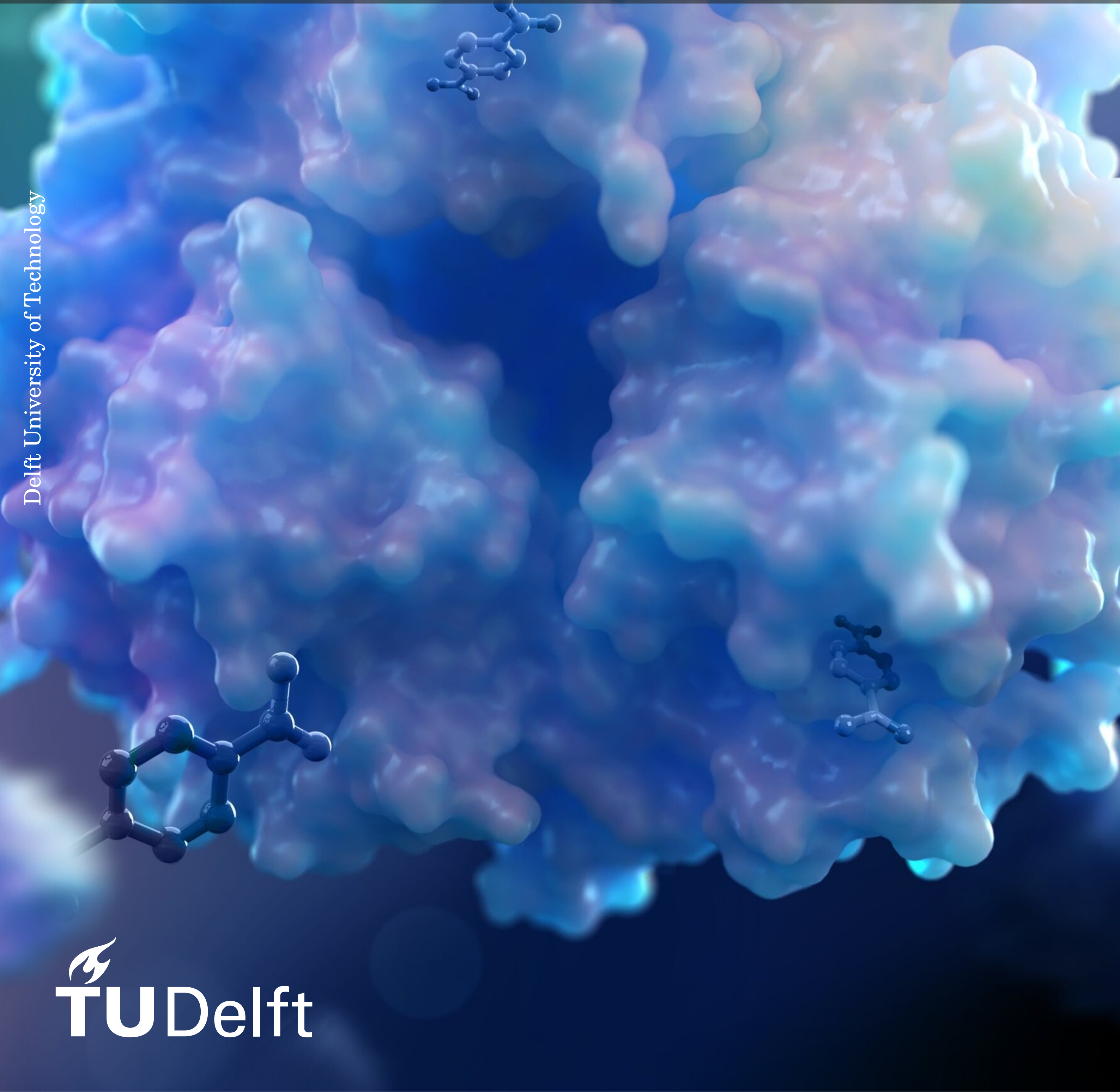
Cross diffusion in single-step enzymatic reactions

Bachelor's Thesis

Samuel Zegers

Delft University of Technology



TU Delft

Cross diffusion in single-step enzymatic reactions

by

Samuel Zegers

Student Name	Student Number
Samuel Zegers	4997654

Instructors: A.B.T. Barbaro, C. R. Kleijn
Faculty: Faculty of Applied Sciences and
Faculty of Electrical Engineering, Mathematics and Computer Sciences,
Delft, The Netherlands

Cover: The TPADO enzyme. Copyright: Rita Clare, Montana State University

Summary

In 2010, Sen et al. found an anomaly in the 'random' movement [18]. They found that in the presence of the reactant particles, a catalyst protein they were studying called *Urease* moved faster by about 28%. The first theories that emerged to explain this phenomenon were based on the exothermicity of the reactions [20, 11]. This explanation did not explain the anomalous movement of endothermic enzymes or movement toward reactant particels. In 2018 however, Canalejo et al. developed a model which could both explain enhanced diffusion and unify movement towards or away from the reactants [1]. In this report, we have derived an agent-based model for the diffusion of enzymes, their substrate and their product, which bases itself on interactions described by Canalejo et al. [1]. However, the model in this report is at the particle level, which means that the stochasticity is not lost. The model showed a phase transition, dependent on the critical parameter β. Using the model, the diffusion coefficient of urease enzymes undergoing repellent interactions with urea was calculated using a mean square distance method. The parameters used in the simulations to determine the diffusion coefficient were: $l = 7.43 \cdot 10^{-3}\,\text{cm}$, $C_s = 1$, $C_e = 43.4$, $k_{cat} = 5913\text{s}^{-1}$, $\gamma_s = 1\text{s}^{-1}$, $K_m = 4.2\,\text{mM}$. This resulted in an enhanced diffusion coefficient of $D' = 4.36 \cdot 10^{-7}\,\frac{\text{cm}^2}{\text{s}}$, a 37% increase from the regular diffusion coefficient. The pattern formation provides an experimental criterion to check the model against.

The continuum limit of this model was derived as well. The equilibrium solutions of the continuum model were calculated and checked against the discrete model. A linear stability analysis was also done to calculate the critical parameter β for which pattern formation occurred.

Contents

1

Introduction

The cells in our bodies work tirelessly day and night to provide us with the organic compounds we need to function. During the day, chemicals are being broken down, built up and stored all across our cells. The stars of these processes are the *proteins*: macro-molecules who accommodate almost all the chemistry in our cells and transport their subsequent products to useful locations. Proteins are therefore one of the main interests in modern molecular biology. A particularly interesting question is one regarding the movement of catalysing proteins: how are they moving? The answer to this question would not only help us understand more of the molecular biology around us, but it would also help in the engineering of nanomotors: a nanoscale particle able to move on its own by converting available energy into motion [23]. Because the clear application of nanomotors is in delivery of medicine, the straightforward blueprint is in our own cells. The first hurdle to overcome was to explain the *how* of the movement. Since our cells are low-Reynolds number environments, which means highly viscous conditions, it was necessary to study what constraints this put on the possible movement. In his influential 1977 paper *Life at Low Reynolds Number*, Edward Purcell derived two effects of the highly viscous environment on the possibility of movement. First of all, he showed that it was necessary for the movement to be non-reciprocal [19]. The example he gave to illustrate this was the clam: opening up will move the clam forward, but closing up will take it right back to its initial location. Secondly, he showed that no inertia was carried by movement in such an environment: the clam cannot simply continue drifting through the medium after opening up. This second criterion implies that each time a protein is propelled forward, it is as if it takes a single step. Combined with the fact the protein randomly rotates, Purcell could explain the proteins' random movement.

In 2010 however, Sen et al. found an anomaly in the 'random' movement [18]. They found that in the presence of the reactant particles, a catalyst protein they were studying called *Urease* moved faster by about 28%. Furthermore, they found that the amount the movement was enhanced followed the rate of catalysis. The first theories that emerged to explain this phenomenon were based on the exothermicity of the reactions [20, 11]. These theories argued that the heat released by exothermic reactions propelled the enzymes further than usual. This of course explained the dependence on the concentration of the reactants. After all, more reactions cause more heat to be generated and thus cause more movement. However, the dependence on exothermicity was disproved in 2017 when Illien et al. showed that an endothermic protein still showed enhanced movement on the order of 30% [14].

The question now concerned the *where* of the movement, since the new competing theories argued on whether the proteins move away from or towards the reactants. In 2018 Canalejo et al. developed a model which could both explain enhanced diffusion and unify movement towards or away from the reactants [1]. Their model attributed the movement of the proteins to interactions that they undergo with the reactants. This also met the criterion of non-reciprocity easily, since it is entirely dependent on the protein's interactions with its environment. In 2020 Giunta et al. analysed the predicted distributions numerically in a 1d setting [10]. The paper by Giunta et al. showed that pattern formation occurred for certain transport models due to feedback from the reaction, which separated the proteins and reactants. This raised the question if such pattern formation would also occur in 2D, reminiscent of territorial models such as the one developed by Alsenafi et al. in 2018 [3].

In this report, cross-diffusion in enzymatic reactions will be analysed in 3 steps. First of all, a theoretical background will be given in section 2 on the catalyst proteins and the partial differential equation that is used to model random movement through a medium. The parameter of interest, the *diffusion coefficient*, which quantifies the movement speed will also be introduced. The partial differential equation will then be extended by adding the terms suggested by Canalejo et al. [1]. Next, a novel on-lattice particle model will be developed in section 3, based on the work by Alsenafi et al [4]. This is a particle based model where, instead of looking at the ensemble, the stochasticity of each molecule considered individually. After the model has been derived, simulations will be run with real-life parameters and the results will be compared to the anomalous increase in diffusion coefficient measured in experiments by Sen et al. and Ilien et al [11, 14] in chapter 4. Ultimately, the continuum limit of the lattice model will be derived in section 5. The continuum limit of the lattice model will be compared to the differential equations derived by Canalejo et al. and a linear stability analysis will be done to predict the threshold magnitude of the interactions at which pattern formation begins to occur

2
Theoretical Basis

In this section we will introduce the theoretical basis for the rest of this report. We will start off by introducing enzymes and explaining how they work. After that, diffusion will be discussed. Consequently, the theory of diffusion will extended using novel methods regarding interactions between diffusive particles.

2.1. Enzymes

As mentioned in the introduction, proteins are vital to all life on earth. Over billions of years of evolution they have evolved into the most complex organic molecules on earth, performing some of the most intricate processes in our bodies. They transcribe our genetic code, transport organelles through our cells and catalyse many chemical processes [2]. The vast subclass of protein catalysts are called enzymes and are the most effective catalysts known today. The effectiveness of enzymes, and proteins in general, can be attributed to the fact that they are dynamic molecules. Their shape changes due to chemical events, which means they both influence and are influenced by the chemistry around them. Therefore, the enzyme's interaction with other molecules around it establishes its properties. The general idea of what enzymes do to molecules is pretty straightforward: they bind to each other. In the case that a molecule is bound to an enzyme it is called a ligand. Enzymes are extremely selective in which ligands are allowed. Which molecules can and cannot bond depends on a unique combination of electrostatic interactions, hydrogen bonds, Van Der Waals bonds and interactions based on polarity. Once a suitable molecule is encountered by an enzyme, all the aforementioned interactions cause the ligand to bond non-covalently to the enzyme in a cavity called the *binding site*. As soon as one or more ligands are bonded, the enzyme kick-starts chemical reactions that convert substrate into a product. The reason for their high conversion rate is twofold. First of all, they simply increase the concentration of the substrate near the binding site while also holding them in the right place for the reaction to occur. But, more importantly, enzymes lower the activation energy of the chemical reaction that the substrate undergoes. During the reaction the substrate molecule's shape and electron distribution changes until it has reached an unstable state called the *transition state*. The enzyme binds tighter to this transition state, which lowers the activation energy. Consequently,

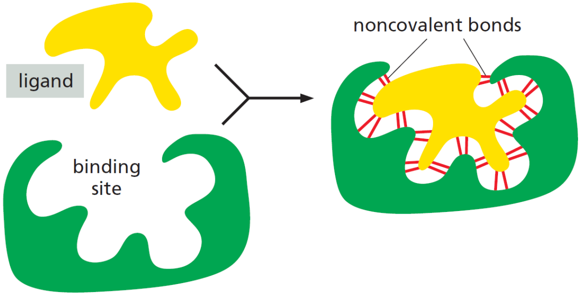


Figure 2.1: A graphic showing ligand engaging in bonds with an enzyme. *Microbiology of the Cell, Alberts* [2].

this energy is more easily reached and the reaction is sped up. In total, the general reaction path can be written as:

$$E + S \rightarrow ES \rightarrow EP \rightarrow E + P. \tag{2.1}$$

where E is the enzyme, S the substrate and P the product.

2.1.1. Michaelis-Menten kinetics

Since enzymes are very important in many biochemical applications, such as industrial-scale production of chemicals, it is valuable to know how enzymes react to different chemical environments. In particular, it is useful to know what influences the production rate, both qualitatively and quantatively. In this report we will be considering single-step enzymatic reactions. That is, only one substrate is bound to the enzyme and subsequently converted into product. This interaction can be written as follows:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_{cat}} E + P \tag{2.2}$$

where k_1 and k_{-1} are the rates of binding and release. k_{cat} is then the rate at which the substrate is converted into the product and decoupled. The rate, V, of this reaction is assumed to be linearly dependent on k_{cat} and $[ES]$ by the law of mass action:

$$V = k_{cat}[ES] \tag{2.3}$$

where $[ES]$ is the concentration of the enzyme-substrate complex. This assumption holds since we assume that the medium allows free diffusion. Next, we would like to write $[ES]$ in terms of k_1, k_{-1} and the concentrations of the enzyme and substrate. To do this we will look at a steady state situation, in terms of $[ES]$. We may analyse the system as steady state if we assume that the substrate concentration is significantly higher than the enzyme concentration and that $E + P$ rarely ever reverts back to ES and subsequently $E + S$. Under the assumption of steady state in $[ES]$ it holds that:

$$[E] = [E_0] - [ES] \tag{2.4}$$

so, the free enzyme concentration is equal to difference in initial enzyme concentration and enzyme-substrate concentration. This means that the rate of ES break-

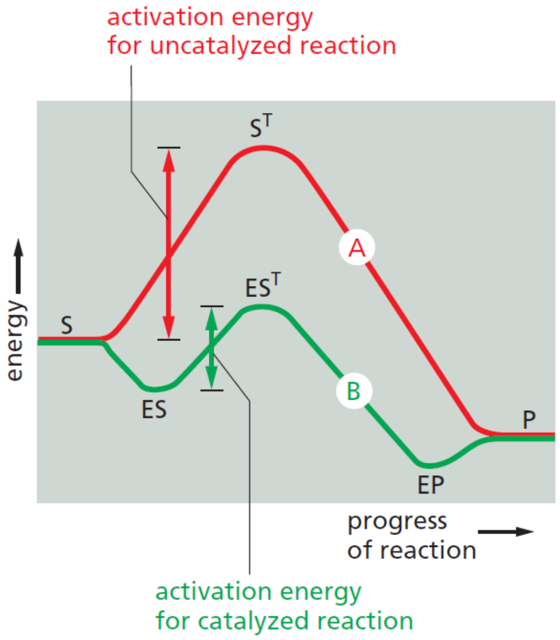


Figure 2.2: An illustration of the enzyme's effect on activation energy. Due to the fact that the substrate is bound tight to the enzyme, ES is a lower energy state than S. The substrate therefore readily engages with the enzyme. Furthermore, the transition state ES^T is also lower in energy due to the reasons mentioned in the text. The energy released during the process uncouples the product and the enzyme.
Microbiology of the Cell, Alberts [2].

down matches the rate of ES creation. This can be written as:

$$[ES] = \frac{k_1}{k_{-1} + k_{cat}}[E][S] = \frac{k_1}{k_{-1} + k_{cat}}([E_0] - [ES])[S] \tag{2.5}$$

If we write $K_m = \frac{k_{-1}+k_{cat}}{k_1}$,then:

$$[ES] = \frac{1}{K_m}[E][S] = \frac{1}{K_m}([E_0] - [ES])[S] \tag{2.6}$$

Then this can be reduced to:

$$[ES] = \frac{[E_0][S]}{K_m + [S]} \tag{2.7}$$

Hence, from equation 2.3 we get the Michaelis-Menten formula for V:

$$\begin{aligned} V &= \frac{k_{cat}[E_0][S]}{K_m + [S]} \\ &= \frac{k_{cat}[E_0]}{1 + \frac{K_m}{[S]}} \end{aligned} \tag{2.8}$$

At first glance, we see that this equation predicts that if S increases, V approaches a maximum: $V_{max} = k_{cat}[E_0]$. Furthermore, if $\frac{K_m}{[S]}$ is very small, $\frac{1}{1+\frac{K_m}{[S]}} \approx 1$ and

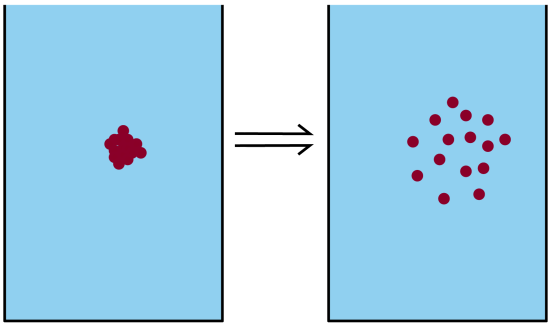


Figure 2.3: A diffusive process. We see that just by drifting away from the centre, the particles move from a region of high concentration to low concentration. Eventually this produces an equilibrium state of even distribution in the container.

$V \approx V_{max}$. If $[S]$ is very big, the substrate supply is high and the reaction speed is not limited by the substrate. Therefore, we expect the reaction to approach V_{max}. However $V \approx V_{max}$ when K_m is very small as well. This implies that K_m tells us something about how tight the substrate binds to the enzyme, since in section 2.1 it was explained that the tightness of binding is intrinsically linked to a decrease in activation energy and thus an increase in reaction speed.

2.2. Enhanced Diffusion

When preparing a cup of tea, we usually start shaking the teabag as soon as it hits the warm water. It just mixes better that way. However, you could just as well keep the bag still. Somehow, the particles spread on their own from a region of high concentration, the teabag, to a region of low concentration, the rest of the cup. This phenomenon is what we call diffusion. The first step in deriving a mathematical description of this phenomenon is looking at the teabag just after it has been immersed and the water has settled again. Inside the teabag the concentration of particles is very high, whereas just outside the teabag it must be low. At this point of time then, there is a significantly higher chance of a particle leaving the teabag than one entering it. In other words, the flux through the teabag is proportional to the difference in concentration between the outside and the inside, pointing down the gradient. This observation yields us the following expression:

$$J = -D\nabla\phi \tag{2.9}$$

where J denotes the flux, ϕ the concentration and D a proportionality constant called the *diffusivity*. Of course, the overall total concentration of tea particles must stay the same throughout our system. Therefore, the change in concentration at the teabag must equal the out or inflow through the teabag, which gives that:

$$\frac{\partial\phi}{\partial t} = -\nabla\cdot(D\nabla\phi)\,. \tag{2.10}$$

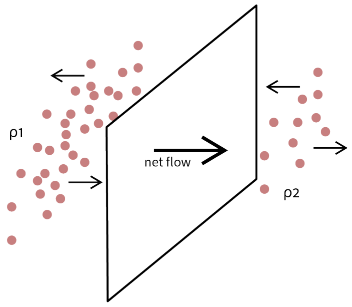


Figure 2.4: A visualisation of a concentration difference driving the flux through a surface. The fact that there are more particles to the left of the surface, means that there is a higher probability of a particle moving from left to right. This creates a net flow towards the right. In other words, the magnitude of the flux is dependent on the concentration difference and its direction points towards low concentration.

