreement is observed. In Figure 6(b), the histogram shows the number (after scaling) of molecule $C$ in each subvolume. The red curve is the analytical solution of the diffusion equation computed by the formula (32). The results obtained from the mesoscopic simulation and the solution of the Smoluchowski equation in the microscopic model converge perfectly. In addition, we can also compare the number of $C$ in the end of the simulation. The number of $C$ in the mesoscopic model at time $t = 0.1$ is $n_{C,meso} = 284.88$. The number is not an integer because it is the average value of 100 identical simulations. In the microscopic model, the CDF of finding molecule $B$ at position $r$ reaches 0.9715 on the boundary which gives the probability of finding a molecule on $EF$ is $1 - 0.9715 = 0.0285$. Since there are 10000 molecule $B$ in space at time $t = 0$, it predicts that the number of molecule $C$ at the end of the simulation is $0.0285 \times 10000 = 285$. The relative error of the number of molecule $C$ in the mesoscopic simulation compared with the number of molecule $C$ in the microscopic model is $\frac{|284.88 - 285|}{285} \approx 0.04\%$ which is quite satisfactory.

The formula in [5] gives $k_{a,meso} = k_{d,meso} = 16.4352$ as shown in Table 2. If we use those values in the mesoscopic simulation with other parameters the same as the simulation above, the number of molecule $C$ is 284.65 which is also quite close to the number predicted by the microscopic model.

### 3.3.2 Experiment 2

In this experiment, we still use 10000 molecules $B$ in the beginning, but with a larger $r_0$ as a starting position. In the mesoscopic simulation, 10000 molecules $B$ are placed in the 4215th voxel in space. The coordinate of the midpoint of the 4215th voxel is $(0.725L, 0.525L, 0.525L)$. The equivalent initial position in the microscopic model is $r_0 = 4.8990 \cdot 10^{-7}$. In this case,
Figure 7: (a) Distribution of molecule $B$ (b) Distribution of molecule $C$

$r_0 \approx 3.3h$. The other parameters are the same as in Experiment 1. The results are displayed in Figure 7.

Figure 7(a) shows the distribution of molecule $B$ in the mesoscopic simulation and the solution of the corresponding Smoluchowski equation. Very good agreement is observed again. The number of molecule $C$ predicted by the microscopic model is 74. In the mesoscopic simulation, the number of $C$ is 74.35. However, if we use the formula in [5] to compute $k_{a, \text{meso}}$ and $k_{d, \text{meso}}$, the number of $C$ is 71.63. It is still an acceptable result but the relative error becomes larger. Thus, the formula in [12] is preferred in our experiments.

3.3.3 Experiment 3

In this experiment, we initialize the molecules in a special position. In the mesoscopic model, at time $t = 0$ 10000 molecules $B$ are placed in the 4211th voxel in space. Line segment $EF$ goes through the midpoint of the 4211th voxel. Thus, the initial position of the molecule in the microscopic model is $r_0 = 0$ which is an invalid value. We need to give an appropriate value for $r_0$ in the microscopic model if the line segment $EF$ goes through the subvolume where molecule $B$ is initialized in the mesoscopic model.

The smallest value of $r_0$ in the mesoscopic model is $\sigma$. $r_0 = \sigma$ means that initially molecule $B$ touches the polymer but is not associated with the polymer. Figure 8(a) shows the comparison between the results obtained from the mesoscopic model and microscopic model with $r_0 = \sigma$. We observe that the results do not converge. The result in the microscopic model reaches its peak at a smaller value of $r$ than the result in the mesoscopic result does. The number of molecule $C$ is 1921 in the microscopic model and 400.82 in the mesoscopic model. They do not converge either. However, it is not surprising to have such results. In the mesoscopic model, the molecules in the same voxel are not distinguishable. For the molecules in the same voxel, the distances between the molecules and the polymer $EF$ are the same and in the simulation, are approximated to the distance between the midpoint of the voxel
and the polymer $EF$, which is denoted by $r_m$. If the polymer $EF$ does not go through the midpoint of the voxel (especially when the polymer is far away from the midpoint), some of the molecules in the voxel have a longer distance to $EF$ than $r_m$, and some of the molecules in the voxel have a shorter distance to $EF$ than $r_m$. Since the molecules in the same voxel are assumed to be uniformly distributed, $r_m$ turns out a good approximation. However, if $r_m = 0$, the real distance between each molecule and the polymer must be greater than or equal to $r_m$. In this case, $r_m = 0$ is too small to be a good approximation. In another word, uniform distribution can not resolve initial distribution close to the polymer. The number of molecule $C$ obtained from the two models coincides with this argument.

By randomly generating the coordinates of 10000 molecules in the same voxel, we compute the distance between the molecule and the polymer $EF$. $r_{avg}$ is defined as the average value of the 10000 distances. We have $r_{avg} = 5.5987 \cdot 10^{-8}$ and take $r_0 = r_{avg} \approx 0.4h$. Figure 8(b) shows the comparison between the results obtained from the mesoscopic model and microscopic model with $r_0 = r_{avg} = 5.5987 \cdot 10^{-8}$. Good agreement is observed here. The number of molecule $C$ is 381 in the microscopic model and 400.82 in the mesoscopic model. The convergence here is not as good as in the previous two experiments in this section, but still satisfactory. More mathematical analysis is needed to get a more accurate result. Another approach is to use a different discretization or a different size of the simulation domain $\Omega$ to avoid such situation.

4 The polymer modeled as two parallel line segments

In this section, there are two polymers in the model. The purpose is to study the influence between them. Each of the polymers is modeled as a line segment. The line segments can
be placed anywhere inside the simulation domain $\Omega$ in the mesoscopic model. However, in the microscopic model the governing PDE becomes rather complicated at the boundary. In order to compare easily the results obtained from the two models, we use two parallel line segments in both models. In this case, we can use the bipolar cylindrical coordinate system in the microscopic model which reduces significantly the numerical complexity of the problem.

