Molecular Communications

Ömer Faruk Ünal

November 13, 2023

Abstract

This paper explores molecular communication, simulating how particles move, reflect and absorb in both 3D and 2D environments. It analyzes the behavior of molecules around absorbing surfaces and reflecting lines. By simulating their stochastic movement, it aims to unravel the dynamics of molecular communication channels, providing insights into their workings.

1 Introduction

Molecular communication, a rapidly growing field at the intersection of nanotechnology and communication systems, explores the transmission of information through the use of molecules as carriers. Understanding the dynamics of molecule diffusion and absorption within a three-dimensional space is crucial in designing efficient communication systems based on this paradigm.

The simulation of molecule diffusion and its interaction with receptors or absorbers, particularly in a 3D infinite space, is a fundamental aspect of modeling molecular communication channels. This simulation involves the stochastic movement of molecules, influenced by factors such as the diffusion coefficient, time steps, emission point, and the characteristics of the receiver or absorber.

The simulation considers various parameters, such as the diffusion coefficient, receiver radius, emission point, and the number of molecules, to analyze the cumulative absorption of molecules over time. Moreover, in real-life scenarios, molecules interact with various obstacles and can be reflected by these elements. In this document, a line in a 2D environment will be simulated to account for these occurrences and examine how molecular absorption changes in such conditions.

This document aims to explore the simulation process of molecule diffusion towards a spherical and circular absorber, detailing the computational steps involved in simulating the absorption of molecules over discrete time intervals and the numerical modeling of molecule movement within a 3D space and 2D space with line obstacle. The rest of this paper is organized in the following manner. Section 2 presents an overview of the molecular systems. Section 3 presents system model section explaining the topology of the system, 3D and 2D simulations, Section 4 shares the numerical results and plots.

2 Introduction to Molecular Systems

Molecular communication represents a relatively new communication paradigm that deviates from traditional systems. In conventional communication, electromagnetic waves or electrons serve as the primary carriers, whereas in molecular systems, molecules themselves are the fundamental information carriers. The traditional process involves encoding the text, voice or video based information in the sender side and subsequently decoding information at the receiver's end. Conversely, in molecular communication, molecules incite biochemical reactions within the receiver, regenerating the sender's status.

Despite the comparatively slower speed and small communication range of molecular communication systems compared to traditional methods, they possess certain advantages. They can carry information that traditional systems cannot, such as the chemical status of living organisms, or transmit signals in environments where conventional communication methods are impractical, such as biological entities. Molecular communication introduces a unique approach, diverging from traditional methods, to transmit information using the language of molecules, unlocking new possibilities and applications across various domains.

There are many proposed systems about molecu-

lar communication. Among this systems Molecular Communication via Diffusion is an energy-efficient and effective method. [1] In Molecular Communication via Diffusion (MCvD), data transmission occurs through the movement of molecules across the environment. This system consists of five primary stages: encoding, emission (transmission), movement through the environment, absorption (reception), and decoding. [2]

In this paper, our first simulation will focus on calculating the probability of particle absorption emitted from a specific point in a 3D environment, following the diffusion law and stochastic movement. Subsequently, we will explore the implications of the presence of obstacles surrounding these particles. For this latter part, we will transition to a 2D environment, utilizing an infinite line as a simplified representation of an obstacle.

3 System Model

For the Molecular Communication via Diffusion, first we will examine in relatively simplified simulation in 3D without reflecting surface.

3.1 Diffusion Simulations in 3D

For this part, we will assume the molecular channel operates within a 3-D space. The transmitter, a point source at distance d from the closest point of the surface of absorbing receiver, emits particles to carry information to the receiver. Each molecule moves independently within the channel following solely Brownian motion. The receiver, a 3-D sphere with a radius r and its center located at rx, has fully absorbing boundaries. This means that any molecule colliding with the sphere's surface is absorbed. The experiment shown in Figure 1.

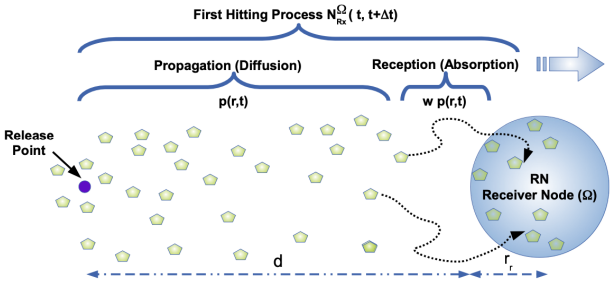

Figure 1: Representation of the 3D Molecular Communication Channel [3]

To simulate the process, we'll establish the initial positions of the particles using an emission point. The receiver, along with it's boundaries, will also be defined. For each small time step, the particles will be randomly moved, adhering to Brownian motion. This will involve generating Gaussian random numbers with a mean of 0 and a standard deviation of $\sqrt{2 \cdot D \cdot \Delta t}$. The simulation will continuously check if the particles are absorbed. Absorbed particles will be removed from the environment, while particles that remain unabsorbed will continue to move in each small time step until the simulation concludes.