In case the diffusivity is a constant, this relation may be written as:

$$\frac{\partial \phi}{\partial t} = -D\nabla^2 \phi. \tag{2.11}$$

Therefore, by considering the flux of the concentration and conservation of mass, it is possible to derive an equation which can describe the motion of the tea particles. The same equation 2.11, called Fick's second law, can be used to describe the motion of enzymes in a medium [13]. However, as is mentioned in the introduction, it was found that the diffusion coefficient of certain enzymes such as Urease change in the presence of their substrate. This means that equation 2.11 does not describe all enzyme behaviour, since equation 2.9 does not take substrate dependency into account. Furthermore, the substrate reacting into the product should also be included into the equations. Consider the following: the enzyme is not freely dissolved in the water on its own, which means it can interact with its environment. First of all, it encounters various *non-specific interactions*, which include any attractive or repulsive forces that do not specifically play a role in ligand binding. They include but are not restricted to van der Waals, steric and electrostatic interactions. Besides non-specific interactions, it also encounters *specific interactions*, which are crucial in ligand binding at the binding site. These specific interactions are usually hydrogen bonds, hydrophobic interactions and temporary covalent bonds, which are able to change the shape of the enzyme [1].

2.2.1. Non-specific interactions

The effect of non-specific interactions on diffusion can be described by diffusiophoresis: the diffusive motion of a colloidal particle in response to the concentration gradient of molecular solute due to surface interactions [15], [7]. In our case, the colloidal particle and the molecular solute are the enzyme and substrate respectively. The substrate is several orders of magnitude smaller than the enzyme. Near the enzyme's surface the system can therefore be interpreted as a flat plate with the substrate molecules experiencing a potential $\phi(y)$, where y is the distance from the plate. The potential causes a force to be exerted on the substrate molecules. This force causes the substrate molecules to either move towards the surface in case of an attractive potential, or to move away from the surface in case of a repellent potential. The migration of substrate molecules causes a pressure gradient to appear near the enzymes surface. This in turn generates a drift velocity for the enzyme:

$$\mathbf{v} = \frac{k_b T}{\eta}\left[\int_0^\infty y\left(e^{-\phi(y)/k_b T} - 1\right) dy\right] \nabla \rho_s, \tag{2.12}$$

where η is the viscosity of the medium, k_b the Boltzmann constant, T the temperature, $\phi(y)$ the potential of the non-specific interactions and ρ_s the concentration of the substrate. The term inside the square brackets is often notated as λ^2 and called the *Derjaguin length*. The Derjaguin length characterises the distance at which non-specific interactions play a role and is typically on the order of several Angström. A more in-depth derivation of equation (2.12) is provided in appendix E. If we now combine this new velocity term with equation 2.9 we get the new modified flux term in equation 2.13.

$$J' = -D\nabla\rho_e + \frac{k_b T}{\eta}\lambda^2 \nabla\rho_s \tag{2.13}$$

2.2.2. Specific interactions

The effect of specific interactions comes from incorporating the different diffusion coefficients of the bound and free enzyme, and Michaelis-Menten kinetics. If we assume that the binding interactions occur at very short range and that the rate of binding and unbinding are again k_1 and k_{-1} respectively, then we get the coupled equations 2.14, 2.15 for the free and bound enzyme populations ρ_{e_f} and ρ_{e_b}:

$$\frac{\partial \rho_{e_f}}{\partial t} = \nabla \cdot \left[-D_f \nabla \rho_{e_f} + \frac{k_b T}{\eta}\lambda_f^2 \nabla s\right] - k_1 \rho_{e_f}\rho_s + k_{-1}\rho_{e_c}, \tag{2.14}$$

$$\frac{\partial \rho_{e_b}}{\partial t} = \nabla \cdot \left[-D_b \nabla \rho e_c + \frac{k_b T}{\eta}\lambda_b^2 \nabla s\right] + k_1 \rho_{e_f}\rho_s - k_{-1}\rho_{e_c}. \tag{2.15}$$

Now, recall from section 2.1.1 that we may apply the Michaelis-Menten equations only if the system is in a local steady state. Although the enzymes move due to diffusion, its timescale is significantly longer than the rate of binding and unbinding. Therefore, we assume that before the enzyme moves, its enzyme kinetics have reached the Michaelis-Menten regime. Thus, we apply $k_1\rho_{e_f}\rho_s \approx k_{-1}\rho_{e_c}$ to find the typical relations between the total enzyme population, ρ_e, and the bound or free enzyme populations:

$$\rho_{e_f} = \frac{K_m}{K_m + \rho_s}\rho_e \quad \text{and} \quad \rho_{e_b} = \frac{\rho_s}{K_m + \rho_s}\rho_e \tag{2.16}$$

Using the Michaelis-Menten relations on 2.14 and 2.15, we find that:

$$\rho_e = \rho_{e_f} + \rho_{e_b} = \nabla \cdot \left[-D\nabla\rho_e + (V_{ph} + V_{bi})\rho_e\right], \tag{2.17}$$

Where V_{ph} is the diffusiophoresis term due to non-specific interactions:

$$V_{ph} = \left(\lambda_f^2 + (\lambda_b^2 - \lambda_f^2) \frac{\rho_s}{K_m + \rho_s} \right) \frac{k_b T}{\eta} \nabla \rho_s, \tag{2.18}$$

and V_{bi} is the drift velocity due to the specific binding and unbinding interactions:

$$V_{bi} = (\lambda_b^2 - \lambda_f^2) \nabla \left[\frac{s}{K_m + \rho_s} \right]. \tag{2.19}$$

In total, 2.17, 2.18 and 2.19 will describe our system. Of note is of course the dependence of equation 2.17 on the gradient of the substrate. This means that when taking into account more complex interactions, the resulting differential equation describes a substrate gradient dependent movement of the enzyme. These equations and their implications are the foundation for the discrete model that will be described in this report.

3
Discrete Model

Now that the theoretical basis has been discussed, the mathematical model will be discussed. The system in equation (2.17) has so far only been evaluated in 1D at the distribution scale. However, a 2D model at the particle scale will be derived in this report. The model will be constructed in three steps, where the first two steps are analogous to how the model is setup in [3]. That is, the initial conditions of the enzymes, substrate and product will be discussed first. After the setting is clear, the movement dynamics will be explained. Lastly, the novel reaction and catalysis step from substrate to product will be addressed.

3.1. The Lattice and Initialisation

We start by defining the lattice where all movement and reactions will be taking place. To that end, let G be an $L \times L$ rectangular grid lattice with periodic boundary conditions and spatial step $l = \frac{1}{L}$ such that $x, y \in \{0, l, 2l, ..., \frac{1-l^2}{l}\}$. Furthermore, the simulations will start at $t = 0$ and time will increase in discrete increments of δt. On the lattice, there are three types of particles present: substrate, enzymes and product. They are denoted by S, E and P, respectively. Let the amount of substrate, enzyme and product at a certain location and time (x, y, t) then be designated by $s(x, y, t)$, $e(x, y, t)$ and $p(x, y, t)$. Therefore, the total amount of particles in the system at time t, which will be referred to as $N(t)$, is $N(t) = \sum_{(x,y)} s(x, y, t) + e(x, y, t) + p(x, y, t)$. At each new instance of the simulation, the substrate and enzyme will be randomly distributed over the lattice using a multinomial distribution. Furthermore, the simulation will start with no product on the lattice at time $t = 0$. For each location (x, y), there is no exclusion of particles. This means that multiple particles can be in the same spot together. This is necessary, since reactions between substrate and enzyme to product are only only defined for when they are at the same location, of course. Finally, let us introduce the notation $(\tilde{x}, \tilde{y})$ as all sites neighbouring (x, y)

3.2. Movement Dynamics

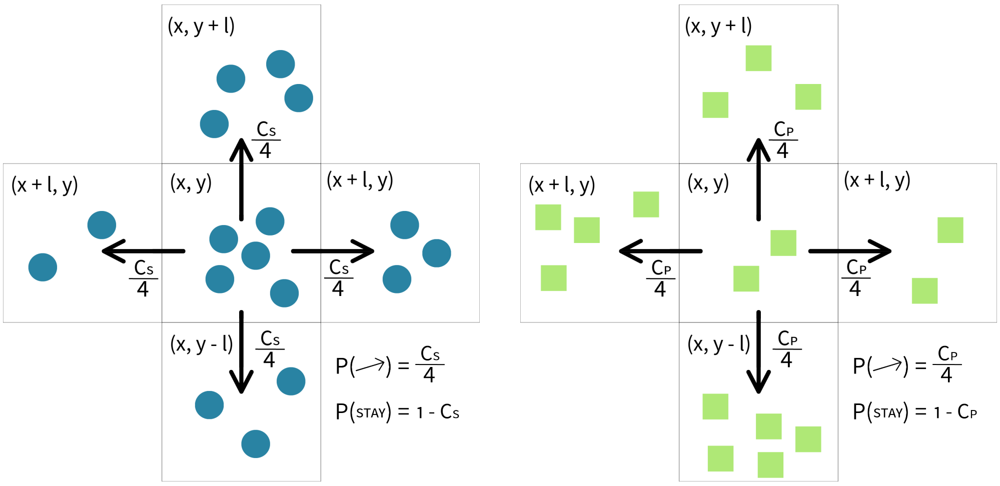
As discussed in the theory section, the particles diffuse on the lattice over time. This will be modeled using a random walk. A random walk is the obvious choice, since it describes the dynamics of a particle randomly moving over the lattice. A set of random walks thus describes group of particles randomly displacing through a medium which is how diffusion was characterised in section 2.2. During a random walk, a particle must move each time step from (x, y) to one of its neighbours $\{(x + l, y), (x - l, y), (x, y + l), (x, y - l)\}$ or stay at its current position. Each of these possibilities has their own respective probability of occurrence between t and $t + \delta t$. The substrate and product undergo unbiased random walks. Therefore, for the substrate and product the probability of movement to $(x_2, y_2) \in (\tilde{x}, \tilde{y}) \sim (x_1, y_1)$ is defined as:

$$\begin{aligned} \mathbb{P}_s(x_1 \to x_2, y_1 \to y_2, t) &= C_s/4 \\ \mathbb{P}_p(x_1 \to x_2, y_1 \to y_2, t) &= C_p/4 \end{aligned} \tag{3.1}$$

where $C_s, C_p \in [0, 1]$, such that:

$$\begin{aligned} \mathbb{P}_s(\{\text{No movement}\}) &= 1 - C_s \\ \mathbb{P}_p(\{\text{No movement}\}) &= 1 - C_p \end{aligned} \tag{3.2}$$

The enzyme's movement, on the other hand, is influenced by the amount of substrate at the neighbouring sites. Indeed, as mentioned in section 2.2 both specific and nonspecific interactions create diffusion terms dependent on $s(x, y, t)$ Therefore, this probability must be dependent on the substrate density. We would like to keep the property of maximum entropy increase each time step from the unbiased random walk, since entropy increase is what drives diffusion in the first place. This leads us to define the following:



(a) Substrate molecules. **(b)** Product molecules.

Figure 3.1: An visual representation of the movement probabilities as defined in equation (3.1). We see that the probability of moving towards a neighbouring site is not dependent on the concentration of other compounds.

$$\mathbb{P}_e(x_1 \to x_2, y_1 \to y_2, t) = C_e \frac{e^{-\frac{\beta}{L^2} s(x_2, y_2, t)}}{\sum_{(\tilde{x}, \tilde{y}) \sim (x_1, y_1)} e^{-\frac{\beta}{L^2} s((\tilde{x}, \tilde{y}), t)}}, \tag{3.3}$$

where we once again take $C_e \in [0,1]$, such that:

$$\mathbb{P}_e(\{\text{No movement}\}) = 1 - C_e \tag{3.4}$$

This is the Boltzmann distribution, well-known from statistical mechanics. Which possesses exactly the property of maximum entropy we desire. In equation (3.3), β is constant where $\beta \in \mathbb{R}$. If $\beta > 0$, then the interaction between the enzyme and the substrate causes the enzyme to drift away from an increasing substrate gradient. If $\beta < 0$, the enzymes will move towards an increasing substrate gradient. From these definitions, we also see that the probability of any of the possibilites happening each time step is

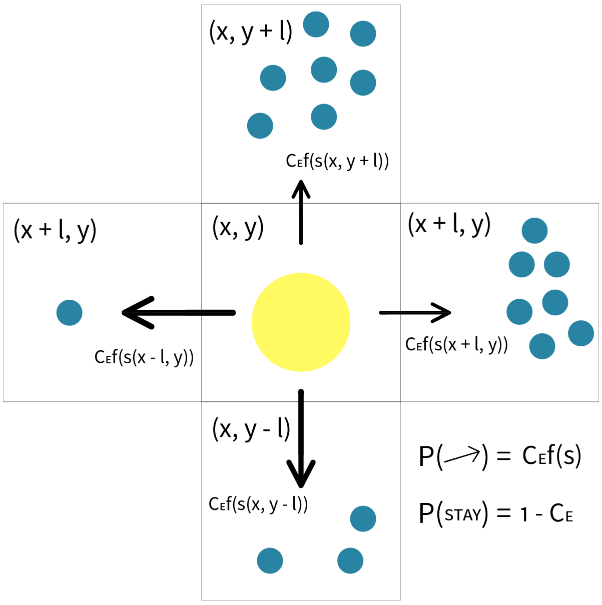


Figure 3.2: A visual representation of the enzyme's movement probability. In this case, $f(x,y,t)$ corresponds to the fraction in equation (3.3) and $\beta < 0$. The arrows pointing towards the locations with less substrate are bigger, indicating that the probability of moving depends on the substrate concentration ath that particular site.

$$\mathbb{P}_{s,e,p}(x_1 \to \tilde{x}, y_1 \to \tilde{y}, t) + \mathbb{P}_{s,e,p}(\{\text{No movement}\}) = C_{s,e,p} + \sum_{(\tilde{x},\tilde{y})\sim(x,y)} \mathbb{P}_{s,e,p} = 1 \tag{3.5}$$

which makes $\mathbb{P}_s$, $\mathbb{P}_e$ and $\mathbb{P}_e$ movement probabilities.

3.3. Reactions

In the theoretical toolbox we discussed in section 2.1.1, Michaelis-Menten kinetics were explained. From this framework, the enzymatic reaction will be modeled. Consider an arbitrary location, (x,y) at time t on G. For an adequate timestep, $k_{cat}\delta t < 1$, the bonding and subsequent reaction to the enzyme can be interpreted as a Bernouilli random variable with reacting probability

$$\mathbb{P}_{\text{r}} = \delta t \cdot k_{cat} \frac{s(x,y)}{K_m + s(x,y)} \tag{3.6}$$

Indeed, for the aforementioned criterion the enzyme either converts a substrate particle between t and $t + \delta t$ or not. Therefore, since the amount of enzymes at our coordinate (x, y, t) is given by $e(x, y, t)$ and each reaction is modeled as a Bernoulli random variable.

Theorem 3.3.1. *Let $X_1, X_2, ..., X_N$ be independent identically distributed Bernoulli random variables, such that $\mathbb{P}(\{X_i = 1\}) = p$ and $\mathbb{P}(\{X_i = 0\}) = 1 - p$, then $\sum_{i=1}^{N} X_i$ is Binomially distributed with parameters N and p. We write: $\sum_{i=1}^{N} X_i \sim \mathbf{Bin}(N, p)$*

Proof. We will prove the theorem by showing that the characteristic functions match and applying the uniqueness theorem. Recall that the characteristic function of one of the Bernoulli variables is:

$$\mathbb{E}(e^{itX_r}) = (1-p)e^{it\cdot 0} + pe^{it\cdot 1} = (1-p) + pe^{it}. \tag{3.7}$$

Next, we will use independence to derive the characteristic function of the sum:

$$\begin{aligned} \mathbb{E}(e^{it\sum_{r=1}^{N} X_r}) &= \prod_{r=1}^{N} \mathbb{E}(e^{itX_r}) \\ &= \left[(1-p) + pe^{it}\right]^N \\ &= \sum_{r=1}^{N} e^{itr} \left[\binom{N}{r} (1-p)^{N-r} (p)^r\right] \end{aligned} \tag{3.8}$$

Notice that the term in between the square brackets is the probability mass function of a Binomial random variable. By the expectation rule, we see that the entire expression may thus be written as $\mathbb{E}(e^{itY})$, where $Y \sim \mathbf{Bin}(N, p)$ Therefore, by the uniqueness of the characteristic function, we have shown that $\sum_{i=1}^{N} X_i \sim \mathbf{Bin}(N, p)$. □

We can now deduce that at each coordinate (x, y, t) the number of reactions R is distributed as $R \sim \mathbf{Bin}(e(x, y, t), k_{cat}\delta t \frac{s(x,y)}{K_m + s(x,y)})$. From the property that the binomial distribution is a sum of independent identically distributed Bernoulli random variables, we can determine its expected value quickly by noting that:

$$\begin{aligned} \mathbb{E}(R) &= \prod_{i=1}^{e(x,y,t)} k_{cat}\delta t \frac{s(x, y)}{K_m + s(x, y)} \\ &= \delta t \cdot k_{cat} e(x, y, t) \frac{s(x, y, t)}{K_m + s(x, y)} \end{aligned} \tag{3.9}$$

It follows that the expected value of the product at coordinate $(x, y, t + \delta t)$ is given by

$$\begin{aligned} p(x, y, t + \delta t) &= p(x, y, t) + \mathbb{E}(R) \\ &= p(x, y, t) + \delta t \cdot k_{cat} e(x, y, t) \frac{s(x, y, t)}{K_m + s(x, y)} \end{aligned} \tag{3.10}$$

and conversely the expected value of the substrate at time $t + \delta t$ then turns out to be

$$\begin{aligned} s(x,y,t+\delta t) &= s(x,y,t) - \mathbb{E}(R) \\ &= s(x,y,t) - \delta t \cdot k_{cat} e(x,y,t) \frac{s(x,y,t)}{K_m + s(x,y,t)} \end{aligned} \tag{3.11}$$

3.4. Substrate and Product transport

Since we assume Michaelis-Menten interactions, there must be a significant inflow of substrate. Otherwise it is not accurate to describe the system as steady-state. After all, the substrate concentration is then depleted over time by the reaction. In real-life experiments this is achieved by adding a substrate reservoir, which is separated from the reaction chamber by a semi-permeable membrane and kept at a constant substrate concentration. A semi-permeable is a membrane which only permits transport of specific molecules by osmosis. In our case, that would mean that the membrane would permit transport of substrate and product, but not of the enzyme. In practice, this would be realised by making the holes in the membrane too small for the enzyme to pass through. As mentioned in section 2.2.1, the enzyme is a macro-molecule which is several orders of magnitude bigger than both substrate and product. Consider a lattice site (x, y) at time t. In between t and $t + \delta t$, two processes take place: substrate is transported from the reservoir into the reaction chamber and vice versa. Therefore, to model this behaviour mathematically, for each arbitrary lattice location (x, y) at time t we would like to have a random variable, which is dependent on the substrate concentration in the reaction chamber $s(x, y, t)$, the substrate concentration in the reservoir s_R and an osmosis rate per particle γ_s. Again, each transition per particle can be seen as a Bernoulli event with probability $\delta t \cdot \gamma_s$. Therefore, both the inflow and outflow are sums of Bernoulli random variables, comparable to how reactions were treated in the previous section. Therefore, the inflow and outflow are distributed as $I_s \sim \mathbf{Bin}(s_R, \delta t \cdot \gamma_s)$ and $O_s \sim \mathbf{Bin}(s(x, y, t), \delta t \cdot \gamma_s)$ respectively.
The net flow of substrate out of (x, y, t) is then $F_s = O_s - I_s$. This allows us to calculate the expected value of s due to F_s at time $t + \delta t$, using the linearity of the expectation operator and the same method as in equation 3.9, such that:

$$s(x,y,t+\delta t) = s(x,y,t) - \delta t \cdot \gamma_s \left(s(x,y,t) - s_r \right). \tag{3.12}$$

Notice that the term inside the brackets can either be negative or positive. It is positive when the concentration at (x, y) is higher than in the reservoir, which causes substrate to flow into the reservoir. Conversely, a substrate concentration at (x, y) which is lower than in the reservoir causes a net flow into the reaction chamber. Thus, this new influx term keeps the substrate concentration in the reaction chamber constant, which is the behaviour that we wanted to model.
The product is also transported away from the reaction chamber. Just like the substrate, its molecular radius is several orders smaller than the enzyme, which means it can pass through the semi-permeable membrane. This means that we can analyse and model the situation in the same way as for the substrate. The only difference is that we assume that near the membrane, the product density in

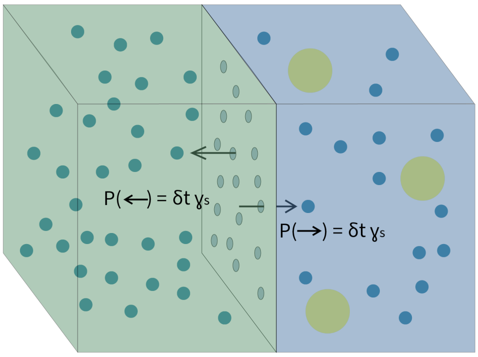


Figure 3.3: A visual representation of the semi-permeable membrane and how a substrate molecule can pass through it with probability P $= \delta t \cdot \gamma_s$. The green part denotes the reservoir and the blue part the reaction chamber. This illustration also shows the distinguishing property of semi-permeable membranes. The holes are too small for the enzyme to pass through.

the reservoir is zero. In an experimental context, this is readily achieved, since the mixture in the reservoir must be refreshed anyway to keep the substrate concentration constant. In other words, we now want to find a random variable that is dependent on $p(x, y, t)$ and a rate of osmosis γ_p.