4.1 Mesoscopic simulation

We use the same simulation domain and discretization as in the previous section. Two line segments are denoted by $EF$ and $E'F'$, respectively. Each one is partitioned into $n_s$ non-overlapping identical subsegments. We employ Algorithm 2 to simulate the reaction-diffusion processes in this model. The quotient matrix $Q$ ($N$-by-$2n_s$) has to be computed accordingly. An algorithm of computing the quotient matrix $Q$ explicitly for this model is summarized in Algorithm 4. In Algorithm 4, we take advantage of the special structure of the polymers to

**Algorithm 4** Computing the quotient matrix $Q$ for polymers modeled as two parallel line segments

1. **Initialization:** $Rm = zeros(N, 2n_s)$, $t = 0$, $N_r = 10^7$.

2. Generate three random numbers $r1$, $r2$ and $r3$ which are uniformly distributed between 0 and $L$. Thus, $G = (r1, r2, r3)$ is a random point inside $\Omega$.

3. Calculate the distance $r_{G1}$ between the point $G$ and the line $EF$.

4. If $r_{G1} > r_c$, then the point $G$ is outside the cylinder $EF$. Go to (5). Otherwise, the point $G$ is inside the cylinder $EF$. Go to (7).

5. Calculate the distance $r_{G2}$ between the point $G$ and the line $E'F'$.

6. If $r_{G2} > r_c$, then the point $G$ is outside the cylinder $E'F'$. Update $t := t + 1$ and go back to (2). Otherwise, the point $G$ is inside the cylinder $E'F'$. Go to (7).

7. Update $t := t + 1$. Determine the index of the voxel where the point $G$ is. The index is denoted by $i$.

8. Determine the index of the subvolume where the point $G$ is. The index is denoted by $j$.

9. Update $Rm(i, j) = Rm(i, j) + 1$. Go to (2) if $t \leq N_r$.

10. Compute $Q(i, j) = \frac{Rm(i, j)}{\sum_{i=1}^{N} Rm(i, j)}$ for $i=1,2,\ldots,N$ and $j=1,2,\ldots,2n_s$. 

27
achieve a more computationally efficient implementation. Though there are two line segments in the model, the data of the molecules on the polymers is stored in a single $2n_s$-by-1 vector which is compatible with the stochastic simulation algorithm. At the end of the simulation, we obtain the distribution of the molecules in space as well as on the polymers.

4.2 Microscopic model

In the previous section we found that at time $t$ the PDF of finding a molecule at position $r$ satisfies the Smoluchowski equation, which is analogous to the heat equation. The main difficulty for this model is to formulate the boundary conditions. It is no longer possible to use the polar coordinates since there are two line segments embedded in the 3D space. However, we can use the bipolar cylindrical coordinate system $(u, v, z)$ which is defined as

$$x = a \frac{\sinh v}{\cosh v - \cos u},$$  \hspace{1cm} (35)

$$y = a \frac{\sin u}{\cosh v - \cos u},$$  \hspace{1cm} (36)

$$z = z,$$  \hspace{1cm} (37)

where $u \in [0, 2\pi]$, $v \in (-\infty, \infty)$ and $z \in (-\infty, \infty)$. The following identities show that the curves of constant $u$ and $v$ are circles in $xy$-plane

$$x^2 + (y - a \cot u)^2 = a^2 \csc^2 u,$$  \hspace{1cm} (38)

$$(x - a \coth v)^2 + y^2 = a^2 \csch^2 v. \hspace{1cm} (39)$$

Equation (39) is the function of a circle whose center is $(a \coth v, 0)$ and radius is $acsch v$. Thus, by choosing an appropriate value of $a$, $v$ and a constant $z$ we have two cylinders in the 3D space and can state the boundary conditions easily there. The illustration of bipolar cylindrical coordinates from [15] is shown in Figure 9(a).

In the bipolar cylindrical coordinate system, equation (39) gives that the radius of the cylinder is $acsch v$. The distance between the centers of the cylinders is $\frac{L}{q}$ in the $xy$-system in the mesoscopic model while it is $2a \coth v$ in the microscopic model. Thus we have the following equations

$$\begin{cases}
\sigma = acschv \\
\frac{L}{q} = 2a \coth v
\end{cases} \hspace{1cm} (40)$$

whose solution is

$$\begin{cases}
v = \pm \ln \left(\frac{L + \sqrt{L^2 + 4q^2 a^2}}{2q a}\right) \\
a = \frac{\sigma}{acsch v}
\end{cases} \hspace{1cm} (41)$$
Thus, the boundary conditions can be posed at $v_1 = \ln \left( \frac{L + \sqrt{L^2 + 4q^2\sigma^2}}{2q\sigma} \right)$ and $v_2 = -\ln \left( \frac{L + \sqrt{L^2 + 4q^2\sigma^2}}{2q\sigma} \right)$. The governing PDE in the $uv$-space is

$$\frac{\partial p}{\partial t} = f(u, v) \left( \frac{\partial^2 p}{\partial u^2} + \frac{\partial^2 p}{\partial v^2} \right)$$

with initial condition

$$p(u, v, t_0|u_{i0}, v_{j0}, t_0) = \delta(u - u_0)\delta(v - v_0)$$

and boundary conditions

$$2\pi\sigma D[g_1(u, v)\frac{\partial p}{\partial u} + g_2(u, v)\frac{\partial p}{\partial v}]|_{v=v_1} = k_d p(u, v_1, t),$$

$$2\pi\sigma D[g_1(u, v)\frac{\partial p}{\partial u} + g_2(u, v)\frac{\partial p}{\partial v}]|_{v=v_2} = k_d p(u, v_2, t),$$

$$p(0, v, t) = p(2\pi, v, t),$$

where

$$f(u, v) = D \frac{(\cosh v - \cos u)^2}{a^2},$$

$$g_1(u, v) = -\frac{(\cosh v - \cos u)\sqrt{(\sinh v)^2 + (\sin u)^2}}{a \sin u \cosh v},$$

$$g_2(u, v) = -\frac{(\cosh v - \cos u)\sqrt{(\sinh v)^2 + (\sin u)^2}}{a \sin v \cos u},$$

$u \in [0, 2\pi]$ and $v \in [v_1, v_2]$. In the simulation, the molecules are usually placed in the 3D space initially. In order to observe a significant influence of $EF$ and $E^\prime F^\prime$, we set $k_{d,\text{micro}} = 0$.