Function `simulate`
- total_absorbed = 0;
- `initialize_particles()`;
- **for** *each time step* **do**
  - `move_particles()`;
  - `absorb_particles()`;
  - update timeline;
- **end**
- **return** simulation result;

Function `initialize_particles`
- Calculate sigma;
- Set initial molecule coordinates;
- Set receiver membrane boundary;

Function `move_particles`
- Generate random Gaussian(0, sigma) movement for each molecule;
- Update molecule positions based on movement;

Function `absorb_particles`
- Calculate distance of molecules from receiver;
- Create mask for particles outside the receiver;
- **for** *each particle outside the membrane* **do**
  - Increase total absorbed count;
  - Remove particles inside the receiver;
- **end**

Algorithm 1: Absorbing Sphere Diffusion Simulator in 3D Without Reflecting Line Pseudo Code

We expect our experimental results to converge with the analytic solution (Equation 1) for this problem, which provides the cumulative number of received molecules over time.

$$N^{R_x}(t) = N^{T_x} F(t) = N^{T_x} \frac{r_x}{r_x + d} \operatorname{erfc}\left(\frac{d}{\sqrt{4Dt}}\right) \quad (1)$$

Where r_x = radius of absorber, N^{T_x} = number of molecules, d = distance between absorber and emission point, D = Diffusion coefficient

3.2 Diffusion simulations in 2D with Reflecting Surface

In this section, our experimental setup and initial assumptions largely mimic the 3D environment scenario. However, we now transition to a 2D environment where a reflecting line is introduced. This is illustrated in Figure 2.

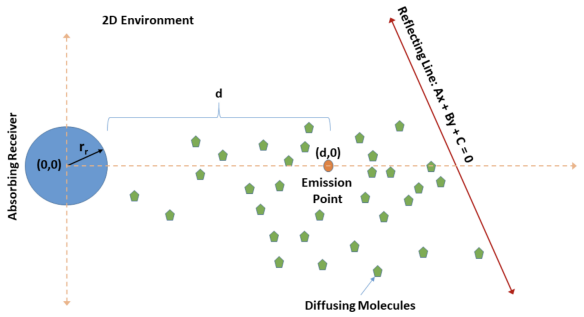

Figure 2: Representation of the 2D Molecular Communication Channel with Reflecting Line [3]

The simulation process largely same with the 3D case. We need to transition our parameters from 3D to 2D and modify particle movement from three dimensions to two. We also introduce a reflecting line defined by parameters A, B, and C.

Regarding the subsequent movements of particles colliding with the line, we'll consider two different strategies: the roll-back and reflection with respect to the line. In the roll-back strategy, particles revert to their previous positions, as if they hadn't hit the line. In the reflection with respect to the line strategy, for simplicity, particles are assumed to hit the line precisely between time t and t + ϵ (t + $\frac{\epsilon}{2}$) and are accordingly reflected.

After changing the simulation dimensions, we will add the 'reflect_particles' method to execute the 2D simulation with a reflecting line. As our time-frame must remain discrete, we'll determine if a particle has hit the line by checking the particle's positional sign with respect to the line equation (e.g., $Ax+By+C > 0$). Should this sign change after movement, it indicates a collision with the line, requiring a reflection based on the chosen strategy.

Function `simulate`
  total_absorbed = 0;
  `initialize_particles()`;
  for *each time step* **do**
    `move_particles()`;
    `reflect_particles()`;
    `absorb_particles()`;
    update timeline;
  end
  return simulation result;
Function `reflect_particles`
  for *each particle* **do**
    if `check_collision()` **then**
      Reflect according to given strategy
    end
  end
Function `check_collision`
  sign_before = Check position sign before movement with respect to line equation;
  sign_after = Check position sign after movement with respect to line equation;
  return sign_before * sign_after ¡ 0

Algorithm 2: Absorbing Sphere Diffusion Simulator in 2D With Reflecting Line Pseudo Code

So for reflection part, if strategy is roll_back the position of the particles that reflected after movement should be reverted to their previous positions.

$$(x, y) = (x_0, y_0) \tag{2}$$

And if strategy is reflection_wrt_line, assuming the hit point to the line is the position at t + $\epsilon/2$, the formula for the new position is

$$x = x_0 - \frac{2A(Ax_0 + By_0 + C)}{(A^2 + B^2)} \tag{3}$$

$$y = y_0 - \frac{2B(Ax_0 + By_0 + C)}{(A^2 + B^2)} \tag{4}$$

This equation simply reflect the particle at t + ϵ with respect to line. Where (x, y) = new position, (x_0, y_0) = old position, A, B, C = line parameters with equation $Ax + By + C = 0$

4 Numerical Results and Plots

The given algorithms have been implemented, and the results of the stochastic simulations with various parameters are provided.