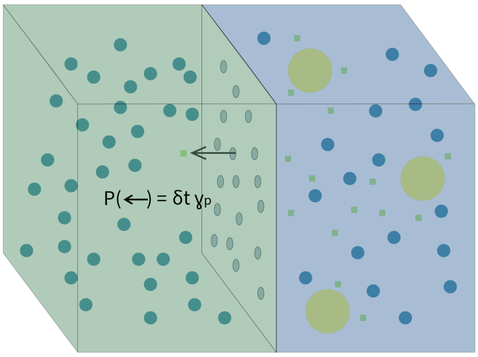


Figure 3.4: A visual representation of the semi-permeable membrane after some product has formed. A product molecule can pass through it with probability P $= \delta t \cdot \gamma_p$. The green and the blue parts denote the reservoir and reaction chamber respectively. Once again we see that the enzymes cannot pass through the membrane. The reservoir is kept at zero product concentration, such that there is no transport of product into the reaction chamber.

Therefore, the net product flow out of the reaction chamber is $F_p = O_p$, where $O_p \sim \textbf{Bin}(p(x, y, t), \delta t \cdot \gamma_p)$. The expected value of p due to F_p at $t + \delta t$ is thus:

$$p(x, y, t + \delta t) = p(x, y, t) - \delta t \cdot \gamma_p p(x, y, t). \tag{3.13}$$

4 Simulations

The resulting model from section 3 will be tested using simulations. First, the model will be explored by varying the β to see what effect it has on the simulations. After this behaviour has been discussed, the model will be applied to the problem of anomalous diffusion by enzymes in the presence of a substrate gradient.

4.1. Varying β

We will begin by showing how the model behaves under different values of β. This parameter determines the amount of bias in the random walk steps, as explained in section 3.2. If we look at equation (3.3), we expect that for small values of β, there will be no pattern formation. In this case the probability of moving to a neighbouring site will be approximately $\frac{C_e}{4}$. Therefore, the random walk will be almost unbiased. Indeed, in figure 4.1, we see that there is no indication of separation for $\beta = 10^{-5}$. All the particles are still well-mixed after 10.000 time steps. The result resembles a regular random walk. Next, we run the model for a value β which is ten times higher, but keep the rest of the parameters the same as in figure 4.1. We see that for $\beta = 10^{-4}$ a patch has appeared in the bottom right cor-

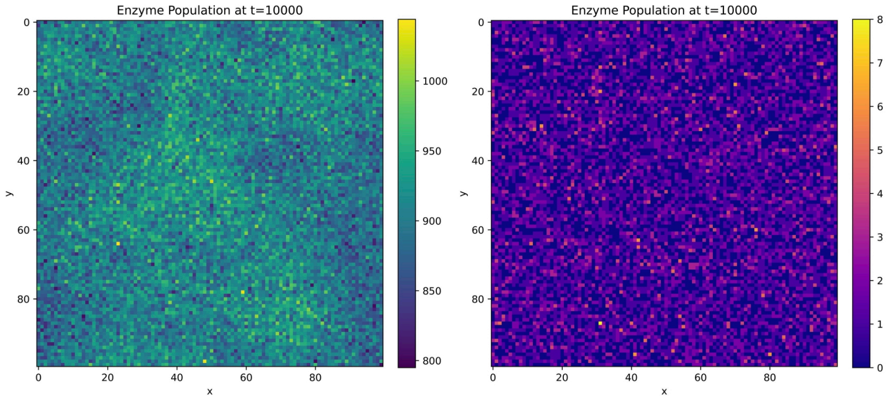


Figure 4.1: The substrate and product lattice after 10000 time steps. In this simulation $e_{tot} = 10^4$, $s_{tot} = 10^7$, $\delta t = 1$, $k_{cat} = 1$, $\gamma_s = 10^{-4}$, $C_s = 1$, $C_e = \frac{1}{10}$, $L = 100$ and $\beta = 10^{-5}$. No pattern has formed.

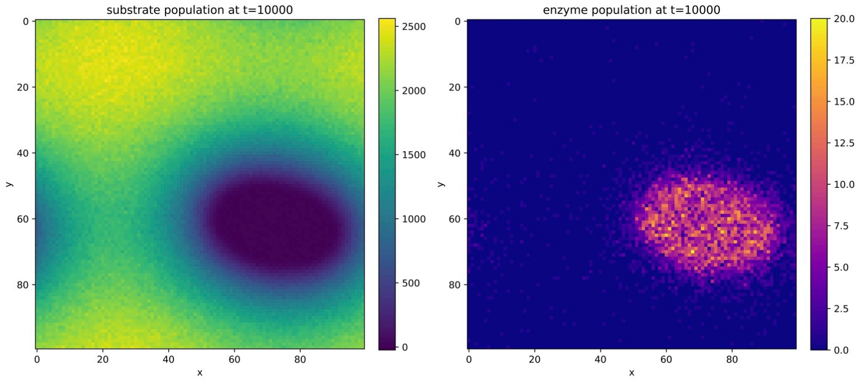
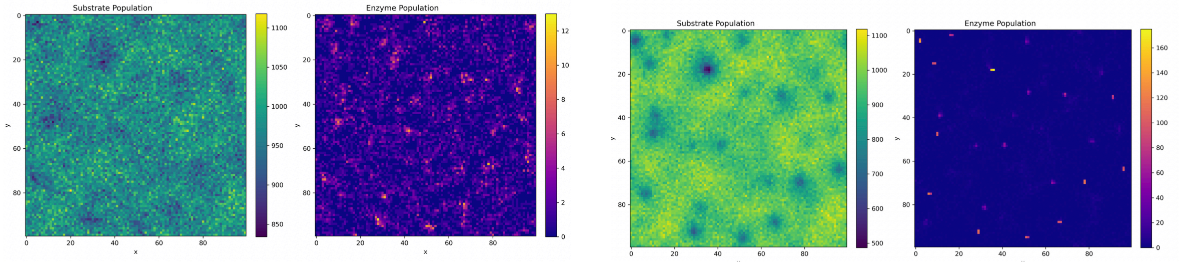


Figure 4.2: The substrate and product lattice after 10000 time steps. In this simulation $e_{tot} = 10^4$, $s_{tot} = 10^7$, $\delta t = 1$, $k_{cat} = 1$, $\gamma_s = 10^{-4}$, $C_s = 1$, $C_e = \frac{1}{10}$ and $\beta = 10^{-5}$. A big patch has formed in the right-hand corner. The substrate and enzyme thus segregate for these parameters.

ner. Increasing β thus has an effect on the amount of organisation in the system. If we increase beta even further to 10^{-1}, the enzymes and the substrate not only separate from each other, but the amount of separation is increased as well. The enzymes now order themselves into a *rice grain* like formation, as seen in figure 4.3.



(a) $t = 250 \cdot \delta t$ **(b)** $t = 500 \cdot \delta t$

Figure 4.3: The substrate and product lattice after 10000 time steps. In this simulation $e_{tot} = 10^4$, $s_{tot} = 10^7$, $\delta t = 1$, $k_cat = 1$, $\gamma_s = 10^{-4}$, $C_s = 1$, $C_e = \frac{1}{10}$ and $\beta = 10^{-1}$. First, the enzymes accumulate into small patches. After this has happened successfully, the enzymes keep on trying to form smaller patches until they have reached the rice grain stage.

Since β now heavily skews the probabilities, the enzymes only move to their respective local minimum of substrate density. First, this causes the dots to appear around $t = 250$. In the end the high value of β ensures that the enzymes oscillate between two lattice sites, forming the *rice grains*.

To talk more quantitatively about the amount of organisation in the system, we introduce the order parameter from appendix F. Plotting the order parameter for the simulation from figure 4.1 yields what we expect: the order parameter stays zero.

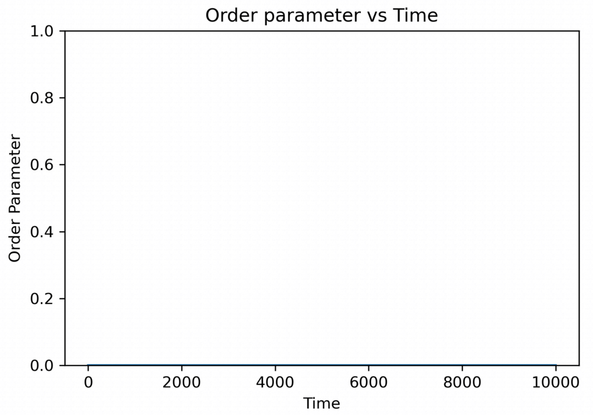


Figure 4.4: The order parameter after 10000 time steps. In this simulation $e_{tot} = 10^4$, $s_{tot} = 10^7$, $\delta t = 1$, $k_{cat} = 1$, $\gamma_s = 10^{-4}$, $C_s = 1$, $C_e = \frac{1}{10}$ and $\beta = 10^{-6}$. The system stays well mixed and thus the order parameter stays around zero.

However, the system in figure 4.3 separates into the rice grains. This is a highly ordered state. As expected, the order parameter almost goes to one.

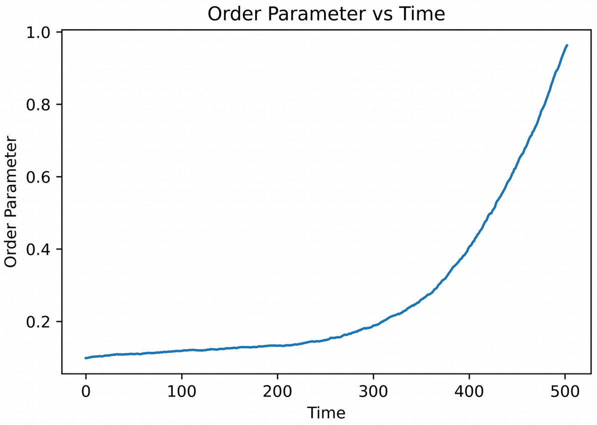


Figure 4.5: The order parameter after 500 time steps. In this simulation $e_{tot} = 10^4$, $s_{tot} = 10^7$, $\delta t = 1$, $k_{cat} = 1$, $\gamma_s = 10^{-4}$, $C_s = 1$, $C_e = \frac{1}{10}$ and $\beta = 10^{-1}$. The system approaches a completely separated state, therefore the order parameter approaches one.

4.2. Enhanced Diffusion Coefficient

In this section, we will examine the effect of the biased random walk on the diffusion coefficient of an enzyme. To do that, we must first choose an enzyme to simulate. In this report we will look at the enzyme *urease*. This enzyme catalyses the hydrolysis of urea into ammonia and carbon dioxide [2]. To determine the diffusion coefficient of urease, an ensemble of 10 simulations were run using simulation parameters that reflect the real-life properties of urease. In this case, we use the parameters for urease found in jack bean *Canavalia ensiformis* to compare to Riedel et al. [20]. The diffusion coefficient of urea is $1.38 \cdot 10^{-5} \frac{\mathrm{cm}^2}{\mathrm{s}}$ [22]. If we take $C_s = 1$, the squared distance that is travelled each time step is l^2. Following appendix D:

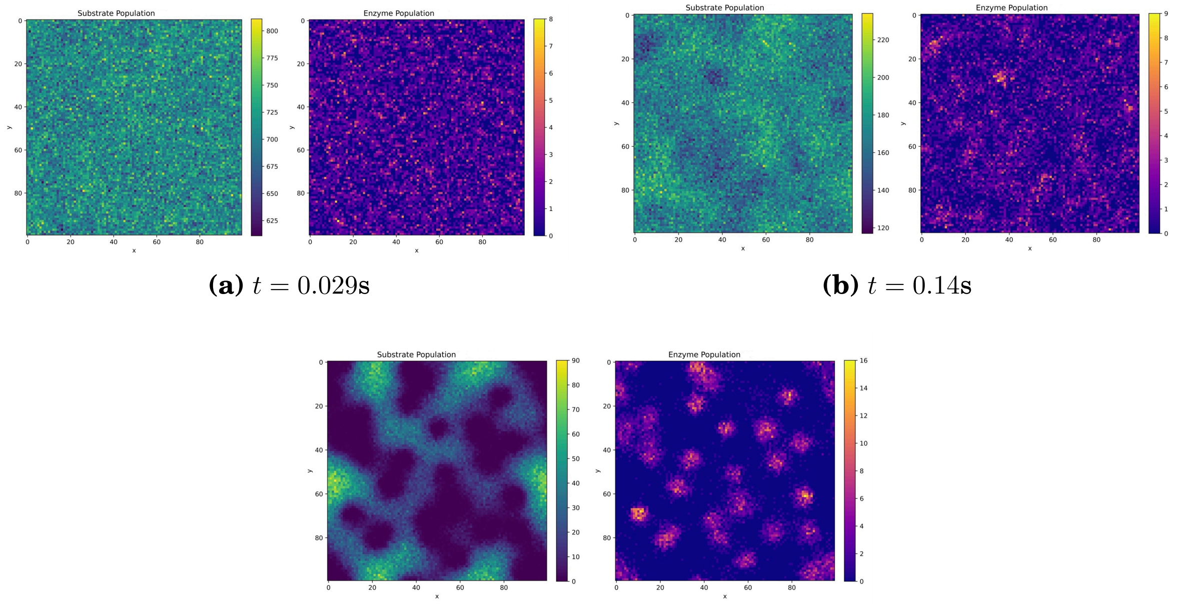
$$l = \sqrt{4D_s \cdot 1} = 7.43 \cdot 10^{-3}\,\mathrm{cm} \tag{4.1}$$

The diffusion coefficient of urease from jack bean is $3.18 \cdot 10^{-7} \frac{\mathrm{cm}^2}{\mathrm{s}}$. This means that:

$$C_e = \frac{D_s}{D_e} = 43.4 \tag{4.2}$$

Furthermore, we let $k_{cat} = 5913\mathrm{s}^{-1}$, $\delta t = \frac{1}{3500}\mathrm{s}$ and $K_m = 4.2\mathrm{mM}$, following Krajewska [16]. The ratio between substrate and product is taken to be the same as in Riedel et al.: 10^6 [20]. We let $\gamma_s = 1\mathrm{s}^{-1}$ and take the Derjaguin length to be $\lambda_c^2 = \lambda_f^2 = -1$Å, which is a typical size [7, 12]. This means that $\beta = 1.76 \cdot 10^{-4}$.

After running the ensemble simulations with the parameters that were mentioned earlier, we get the following results. If we look at the snapshots randomly picked from the ensemble simulations, we see that the enzymes start accumulating quickly into spots of low substrate density. After a short while, this means that they start depleting areas on the lattice of substrate, causing them to be able to spread out further. The effect on the mean squared difference is shown in figure 5.2.



(a) $t = 0.029\mathrm{s}$

(b) $t = 0.14\mathrm{s}$

(c) $t = 0.36\mathrm{s}$

Figure 4.6: The time evolution of the substrate and enzyme lattice. The parameters are: $l = 7.43 \cdot 10^{-3}\,\mathrm{cm}$, $C_s = 1$, $C_e = 43.4$, $k_{cat} = 5913\mathrm{s}^{-1}$, $\gamma_s = 1\mathrm{s}^{-1}$, $K_m = 4.2\mathrm{mM}$ and $\beta = 1.76 \cdot 10^{-4}$

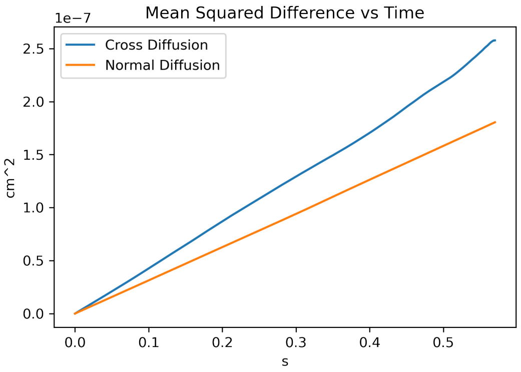


Figure 4.7: The mean squared difference plotted against the time shift. The blue line corresponds to the ghost-particles who walk dependent on equation (3.3), while the yellow line shows the ensemble average of independently random walking ghost-particles. The cross diffusion slope is steeper compared to the slope of the regular diffusion. The magnitude of this difference is in the order as predicted by Riedel et al. [20].

The slope of the mean squared difference as calculated from the ghost particles undergoing regular diffusion returns the same value as chosen before the simulation. The slope as returned by a least squares fit is:

$$D' = 4.36 \cdot 10^{-7}\,\frac{\mathrm{cm}^2}{\mathrm{s}} \tag{4.3}$$

This is an increase of 37%. This is in the same order as the increase that was found by Riedel et al. in 2015. They found that the enzyme's diffusion coefficient had increased by 28%. Therefore, non-specific interactions could play a role in the anomalous diffusion of urease. The behaviour of the model also implies that it can be tested experimentally if non-specific interactions play a role in the enhanced diffusion of enzymes, since this model demonstrates pattern formation.

5

Continuum model

The continuum limit of the discrete model constructed in section 3 will be discussed in this section. After the continuum limit has been derived, the equilibrium solutions will be calculated. A linear stability analysis will then be done on the continuum equations to calculate a critical β for which a phase transition will occur.