We use finite difference method to solve the PDE. The main difficulty comes from the function $f(u, v)$, $g_1(u, v)$ and $g_2(u, v)$ which are shown in Figure 9(b) and Figure 10.

$f(u, v)$ Figure 9(b) shows that the function $f(u, v)$ is close to 0 when $v$ is away from the boundary. However, it grows rapidly as $v$ reaches the boundary.

$g_1(u, v)$ and $g_2(u, v)$ Figure 10(a) shows the function $g_1(u, v_1)$, $u \in \left[ \frac{\pi}{2\pi}, \frac{\pi}{2\pi} + 2\pi \right]$. We shift $u$ slightly to the right to avoid $u = 0$ and $u = 2\pi$ where the function is unbounded. We must avoid $u = \pi$ to be one of the grid points. The same explanation goes to the other three figures.

An implicit method has the advantage of unconditional stability. However, it is only applied easily to the interior points of the mesh after discretization in space. Because of the functions $f(u, v)$, $g_1(u, v)$ and $g_2(u, v)$, a small time step has to be taken also in an implicit method for
Figure 9: (a) Bipolar cylindrical coordinate system (b) Function $f(u, v)$

Figure 10: Functions defining the boundary conditions in (48) and (49)
the accuracy. In addition, we have to solve a linear system in each time step which makes an implicit method more computationally expensive. Thus, we use an explicit method to solve the equation.

The time interval \([0, T]\) is divided into \(N\) identical subintervals with length \(\Delta t = \frac{T}{N}\). The spatial grid is uniform with grid points \((u_i, v_j)\) where \(i = 1, 2, 3, ..., N_u\) and \(j = 1, 2, 3, ..., N_v\). We assume that \(N_u\) and \(N_v\) are odd numbers so that the spatial grid contains \((\pi, 0)\) which is the midpoint of the domain. Using a forward difference at time \(t_n\) and a second-order central difference for the space derivative at position \((u_i, v_j)\), equation (42) is discretized as

\[
\frac{p_{i,j}^{n+1} - p_{i,j}^n}{\Delta t} = f(u_i, v_j) \left( \frac{p_{i+1,j}^n - 2p_{i,j}^n + p_{i-1,j}^n}{(\Delta u)^2} + \frac{p_{i,j+1}^n - 2p_{i,j}^n + p_{i,j-1}^n}{(\Delta v)^2} \right),
\]

(50)

where \(\Delta u = u_2 - u_1\) and \(\Delta v = v_2 - v_1\). After rearranging the terms we obtain

\[
p_{i,j}^{n+1} = f(u_i, v_j) \frac{\Delta t}{(\Delta u)^2} (p_{i+1,j}^n - 2p_{i,j}^n + p_{i-1,j}^n) + f(u_i, v_j) \frac{\Delta t}{(\Delta v)^2} (p_{i,j+1}^n - 2p_{i,j}^n + p_{i,j-1}^n) + p_{i,j}^n.
\]

(51)

When computing \(p_{i,j}^{n+1}\), \(p_{N_u,j}^{n+1}\), \(p_{i,1}^{n+1}\) and \(p_{i,N_v+1}^{n+1}\), the values of \(p_{0,j}^n\), \(p_{N_u+1,j}^n\), \(p_{i,0}^n\) and \(p_{i,N_v+1}^n\), respectively, are needed. Those points are outside the initial spatial grid, but can be obtained by the boundary conditions. The finite difference approximations of (44) and (45) are

\[
2\pi\sigma D \left( g_1(i, 1) \frac{p_{i+1,1}^n - p_{i-1,1}^n}{2\Delta u} + g_2(i, 1) \frac{p_{i,2}^n - p_{i,0}^n}{2\Delta v} \right) = k_a p_{i,1}^n,
\]

(52)

\[
2\pi\sigma D \left( g_1(i, N_v) \frac{p_{i+1,N_v}^n - p_{i-1,N_v}^n}{2\Delta u} + g_2(i, N_v) \frac{p_{i,N_v+1}^n - p_{i,N_v-1}^n}{2\Delta v} \right) = k_a p_{i,N_v}^n,
\]

(53)

which give

\[
p_{i,0}^n = p_{i,2}^n - \frac{\Delta v}{\pi\sigma D g_2(i, 1)} k_a p_{i,1}^n + \frac{\Delta v g_1(i, 1)}{\Delta u g_2(i, 1)} p_{i+1,1}^n - \frac{\Delta v g_1(i, 1)}{\Delta u g_2(i, 1)} p_{i-1,1}^n,
\]

(54)

\[
p_{i,N_v+1}^n = p_{i,N_v-1}^n + \frac{\Delta v}{\pi\sigma D g_2(i, N_v)} k_a p_{i,N_v}^n - \frac{\Delta v g_1(i, N_v)}{\Delta u g_2(i, N_v)} p_{i+1,N_v}^n + \frac{\Delta v g_1(i, N_v)}{\Delta u g_2(i, N_v)} p_{i-1,N_v}^n.
\]

(55)

The periodic boundary condition (46) gives

\[
p_{0,j}^n = p_{N_u-1,j}^n,
\]

(56)

\[
p_{N_u+1,j}^n = p_{N_u,j}^n.
\]

(57)

The explicit method is known to be numerically stable and convergent when

\[
f_{\max} \cdot \left( \frac{\Delta t}{(\Delta u)^2} + \frac{\Delta t}{(\Delta v)^2} \right) \leq \frac{1}{2},
\]

(58)
where \( f_{\text{max}} = \max_{i=1,2,...,N_u} f(u_i, v_j) \). However, an even smaller time step may be required in practice due to the boundary condition. It is difficult to derive a strong stability condition for this problem.