4.1 3D Without Reflecting Line

```
Parameter Set Task1-1:
sim_params.rx_center = [0, 0, 0];
sim_params.rx_r_inMicroMeters = 5;
sim_params.rx_tx_distance = 5;
sim_params.tx_emission_pt = [10, 0, 0];
sim_params.D_inMicroMeterSqrPerSecond = 75;
sim_params.tend = 0.4;
sim_params.delta_t = 0.0001;
sim_params.num_molecules = 50000;
```

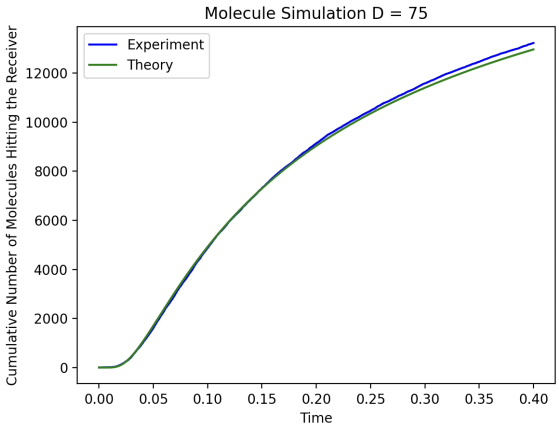

Figure 3: 3D Simulation with D = 75

```
Parameter Set Task1-2:
sim_params.rx_center = [0, 0, 0];
sim_params.rx_r_inMicroMeters = 5;
sim_params.rx_tx_distance = 5;
sim_params.tx_emission_pt = [10, 0, 0];
sim_params.D_inMicroMeterSqrPerSecond = 200;
sim_params.tend = 0.4;
sim_params.delta_t = 0.0001;
sim_params.num_molecules = 50000;
```

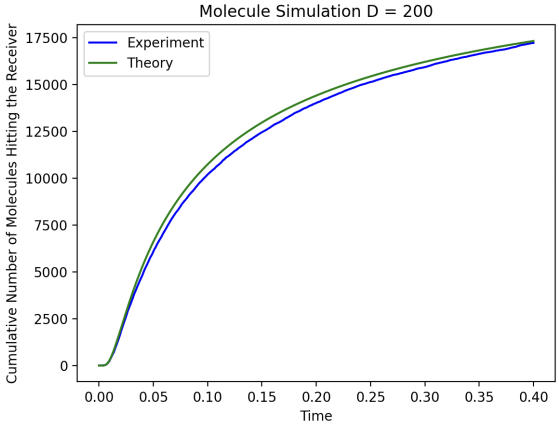

Figure 4: 3D Simulation with D = 200

In the graphs, it's evident that our experiments align with the theoretical values. Also it can be seen that as the diffusion coefficient increased, the number of absorbed molecules also increased.

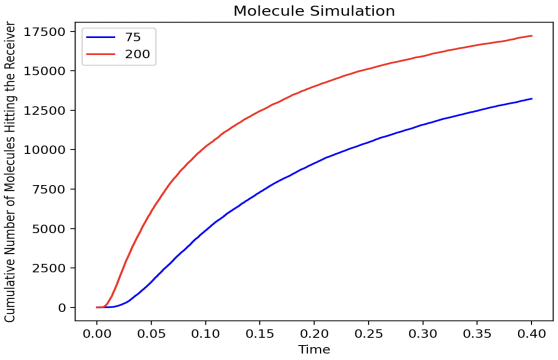

Figure 5: 3D Simulation with D = 75 and D = 200 together

In this figure, the results of two experiments are plotted. With the increase in the diffusion coefficient, the standard deviation in the random movement of particles also increases. Consequently, particles is able to reach the absorbing molecule faster. Eventually number of molecules absorbed will be same (given by Equation 1), but the one with higher diffusion coefficient will reach steady-state faster.