5.1. Limit derivation

We will start by deriving the expected values for particle densities at the site $\mathbf{v} \in \mathbb{R}^n$. To convert to particle densities, we divide e, s and p by L^2. We denote the densities by ρ_s, ρ_e and ρ_p. Using the definition of expectation, the general expression for the density of $\rho_i \in \{\rho_e, \rho_s, \rho_p\}$ at time $t + \delta t$ is given by:

$$\begin{aligned}\rho_i(\mathbf{v}, t+\delta t) = \rho_i(\mathbf{v}, t) &+ \sum_{\tilde{\mathbf{v}}\sim\mathbf{v}} \rho_i(\tilde{\mathbf{v}}, t)\mathbb{P}_i(\tilde{\mathbf{v}} \to \mathbf{v}) - \rho_i(\mathbf{v}, t) \sum_{\tilde{\mathbf{v}}\sim\mathbf{v}} \mathbb{P}_i(\mathbf{v} \to \tilde{\mathbf{v}}) \\ &+ \mathbb{E}(\{j \to i | j \neq i\}) - \mathbb{E}(\{i \to j | i \neq j\}) \\ &+ \mathbb{E}(\{\emptyset \longleftrightarrow i\}).\end{aligned} \tag{5.1}$$

$\sum_{\tilde{\mathbf{v}}\sim\mathbf{v}} \rho_i(\tilde{\mathbf{v}}, t)\mathbb{P}_i(\tilde{\mathbf{v}} \to \mathbf{v})$ corresponds to the expected density of particles moving from $\tilde{\mathbf{v}}$ into $\mathbf{v}$. $\rho_i(\mathbf{v}, t) \sum_{\tilde{\mathbf{v}}\sim\mathbf{v}} \mathbb{P}_i(\mathbf{v} \to \tilde{\mathbf{v}})$ denotes the expected density of particles moving away from $\mathbf{v}$ into $\tilde{\mathbf{v}}$. The term $\mathbb{E}(\{j \to i | j \neq i\})$ signifies the expected value of possible reactions from ρ_j into ρ_i. Conversely, $\mathbb{E}(\{i \to j | i \neq j\})$ indicates the reactions from ρ_i into ρ_j. The last term, $\mathbb{E}(\{\emptyset \longleftrightarrow i\})$, denotes the inflow or outflow of ρ_i in or out of the system. Each time step, the particles are forced to move to a neighbouring point on the lattice or stay at the site with probabilities C_i and $1 - C_i$ respectively. This means that $\sum_{\tilde{\mathbf{v}}\sim\mathbf{v}} \mathbb{P}_i(\tilde{\mathbf{v}}) = C_i$. Therefore, equation 5.1 can be simplified to:

$$\begin{aligned}\rho_i(\mathbf{v}, t+\delta t) = \; &(1 - C_i)\, \rho_i(\tilde{\mathbf{v}}, t) + \sum_{\tilde{\mathbf{v}}\sim\mathbf{v}} \rho_i(\tilde{\mathbf{v}}, t)\mathbb{P}_i(\tilde{\mathbf{v}} \to \mathbf{v}) + \mathbb{E}(\{j \to i | j \neq i\}) \\ &- \mathbb{E}(\{i \to j | i \neq j\}) + \mathbb{E}(\{\emptyset \to i\}) - \mathbb{E}(\{i \to \emptyset\}).\end{aligned} \tag{5.2}$$

5.1.1. Discrete Laplacian

We will derive an expression for the discrete Laplacian in $\mathbb{R}^n$ to replace the contribution of neighbouring sites with information about the second partial derivatives. For an arbitrary function $f \in C^2(\mathbb{R}^n)$:

$$\Delta f(\mathbf{x}) = \sum_{i=1}^{n} \frac{\partial^2 f(\mathbf{x})}{\partial x_i^2} \tag{5.3}$$

Approximating any one of the first partial derivatives with a backward finite difference gives:

$$\frac{\partial f(x_i)}{\partial x_i} \approx \frac{f(x_i) - f(x_i - h)}{h}. \tag{5.4}$$

The second derivative can then be approximated by a forward difference:

$$\frac{\partial^2 f(x_i)}{\partial x_i^2} \approx \frac{\frac{\partial f(x_i+h)}{\partial x_i} - \frac{\partial f(x_i)}{\partial x_i}}{h}. \tag{5.5}$$

Substituting equation 5.4 into the approximation for the second derivative yields:

$$\frac{\partial^2 f(x_i)}{\partial x_i^2} \approx \frac{f(x_i + h) + f(x_i - h) - 2f(x_i)}{h^2}. \tag{5.6}$$

Notice that $(x_i + h)$ and $(x_i - h)$ are the neighbouring points to (x_i). If we switch to the notation where $\sum_{\tilde{x_i} \sim x_i}$ is the sum over the neighbouring points in the x_i direction it follows that:

$$\frac{\partial^2 f(x_i)}{\partial x_i^2} \approx \frac{\sum_{\tilde{x_i} \sim x_i} f(\tilde{x_i}) - 2nf(x_i)}{h^2}. \tag{5.7}$$

To see what order error this term leaves, we will Taylor expand $f(x_i + h)$ and $f(x_i - h)$ in x_i:

$$f(x_i + h) = f(x_i) + h\frac{\partial f(x_i)}{\partial x_i} + \frac{h^2}{2}\frac{\partial^2 f(x_i)}{\partial x_i^2} + \frac{h^3}{6}\frac{\partial^3 f(x_i)}{\partial x_i^3} + \mathcal{O}(h^4) \tag{5.8}$$

and

$$f(x_i - h) = f(x_i) - h\frac{\partial f(x_i)}{\partial x_i} + \frac{h^2}{2}\frac{\partial^2 f(x_i)}{\partial x_i^2} - \frac{h^3}{6}\frac{\partial^3 f(x_i)}{\partial x_i^3} + \mathcal{O}(h^4). \tag{5.9}$$

From these Taylor expansions we see that:

$$f(x_i + h) + f(x_i - h) - 2f(x_i) = h^2\frac{\partial^2 f(x_i)}{\partial x_i^2} + \mathcal{O}(h^4). \tag{5.10}$$

Combining equation (5.10) and (5.6) therefore gives:

$$\frac{f(x_i + h) + f(x_i - h) - 2f(x_i)}{h^2} - \frac{\partial^2 f(x_i)}{\partial x_i^2} \in \mathcal{O}(h^2). \tag{5.11}$$

Thus, to derive the final approximation to equation 5.3, we substitute every second derivative in the sum for its finite difference to obtain the discrete Laplacian:

$$\Delta f = \frac{\sum_{\tilde{\mathbf{x}}\sim\mathbf{x}} f(\tilde{\mathbf{x}}) - 2nf(\mathbf{x})}{h^2} + \mathcal{O}(h^2). \tag{5.12}$$

Conversely, we now see that any time we encounter a sum over neighbouring sites it can be substituted by:

$$\sum_{\tilde{\mathbf{x}}\sim\mathbf{x}} f(\tilde{\mathbf{x}}) = h^2\Delta f(\mathbf{x}) + 2nf(\mathbf{x}) + \mathcal{O}(h^4). \tag{5.13}$$

5.1.2. The Substrate and Product Equation

We now derive the continuum limit for the substrate and product densities, ρ_s and ρ_p. First note that the transition probability of moving to an arbitrary neighbouring site in $\mathbb{R}^n$ is $\frac{C_s}{2n}$, since the substrate move one step at a time through the medium without restriction. This is a generalisation to $\mathbb{R}^n$ of what was discussed in section 3.2 with regard to equation 3.1. Including the reactions and permeation through the membrane, equation (5.1) reduces to

$$\begin{aligned}\rho_s(\mathbf{v}, t+\delta t) - \rho_s(\mathbf{v}, t) = &\left(\frac{C_s}{2n}\sum_{\tilde{\mathbf{v}}\sim\mathbf{v}} \rho_s(\tilde{\mathbf{v}}, t)\right) - C_s\rho_s(\mathbf{v}, t)\\ &- \mathbb{E}(\{s \to p\}) + \mathbb{E}(\{\emptyset \longleftrightarrow s\})\end{aligned} \tag{5.14}$$

in case of the substrate and

$$\begin{aligned}\rho_p(\mathbf{v}, t+\delta t) - \rho_p(\mathbf{v}, t) = &\left(\frac{C_p}{2n}\sum_{\tilde{\mathbf{v}}\sim\mathbf{v}} \rho_p(\tilde{\mathbf{v}}, t)\right) - C_p\rho_p(\mathbf{v}, t)\\ &+ \mathbb{E}(\{s \to p\}) + \mathbb{E}(\{\emptyset \longrightarrow p\})\end{aligned} \tag{5.15}$$

for the product. Using equation (5.12) to substitute the contribution of neighbouring sites to obtain an expression only dependent on $\mathbf{v}$:

$$\begin{aligned}\rho_s(\mathbf{v}, t+\delta t) - \rho_s(\mathbf{v}, t) = &\frac{C_s}{2n}\left[l^2\Delta\rho_s + 2n\rho_s\right] - C_s\rho_s\\ &- \mathbb{E}(\{s \to p\}) + \mathbb{E}(\{\emptyset \longleftrightarrow s\}) + \mathcal{O}(l^4)\end{aligned} \tag{5.16}$$

and

$$\begin{aligned}\rho_p(\mathbf{v}, t+\delta t) - \rho_p(\mathbf{v}, t) = &\frac{C_p}{2n}\left[l^2\Delta\rho_p + 2n\rho_p\right] - C_p\rho_p\\ &+ \mathbb{E}(\{s \to p\}) + \mathbb{E}(\{\emptyset \longleftrightarrow p\}) + \mathcal{O}(l^4).\end{aligned} \tag{5.17}$$

The expected values are given by equations (3.10), (3.11) and 3.13 such that:

$$\rho_s(\mathbf{v}, t+\delta t) - \rho_s(\mathbf{v}, t) = \frac{l^2 C_s}{2n}\Delta\rho_s - \delta t \cdot \rho_e k_{cat}\frac{\rho_s}{K_m + \rho_s} - \delta t \cdot \gamma_s\left(\rho_s - s_R\right) + \mathcal{O}(l^4) \tag{5.18}$$

and

$$\rho_p(\mathbf{v}, t+\delta t) - \rho_p(\mathbf{v}, t) = \frac{l^2 C_p}{2n}\Delta\rho_p + \delta t \cdot \rho_e k_{cat}\frac{\rho_s}{K_m + \rho_s} - \delta t \cdot \gamma_p \rho_p + \mathcal{O}(l^4). \tag{5.19}$$

Dividing 5.18 and 5.19 by δt gives:

$$\frac{\rho_s(\mathbf{v}, t+\delta t) - \rho_s(\mathbf{v}, t)}{\delta t} = \frac{l^2 C_s}{2n\delta t}\Delta\rho_s - \rho_e k_{cat}\frac{\rho_s}{K_m + \rho_s} - \gamma_s\left(\rho_s - s_R\right) + \mathcal{O}(l^4) \tag{5.20}$$

and

$$\frac{\rho_p(\mathbf{v}, t+\delta t) - \rho_p(\mathbf{v}, t)}{\delta t} = \frac{l^2 C_p}{2n\delta t}\Delta\rho_p + \rho_e k_{cat}\frac{\rho_s}{K_m + \rho_s} - \gamma_p \rho_p + \mathcal{O}(l^4). \tag{5.21}$$

Now we take $\delta t \to 0$ and $l \to 0$ in such a way that $\frac{l^2 C_s}{\delta t} \to D_s$ and $\frac{l^2 C_p}{\delta t} \to D_p$. Then we obtain:

$$\frac{\partial\rho_s(\mathbf{v}, t)}{\partial t} = \frac{D_s}{2n}\Delta\rho_s - \rho_e k_{cat}\frac{\rho_s}{K_m + \rho_s} - \gamma_s\left(\rho_s - s_R\right) \tag{5.22}$$

and

$$\frac{\partial\rho_p(\mathbf{v}, t)}{\partial t} = \frac{D_p}{2n}\Delta\rho_p + \rho_e k_{cat}\frac{\rho_s}{K_m + \rho_s} - \gamma_p \rho_p \tag{5.23}$$

as the limiting equations.

5.1.3. Enzymatic Transition Probability

We have defined the transition probability for the enzyme in equation (3.3). This probability is dependent on location, time and substrate density at the neighbouring sites. Therefore, we cannot derive the continuum equations as easily as for the substrate and product. We will proceed by simplifying (3.3) by substituting the discrete Laplacian in the denominator of equation (3.3) for equation (5.13):

$$\begin{aligned}\mathbb{P}_e(\mathbf{x} \to \mathbf{y}) &= C_e \frac{e^{-\beta\rho_s(\mathbf{y},t)}}{\sum_{\tilde{\mathbf{x}}\sim\mathbf{x}} e^{-\beta\rho_s(\mathbf{x},t)}} \\ &= C_e \frac{e^{-\beta\rho_s(\mathbf{y},t)}}{2n\, e^{-\beta\rho_s(\mathbf{x},t)} + l^2\Delta e^{-\beta\rho_s(\mathbf{x},t)}} \\ &= C_e e^{-\beta\rho_s(\mathbf{y},t)}\left[\frac{e^{\beta\rho_s(\mathbf{x},t)}}{2n + l^2\{(\nabla(\beta\rho_s(\mathbf{x},t)))^2 - \Delta(\beta\rho_s(\mathbf{x},t))\}}\right].\end{aligned} \tag{5.24}$$

Next, we will Taylor expand the term in between the square brackets. For this we will use that

$$\frac{1}{a+h} = \frac{1}{a} - \frac{h}{a^2} + \mathcal{O}(h^2). \tag{5.25}$$

This expression is derived in Appendix B. If we apply equation (5.25) to equation (5.24) it follows that:

$$\mathbb{P}_e(\mathbf{x} \to \mathbf{y}) = C_e e^{-\beta\rho_s(\mathbf{y},t)}\left[\frac{e^{\beta\rho_s(\mathbf{x},t)}}{2n}\left(1 - \frac{l^2}{2n}\{\|\nabla(\beta\rho_s(\mathbf{x},t))\|^2 - \Delta(\beta\rho_s(\mathbf{x},t))\}\right)\right]. \tag{5.26}$$

For shorthand notation we will refer to the term in square brackets as $T_e(\mathbf{x},t)$:

$$T_e(\mathbf{x},t) = \frac{e^{\beta\rho_s(\mathbf{x},t)}}{2n}\left(1 - \frac{l^2}{2n}\{\|\nabla(\beta\rho_s(\mathbf{x},t))\|^2 - \Delta(\beta\rho_s(\mathbf{x},t))\}\right). \tag{5.27}$$

It is useful to know what $\nabla T_e(\mathbf{x},t)$ and $\Delta T_e(\mathbf{x},t)$ are for the continuum derivation. First, the gradient will be calculated:

$$\begin{aligned}\nabla T_e &= \nabla\left[\frac{e^{\beta\rho_s}}{2n}\left(1 - \frac{l^2}{2n}\{\|\nabla(\beta\rho_s)\|^2 - \Delta(\beta\rho_s)\}\right)\right]\\ &= \frac{e^{\beta\rho_s}}{2n}\Bigg(\nabla(\beta\rho_s) - \frac{l^2}{2n}\Bigg[\nabla(\beta\rho_s))^3 - \nabla^3(\beta\rho_s)\\ &\qquad +\nabla(\beta\rho_s)\Delta(\beta\rho_s)\Bigg]\Bigg)\\ &= \frac{e^{\beta\rho_s}}{2n}\nabla(\beta\rho_s) + \mathcal{O}(l^2).\end{aligned} \tag{5.28}$$

The Laplacian of T_e can now be calculated:

$$\begin{aligned}\Delta T_e &= \nabla\cdot\nabla T_e\\ &= \frac{e^{\beta\rho_s}}{2n}\Bigg(((\nabla(\beta\rho_s))^2 + \Delta(\beta\rho_s)) - \frac{l^2}{2n}(4\|\nabla(\beta\rho_s)\|^2\Delta(\beta\rho_s)\\ &\qquad + \|\Delta(\beta\rho_s)\|^2 + (\nabla(\beta\rho_s))^4 - \nabla^4(\beta\rho_s)\Bigg) + \mathcal{O}(l^4).\\ &= \frac{e^{\beta\rho_s}}{2n}\left(\|\nabla(\beta\rho_s)\|^2 + \Delta(\beta\rho_s)\right) + \mathcal{O}(l^2)\end{aligned} \tag{5.29}$$

5.1.4. The Enzyme Equation

The enzyme does not react or permeate through the membrane, so equation 5.1 reduces to:

$$\rho_e(\mathbf{v},t+\delta t) = \rho_e(\mathbf{v},t) + \sum_{\tilde{\mathbf{v}}\sim\mathbf{v}}\rho_e(\tilde{\mathbf{v}},t)\mathbb{P}_e(\tilde{\mathbf{v}}\to\mathbf{v}) - C_e\rho_e(\mathbf{v},t) + \mathcal{O}(l^4). \tag{5.30}$$

Subtracting $\rho_e(\mathbf{v},t)$ and dividing by δt yields:

$$\frac{\rho_e(\mathbf{v},t+\delta t) - \rho_e(\mathbf{v},t)}{\delta t} = \frac{1}{\delta t}\left[\sum_{\tilde{\mathbf{v}}\sim\mathbf{v}}\rho_e(\tilde{\mathbf{v}},t)\mathbb{P}_e(\tilde{\mathbf{v}}\to\mathbf{v}) - C_e\rho_e(\mathbf{v},t)\right] + \mathcal{O}(\frac{l^4}{\delta t}). \tag{5.31}$$

Substituting $\mathbb{P}_e(\tilde{\mathbf{v}}\to\mathbf{v})$ by equation (5.24) gives:

$$\frac{\rho_e(\mathbf{v},t+\delta t) - \rho_e(\mathbf{v},t)}{\delta t} = \frac{1}{\delta t}\left[C_e e^{-\beta\rho_s(\mathbf{v},t)}\left(\sum_{\tilde{\mathbf{v}}\sim\mathbf{v}}\rho_e(\tilde{\mathbf{v}},t)T_e(\tilde{\mathbf{v}},t)\right) - C_e\rho_e(\mathbf{v},t)\right] + \mathcal{O}(\frac{l^4}{\delta t}). \tag{5.32}$$

Using the discrete Laplacian once more to replace the summation over neighbouring sites we obtain a right-hand side only dependent on $(\mathbf{v},t)$:

$$\frac{\rho_e(\mathbf{v},t+\delta t) - \rho_e(\mathbf{v},t)}{\delta t} = \frac{1}{\delta t}\left[C_e - e^{\beta\rho_s}\left(2n\rho_e T_e + l^2\Delta(\rho_e T_e)\right) - C_e\rho_e\right] + \mathcal{O}(\frac{l^4}{\delta t}). \tag{5.33}$$

We can now substitute for $T_e(\mathbf{v},t)$ in the first term, giving:

$$\begin{aligned}\frac{\rho_e(\mathbf{v},t+\delta t)-\rho_e(\mathbf{v},t)}{\delta t} &= \frac{1}{\delta t}\left[C_e\rho_e\left(1-\frac{l^2}{2n}\{\|\nabla(\beta\rho_s)\|^2-\Delta(\beta\rho_s)\}\right)\right.\\ &\left.\quad + C_e l^2 e^{-\beta\rho_s}\Delta(\rho_e T_e) - C_e\rho_e\right] + \mathcal{O}(l^4)\\ &= \frac{l^2C_e}{\delta t}\left[-\frac{\rho_e}{2n}\left(\|\nabla(\beta\rho_s)\|^2-\Delta(\beta\rho_s)\right)+e^{-\beta\rho_s}\Delta(\rho_eT_e)\right]+\mathcal{O}(\frac{l^4}{\delta t}).\end{aligned} \tag{5.34}$$

Using the product rule on $\Delta(\rho_eT_e)$ we see that:

$$\Delta(\rho_eT_e) = T_e\Delta\rho_e + 2\nabla T_e\nabla\rho_e + \rho_e\Delta T_e. \tag{5.35}$$