In order to compare with the solutions obtained from the mesoscopic simulation, the PDF \( p_{u,v,t} \) in the bipolar cylindrical coordinate system has to be transferred to the function \( p_{x,y,t} \) in the Cartesian coordinate system. The transformation can be done by a built-in function \texttt{TriScatteredInterp} in MATLAB which is used in the experiment in this section.

4.3 Comparison

In this section, we compare the results obtained from the mesoscopic simulation and the microscopic solution with respect to the distribution of the molecules. The model parameters are \( L = 3 \cdot 10^{-6}, \sigma = 1.005 \cdot 10^{-9}, n = 20, n_s = 20, T = 0.1, D = 10^{-12} \) and \( k_{d,\text{micro}} = 0 \). The coordinates of the lines are \( E = (\frac{L}{2}, L, \frac{2L}{3}), F = (\frac{L}{2}, 0, \frac{2L}{3}), E' = (\frac{L}{2}, L, \frac{L}{3}) \) and \( F' = (\frac{L}{2}, 0, \frac{L}{3}) \). The distance between \( EF \) and \( E'F' \) is \( \frac{L}{2} \) so that \( q = 3 \) in equation (40). The illustration of the simulation domain is shown in Figure 11.

4.3.1 Experiment 1

At time \( t = 0 \), 10000 molecules \( \mathcal{B} \) are placed at the midpoint of the simulation domain \( \Omega \). Thus, in the microscopic model the initial position of the molecule is \( (\pi, 0) \). However, the midpoint of \( \Omega \) does not coincide with any midpoint of the subvolume in the mesoscopic model since \( n = 20 \) is an even number. In this case, 10000 molecules are uniformly distributed in the eight voxels which are around the midpoint of \( \Omega \) with 1250 molecules in each of them.
The association rate in the microscopic model is $k_{a,\text{micro}} = 5 \cdot 10^{-13}$. By using the formula in [12], we have $k_{a,\text{meso}} = 17.5155$ and $k_{d,\text{meso}} = 0$.

Figure 12(a) shows the solution of the PDE (42) in $uv$-plane in the microscopic model. The peak of the solution is at the point $(\pi, 0)$ where the molecule is initialized. The solution is 0 at the point $(0, 0)$ and $(2\pi, 0)$ because of the boundary condition of variable $u$. The effect of the polymers can be observed at the boundary of $v$. The solution is symmetric with respect to $u = \pi$ and $v = 0$ due to the position of the polymers and the initialization of the molecules. From the graph point of view, this is a reasonable result for equation (42). Figure 12(b) is the projection of Figure 12(a) in the $uv$-plane which is generated by the function \textit{contour} in MATLAB.

By using the function \textit{TriScatteredInterp}, we transfer the PDF $p_{u,v,t}$ in the bipolar cylindrical coordinate system to $p_{x,z,t}$ in the Cartesian coordinate system. The illustration of $p_{x,z,t}$ is shown in Figure 13(a). We observe that it is almost a normal distribution. It should be a normal distribution if there is no polymer in the model. We expect that there would be perturbations around the polymers, but they are not visible. This may be due to the low association rate we use in this experiment. Also, the initial position of molecule $B$ is further away from the polymers than that in the experiments in Section 3. In the next experiment, we use a higher association rate in order to observe the perturbation.

Figure 13(b) shows the distribution of molecule $B$ in the mesoscopic model. It is almost a normal distribution. We can not observe any perturbation here either. Comparing the results obtained from two models, we find that they are almost the same, but $p_{u,v,t}$ is slightly larger than $p_{x,z,t}$. This is due to the coarse mesh ($81 \times 51$) we use to discretize the PDE. A more accurate result can be obtained by refining the mesh close to $v = v_1$ and $v_2$, see the singularities in Figure 10.
Figure 13: (a) Solution of the Smoluchowski equation in Cartesian coordinates (b) Solution from the mesoscopic model

Figure 14(a) shows the distribution of molecule $\mathcal{B}$ in the $xy$-plane in the mesoscopic simulation and Figure 14(b) shows the distribution of molecule $\mathcal{B}$ in the $yz$-plane in the mesoscopic simulation. Both of them are almost normal distribution. In these two figures, the perturbations of the polymers are distributed along the $y$ axis. In this case, we can not observe the perturbation unless a large number of molecules $\mathcal{B}$ are associated with the polymer in the end of the simulation. In conclusion, the results obtained from the two models coincide with each other.

4.3.2 Experiment 2

In this experiment, the same values of the parameters are used in the simulation except for a larger association rate in order to observe the perturbation around the polymers. We use $k_{a,\text{micro}} = 5 \times 10^{-7}$ and $k_{d,\text{micro}} = 0$ which yield $k_{a,\text{meso}} = 82.6965$ and $k_{d,\text{meso}} = 0$. These values are out of range in a real living cell. Thus, this experiment is constructed to test the algorithm rather than to solve a real problem.

We use Algorithm 2 for simulation of the reaction-diffusion processes in the domain $\Omega$. Figure 15(a), Figure 15(b) and Figure 15(c) show the distribution of molecule $\mathcal{B}$ in $xz$-plane, $xy$-plane and $yz$-plane in the mesoscopic model, respectively. In Figure 15(a), the perturbations are observed around the point $(1.5, 1) \times 10^{-6}$ and $(1.5, 2) \times 10^{-6}$. These are exactly the positions of the polymers. The perturbation along a single polymer is accumulated along the $y$-axis so that it is large enough to be visible. The perturbations in Figure 15(b) and Figure 15(c) are not clear because they are distributed along the polymer. An even larger association rate is needed to observe the perturbations there. However, there is a limit of the reaction rate in the mesoscopic model [12] which is also shown in Table 2. The association rate in this experiment is quite close to its limit. A feasible method is to initialize molecule $\mathcal{B}$ at a position which is closer to one of the polymers. We expect to observe a significant
Figure 14: (a) Solution of the Smoluchowski equation in $xy$-plane in the Cartesian coordinates 
(b) Solution of the Smoluchowski equation in $yz$-plane in the Cartesian coordinates

Figure 15: Solutions of the mesoscopic simulation of the model with two parallel line segments in $xz$, $xy$ and $yz$-plane

perturbation around that polymer. However, the solution is no longer symmetric. And in this case, it is analogous to the experiments in the previous section with a single line segment in the model.