Plotting the values in Equation one gives

$$\lim_{t\to\infty} 50000 \cdot \frac{5}{5+5} \cdot erfc(\frac{d}{\sqrt{4Dt}})$$

$$\lim_{t\to\infty} erfc(\frac{d}{\sqrt{4Dt}}) = \lim_{t\to\infty} erfc(0) = 1$$

$$\lim_{t\to\infty} N^{R_x}(t) = 25000$$

4.2 2D With Reflecting Line

```
Parameter Set Task2-1:
sim_params.rx_center = [0, 0, 0];
sim_params.rx_r_inMicroMeters = 5;
sim_params.rx_tx_distance = 5;
sim_params.tx_emission_pt = [10, 0, 0];
sim_params.D_inMicroMeterSqrPerSecond = 75;
sim_params.reflecting_line_eqn_A = 1;
sim_params.reflecting_line_eqn_B = 1;
sim_params.reflecting_line_eqn_C = -15;
sim_params.reflection_strategy = 'roll-back';
sim_params.tend = 0.6;
sim_params.delta_t = 0.0001;
sim_params.num_molecules = 50000;
```

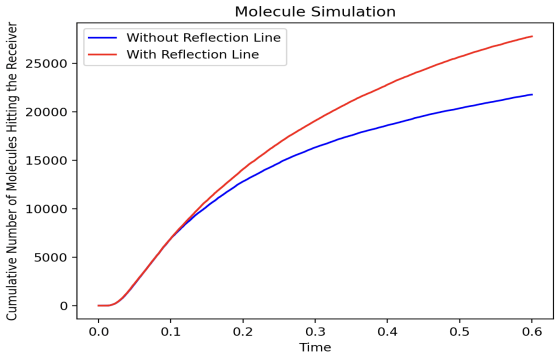

Figure 6: 2D Simulation with reflecting line and without reflecting line

```
Parameter Set Task2-2:
sim_params.rx_center = [0, 0, 0];
sim_params.rx_r_inMicroMeters = 5;
sim_params.rx_tx_distance = 5;
sim_params.tx_emission_pt = [10, 0, 0];
sim_params.D_inMicroMeterSqrPerSecond = 75;
sim_params.reflecting_line_eqn_A = 1;
sim_params.reflecting_line_eqn_B = 1;
sim_params.reflecting_line_eqn_C = -18;
sim_params.reflection_strategy = 'roll-back';
sim_params.tend = 0.6;
sim_params.delta_t = 0.0001;
sim_params.num_molecules = 50000;
```

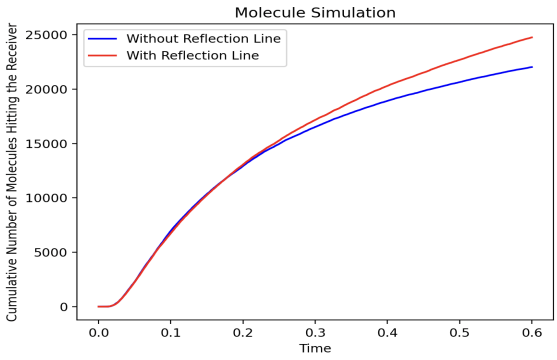

Figure 7: 2D Simulation with reflecting line and without reflecting line

```
Parameter Set Task2-3:
sim_params.rx_center = [0, 0, 0];
sim_params.rx_r_inMicroMeters = 5;
sim_params.rx_tx_distance = 5;
sim_params.tx_emission_pt = [10, 0, 0];
sim_params.D_inMicroMeterSqrPerSecond = 75;
sim_params.reflecting_line_eqn_A = 1;
sim_params.reflecting_line_eqn_B = 0;
sim_params.reflecting_line_eqn_C = -15;
sim_params.reflection_strategy = 'roll-back';
sim_params.tend = 0.6;
sim_params.delta_t = 0.0001;
sim_params.num_molecules = 50000;
```

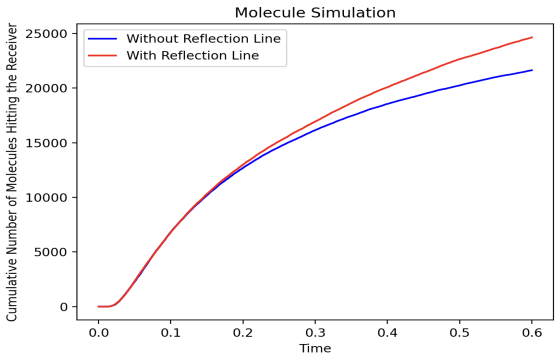

Figure 8: 2D Simulation with reflecting line and without reflecting line

With reflecting line number of absorbed molecules increased, because some of the molecules going to another direction are reflected.