Substituting equations (5.28) and (5.29) into (5.35) gives

$$\Delta(\rho_eT_e) = \frac{e^{\beta\rho_s}}{2n}\Delta\rho_e + \frac{2e^{\beta\rho_s}}{2n}\nabla(\beta\rho_s)\nabla\rho_e + \frac{e^{\beta\rho_s}}{2n}\rho_e\left(\Delta(\beta\rho_s)+\|\nabla(\beta\rho_s)\|^2\right)+\mathcal{O}(l^2), \tag{5.36}$$

which can be rearranged to

$$\Delta(\rho_eT_e) = \frac{e^{\beta\rho_s}}{2n}\left[\Delta\rho_e+\rho_e\left(\Delta(\beta\rho_s)+\|\nabla(\beta\rho_s)\|^2\right)+\nabla(\beta\rho_s)\nabla\rho_e\right]+\mathcal{O}(l^2). \tag{5.37}$$

Plugging this into equation 5.34 produces:

$$\begin{aligned}\frac{\rho_e(\mathbf{v},t+\delta t)-\rho_e}{\delta t} =& \frac{l^2C_e}{2n\delta t}\left[-\rho_e\left(\|\nabla(\beta\rho_s)\|^2-\Delta(\beta\rho_s)\right)+\Delta\rho_e\right.\\ &+\rho_e\left(\|\nabla(\beta\rho_s)\|^2+\Delta(\beta\rho_s)\right)+2\nabla(\beta\rho_s)\nabla\rho_e]+\mathcal{O}(\frac{l^4}{\delta t})\end{aligned} \tag{5.38}$$

Rearranging the terms and taking the gradient out of the square brackets gives:

$$\frac{\rho_e(\mathbf{v},t+\delta t)-\rho_e(\mathbf{v},t)}{\delta t} = \frac{l^2C_e}{2n\delta t}\nabla\cdot\left[\nabla\rho_e+2\rho_e\nabla(\beta\rho_s)\right]+\mathcal{O}(\frac{l^4}{\delta t}). \tag{5.39}$$

Once again taking $\delta t \to 0$ and $l \to 0$ so that $\frac{l^2C_e}{\delta t} \to D_e$ leads us to the final expression, which is:

$$\frac{\partial\rho_e(\mathbf{v},t)}{\partial t} = \frac{D_e}{2n}\nabla\cdot\left[\nabla\rho_e+2\rho_e\nabla(\beta\rho_s)\right]. \tag{5.40}$$

5.2. Total System

If we combine equations (5.22), (5.23) and (5.40) into one system we get the following:

$$\begin{cases}\frac{\partial\rho_s}{\partial t} &= \frac{D_s}{2n}\Delta\rho_s - \rho_e k_{cat}\frac{\rho_s}{K_m+\rho_s} - \gamma_s\left(\rho_s - s_R\right),\\ \frac{\partial\rho_e}{\partial t} &= \frac{D_e}{2n}\nabla\cdot\left[\nabla\rho_e+2\rho_e\nabla(\beta\rho_s)\right],\\ \frac{\partial\rho_p}{\partial t} &= \frac{D_p}{2n}\Delta\rho_p + \rho_e k_{cat}\frac{\rho_s}{K_m+\rho_s} - \gamma_p\rho_p.\end{cases} \tag{5.41}$$

In the equations for the substrate and the product, we recognise the first term as Fick's law like in equation 2.11. Furthermore, we see that the second term subtracts substrate from the substrate equation at the same rate as it adds product to the product equation. The rate at which this occurs is of course the Michaelis-Menten term as discussed in section 2.1.1. We can also recognise the last term in the substrate and the product equations as the permeation of each compound through the semi-permeable membrane with rate γ_s and γ_p respectively. The permeation term in the substrate equation retains the possibility of being either positive or negative, while the product is only transported away from the reaction chamber. Finally, we see that the enzyme's differential equation resembles equation (2.17). There is a subtle difference, however. Instead of the drift velocity terms there is a β in the gradient term with the enzyme concentration. For β a constant, we see that this corresponds to the situation where the Derjaguin lengths for the bound and free enzyme have the same value, that is $\lambda_b^2 = \lambda_f^2$. We then see that $V_{ph} = \lambda_f^2 \frac{k_b T}{\eta}$ in equation (2.18) and $V_{bi} = 0$ in equation (2.19). In this case, equation (2.17) corresponds to the enzyme's evolution equation in system (5.41) above, where $\beta = \lambda_f^2 \frac{k_b T}{\eta}$. If the Derjaguin lengths are not the same however, an ODE can be solved for $\beta(\mathbf{v}, t)$ such that it corresponds to the drift velocity terms when the gradient is further expanded using the product rule. This will be elaborated on in the discussion section of this report. For the nect sections, we will assume that $\lambda_b = \lambda_f$ in the next sections. Furthermore, to compare to the discrete 2-dimensional model from section 3, we let $n = 2$.

5.2.1. Equilibrium solution

To derive an equilibrium solution to the system of equations 5.41, we solve the equations under the condition that $\frac{\partial \rho_i}{\partial t} = 0$ [13]. Thus we need to solve for $\bar{\rho_s}$, $\bar{\rho_e}$ and $\bar{\rho_p}$ such that:

$$\begin{cases} 0 = \frac{D_s}{4}\Delta\bar{\rho_s} - \bar{\rho_e} k_{cat}\frac{\bar{\rho_s}}{K_m + \bar{\rho_s}} - \gamma_s\left(\bar{\rho_s} - s_R\right), \\ 0 = \frac{D_e}{4}\nabla \cdot \left[\nabla\bar{\rho_e} + 2\bar{\rho_e}\beta\nabla\bar{\rho_s}\right], \\ 0 = \frac{D_p}{4}\Delta\bar{\rho_p} + \bar{\rho_e} k_{cat}\frac{\bar{\rho_s}}{K_m + \bar{\rho_s}} - \gamma_p\bar{\rho_p}. \end{cases} \tag{5.42}$$

We start by looking at the enzyme's equation. If we assume that $\bar{\rho_e} \neq 0$ is a constant, then it follows that

$$\nabla \cdot (2\bar{\rho_e}\beta\nabla\bar{\rho_s}) = 0, \tag{5.43}$$

since, $\nabla\bar{\rho_e} = 0$ if ρ_e is constant. Taking the divergence operator into the brackets gives:

$$2\beta\nabla\bar{\rho_e} \cdot \nabla\bar{\rho_s} + 2\bar{\rho_e}\beta\Delta\bar{\rho_s} = 0, \tag{5.44}$$

which similarly simplifies to:

$$2\bar{\rho_e}\beta\Delta\bar{\rho_s} = 0. \tag{5.45}$$

We assumed a non-trivial homogeneous solution for $\bar{\rho_e}$, which together with the fact that $\beta > 0$ results in the conclusion that $\Delta\bar{\rho_s} = 0$. Thus, the substrate's equation reduces to:

$$\bar{\rho_e} k_{cat} \frac{\bar{\rho_s}}{K_m + \bar{\rho_s}} + \gamma_s \left(\bar{\rho_s} - s_R\right) = 0 \tag{5.46}$$

Multiplying through by $K_m + \bar{\rho_s}$ gives that:

$$\bar{\rho_e} k_{cat} \bar{\rho_s} + \gamma_s \left(\bar{\rho_s}^2 + (K_m - s_R)\bar{\rho_s} - s_R K_m\right) = 0 \tag{5.47}$$

Dividing by γ_s and completing the square yields:

$$\left(\bar{\rho_s} + \frac{1}{2}\left[\bar{\rho_e}\frac{k_{cat}}{\gamma_s} + (K_m - s_R)\right]\right)^2 = s_R K_m + \frac{1}{4}\left[\bar{\rho_e}\frac{k_{cat}}{\gamma_s} + (K_m - s_R)\right]^2 \tag{5.48}$$

From this, the final expression for the substrate's equilibrium solution $\bar{\rho_s}$ is deduced from the fact that we require the answer to be non-negative to be physically accurate:

$$\bar{\rho_s} = \frac{1}{2}\left(s_R - K_m - \frac{k_{cat}}{\gamma_s}\bar{\rho_e} + \sqrt{4 s_R K_m + \left[\frac{k_{cat}}{\gamma_s}\bar{\rho_e} + (K_m - s_R)\right]^2}\right) \tag{5.49}$$

If we move on to the evolution equation for the product, we can easily derive an equilibrium solution if we assume that it is constant as well. The PDE then reduces to:

$$0 = \bar{\rho_e} k_{cat} \frac{\bar{\rho_s}}{K_m + \bar{\rho_s}} - \gamma_p \bar{\rho_p}. \tag{5.50}$$

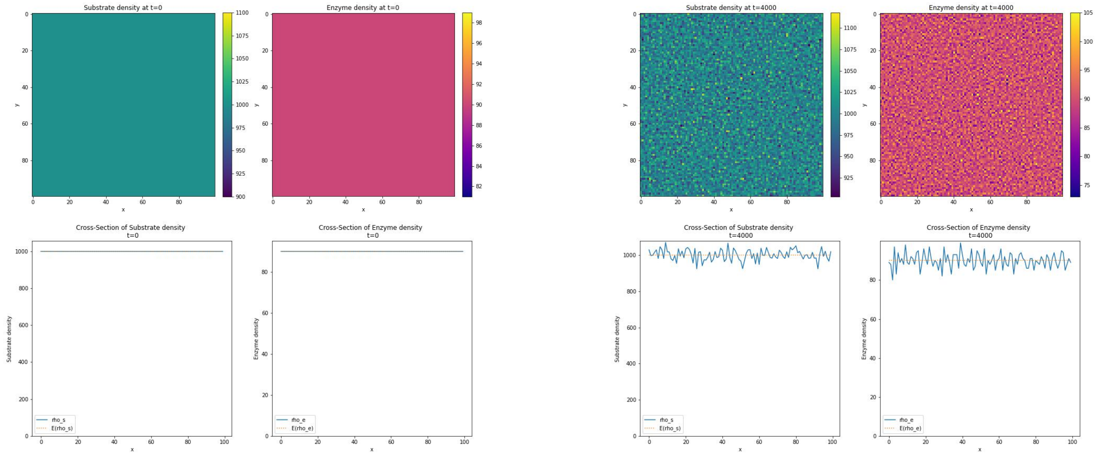
Using the fact that $\bar{\rho_e} k_{cat} \frac{\bar{\rho_s}}{K_m + \bar{\rho_s}} + \gamma_s \left(\bar{\rho_s} - s_R\right) = 0$ from equation (5.46), we can add $\gamma_s \left(\bar{\rho_s} - s_R\right)$ to both sides of equation (5.50) to get:

$$\gamma_p \bar{\rho_p} = -\gamma_s \left(\bar{\rho_s} - s_R\right) \tag{5.51}$$

We can now substitute this expression for $\bar{\rho_s}$ into equation (5.49) and divide by γ_p, such that:

$$\bar{\rho_p} = \frac{\gamma_s}{2\gamma_p}\left(s_R + K_m + \frac{k_{cat}}{\gamma_s}\bar{\rho_e} - \sqrt{4 s_R K_m + \left[\frac{k_{cat}}{\gamma_s}\bar{\rho_e} + (K_m - s_R)\right]^2}\right) \tag{5.52}$$

The continuum derivation can be checked using the equilibrium solution that was derived by starting a simulation with constant distributions of $\bar{\rho_e}$, $\bar{\rho_p}$ and $\bar{\rho_p}$ related to each other by equations (5.49) and (5.52). After 4000 timesteps, pattern formation would have usually occurred already. However, as seen in figure 5.1, the distributions of both enzyme and substrate remain almost constant. The maximum deviation from the original constant distribution is 0.7% for the ρ_s and 1.0% for ρ_e. This indicates that the steady state behaviour of the particle model is akin to the equilibrium situation of the continuum model.



(a) t = 0

(b) t = 4000

Figure 5.1: A simulation of the equilibrium solution of the enzyme and the substrate. In this simulation $e_{tot} = 10^6$, $s_{tot} = 11 * 10^6$, $\delta t = 1$, $k_{cat} = 1$, $\gamma_s = 10^{-4}$, $C_s = 1$, $C_e = \frac{1}{10}$ and $\beta = 10^{-1}$.

5.2.2. Linear stability analysis

Now that we have derived and verified the equilibrium solution to the continuum model, we can perform a linear stability analysis to calculate when the phase transition occurs. The perturbations that will be applied to the system have the form:

$$\begin{cases} \rho_s = \bar{\rho_s} + \delta_s e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}, \\ \rho_e = \bar{\rho_e} + \delta_e e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}, \\ \rho_p = \bar{\rho_p} + \delta_p e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}, \end{cases} \tag{5.53}$$

where $\delta_i \ll 1$, $\bar{\rho_i} \in \{\bar{\rho_e}, \bar{\rho_s}, \bar{\rho_p}\}$ correspond to the equilibrium solutions derived in section 5.2.1 and $\mathbf{v} = [x, y]^T$. This method is equivalent to the approach taken in [3, 10, 21]. Substituting the equations in (5.53) into the evolution equation for ρ_s from equation (5.41) gives:

$$\begin{aligned} \frac{\partial}{\partial t}(\bar{\rho_s} + \delta_s e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}) =& \frac{D_s}{4}\Delta(\bar{\rho_s} + \delta_s e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}) \\ &- k_{cat}(\bar{\rho_e} + \delta_e e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}) \frac{\bar{\rho_s} + \delta_s e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}}{K_m + \bar{\rho_s} + \delta_s e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}} \\ &- \gamma_s \left(\bar{\rho_s} + \delta_s e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}} - s_R\right). \end{aligned} \tag{5.54}$$

To linearise this equation, we substitute $\frac{1}{K_m + \bar{\rho_s} + \delta_s e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}}$ for its Taylor expansion, derived in appendix B: $\frac{1}{K_m + \bar{\rho_s} + \delta_s e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}} = \frac{1}{K_m + \bar{\rho_s}} - \frac{\delta_s e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}}{(K_m + \bar{\rho_s})^2} + \mathcal{O}(\delta_s^2)$. This results in the following expression:

$$\begin{aligned}\frac{\partial}{\partial t}(\bar{\rho_s}+\delta_s e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}}) =&\frac{D_s}{4}\Delta(\bar{\rho_s}+\delta_s e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}})\\ &-k_{cat}(\bar{\rho_e}+\delta_e e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}})(\bar{\rho_s}+\delta_s e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}})\left(\frac{1}{K_m+\bar{\rho_s}}-\frac{\delta_s e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}}}{(K_m+\bar{\rho_s})^2}+\mathcal{O}(\delta_s^2)\right)\\ &-\gamma_s\left(\bar{\rho_s}+\delta_s e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}}-s_R\right).\end{aligned} \tag{5.55}$$

Using the fact that $\frac{\bar{\rho_s}}{\partial t}=\frac{D_s}{2n}\Delta\bar{\rho_s}-\bar{\rho_e}k_{cat}\frac{\bar{\rho_s}}{K_m+\bar{\rho_s}}-\gamma_s\left(\bar{\rho_s}-s_R\right)=0$, since $(\bar{\rho_e},\bar{\rho_s},\bar{\rho_p})$ is an equilibrium solution, and neglecting the higher order δ_i terms, we find:

$$\alpha\delta_s e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}}=\left(-\frac{D_s}{4}|\mathbf{k}|^2\delta_s-k_{cat}\frac{\bar{\rho_e}\delta_s+\bar{\rho_s}\delta_e}{K_m+\bar{\rho_s}}+k_{cat}\bar{\rho_e}\bar{\rho_s}\delta_s\frac{1}{(K_m+\bar{\rho_s})^2}-\gamma_s\delta_s\right)e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}}. \tag{5.56}$$

For clarity, we will write $F(x)=\frac{x}{K_m+x}$ and $F'(x)=\frac{d}{dx}\frac{x}{K_m+x}=\frac{K_m}{(K_m+x)^2}$, such that:

$$\alpha\delta_s=\left(-\frac{D_s}{4}|\mathbf{k}|^2-k_{cat}\frac{\bar{\rho_e}}{K_m+\bar{\rho_s}}+\frac{k_{cat}}{K_m}\bar{\rho_e}\bar{\rho_s}F'(\bar{\rho_s})-\gamma_s\right)\delta_s-k_{cat}\delta_e F(\bar{\rho_s}). \tag{5.57}$$

This is the furthest that we can simplify this expression. Next, we will consider the evolution equation for ρ_p. Again, substituting for the perturbed solutions gives:

$$\begin{aligned}\frac{\partial}{\partial t}(\bar{\rho_p}+\delta_p e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}}) =&\frac{D_p}{4}\Delta(\bar{\rho_p}+\delta_p e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}})\\ &+k_{cat}(\bar{\rho_e}+\delta_e e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}})\frac{\bar{\rho_s}+\delta_s e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}}}{K_m+\bar{\rho_s}+\delta_s e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}}}\\ &-\gamma_p(\bar{\rho_p}+\delta_p e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}}).\end{aligned} \tag{5.58}$$

Using the Taylor expansion, as we did for equation (5.55), yields:

$$\begin{aligned}\frac{\partial}{\partial t}(\bar{\rho_p}+\delta_p e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}}) =&\frac{D_p}{4}\Delta(\bar{\rho_p}+\delta_p e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}})\\ &+k_{cat}(\bar{\rho_e}+\delta_e e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}})(\bar{\rho_s}+\delta_s e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}})\left(\frac{1}{K_m+\bar{\rho_s}}-\frac{\delta_s e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}}}{(K_m+\bar{\rho_s})^2}+\mathcal{O}(\delta_s^2)\right)\\ &-\gamma_p\left(\bar{\rho_p}+\delta_p e^{\alpha t}e^{i\mathbf{k}\cdot\mathbf{v}}\right).\end{aligned} \tag{5.59}$$

Using the properties of the equilibrium solution and once again neglecting higher order terms of δ_i results in the final equation:

$$\alpha\delta_p=-\left(\frac{D_p}{4}|\mathbf{k}|^2+\gamma_p\right)\delta_p+\left(k_{cat}\frac{\bar{\rho_e}}{K_m+\bar{\rho_s}}-\frac{k_{cat}}{K_m}\bar{\rho_e}\bar{\rho_s}F'(\bar{\rho_s})\right)\delta_s+k_{cat}\delta_e F(\bar{\rho_s}). \tag{5.60}$$

Lastly, we will examine the evolution equation for ρ_e. If we substitute the equilibrium solutions into equation (5.40), under the assumption of constant β, we obtain:

$$\begin{aligned}\frac{\partial}{\partial t}(\bar{\rho_e} + \delta_e e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}) &= \frac{D_e}{4}\nabla\cdot\left[\nabla(\bar{\rho_e} + \delta_e e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}) + 2\beta(\bar{\rho_e} + \delta_e e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}})\nabla(\bar{\rho_s} + \delta_s e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}})\right] \\ &= \frac{D_e}{4}\nabla\cdot\left[\nabla(\bar{\rho_e}) + \delta_e i\mathbf{k} e^{\alpha t} e^{ikx} + 2\beta(\bar{\rho_e} + \delta_e e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}})\nabla(\bar{\rho_s} + \delta_s e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}).\right]\end{aligned} \tag{5.61}$$

Due to the fact that $\frac{\partial\bar{\rho_e}}{\partial t} = \frac{D_e}{2n}\nabla\cdot\left[\nabla\bar{\rho_e} + 2\bar{\rho_e}\beta\nabla\bar{\rho_s}\right] = 0$, the expression reduces to:

$$\alpha\delta_e e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}} = \frac{D_e}{4}\left(-|\mathbf{k}|^2\delta_e - 2|\mathbf{k}|^2\beta\bar{\rho_e}\delta_s\right) e^{\alpha t} e^{i\mathbf{k}\cdot\mathbf{v}}. \tag{5.62}$$

Therefore:

$$\alpha\delta_e = \frac{D_e}{4}\left(-|\mathbf{k}|^2\delta_e - 2|\mathbf{k}|^2\beta\bar{\rho_e}\delta_s\right). \tag{5.63}$$