In the microscopic model, $k_{a,meso} = 5 \cdot 10^{-7}$ makes it practically impossible to solve the PDE. Assume we use the same mesh as in the first experiment, then in (55) we have $\Delta v k_a \gg \pi \sigma D g_2(i, N_v)$. This inequality makes the solution go to infinity after a few time steps regardless of the stability condition for the interior points of the mesh. The only way to avoid the infinite solution is to use an extremely small time step requiring much work on high performance computing and parallel computers. However, it is not worth doing so since the mesoscopic simulation is a much better approach to handle this problem.

35
In the previous sections, the experiments have shown that the mesoscopic simulation algorithm is accurate and convergent to the microscopic solution. In this section, we test the algorithm with a more complicated model. Curvature is involved in modeling the polymer. In most cases, solving the Smoluchowski equation in the microscopic model is almost infeasible due to the complicated boundary conditions of the PDE and the requirements of the mesh there. The mesoscopic simulation is robust and can handle a much wider range of problems.

5.1 A polymer is modeled as a circle

In the first stage, the polymer is modeled as a circle. A more complicated geometry is treated later in this section.

5.1.1 Mesoscopic simulation

We use the same simulation domain $\Omega$ and its discretization as in the previous sections. A circle with radius $R$ and center $O$ is embedded in the 3D space. Algorithm 1 is suitable for computing the quotient matrix $Q$. However, we take advantage of the spetial geometry of the polymer since a circle has a highly symmetric shape. We analyze the discretization of the circle and propose an algorithm explicitly for computing the quotient matrix $Q$ for this model which is more computationally efficient than Algorithm 1.

The discretization of the circle is shown in Figure 16(a). The circle in blue is discretized into identical arcs which are approximated by line segments (red dashes). We assume that
the number of arcs is \( n_s \) so that the angle \( \theta \) is given by \( \theta = \frac{2\pi}{n_s} \). We denote the line segments by \( sl_1, sl_2, ..., sl_{n_s} \). The intersection between \( sl_1 \) and \( sl_{n_s} \) is considered to be the starting point of the polygon and is denoted by \( A_0 \). As stated in the previous sections, the molecule in space may associate with the polymer if the distance between them is less than \( r_c \). Thus, each of the segments \( sl \) is surrounded by a smooth cylindrical surface to compose a new geometry whose projection in 2D is displayed in Figure 16(a), the quadrilateral \( a1a2a3a4 \). The subvolumes are denoted by \( sv_1, sv_2, ..., sv_{n_s} \). Figure 16(b) shows the polymer with its cylindrical surface in 3D. The big red circle which is denoted by \( \hat{C} \) models the polymer and it becomes the blue torus in the mesoscopic model. The radius of the small red circle is \( r_c = \frac{h}{\sqrt{\pi}} \) as stated in Section 2.1.

We use Algorithm 2 to simulate the reaction-diffusion processes in this model. A more computationally efficient algorithm of computing the quotient matrix \( Q \) is developed. Whether a random point is inside a subvolume or not is determined with the help of the distance between the point and the line segments. For a molecule inside one of the subvolumes, the main difficulty is to determine its index. This can be done by considering the polymer to be in a polar coordinate system and using the angular coordinate \( \theta \). The procedure of computing the quotient matrix \( Q \) explicitly for the model in this section is summarized in Algorithm 5.

Algorithm 5 Computing the quotient matrix \( Q \) for the polymer modeled as a circle

1. **Initialization:** \( Rm = zeros(N, n_s) \), \( t = 0 \), \( N_r = 10^7 \).

2. Generate three random numbers \( r_1, r_2 \) and \( r_3 \) which are uniformly distributed between 0 and \( L \). Thus, \( G = (r_1, r_2, r_3) \) is a random point inside \( \Omega \). Set \( k = 1 \).

3. If \( k \leq n_s \), calculate the distance \( r_{G_k} \) between the point \( G \) and segment \( s_k \); otherwise update \( t := t + 1 \) and go back to step (2).

4. If \( r_{G_k} > r_c \), the point \( G \) is outside the subcylinder \( sc_k \). Update \( k := k + 1 \) and go back to step (3). Otherwise, the point \( G \) may be inside the cylinder \( sc_k \). Go to step (5).

5. Calculate the angle \( \theta \) between two rays \( OG_k \) and \( OA_0 \). Point \( G \) is inside subcylinder \( sc_k \) if \( \frac{2\pi}{n_s} (k - 1) \leq \theta < \frac{2\pi}{n_s} k \), go to step (6); otherwise update \( k := k + 1 \) and go back to step (3).

6. Determine the index of the subvolume \( i \) where the point \( G \) is, with the help of its coordinate.

7. Update \( Rm(i, k) := Rm(i, k) + 1 \), \( t := t + 1 \). Go back to (2) if \( t \leq N_r \).

8. Compute \( Q(i, j) = \frac{V_i}{V_j} = \frac{\sum_{i=1}^{N} Rm(i,j)}{\sum_{i=1}^{N} Rm(1,j)} \) for \( i = 1, 2, ..., N \) and \( j = 1, 2, ..., n_s \).
5.1.2 Microscopic model on the circle

The PDF of the molecules in space is the solution of the Smoluchowski equation as shown in (13). However, it is difficult to state the boundary conditions on a torus in a finite difference method even in a toroidal coordinate system, and impossible to do so with a more complicated geometry. Thus, we only simulate the reaction-diffusion processes mesoscopically and analyze the results.

The pure diffusion process on the circle follows the normal distribution and the PDF can be computed analytically as shown in (32). This is used to compare with the distribution obtained from the mesoscopic simulation in the following experiments.