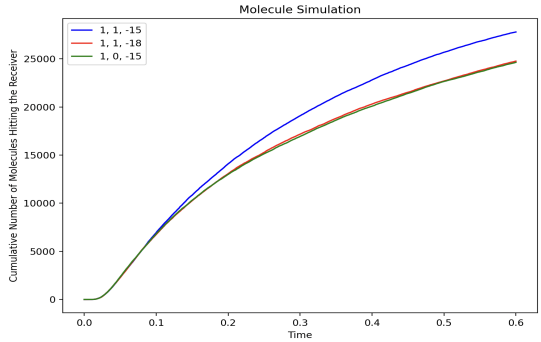

Figure 9: 2D Simulation with 3 different reflecting lines

And in this graph results of different reflecting lines with different parameters are shown. The position of reflecting line affects how fast molecules can be absorbed.

All of these simulations thus far have been conducted using the roll-back strategy, a mere approximation of reflection. Moving forward, we intend to perform simulations utilizing reflecting lines to observe the potential differences between these strategies and if they yield the same results.

```
Parameter Set Task3-1:
Same as Parameter Set Task2-1 except

sim_params.reflection_strategy =
'reflection_wrt_line';
```

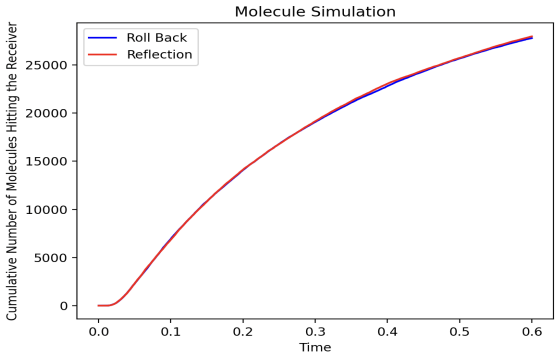

Figure 10: 2D Simulation with roll back and reflection wrt line strategy

```
Parameter Set Task3-2:
Same as Parameter Set Task2-2 except

sim_params.reflection_strategy =
'reflection_wrt_line';
```

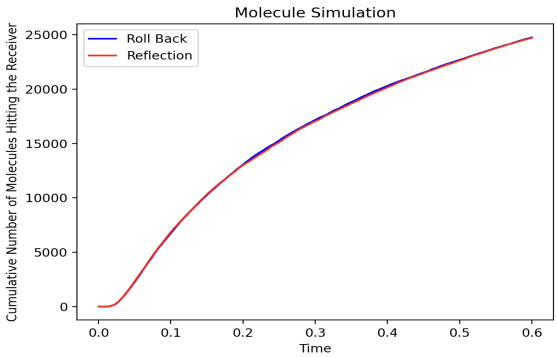

Figure 11: 2D Simulation with roll back and reflection wrt line strategy

```
Parameter Set Task3-3:
Same as Parameter Set Task2-3 except

sim_params.reflection_strategy =
'reflection_wrt_line';
```

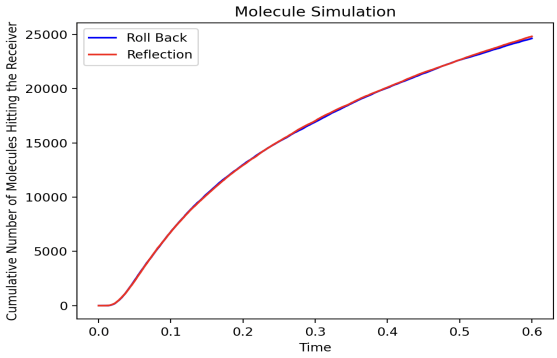

Figure 12: 2D Simulation with roll back and reflection wrt line strategy

As can be seen, the graphs depict that both the reflection and roll-back strategies yield similar results. While there are slight differences, they're well within the realm of expected randomness. We can conclude that, in this experiment, both the roll-back and reflection with respect to line strategies essentially produce the same outcome.

References

[1] M. S. Kuran, H. B. Yilmaz, T. Tugcu, and B. Ozerman, "Energy model for communication via diffusion in nanonetworks," Nano Comm. Networks, vol. 1, no. 2, pp. 86–95, 2010.

[2] H. B. Yilmaz, A. C. Heren, T. Tugcu, and C.-B. Chae, "Threedimensional channel characteristics for molecular communications with an absorbing receiver," IEEE Commun. Lett., vol. 18, no. 6, pp. 929–932, Jun. 2014.

[3] H. B. Yilmaz, (2023, October). Particle Based Simulations [Lecture slides]. Faculty of Computer Engineering, Bogazici University.