We can combine equations (5.57), (5.60) and (5.63) into an eigenvalue equation $M\vec{\delta} = \alpha\vec{\delta}$:

$$\alpha\begin{bmatrix}\delta_e \\ \delta_s \\ \delta_p\end{bmatrix} = \begin{bmatrix} -\frac{D_e}{4}|\mathbf{k}|^2 & -\frac{D_e}{2}\beta\bar{\rho_e}|\mathbf{k}|^2 & 0 \\ -k_{cat}\delta_e F(\bar{\rho_s}) & \left(-\frac{D_s}{4}|\mathbf{k}|^2 - k_{cat}\frac{\bar{\rho_e}}{K_m+\bar{\rho_s}} + \frac{k_{cat}}{K_m}\bar{\rho_e}\bar{\rho_s}F'(\bar{\rho_s}) - \gamma_s\right) & 0 \\ k_{cat}F(\bar{\rho_s}) & \left(k_{cat}\frac{\bar{\rho_e}}{K_m+\bar{\rho_s}} - \frac{k_{cat}}{K_m}\bar{\rho_e}\bar{\rho_s}F'(\bar{\rho_s})\right) & -\frac{D_p}{4}|\mathbf{k}|^2 - \gamma_p \end{bmatrix}\begin{bmatrix}\delta_e \\ \delta_s \\ \delta_p\end{bmatrix}. \tag{5.64}$$

To find the possible instability of the system, we need to solve this eigenvalue equation and thus find its normal modes. If there exists an eigenvalue α such that $\mathrm{Re}(\alpha) > 0$, then the perturbations will not decay over time and we will have found an instability. Furthermore, we can make the following observation: the product does not play any role in the pattern formation. This means that the instability must occur in the (δ_e, δ_s) subspace. We can then find the relevant eigenvalues by demanding non-trivial solutions, such that $|M - I\alpha| \neq 0$. In the (δ_e, δ_s) subspace this yields:

$$\begin{vmatrix} -\frac{D_e}{4}|\mathbf{k}|^2 - \alpha & -\frac{D_e}{2}\beta\bar{\rho_e}|\mathbf{k}|^2 \\ -k_{cat}F(\bar{\rho_s}) & \left(-\frac{D_s}{4}|\mathbf{k}|^2 - k_{cat}\left(\frac{\bar{\rho_e}}{K_m+\bar{\rho_s}} + \frac{\bar{\rho_e}\bar{\rho_s}}{K_m}F'(\bar{\rho_s})\right) - \gamma_s - \alpha\right)\end{vmatrix} = 0. \tag{5.65}$$

We use the well-known formula for the determinant of a 2×2 matrix to find that:

$$\begin{aligned}&\frac{D_s D_e}{16}|\mathbf{k}|^4 + k_{cat}\frac{D_e}{4}|\mathbf{k}|^2\left(\frac{\bar{\rho_e}}{K_m+\bar{\rho_s}} - \frac{\bar{\rho_e}\bar{\rho_s}}{K_m}F'(\bar{\rho_s})\right) + \gamma_s\frac{D_e}{4}|\mathbf{k}|^2 + \frac{D_e}{4}|\mathbf{k}|^2\alpha + \frac{D_s}{4}|\mathbf{k}|^2\alpha \\ &+ \alpha k_{cat}\left(\frac{\bar{\rho_e}}{K_m+\bar{\rho_s}} - \frac{\bar{\rho_e}\bar{\rho_s}}{K_m}F'(\bar{\rho_s})\right) + \alpha\gamma_s + \alpha^2 - k_{cat}F(\bar{\rho_s})\frac{D_e}{2}|\mathbf{k}|^2\beta\bar{\rho_e} = 0.\end{aligned} \tag{5.66}$$

Next, we group terms in powers of α:

$$\begin{aligned}&\alpha^2 + \alpha\left(\frac{D_s+D_e}{4}|\mathbf{k}|^2 + k_{cat}\left(\frac{\bar{\rho_e}}{K_m+\bar{\rho_s}} - \frac{\bar{\rho_e}\bar{\rho_s}}{K_m}F'(\bar{\rho_s})\right) + \gamma_s\right) \\ &- k_{cat}F(\bar{\rho_s})\frac{D_e}{2}|\mathbf{k}|^2\beta\bar{\rho_e} + \frac{D_s D_e}{16}|\mathbf{k}|^4 + k_{cat}\frac{D_e}{4}|\mathbf{k}|^2\left(\frac{\bar{\rho_e}}{K_m+\bar{\rho_s}} - \frac{\bar{\rho_e}\bar{\rho_s}}{K_m}F'(\bar{\rho_s})\right) + \gamma_s\frac{D_e}{4}|\mathbf{k}|^2 = 0.\end{aligned} \tag{5.67}$$

We can then complete the square to get:

$$\left(\alpha + \frac{1}{2}\left[\frac{D_s + D_e}{4}|\mathbf{k}|^2 + k_{cat}\left(\frac{\bar{\rho}_e}{K_m + \bar{\rho}_s} - \frac{\bar{\rho}_e\bar{\rho}_s}{K_m}F'(\bar{\rho}_s)\right) + \gamma_s\right]\right)^2 =$$
$$k_{cat}F(\bar{\rho}_s)\frac{D_e}{2}|\mathbf{k}|^2\beta\bar{\rho}_e - \frac{D_sD_e}{16}|\mathbf{k}|^4 - k_{cat}\frac{D_e}{4}|\mathbf{k}|^2\left(\frac{\bar{\rho}_e}{K_m + \bar{\rho}_s} - \frac{\bar{\rho}_e\bar{\rho}_s}{K_m}F'(\bar{\rho}_s)\right) - \gamma_s\frac{D_e}{4}|\mathbf{k}|^2$$
$$+ \left(\frac{1}{2}\left[\frac{D_s + D_e}{4}|\mathbf{k}|^2 + k_{cat}\left(\frac{\bar{\rho}_e}{K_m + \bar{\rho}_s} - \frac{\bar{\rho}_e\bar{\rho}_s}{K_m}F'(\bar{\rho}_s)\right) - \gamma_s\right]\right)^2. \tag{5.68}$$

Taking the square root of both sides gives the eigenvalues:

$$\alpha_{1,2} = -\frac{1}{2}\left[\frac{D_s + D_e}{4}|\mathbf{k}|^2 + k_{cat}\left(\frac{\bar{\rho}_e}{K_m + \bar{\rho}_s} - \frac{\bar{\rho}_e\bar{\rho}_s}{K_m}F'(\bar{\rho}_s)\right) + \gamma_s\right] \pm$$
$$\left[k_{cat}F(\bar{\rho}_s)\frac{D_e}{2}|\mathbf{k}|^2\beta\bar{\rho}_e - \frac{D_sD_e}{16}|\mathbf{k}|^4 - k_{cat}\frac{D_e}{4}|\mathbf{k}|^2\left(\frac{\bar{\rho}_e}{K_m + \bar{\rho}_s} - \frac{\bar{\rho}_e\bar{\rho}_s}{K_m}F'(\bar{\rho}_s)\right)\right.$$
$$\left.-\gamma_s\frac{D_e}{4}|\mathbf{k}|^2 + \left(\frac{1}{2}\left[\frac{D_s + D_e}{4}|\mathbf{k}|^2 + k_{cat}\left(\frac{\bar{\rho}_e}{K_m + \bar{\rho}_s} - \frac{\bar{\rho}_e\bar{\rho}_s}{K_m}F'(\bar{\rho}_s)\right) + \gamma_s\right]\right)^2\right]^{\frac{1}{2}}. \tag{5.69}$$

As mentioned earlier, we want to find the critical conditions, such that at least one of the eigenvalues is positive and a phase transition appears. Looking at the expressions for α_1 and α_2, we see that only α_1 can become positive. This happens if:

$$k_{cat}F(\bar{\rho}_s)\frac{D_e}{2}|\mathbf{k}|^2\beta\bar{\rho}_e - \frac{D_sD_e}{16}|\mathbf{k}|^4 - k_{cat}\frac{D_e}{4}|\mathbf{k}|^2\left(\frac{\bar{\rho}_e}{K_m + \bar{\rho}_s} - \frac{\bar{\rho}_e\bar{\rho}_s}{K_m}F'(\bar{\rho}_s)\right) - \gamma_s\frac{D_e}{4}|\mathbf{k}|^2 > 0. \tag{5.70}$$

If we rewrite this in terms of the frequency of the perturbation, we get the following inequality:

$$|\mathbf{k}|^2 < \frac{8k_{cat}\bar{\rho}_e}{D_s}F(\bar{\rho}_s)\beta - \frac{4k_{cat}}{D_s}\left(\frac{\bar{\rho}_e}{K_m + \bar{\rho}_s} - \frac{\bar{\rho}_e\bar{\rho}_s}{K_m}F'(\bar{\rho}_s)\right) - \frac{4\gamma_s}{D_s}. \tag{5.71}$$

Taking everything together into the same fraction gives:

$$|\mathbf{k}|^2 < \frac{8k_{cat}\bar{\rho}_e\bar{\rho}_s\beta(K_m + \bar{\rho}_s) - 4k_{cat}\bar{\rho}_e(K_m + \bar{\rho_s}) + k_{cat}\bar{\rho}_e\bar{\rho}_s - 4\gamma_s(K_m + \bar{\rho}_s)^2}{D_s(K_m + \bar{\rho}_s)^2}, \tag{5.72}$$

where both $|\mathbf{k}|^2 > 0$ and $D_s(K_m + \bar{\rho}_s)^2 > 0$. This means that it must hold that:

$$8k_{cat}\bar{\rho}_e\bar{\rho}_s\beta(K_m + \bar{\rho}_s) > 4k_{cat}\bar{\rho}_e(K_m + \bar{\rho_s}) - 4k_{cat}\bar{\rho}_e\bar{\rho}_s + 4\gamma_s(K_m + \bar{\rho}_s)^2. \tag{5.73}$$

Or to write the condition in terms of beta:

$$\beta > \frac{1}{2\bar{\rho}_s} - \frac{1}{2}\frac{1}{K_m + \bar{\rho}_s} + \frac{\gamma_s(K_m + \bar{\rho}_s)}{2k_{cat}\bar{\rho}_s\bar{\rho}_e}. \tag{5.74}$$

We can now show that we expect the urease molecule to undergo a phase transition.

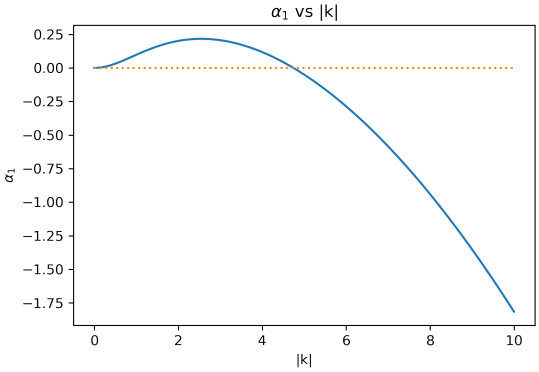


Figure 5.2: A plot of α_1 against $|k|$, for the values: $l = 7.43 \cdot 10^{-3}$ cm, $C_s = 1$, $C_e = 43.4$, $k_{cat} = 5913\text{s}^{-1}$, $\gamma_s = 1\text{s}^{-1}$, $K_m = 4.2\text{mM}$. α_1 is positive for any value lower than $|k| \approx 5$. Thus a phase transition is expected

The eigenvalue permits positive values, which means that for a perturbation of the allowed frequency, a phase transition will occur. This can be seen in figure 4.6. From (5.74), we can calculate that we expect the critical value for β is $\beta = 8.4 \cdot 10^{-5}$. This corresponds to a Derjaguin length of $lambda_c^2 = \lambda_f^2 \approx -0.5\,\text{Å}$.

6
Discussion

6.1. Critical β

There was no time to check the critical value for β against the order parameter in the discrete model. This is an especially interesting point, since it would validate the derivation of the continuum limit further than the equilibrium limit. This is a very computationally heavy task though, since checking the phase transition near the critical point takes a long time. Due to the fact that the code was slow already, it was decided to leave it as further research.

6.2. Porting Code to C

As mentioned in the last section, the code that the discrete model ran on was slow. This is due to how Python works. The code in python in interpreted before it is executed. Therefore, some programs might turn out to be very slow. Despite the fact that the code was optimised during this project, it still took 2 seconds per iteration or more to run very intensive simulations. This obviously restricted some of the simulation parts. Therefore, it is recommended to port the code into C or something else as quick. This could open up the door to very extensive ensemble simulations, which would be good to extend the section on the determination of the critical β of a system.

6.3. Substrate dependent β

As mentioned in section 5.2, β can become dependent on location and time when the Derjaguin lengths of the free and bound enzyme are no longer the same. This occurs generally if the binding or unbinding of a ligand changes the surface chemistry of the enzyme [15]. If λ_f^2 and λ_b^2 are distinct values, we see from equations 2.18 and 2.19 that the cross diffusion term will be:

$$\left(\left[\lambda_f^2 + (\lambda_b^2 - \lambda_f^2)\frac{\rho_s}{K_m + \rho_s}\right]\frac{k_b T}{\eta} + (\lambda_b^2 - \lambda_f^2)\frac{K_m}{(K_m + \rho_s)^2}\right)\rho_e \nabla \rho_s. \tag{6.1}$$

Therefore, for equation 5.40 to agree with 6.1, it must hold that:

$$2\rho_e \nabla(\beta \rho_s) = \left(\left[\lambda_f^2 + (\lambda_b^2 - \lambda_f^2)\frac{\rho_s}{K_m + \rho_s}\right]\frac{k_b T}{\eta} + (\lambda_b^2 - \lambda_f^2)\frac{K_m}{(K_m + \rho_s)^2}\right)\rho_e \nabla \rho_s. \tag{6.2}$$

Assuming β to be a function of ρ_s, we can use the product rule on the left side to see that:

$$2\left(\beta+\rho_s\frac{d\beta}{d\rho_s}\right)\rho_e\nabla\rho_s=\left(\left[\lambda_f^2+(\lambda_b^2-\lambda_f^2)\frac{\rho_s}{K_m+\rho_s}\right]\frac{k_bT}{\eta}+(\lambda_b^2-\lambda_f^2)\frac{K_m}{(K_m+\rho_s)^2}\right)\rho_e\nabla\rho_s. \tag{6.3}$$

There are two options for the two sides to be equal. First, it could be that $\rho_e\nabla\rho_s=0$. From section 5.2.1, we see that this corresponds to an equilibrium situation, of system 5.41. This means that the uniformly distributed equilibrium solutions discussed in section 5.2.1 are also equilibrium solutions when β is a function of ρ_s Another non-trivial solution occurs when:

$$2\left(\beta+\rho_s\frac{d\beta}{d\rho_s}\right)=\left[\lambda_f^2+(\lambda_b^2-\lambda_f^2)\frac{\rho_s}{K_m+\rho_s}\right]\frac{k_bT}{\eta}+(\lambda_b^2-\lambda_f^2)\frac{K_m}{(K_m+\rho_s)^2}. \tag{6.4}$$

Now notice that $\left(\beta+\rho_s\frac{d\beta}{d\rho_s}\right)=\frac{d}{d\rho_s}(\beta\rho_s)$. Substituting this into equation 6.4 yields the following differential equation:

$$2\frac{d}{d\rho_s}(\beta\rho_s)=\left[\lambda_f^2+(\lambda_b^2-\lambda_f^2)\frac{\rho_s}{K_m+\rho_s}\right]\frac{k_bT}{\eta}+(\lambda_b^2-\lambda_f^2)\frac{K_m}{(K_m+\rho_s)^2}. \tag{6.5}$$

Taking the integral gives us the solution:

$$2\beta\rho_s=\int\left[\lambda_f^2+(\lambda_b^2-\lambda_f^2)\frac{\rho_s}{K_m+\rho_s}\right]\frac{k_bT}{\eta}+(\lambda_b^2-\lambda_f^2)\frac{K_m}{(K_m+\rho_s)^2}\,\mathrm{d}\rho_s. \tag{6.6}$$

This integral can be solved analytically. Long division gives the antiderivative of $\frac{\rho_s}{K_m+\rho_s}$:

$$\begin{aligned}\int\frac{\rho_s}{K_m+\rho_s}\,\mathrm{d}\rho_s&=\int 1-\frac{K_m}{K_m+\rho_s}\,\mathrm{d}\rho_s\\&=\rho_s-K_m\ln(K_m+\rho_s).\end{aligned} \tag{6.7}$$

Using this, we solve for $2\beta\rho_s$:

$$2\beta\rho_s=\left[\lambda_b^2\rho_s+(\lambda_f^2-\lambda_b^2)\ln(K_m+\rho_s)\right]+(\lambda_f^2-\lambda_b^2)\frac{K_m}{(K_m+\rho_s)^2}+C. \tag{6.8}$$

This implies that our model will agree with the model from 2.17 if β satisfies the following:

$$\beta=\frac{1}{2}\left[\lambda_b^2+(\lambda_f^2-\lambda_b^2)\frac{\ln(K_m+\rho_s)}{2}\right]+\frac{1}{2}(\lambda_f^2-\lambda_b^2)\frac{K_m}{(K_m+\rho_s)^2}+\frac{C}{\rho_s}. \tag{6.9}$$

There are infinitely many solutions for β in the form of equation 6.9, since the constant is arbitrary. It is a linear and constant term in each of the exponents in equation 3.3 that means that the contribution of C occurs in both the numerator and the denominator. Therefore, it is cancels and is thus arbitrary. Implementing this β into the model can be used to study enzymes for which the interaction length of the non-specific interaction is different for the bound and free enzyme.

6.4. 3D-Extension

The model that was programmed in this report was restricted to 2D. Real life interactions in cell of course happen in a 3D space. It would be interesting to see what a 3D-extension to the model would mean for the phase transition. Multiple questions arise immediately. Do the rice-grains appear again? How does the model react to the membrane being on the outside of the system? Does the system show instabilities at all? The 3D behaviour of the system would certainly help expand the realism.

7

Conclusion

In this report, we have derived an agent-based model for the diffusion of enzymes, their substrate and their product. The basis of this work were the interactions described by Canalejo et al. [1]. The segregating behaviour of the model was shown by looking at simulations. The dependence of the order in the system on β was explained and an order parameter was introduced to quantify the amount of pattern formation in the system.

The diffusion coefficient of urease enzymes undergoing repellent interactions with urea was calculated using a mean square distance method. The parameters used in the simulations to determine the diffusion coefficient were: $l = 7.43 \cdot 10^{-3}\,\text{cm}$, $C_s = 1$, $C_e = 43.4$, $k_{cat} = 5913\text{s}^{-1}$, $\gamma_s = 1\text{s}^{-1}$, $K_m = 4.2\,\text{mM}$ and $\beta = 1.76 \cdot 10^{-4}$. This resulted in an enhanced diffusion coefficient of $D' = 4.36 \cdot 10^{-7}\,\frac{\text{cm}^2}{\text{s}}$, a 37% increase from the regular diffusion coefficient. The behaviour of the discrete model also implied an experimentally testable prerequisite to test the role of non-specific interactions, pattern formation.

A continuum version of the model was derived after the simulations. The derivation and match of equilibrium solutions showed an agreement with the discrete model. A linear stability analysis provided a critical point for which a larger β causes the system to show pattern formation. The critical value for the urease system was determined to be: $\beta = 8.4 \cdot 10^{-5}$. Future work includes checking the critical β against the discrete model, an extension into 3D and a complication of β to include a dependence on the substrate to include specific interactions into the discrete model as well.

References

[1] Jaime Agudo-Canalejo, Pierre Illien, and Ramin Golestanian. "Phoresis and Enhanced Diffusion Compete in Enzyme Chemotaxis". In: *Nano Letters* 18.4 (Apr. 2018), pp. 2711–2717. ISSN: 1530-6984. DOI: 10.1021/acs.nanolett.8b00717. URL: https://doi.org/10.1021/acs.nanolett.8b00717.