5.1.3 Experiments

Experiment 1 In this experiment, the geometry of the polymer is a circle. The radius of the circle is \( R = 0.239L \) where \( L \) is the side length of the simulation domain. The center of the circle coincides with the midpoint of \( \Omega \). The number of subcylinders \( n_s = 30 \). At time \( t = 0 \), 10000 molecules \( B \) are uniformly distributed in eight voxels around the midpoint of the simulation domain \( \Omega \), with 1250 molecules \( B \) in each of those voxels. The association rate \( k_{a,\text{micro}} = 5 \times 10^{-13} \) in the microscopic model yields \( k_{a,\text{meso}} = 17.5155 \) in the mesoscopic model.

We use the same association rate and dissociation rate so that \( k_{d,\text{micro}} = k_{a,\text{micro}} = 22.2222 \) and \( k_{d,\text{meso}} = k_{a,\text{meso}} = 17.5155 \).

Figure 17 shows the distribution of molecule \( C \) at the end of the simulation. The histogram represents the number of molecule \( C \) in each subvolume and the red horizontal line is the mean value. Due to the symmetry of the geometry and the initial conditions, the number of molecule \( C \) should be almost the same in all subvolumes which can be observed in Figure 17.

Experiment 2 In this experiment, we use the same parameters as in the first experiment, but initialize 10000 molecules \( C \) in the 15\textsuperscript{th} subvolume. The coordinate of the midpoint of
Figure 18: (a) Distribution of molecule $C$ on the circle compared with the analytical solution of the diffusion equation (b) The distribution of molecule $B$ in the $yz$-plane

the 15th subvolume is $(0.2623L, 0.525L, 0.5L)$. The diffusion on a curve is also a normal distribution, the same as the diffusion on a straight line in (32). Thus, we can compare the distribution of molecule $C$ in the mesoscopic model with the corresponding normal distribution. We also expect to observe the perturbation in space caused by the polymer.

Figure 18(a) shows the distribution of molecule $C$ at the end of the simulation. The histogram is the result in the mesoscopic model and the red curve is the analytical solution of the 1D diffusion equation. Very good agreement is observed. Figure 18(b) shows the distribution of molecule $C$ in the $yz$-plane. The projection of the circle in the $yz$-plane is a line segment with length $2R$ where $R$ is the radius of the circle. Similarly, the projection of the torus in the $yz$-plane is a rectangle with length $2R$ and width $2r_c$. In Figure 18(b), the perturbation is observed along $z = 1.5 \cdot 10^{-6}$ where exactly the polymer lies.

5.2 A polymer is modeled as a spiral

Curvature is involved in modeling the polymer in the experiments in the last section. Now we add variation also along the $z$-axis and model the polymer as a spiral. This is a general structure of the polymer. We can also use a more complicated geometry in the simulation. However, it would be difficult to examine and analyze the result. Thus, a spiral is an appropriate geometry to test the algorithm in a general way.
Figure 19: (a) Illustration of DNA [17] (b) Geometry of the spiral (c) Geometry of the spiral composed by random points

5.2.1 Mesoscopic simulation

We use the same domain Ω and its discretization as in the previous section. The function of the spiral shown in Figure 19(b) is

\[
\begin{align*}
x &= \left(\frac{1}{4} \cos s + \frac{1}{2}\right) \cdot L, \\
y &= \left(\frac{1}{4} \sin s + \frac{1}{2}\right) \cdot L, \\
z &= \frac{1}{4\pi} s \cdot L,
\end{align*}
\]

where \(0 \leq s \leq 4\pi\) and \(L\) is the side length of the simulation domain Ω. The spiral is discretized into nonoverlapping identical segments which are approximated by the line segments \(sl_1, sl_2, \ldots, sl_{n_s}\). The line segments are surrounded by the smooth surfaces. The distance between any point on the surface and its corresponding line segment is \(r_c = \frac{h}{\sqrt{\pi}}\) where \(h\) is the side length of the voxel in space. We denote the subvolumes by \(sv_1, sv_2, \ldots, sv_{n_s}\). Algorithm 1 is used to compute the quotient matrix \(Q\) for this model. It is still possible to take advantage of the geometry of the polymer. We aim at testing the algorithm in a general way so that we apply Algorithm 1. Figure 19(c) shows the geometry composed by the random points when computing \(Q\) by Algorithm 1. The shape of a spiral is observed.

5.2.2 Experiments

Experiment 1  The polymer is modeled as a spiral shown in Figure 19(b). The function of the spiral is given by (59). The number of subvolumes after discretization is \(n_s = 40\). In the beginning, 10000 molecules \(C\) are placed in the 20th subvolume. The coordinate of the midpoint of the 20th subvolume is (0.7469L, 0.4609L, 0.4875L). The association rate
where \( k_{a,micro} = 5 \cdot 10^{-13} \) in the microscopic model yields \( k_{a,meso} = 17.5155 \) in the mesoscopic model. We use the same association rate and dissociation rate so that \( k_{d,micro} = k_{a,micro} h^2 = 22.2222 \) and \( k_{d,meso} = k_{a,meso} = 17.5155 \).

Figure 20(a) shows the distribution of molecule \( C \). The histogram is the result in the mesoscopic simulation and the red curve is the solution of the corresponding diffusion equation. Good agreement is observed. Figure 20(b) shows the number of molecule \( B \) in the \( yz \)-plane. It looks like a normal distribution being perturbed. This can be explained by Figure 20(c) where the projection of the spiral in the \( yz \)-plane is shown. The red square is where molecule \( B \) is initialized at the beginning of the simulation. Figure 20(a) shows that molecule \( B \) spreads in a range of approximately \( 2 \cdot 10^{-6} \). Thus, we expect that the perturbation occurs along the line from the point \((0.8, 1.2) \cdot 10^{-6}\) to the point \((2.2, 1.7) \cdot 10^{-6}\). This is exactly what we observe in Figure 20(b).