[2] Bruce Alberts. *Molecular Bioloogy of the Cell*. 6th ed. Garland Science, 2015.

[3] Abdulaziz Alsenafi and Alethea B.T. Barbaro. "A convection–diffusion model for gang territoriality". In: *Physica A: Statistical Mechanics and its Applications* 510 (2018), pp. 765–786. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2018.07.004. URL: https://www.sciencedirect.com/science/article/pii/S0378437118308604.

[4] Abdulaziz Alsenafi and Alethea BT Barbaro. "A multispecies cross-diffusion model for territorial development". In: *Mathematics* 9.12 (2021), p. 1428.

[5] N. L. Carothers. *Real Analysis*. Cambridge University Press, 2000. DOI: 10.1017CBO9780511814228.

[6] Boris Vladimirovich Derjaguin et al. *Surface forces*. Springer, 1987.

[7] BV Derjaguin et al. "Kinetic phenomena in boundary films of liquids". In: *Kolloidn. zh* 9.01 (1947).

[8] A. Einstein. "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen". In: *Annalen der Physik* 322.8 (Jan. 1905), pp. 549–560. DOI: 10.1002/andp.19053220806.

[9] Dominique Ernst and Jürgen Köhler. "Measuring a diffusion coefficient by single-particle tracking: statistical analysis of experimental mean squared displacement curves". In: *Phys. Chem. Chem. Phys.* 15 (3 2013), pp. 845–849. DOI: 10.1039/C2CP43433D. URL: http://dx.doi.org/10.1039/C2CP43433D.

[10] Giovanni Giunta, Hamid Seyed-Allaei, and Ulrich Gerland. "Cross-diffusion induced patterns for a single-step enzymatic reaction". In: *Communications Physics* 3.1 (Sept. 2020), p. 167. ISSN: 2399-3650. DOI: 10.1038/s42005-020-00427-w. URL: https://doi.org/10.1038/s42005-020-00427-w.

[11] Ramin Golestanian. "Enhanced diffusion of enzymes that catalyze exothermic reactions". In: *Physical review letters* 115.10 (2015), p. 108102.

[12] Ramin Golestanian. "Phoretic active matter". In: *arXiv preprint arXiv:1909.03747* (2019).

[13] Richard Haberman. *Applied partial differential equations with Fourier series and boundary value problems*. 5th ed. Pearson, 2012.

[14] Pierre Illien et al. “Exothermicity is not a necessary condition for enhanced diffusion of enzymes”. In: *Nano letters* 17.7 (2017), pp. 4415–4420.

[15] Huan J. Keh. “Diffusiophoresis”. In: *Encyclopedia of Microfluidics and Nanofluidics*. Ed. by Dongqing Li. Boston, MA: Springer US, 2008, pp. 365–369. ISBN: 978-0-387-48998-8. DOI: 10.1007/978-0-387-48998-8_328. URL: https://doi.org/10.1007/978-0-387-48998-8_328.

[16] Barbara Krajewska. “Ureases I. Functional, catalytic and kinetic properties: A review”. In: *Journal of Molecular Catalysis B: Enzymatic* 59.1 (2009), pp. 9–21. ISSN: 1381-1177. DOI: https://doi.org/10.1016/j.molcatb.2009.01.003. URL: https://www.sciencedirect.com/science/article/pii/S1381117709000150.

[17] William E. Langlois and Michel O. Deville. “Lubrication Theory”. In: *Slow Viscous Flow*. Cham: Springer International Publishing, 2014, pp. 229–249. ISBN: 978-3-319-03835-3. DOI: 10.1007/978-3-319-03835-3_9. URL: https://doi.org/10.1007/978-3-319-03835-3_9.

[18] Hari S Muddana et al. “Substrate catalysis enhances single-enzyme diffusion”. In: *Journal of the American Chemical Society* 132.7 (2010), pp. 2110–2111.

[19] Edward M Purcell. “Life at low Reynolds number”. In: *American journal of physics* 45.1 (1977), pp. 3–11.

[20] Clement Riedel et al. “The heat released during catalytic turnover enhances the diffusion of an enzyme”. In: *Nature* 517.7533 (Jan. 2015), pp. 227–230. ISSN: 1476-4687. DOI: 10.1038/nature14043. URL: https://doi.org/10.1038/nature14043.

[21] K.A.J. White, M.A. Lewis, and J.D. Murray. “A Model for Wolf-Pack Territory Formation and Maintenance”. In: *Journal of Theoretical Biology* 178.1 (1996), pp. 29–43. ISSN: 0022-5193. DOI: https://doi.org/10.1006/jtbi.1996.0004. URL: https://www.sciencedirect.com/science/article/pii/S0022519396900041.

[22] Jochen Winkelmann. *Diffusion coefficient of urea in water at infinite dilution: Datasheet from Physical Chemistry · Volume 15B2: “Diffusion in Gases, Liquids and Electrolytes” in SpringerMaterials (https://doi.org/10.1007/978-3-662-54089-3_1386)*. Ed. by M. D. Lechner. accessed 2022-06-28. DOI: 10.1007/978-3-662-54089-3_1386. URL: https://materials.springer.com/lb/docs/sm_lbs_978-3-662-54089-3_1386.

[23] Yifei Zhang and Henry Hess. “Enhanced Diffusion of Catalytically Active Enzymes”. In: *ACS Central Science* 5.6 (2019). PMID: 31263753, pp. 939–948. DOI: 10.1021/acscentsci.9b00228. eprint: https://doi.org/10.1021/acscentsci.9b00228. URL: https://doi.org/10.1021/acscentsci.9b00228.

A

Dimensional Analysis

In this appendix we will analyse the dimensional homogeneity of the system derived in the continuum limit sections of this report. A dimensional analysis is a crucial first step in validating the equations that were derived in this report. We will begin by looking at the main system of partial differential equations for ρ_s, ρ_e and ρ_p, system (5.41). The following physical quantities appear in the PDEs: D_e, D_p, D_s, ρ_e, ρ_p, ρ_s, k_{cat}, K_m, β, γ_p, γ_s and s_R. The units of these quantities are shown in the table below.

Symbol	Definition	Unit
D_e	Enzyme diffusion coefficient	$[\frac{\mathrm{m}^2}{\mathrm{s}}]$
D_p	Product diffusion coefficient	$[\frac{\mathrm{m}^2}{\mathrm{s}}]$
D_s	Substrate diffusion coefficient	$[\frac{\mathrm{m}^2}{\mathrm{s}}]$
k_{cat}	Enzyme turnover number	$[\frac{1}{\mathrm{s}}]$
K_m	Enzyme turnover number	$[\frac{\mathrm{mol}}{\mathrm{L}}]$
s_R	Reservoir substrate density	$[\frac{\mathrm{mol}}{\mathrm{L}}]$
β	Dependence/separation constant	$[\frac{\mathrm{L}}{\mathrm{mol}}]$
γ_p	Product permeation rate	$[\frac{1}{\mathrm{s}}]$
γ_s	Substrate permeation rate	$[\frac{1}{\mathrm{s}}]$
ρ_e	Enzyme density	$[\frac{\mathrm{mol}}{\mathrm{L}}]$
ρ_p	Product density	$[\frac{\mathrm{mol}}{\mathrm{L}}]$
ρ_s	Substrate density	$[\frac{\mathrm{mol}}{\mathrm{L}}]$

The units for β were derived by looking at the enzyme's probability to move to a neighbouring site:

$$\mathbb{P}_e(x_1 \to x_2, y_1 \to y_2, t) = C_e \frac{e^{-\beta s(x_2,y_2,t)}}{\sum_{(\tilde{x},\tilde{y})\sim(x_1,y_1)} e^{-\beta s((\tilde{x},\tilde{y}),t)}}. \tag{A.1}$$

This is a probability, so the total expression should be dimensionless. Therefore it must hold that $\beta s(x, y, t)$ is dimensionless:

$$\Big[\beta s(x,y,t)\Big] = \Big[\beta\Big]\left[\frac{\mathrm{mol}}{\mathrm{L}}\right] = \Big[\quad \cdot \quad\Big]. \tag{A.2}$$

From equation (A.2) it follows that:

$$\left[\beta\right] = \left[\frac{\mathrm{L}}{\mathrm{mol}}\right] \tag{A.3}$$

Now, we can check the dimensions of the PDEs themselves. We begin by looking at equation (5.22):

$$\frac{\partial \rho_s(\mathbf{v},t)}{\partial t} = \frac{D_s}{2n}\Delta\rho_s - \rho_e k_{cat}\frac{\rho_s}{K_m+\rho_s} - \gamma_s\left(\rho_s - s_R\right). \tag{A.4}$$

The units here are:

$$\left[\frac{\mathrm{mol}}{\mathrm{Ls}}\right] = \left[\frac{\mathrm{m}^2}{\mathrm{s}}\right]\left[\frac{\mathrm{mol}}{\mathrm{Lm}^2}\right] - \left[\frac{\mathrm{mol}}{\mathrm{L}}\right]\left[\frac{1}{\mathrm{s}}\right]\frac{\left[\frac{\mathrm{mol}}{\mathrm{L}}\right]}{\left[\frac{\mathrm{mol}}{\mathrm{L}}\right]+\left[\frac{\mathrm{mol}}{\mathrm{L}}\right]} - \left[\frac{1}{\mathrm{s}}\right]\left(\left[\frac{\mathrm{mol}}{\mathrm{L}}\right] - \left[\frac{\mathrm{mol}}{\mathrm{L}}\right]\right). \tag{A.5}$$

Indeed we see that the units on both the lefthand side and righthand side reduce to $\left[\frac{\mathrm{mol}}{\mathrm{Ls}}\right]$, which means equation 5.22 is dimensionally correct. Equation 5.23 can be treated similarly:

$$\frac{\partial \rho_s(\mathbf{v},t)}{\partial t} = \frac{D_s}{2n}\Delta\rho_s - \rho_e k_{cat}\frac{\rho_s}{K_m+\rho_s} - \gamma_s\left(\rho_s - s_R\right). \tag{A.6}$$

Writing down the units, we obtain:

$$\left[\frac{\mathrm{mol}}{\mathrm{Ls}}\right] = \left[\frac{\mathrm{m}^2}{\mathrm{s}}\right]\left[\frac{\mathrm{mol}}{\mathrm{Lm}^2}\right] + \left[\frac{\mathrm{mol}}{\mathrm{L}}\right]\left[\frac{1}{\mathrm{s}}\right]\frac{\left[\frac{\mathrm{mol}}{\mathrm{L}}\right]}{\left[\frac{\mathrm{mol}}{\mathrm{L}}\right]+\left[\frac{\mathrm{mol}}{\mathrm{L}}\right]} - \left[\frac{1}{\mathrm{s}}\right]\left[\frac{\mathrm{mol}}{\mathrm{L}}\right]. \tag{A.7}$$

Again, all the units reduce to $\left[\frac{\mathrm{mol}}{\mathrm{Ls}}\right]$, as expected. Lastly, the equation 5.40 will be examined:

$$\frac{\partial \rho_e(\mathbf{v},t)}{\partial t} = \frac{D_e}{2n}\nabla\cdot\Big(\nabla\rho_e + 2\rho_e\nabla(\beta\rho_s)\Big). \tag{A.8}$$

Analysing the units in this equation yields:

$$\left[\frac{\mathrm{mol}}{\mathrm{Ls}}\right] = \left[\frac{\mathrm{m}^2}{\mathrm{s}}\right]\left[\frac{1}{\mathrm{m}}\right]\cdot\left(\left[\frac{\mathrm{mol}}{\mathrm{Lm}}\right] + \left[\frac{\mathrm{mol}}{\mathrm{L}}\right]\left[\frac{1}{\mathrm{m}}\right]\left[\frac{\mathrm{L}}{\mathrm{mol}}\right]\left[\frac{\mathrm{mol}}{\mathrm{L}}\right]\right). \tag{A.9}$$

Once more, we see that all the terms' units correspond to $\left[\frac{\mathrm{mol}}{\mathrm{Ls}}\right]$. Therefore, the system of PDEs is dimensionally correct.

We can also look at the expressions derived for the homogeneous solutions ($\bar{\rho_e}$, $\bar{\rho_p}$, $\bar{\rho_s}$) and check to see if their units match. First, note that $\bar{\rho_e}$, $\bar{\rho_p}$ and $\bar{\rho_s}$ are concentrations and thus have $\left[\frac{\mathrm{mol}}{\mathrm{L}}\right]$ as their units. Equation 5.49 is:

$$\bar{\rho_s} = \frac{1}{2}\left(s_R - K_m - \frac{k_{cat}}{\gamma_s}\bar{\rho_e} + \sqrt{4 s_R K_m + \left[\frac{k_{cat}}{\gamma_s}\bar{\rho_e} + (K_m - s_R)\right]^2}\right). \tag{A.10}$$

Its terms units are:

$$\left[\frac{\text{mol}}{\text{L}}\right] = \left[\frac{\text{mol}}{\text{L}}\right] - \left[\frac{\text{mol}}{\text{L}}\right] - \frac{\left[\frac{1}{\text{s}}\right]}{\left[\frac{1}{\text{s}}\right]}\left[\frac{\text{mol}}{\text{L}}\right] + \sqrt{\left[\frac{\text{mol}}{\text{L}}\right]\left[\frac{\text{mol}}{\text{L}}\right] + \left(\frac{\left[\frac{1}{\text{s}}\right]}{\left[\frac{1}{\text{s}}\right]}\left[\frac{\text{mol}}{\text{L}}\right] + \left[\frac{\text{mol}}{\text{L}}\right] - \left[\frac{\text{mol}}{\text{L}}\right]\right)^2}. \tag{A.11}$$

It is easily verified that everything before the square root is in units of $\left[\frac{\text{mol}}{\text{L}}\right]$. Both terms inside the square root simplify to $\left[\frac{\text{mol}^2}{\text{L}^2}\right]$ and therefore, all the units match. Similarly, equation 5.52 is:

$$\frac{\bar{\rho}_p = \frac{\gamma_s}{2\gamma_p}\left(s_R + K_m + \frac{k_{cat}}{\gamma_s}\bar{\rho}_e - \sqrt{4s_R K_m + \left[\frac{k_{cat}}{\gamma_s}\bar{\rho}_e + (K_m - s_R)\right]^2}\right)}{.} \tag{A.12}$$

If we look at the units in this equation, we find that:

$$\left[\frac{\text{mol}}{\text{L}}\right] = \left[\frac{\text{mol}}{\text{L}}\right] + \left[\frac{\text{mol}}{\text{L}}\right] + \frac{\left[\frac{1}{\text{s}}\right]}{\left[\frac{1}{\text{s}}\right]}\left[\frac{\text{mol}}{\text{L}}\right] - \sqrt{\left[\frac{\text{mol}}{\text{L}}\right]\left[\frac{\text{mol}}{\text{L}}\right] + \left(\frac{\left[\frac{1}{\text{s}}\right]}{\left[\frac{1}{\text{s}}\right]}\left[\frac{\text{mol}}{\text{L}}\right] + \left[\frac{\text{mol}}{\text{L}}\right] - \left[\frac{\text{mol}}{\text{L}}\right]\right)^2}. \tag{A.13}$$

It is in the same form as equation (5.49) and thus all the units on the right side of the equation again reduce to $\left[\frac{\text{mol}}{\text{L}}\right]$

Lastly, we look at if the units in the inequality for the critical β match each other. Recall that equation (5.74) is:

$$\beta > \frac{1}{2\bar{\rho}_s} + \frac{1}{2}\frac{1}{K_m + \bar{\rho}_s} + \frac{\gamma_s(K_m + \bar{\rho}_s)}{2k_{cat}\bar{\rho}_s\bar{\rho}_e}. \tag{A.14}$$

Using equation (A.3), equation (5.74) gives the following result:

$$\left[\frac{\text{L}}{\text{mol}}\right] > \left[\frac{\text{L}}{\text{mol}}\right] + \left[\frac{\text{L}}{\text{mol}}\right] + \frac{1}{\left[\frac{\text{mol}}{\text{L}}\right] + \left[\frac{\text{mol}}{\text{L}}\right]} + \frac{\left[\frac{1}{\text{s}}\right]\left(\left[\frac{\text{mol}}{\text{l}}\right]\right)}{\left[\frac{1}{\text{s}}\right]\left[\frac{\text{mol}}{\text{l}}\right]\left[\frac{\text{mol}}{\text{l}}\right]}. \tag{A.15}$$

Everything on the right side of the inequality reduces to $\left[\frac{\text{L}}{\text{mol}}\right]$, which is what was expected.

Thus we have checked the dimensionality of all of the equations derived in section 5, and we have found them to be dimensionally correct.

B

Taylor expansion

In this appendix it will be shown that:

$$\frac{1}{a+h} = \frac{1}{a} - \frac{h}{a^2} + \mathcal{O}(h^2) \tag{B.1}$$

using Taylor's theorem [5]. Recall that from Taylor's theorem, it holds that for small values of h,

$$f(a+h) = f(x_i) + h\frac{\partial f(a)}{\partial a} + \mathcal{O}(h^2), \tag{B.2}$$