**Experiment 2** In this experiment, we use the same discretization in space as in Experiment 1. We initialize 10000 molecules \( B \) in the 3793\textsuperscript{th} voxel in space. The coordinate of the midpoint of the 3793\textsuperscript{th} voxel is \((0.625L, 0.475L, 0.475L)\). This voxel does not overlap with the spiraling polymer but is close to the 20\textsuperscript{th} subvolume whose midpoint’s coordinate is \((0.7469L, 0.4609L, 0.4875L)\). The point \((0.625L, 0.475L)\) is inside the projection of the polymer in \( xy \)-plane. In order to observe a significant perturbation in space, we use the reaction parameters \( k_{a,micro} = 5 \cdot 10^{-9} \) and \( k_{d,micro} = 0 \) in the microscopic model which yield \( k_{a,meso} = 82.666 \) and \( k_{d,meso} = 0 \) in the mesoscopic model.

Figure 21(a) shows that the distribution of molecule \( C \) in the mesoscopic simulation does not converge with the microscopic solution of the diffusion equation. It is not surprising to get such a result since the spiraling polymer is not symmetric with respect to the initial position of the molecules. Also, the initial position of the molecules has almost the same distance to several subsegments which helps the molecules spread further on the polymer.

Figure 21(b) shows the number of molecule \( B \) in the \( yz \)-plane. The perturbation is clearly observed and the same explanation can be given as in Experiment 1.
Figure 21: (a) Distribution of molecule $C$ on the spiral comparing with the analytical solution of the diffusion equation (b) Distribution of molecule $B$ in the $yz$-plane

6 Roadblocks

In this section, the polymer models DNA (black spiraling structures in Figure 22(a)). DNA, deoxyribonucleic acid, is a nucleic acid containing the genetic instructions. The structure of the DNA is shown in Figure 19(a). For convenience, we use part of the DNA for analysis as shown in Figure 22(a). There are specific binding sites (BS in Figure 22(a)) and nonspecific binding sites on DNA. A transcription factor (red cylinder in Figure 22(a)), which is also called a sequence-specific DNA-binding factor, is a protein that binds to specific DNA sequences [20]. There are also roadblocks (black cylinders in Figure 22(a)) on DNA to prevent the 1D diffusion process of the transcription factor. A transcription factor can diffuse along the DNA but can not pass the roadblocks. It can also dissociate from the DNA and associate with other parts of the DNA and search for the specific binding site [11]. In a living cell, there are few copy numbers of each transcription factor. We are interested in the time $t_{\text{bind}}$ when the transcription factor binds to the specific binding site.

We use the same $\Omega$ and its discretization as in the previous sections. In the beginning, the DNA is modeled as a straight line denoted by $EF$ whose illustration is shown in Figure 22(b). $EF$ is partitioned into identical subsegments. Two roadblocks (black circles) are on $EF$ at the fixed positions with a distance $L_{rb}$ between each other. A specific binding site (red circle) is on $EF$ at a fixed position and is denoted by $BS$. Transcription factors diffuse in space (blue circles). They are the same as the molecule $B$ in the previous model. We denote them by $B$ here as well. A transcription factor may associate with $EF$ and then diffuse along the DNA. The transcription factor does not change after the association. However, we denote it by $C$ separately to indicate that it is now attached to the polymer. Transcription factor $C$ is the yellow circle in Figure 22(b). A transcription factor $C$ diffuses along $EF$ but is blocked by the roadblocks. It may also dissociate from $EF$ and form a transcription factor $B$. Assume
Figure 22: (a) DNA, transcription factor and specific binding site [14] (b) Model used in the first experiment (c) A transcription factor moves between segments of the DNA [14] (d) Model used in the second experiment
the simulation starts at \( t = 0 \), then the transcription factor binds to the specific binding site at time \( t = t_{\text{bind}} \). The association, dissociation and binding events can be characterized by the following reactions

\[
B + EF \xrightleftharpoons[k_d]{k_a} C + EF, \quad (60)
\]

\[
C + BS \xrightarrow{k_{\text{bind}}} \Phi, \quad (61)
\]

where \( k_a \) and \( k_d \) are the association rate and dissociation rate, respectively. \( k_{\text{bind}} \) is the rate for the event that the transcription factor binds to the specific binding site. In the implementation, the simulation stops immediately when the binding event occurs.

In the experiment, the coordinates of the line segment are \( E = (0, 0, 0) \) and \( F = (L, L, L) \) where \( L \) is the side length of \( \Omega \). Line segment \( EF \) is partitioned into \( n_s = 35 \) identical subsegments. The specific binding site \( BS \) is fixed in the 17th subvolume. The coordinate of the midpoint of the 17th subvolume is \( \left( \frac{33}{70}L, \frac{33}{70}L, \frac{33}{70}L \right) \). There is one road block on each side of \( BS \). By changing the position of the roadblocks, we have different values of \( L_{rb} \). For each \( L_{rb} \), we simulate the time \( t_{\text{bind}} \) and analyze how \( t_{\text{bind}} \) depends on \( L_{rb} \). For each \( L_{rb} \), we simulate 10 transcription factors \( B \) are initialized in the 3790th voxel in space. The coordinate of the 3790th voxel is \( (0.475L, 0.475L, 0.475L) \). The 3790th voxel is overlapped by the 17th subvolume. We expect a short binding time since the initial position of the transcription factor is close to the specific binding site. The other model parameters are \( k_a = k_d = k_{\text{bind}} = 5 \cdot 10^{-13} \), the diffusion constant in space \( D = 10^{-12} \) and the diffusion constant on the DNA \( D_l = 10^{-14} \). In Figure 23(a), the blue line is the average value of \( t_{\text{bind}} \) in 1000 simulations with the corresponding value of \( L_{rb} \). There is no significant dependence of \( t_{\text{bind}} \) on \( L_{rb} \) for this choice of reaction parameters. The red polygonal line is the standard deviation of its corresponding \( t_{\text{bind}} \). We observe that the standard deviation is quite large, even larger than the average of \( t_{\text{bind}} \). Figure 23(b) shows the binding time \( t_{\text{bind}} \) in 1000 simulations when the distance between two roadblocks \( L_{rb} = 2.2269 \cdot 10^{-6}m \). The red line is the mean value of \( t_{\text{bind}} \) in 1000 simulations. We observe that \( t_{\text{bind}} \) varies a lot among simulations and it is not clear in which range of the time that \( t_{\text{bind}} \) most likely lies. In conclusion, there is no significant dependence of \( t_{\text{bind}} \) on \( L_{rb} \) in our parameter range. Hammar, Leroy, Mahmutovic, Marklund, Berg and Elf have developed an analytical formula for this problem in [11]. Our result coincides with that formula. The dependence of \( t_{\text{bind}} \) on \( L_{rb} \) is expected to be observed with a realistic set of parameter values.

Another model that we are interested in involves more complicated reaction-diffusion processes. The simulation domain \( \Omega \) and its discretization are the same as in the previous sections. The polymer is modeled as a line segment \( EF \) where \( E = (0, 0, 0) \) and \( F = (L, L, L) \). The illustration of \( EF \) is shown in Figure 22(d). There is an operator site \( D \) (red) fixed on the polymer. There are also roadblocks \( G \) (black) on the polymer. They can diffuse along the polymer, react with \( D \) to form \( H \), or release from the polymer to become \( F \) which diffuses
in 3D space. The reversible association is also allowed. The reactions can be written as

\[ \begin{align*}
G + D & \overset{k_{a1}}{\underset{k_{d1}}{\rightleftharpoons}} H, \\
F + EF & \overset{k_{a2}}{\underset{k_{d2}}{\rightleftharpoons}} G + EF.
\end{align*} \tag{62} \tag{63} \]

The molecule in yellow is denoted by \( C \). It diffuses on the polymer. \( G \) and \( C \) can not pass each other. However, \( C \) can react with \( D \) and form \( E \) which is fixed at the same position. The reversible reaction is also allowed. This reaction can be written as

\[ C + D \overset{k_{a3}}{\underset{k_{d3}}{\rightleftharpoons}} E. \tag{64} \]

\( C \) can dissociate from the polymer and become \( B \) in the free space. The reaction can be written as

\[ B + EF \overset{k_{a4}}{\underset{k_{d4}}{\rightleftharpoons}} C + EF. \tag{65} \]

\( B \) diffuses in space in the same way as \( F \). When \( B \) moves a distance \( L_a \) away from the polymer, we say that it will not come back to the polymer any more. The simulation stops and the time \( t_a \) is recorded. We expect to find the dependence of \( t_a \) on the initial number of \( F \).

In the experiment, we use the same simulation domain and discretization as in the first model. We use \( L_a = 0.2L \) in the simulation. All the reaction rates are set to be \( 5 \cdot 10^{-7} \) and
Figure 24: (a) $t_a$ in 1000 consecutive simulations with $nF = 1$ (b) $t_a$ in 1000 consecutive simulations with $nF = 1000$

the diffusion rates are $10^{-12}$. At time $t = 0$, one molecule $\mathcal{D}$ and one molecule $\mathcal{C}$ are fixed in the 17th subvolume. We also initialize $\mathcal{F}$ in the 3790th voxel in space. By changing the initial number of $\mathcal{F}$, we aim at finding the dependence of $t_a$ on $nF$.

Figure 24(a) shows $t_a$ in 1000 consecutive simulations with $nF = 1$ and Figure 24(b) shows $t_a$ in 1000 consecutive simulations with $nF = 1000$. The red lines are the average values of each set of simulations, respectively. The average value of $t_a$ in Figure 24(a) is 0.1266s and in Figure 24(b) is 0.1219s. Since $\mathcal{F}$ becomes $\mathcal{G}$ after association with the polymer and $\mathcal{C}$ can not pass $\mathcal{G}$, it is more likely to have $\mathcal{B}$ in the system if $nF$ is large. Consequently, it takes a shorter time for $\mathcal{B}$ to diffuse away from the polymer. The result in the experiment coincides with this argument. However, the variation is large among simulations. This is due to the parameters we use in the experiment.

In this section, we use NVM to solve two problems which are almost impossible to be solved microscopically due to the complexity of the problems. It is shown that the mesoscopic simulation handles more problems than the microscopic simulation.

7 Conclusion

In this thesis, a new model for the stochastic simulation of reaction-diffusion processes in a living cell is proposed and studied. We embed 1D polymers in the 3D space. The polymer has an arbitrary shape. Reaction and diffusion events occur both in space and on the polymers. This model gives a more precise representation of the complicated intracellular structure.

We simulate at a mesoscopic level and compare with microscopic solutions of the Smoluchowski equation. At the mesoscopic level, there are several algorithms for the stochastic simulation of reaction-diffusion processes, such as Gillespie’s SSA [7][8], NSM[3] and NRM[6]. These algorithms can only be used mesoscopically for the model in free space.
Thus, we develop the corresponding algorithm \( NVM \) for the new model in this thesis. \( NVM \) is based on \( NSM \) and the overlap between the mesoscopic polymer model and the voxels in 3D space stored in the quotient matrix \( Q \). We use the Monte Carlo approach to develop an algorithm for computing \( Q \) for a polymer with an arbitrary shape. The algorithm is tested to be correct and computationally efficient. Several simplified versions of the algorithm are also provided for a specific shape of the polymer. We also test \( NVM \) with different shapes of the polymers, such as a straight line, a circle and a spiral. The results in the mesoscopic simulation converge well with the microscopic solution of the Smoluchowski equation. The Smoluchowski equation is solved by finite difference method in a specific coordinate system adapted to the shape of the polymers. Very good agreement is observed between the mesoscopic simulation and the microscopic solution. However, the boundary condition becomes rather complicated when the polymer has an irregular shape. In this case, it is too computationally expensive to solve the Smoluchowski equation. Then the mesoscopic simulation is preferred.

The mesoscopic simulation is robust and can handle more complicated problems as stated in Section 6. It is practically too difficult to solve those problems microscopically. The realistic model parameters have to be obtained from the biologists in order to do more analysis of the results. The future work will be focused on optimizing the model to represent the intracellular structure more precisely and developing the corresponding mesoscopic simulation algorithm accurately and efficiently. If we want to solve the Smoluchowski equation for the model of an arbitrary shaped polymer embedded in space, parallel computing is needed. The results from the biological experiments can also be used for comparison if available.

References