Now let $f(x) = \frac{1}{x}$. Then, $\frac{\partial f(a)}{\partial a} = -\frac{1}{a^2}$. Consequently, equation (B.2) gives

$$\frac{1}{a+h} = \frac{1}{a} - \frac{h}{a^2} + \mathcal{O}(h^2). \tag{B.3}$$

C

Code

```python
import numpy as np
import matplotlib.pyplot as plt
import random
from tqdm import tqdm
from scipy.stats import poisson, binom, bernoulli
from scipy import signal
from scipy.optimize import curve_fit

#defining the partition function that will be used for stepping
def partfunc(beta, lat, loc, size):
    u = np.exp(-(beta)*lat[(loc[0] + 1)%size, loc[1]])
    d = np.exp(-(beta)*lat[(loc[0] - 1)%size, loc[1]])
    l = np.exp(-(beta)*lat[loc[0], (loc[1] - 1)%size])
    r = np.exp(-(beta)*lat[loc[0], (loc[1] + 1)%size])
    z = u + d + l + r
    return(np.array([d/z, (d+r)/z, (d+r+u)/z, (d+r+u+l)/z]))

def order(lat, dep):
    diflat = lat #- dep
    sumlat = lat #+ dep
    filt = np.array([[0, 1, 0],
                  [1,0,1],
                  [0,1,0]])
    convlat = diflat*signal.convolve2d(diflat, filt, mode = 'same', boundary = 'wrap')
    return(np.sum(convlat, axis = None)/(4*np.sum(sumlat, axis = None)**2))

def autocorrFFT(x):
  N=len(x)
  F = np.fft.fft(x, n=2*N)  #2*N because of zero-padding
  PSD = F * F.conjugate()
  res = np.fft.ifft(PSD)
  res= (res[:N]).real   #now we have the autocorrelation in convention B
  n=N*np.ones(N)-np.arange(0,N) #divide res(m) by (N-m)
  return(res/n) #this is the autocorrelation in convention A

def msd_fft(r):
  N=len(r)
  D=np.square(r).sum(axis=1)
  D=np.append(D,0)
  S2=sum([autocorrFFT(r[:, i]) for i in range(r.shape[1])])
  Q=2*D.sum()
  S1=np.zeros(N)
  for m in range(N):
      Q=Q-D[m-1]-D[N-m]
      S1[m]=Q/(N-m)
  return(S1-2*S2)

#the particle class containing all properties we need
class particle:
    #setup functions
    def __init__(self, pop, size):
        self.reset(pop, size)
        self.size = size
        self.pop = pop

    def reset(self, pop, n):
        self.lattice = np.random.multinomial(int(pop/n), [1/n]*n, size = n)

    def latupdate(self, newlat):
        self.lattice = np.copy(newlat)

    def plot(self):
        plt.imshow(self.lattice)

    #walking functions
    def depwalk(self, beta, dep, prop):
        parts = np.dstack(np.nonzero(self.lattice))
        newlat = np.zeros((self.size, self.size))
        for part in parts[0]:
            chance = prop*partfunc(beta, dep, part, self.size)
            for i in range(0,int(self.lattice[part[0], part[1]])):
                dice = np.random.uniform()
                if dice < chance[0]:
                    newlat[(part[0] - 1)%self.size, part[1]] += 1
                elif dice < chance[1]:
                    newlat[part[0], (part[1] + 1)%self.size] += 1
                elif dice < chance[2]:
                    newlat[(part[0] + 1)%self.size, part[1]] += 1
                elif dice <= chance[3]:
                    newlat[part[0], (part[1] - 1)%self.size] += 1
                else:
                    newlat[part[0], part[1]] += 1
        self.latupdate(newlat)
        return(newlat)

    def depwalk2(self, beta, dep, prop):
        parts = np.dstack(np.nonzero(self.lattice))
        newlat = np.zeros((self.size, self.size))
        for loc in parts[0]:
            chance = prop*partfunc(beta, dep, loc, self.size)
            dices = np.random.uniform(size = int(self.lattice[loc[0], loc[1]]))
            additions = np.histogram(dices, bins=[0, chance[0], chance[1], chance[2], chance[3]])
            newlat[(loc[0] - 1)%self.size, loc[1]] += additions[0][0]
            newlat[loc[0], (loc[1] + 1)%self.size] += additions[0][1]
            newlat[(loc[0] + 1)%self.size, loc[1]] += additions[0][2]
            newlat[loc[0], (loc[1] - 1)%self.size] += additions[0][3]
            newlat[loc[0], loc[1]] += self.lattice[loc[0], loc[1]] - np.sum(additions[0])
        self.latupdate(newlat)
        return(newlat)

    def depwalk3(self, beta, dep, prop):
        newlat = np.zeros((self.size, self.size))
        for iy, ix in np.ndindex(self.lattice.shape):
            if self.lattice[iy, ix] <= 0:
                pass
            else:
                chance = prop*partfunc(beta, dep, [iy, ix], self.size)
                dices = np.random.uniform(size = int(self.lattice[iy, ix]))
                additions = np.histogram(dices, bins=[0, chance[0], chance[1], chance[2], chance[3]])
                newlat[(iy - 1)%self.size, ix] += additions[0][0]
                newlat[iy, (ix + 1)%self.size] += additions[0][1]
                newlat[(iy + 1)%self.size, ix] += additions[0][2]
                newlat[iy, (ix - 1)%self.size] += additions[0][3]
                newlat[iy, ix] += self.lattice[iy, ix] - np.sum(additions[0])
        self.latupdate(newlat)
        return(newlat)

    def walk3(self):
        newlat = np.zeros((self.size, self.size))
        for iy, ix in np.ndindex(self.lattice.shape):
            if self.lattice[iy, ix] <= 0:
                pass
            else:
                dices = np.random.uniform(size = int(self.lattice[iy, ix]))
                additions = np.histogram(dices, bins=[0, 0.25, 0.5, 0.75, 1])
                newlat[(iy - 1)%self.size, ix] += additions[0][0]
                newlat[iy, (ix + 1)%self.size] += additions[0][1]
                newlat[(iy + 1)%self.size, ix] += additions[0][2]
                newlat[iy, (ix - 1)%self.size] += additions[0][3]
                #newlat[iy, ix] += self.lattice[iy, ix] - np.sum(additions[0])
        self.latupdate(newlat)
        return(newlat)

    def walk2(self):
        parts = np.dstack(np.nonzero(self.lattice))
        newlat = np.zeros((self.size, self.size))
        for loc in parts[0]:
            dices = np.random.uniform(size = int(self.lattice[loc[0], loc[1]]))
```

```python
            additions = np.histogram(dices, bins=[0, 0.25, 0.5, 0.75, 1])
            newlat[(loc[0] - 1)%self.size, loc[1]] += additions[0][0]
            newlat[loc[0], (loc[1] + 1)%self.size] += additions[0][1]
            newlat[(loc[0] + 1)%self.size, loc[1]] += additions[0][2]
            newlat[loc[0], (loc[1] - 1)%self.size] += additions[0][3]
            #newlat[loc[0], loc[1]] += self.lattice[loc[0], loc[1]] - np.sum(additions[0])
        self.latupdate(newlat)
        return(newlat)

    def walk(self):
        parts = np.dstack(np.nonzero(self.lattice))
        newlat = np.zeros((self.size, self.size))
        for part in parts[0]:
            for i in range(0,int(self.lattice[part[0], part[1]])):
                dice = np.random.uniform()
                if dice < 0.25:
                    newlat[(part[0] - 1)%self.size, part[1]] += 1
                elif dice < 0.5:
                    newlat[part[0], (part[1] + 1)%self.size] += 1
                elif dice < 0.75:
                    newlat[(part[0] + 1)%self.size, part[1]] += 1
                else:
                    newlat[part[0], (part[1] - 1)%self.size] += 1
        self.latupdate(newlat)
        return(newlat)

    #reacting and influx
    def react(self, kcat, dep, prod):
        enz = np.dstack(np.nonzero(dep.lattice))
        for locs in enz[0]:
            reacts = min(poisson.rvs(dep.lattice[locs[0], locs[1]] * kcat), self.lattice[locs[0], locs[1]])
            #reacts = binom.rvs(int(dep.lattice[locs[0], locs[1]]), kcat*(self.lattice[locs[0], locs[1]]/(1+self.lattice[locs[0], locs[1]])))
            self.lattice[locs[0], locs[1]] -= reacts
            prod.lattice[locs[0], locs[1]] += reacts
        return(self.lattice)

    def react2(self, kcat, dep, prod):
        enz = np.dstack(np.nonzero(dep.lattice))
        reacts = np.zeros((self.size, self.size))
        for locs in enz[0]:
            reacts[locs[0], locs[1]] = min(poisson.rvs(dep.lattice[locs[0], locs[1]] * kcat), self.lattice[locs[0], locs[1]])
            #reacts = binom.rvs(int(dep.lattice[locs[0], locs[1]]), kcat*(self.lattice[locs[0], locs[1]]/(1+self.lattice[locs[0], locs[1]])))
        self.lattice = self.lattice - reacts
        prod.lattice = prod.lattice + reacts
        return(self.lattice)

    def react3(self, kcat, km, dep, prod):
        depflat = np.ndarray.flatten(dep.lattice)
        selfflat = np.ndarray.flatten(self.lattice)
        reacts = np.array([binom.rvs(int(depflat[i]), kcat*(selfflat[i]/(km+selfflat[i]))) for i in range(self.size**2)]).reshape((self.size, self.size))
        self.lattice = self.lattice - reacts
        prod.lattice = prod.lattice + reacts
        return(self.lattice)

    def influx(self, gamma, s_r):
        for iy, ix in np.ndindex(self.lattice.shape):
            self.lattice[iy, ix] += poisson.rvs(s_r*gamma) - poisson.rvs(self.lattice[iy, ix]*gamma)
        return(self.lattice)

    def influx2(self, gamma, s_r):
        inflow = np.array([poisson.rvs(s_r*gamma) - poisson.rvs(rho_s*gamma) for rho_s in np.ndarray.flatten(self.lattice)]).reshape((self.size, self.size))
        self.lattice = self.lattice + inflow
        return(self.lattice)

class s_particle:
    def __init__(self, dim, pop):
        self.dim = dim
        self.pop = pop
        self.reset(pop)

    def reset(self, i):
        self.coords = np.empty(i, dtype=np.ndarray)
        for j in range(i):
            self.coords[j] = np.array([np.random.randint(self.dim, size=2)])
        self.histry = np.copy(self.coords)

    def histupdate(self, i):
        for j in range(i):
            self.histry[j] = np.append(self.histry[j], self.coords[j], axis=0)

    def depwalk(self, beta, dep, prop):
        for j in range(self.pop):
            chance = prop*partfunc(beta, dep, self.coords[j][0], self.dim)
            dice = np.random.uniform()
            if dice < chance[0]:
                self.coords[j][0][0] = (self.coords[j][0][0] - 1)%self.dim #-y
            elif dice < chance[1]:
                self.coords[j][0][1] = (self.coords[j][0][1] + 1)%self.dim #+x
            elif dice < chance[2]:
                self.coords[j][0][0] = (self.coords[j][0][0] + 1)%self.dim #+y
            elif dice <= chance[3]:
                self.coords[j][0][1] = (self.coords[j][0][1] - 1)%self.dim #-x
            else:
                pass
        self.histupdate(self.pop)
        return()

    def walk(self, prop):
        for j in range(self.pop):
            dice = np.random.uniform()
            if dice < prop*0.25:
                self.coords[j][0][0] = (self.coords[j][0][0] - 1)%self.dim #-y
            elif dice < prop*0.5:
                self.coords[j][0][1] = (self.coords[j][0][1] + 1)%self.dim #+x
            elif dice < prop*0.75:
                self.coords[j][0][0] = (self.coords[j][0][0] + 1)%self.dim #+y
            elif dice <= prop:
                self.coords[j][0][1] = (self.coords[j][0][1] - 1)%self.dim #-x
            else:
                pass
        self.histupdate(self.pop)
        return()

    def MSD(self):
        MSDi = np.zeros((self.pop, len(self.histry[0])))
        for k in range(self.pop):
            MSDi[k] = msd_fft(self.histry[k])
        MSD = np.mean(MSDi, axis=0)
        return(MSD)
```

D

Determining the Diffusion Coefficient

In the discrete model derived in section 3, we let particles randomly walk across a lattice. In the continuous case, this corresponds to diffusion, as mentioned earlier. Following Einstein, it can be determined that if N particles starting from the origin at time $t = 0$ randomly diffuse in 1D, the solution to the diffusion equation yields [8]:

$$\rho(x,t) = \frac{N}{\sqrt{4\pi Dt}} e^{-\frac{x^2}{4Dt}}. \tag{D.1}$$

This corresponds to a normal distribution with mean zero and variance $\sigma^2 = 2Dt$. This means that the second moment can be determined as:

$$\overline{x}^2 = 2Dt. \tag{D.2}$$

In 2D this relation turns out to be:

$$\overline{x}^2 = 4Dt. \tag{D.3}$$

This means that on our lattice, we can estimate the diffusion coefficient by tracking particles and calculating an ensemble average of their mean squared displacement: $\langle|\mathbf{x}(t+\tau) - \mathbf{x}(t)|^2\rangle$, where $\mathbf{x}(t)$ is the position $(x(t), y(t))$ of a particle on the lattice at time t [9]. The mean squared displacement of particles on the lattice can be calculated using the ensemble average:

$$\langle|\mathbf{x}(t+\tau) - \mathbf{x}(t)|^2\rangle = \frac{1}{N}\sum_{i=1}^{N} |x^i(t+\tau) - x^i(t)|^2 + |y^i(t+\tau) - y^i(t)|^2. \tag{D.4}$$

We can then fit for the diffusion coefficient using the following expression:

$$\langle|\mathbf{x}(t+\tau) - \mathbf{x}(t)|^2\rangle = 4D\tau. \tag{D.5}$$

The enzymes in the model were tracked using a *ghost particle method*. These particles use the same movement dynamics as the other enzymes in the model, but they do not participate in the reactions between the substrate and the enzyme. In this way, an big ensemble could be tested without having to increase the substrate population, which would slow down computations significantly. The ghost particles' initial locations were not randomised across the lattice. They all started at random locations which were at most 5 lattice points away from the origin. This was done to make sure that the ghost particles would not reach the edge in the simulations as this would skew the mean squared difference measurements due to the periodic boundary conditions.

E

Drift-Velocity of Non-Specific interactions

Non-specific interactions cause a drift velocity in response to the concentration gradient of a molecular solute. This drift velocity is derived by analysing the conservation equation of a molecular solute under the influence of a potential $\phi(y)$, where y is the distance from the plate[12]. Let us then analyse a stationary, incompressible medium under constant pressure, in which there is a non-zero substrate gradient, $\nabla\rho_{\mathbf{s}} \neq 0$. The enzyme's surface thus exerts a force on a substrate molecule, which is:

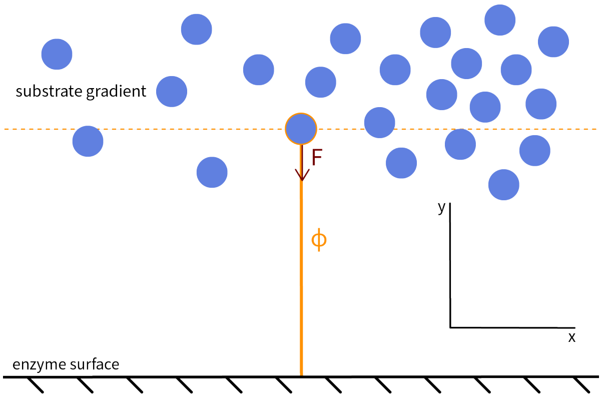


Figure E.1: A zoom-in of an enzyme's surface in a medium with a substrate gradient. The substrate molecule experiences a force **F** due to ϕ.

$$\mathbf{F} = -\nabla\phi(r). \tag{E.1}$$

This means that a group of particles undergo a body force **f**. Therefore, at distance x from the molecule:

$$\mathbf{f} = \rho_s \mathbf{F} = \rho_s \left(-\nabla\phi(r)\right), \tag{E.2}$$

where ρ_s is the concentration of the substrate. Following this, we can write down a general conservation equation in the same sense as equation (2.10). The cause of movement is now not only free diffusion, but also the body force. There is no advection term, since we consider situations where the fluid is stationary. Therefore:

$$\frac{\partial \rho_s}{\partial t} + \nabla \cdot J = 0, \tag{E.3}$$

where J is now written as:

$$J = -D\nabla\rho_s + \frac{D}{k_B T}\mathbf{f}. \tag{E.4}$$

This implies that the divergence of the flux term is:

$$\nabla \cdot J = -D\Delta\rho_s + \frac{D}{k_B T}\nabla \cdot \mathbf{f}, \tag{E.5}$$

assuming that the diffusion coefficient of the substrate is constant. This is a reasonable assumption, since the substrate's diffusive properties only change when it is converted into the product.

The new body force term in equation (E.5) contains a complication. Since the **f** occurs due to the non-specific interactions that were mentioned earlier, it has singular behaviour. The potential used to model the Van Der Waals interaction, for example, has a dependence on $\frac{1}{r^6}$. This means that the substrate particle only suddenly experiences the potential when it comes close enough to the surface of the enzyme. Furthermore, it means that there is a discontinuity in the flux in the neighbourhood of the surface. This is of course a problem if we want to take the gradient. We will try to solve this by using the Stokes flow equation: a linearisation of the Navier-Stokes equations for fluid flow in low Reynolds numbers environments like cells. The equation of motion near the enzyme wall is:

$$-\eta\Delta\mathbf{v} = -\nabla\mathbf{p} + \mathbf{f}, \tag{E.6}$$

where η is the viscosity of the fluid, p the pressure and v the velocity of the fluid [12]. As mentioned earlier, the medium in the systems that we are considering is incompressible, which means that $\nabla \cdot \mathbf{v} = 0$. Therefore, in our situation equation (E.6) reduces to:

$$-\nabla\mathbf{p} + \mathbf{f} = 0. \tag{E.7}$$

Taking the divergence of equation (E.7), we obtain:

$$-\Delta\mathbf{p} + \nabla \cdot \mathbf{f} = 0, \tag{E.8}$$

such that:

$$\begin{aligned} \nabla J &= -D\Delta\rho_s + \frac{D}{k_B T}\Delta\mathbf{p} \\ &= Dk_B T\Delta\left(\mathbf{p} - k_B T\rho_s\right). \end{aligned} \tag{E.9}$$

Very close to the boundary, $J_y = 0$, since the boundary is solid and thus $\frac{\partial(\rho_s)_y}{\partial t} = 0$. In other words, there is no transport from the boundary into the medium. This means that in the y-direction, the substrate density near the enzyme's surface, ρ_s^S, is:

$$-D\frac{\partial\rho_s^S}{\partial y} + \frac{D}{k_B T}\rho_s^S\left(-\frac{\partial\phi}{\partial y}\right) = 0. \tag{E.10}$$

This can be rewritten as:

$$\frac{\partial \rho_s^S}{\partial y} = \frac{D}{k_B T}\rho_s^S\left(-\frac{\partial \phi}{\partial y}\right), \tag{E.11}$$

which is a separable equation with the solution:

$$\rho_s^S = Ce^{-\frac{\phi}{k_B T}}. \tag{E.12}$$

It was mentioned earlier that potential of the non-specific interactions is dependent on $\frac{1}{r^\alpha}$, where $\alpha \geq 1$. The potential then goes to zero very quickly once we are some distance removed from the surface of the enzyme. We can thus separate figure E.1 into two distinct regions: the surface region and the outer region. This is illustrated in figure E.2.

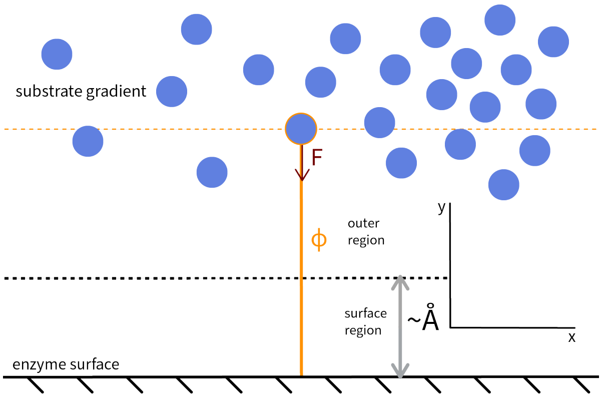


Figure E.2: The region where a substrate molecule undergoes an interaction with the potential ϕ is called the *surface region*. This region is several Angström high.

From this distinction we can determine the boundary conditions that equation (E.13) is subject to. Namely, at the boundary there is no substrate and in the outer regions, where $\phi \approx 0$, the substrate density is simply ρ_s. So, it is clear that $C = \rho_s$ and:

$$\rho_s^S = \rho_s e^{-\frac{\phi}{k_B T}}. \tag{E.13}$$

We can now use equation (E.13) to solve the continuity equation for when the density has reached steady state near the enzyme's surface:

$$\Delta\left(\mathbf{p} - k_B T \rho_s\right) = 0. \tag{E.14}$$

The steady state assumption is permitted, since the surface region is only several Angström wide, while the diffusion coefficient of the substrate is in the order of micrometers. We then make the approximation that $\mathbf{p} - k_B T\rho_s = K$, where K is an arbitrary constant. We do this, because this approximation implies that on a small strip of fluid, the pressure smoothly increases towards the enzyme. In other words, there is a pressure balance. This implies that the pressure near the enzyme surface, $\mathbf{p}^\mathbf{S}$, is:

$$\begin{aligned}\mathbf{p}^\mathbf{S} &= \mathbf{p}_\mathbf{out} + k_B T(\rho_s^S - \rho_s) \\ &= \mathbf{p}_\mathbf{out} + k_B T\rho_s\left(e^{-\frac{\phi}{k_B T}} - 1\right).\end{aligned} \tag{E.15}$$

In the outer layer, the pressure generated by ϕ plays no role. Thus, $\mathbf{p} = \mathbf{p_{out}}$ in this region. However, directly near the surface of the enzyme, there is no substrate gradient. Either there is no substrate at all in the case of repulsive forces, or there is a constant concentration through accumulation from attractive forces. We assumed that $\mathbf{p} - k_B T \rho_\mathbf{s} = K$, which then implies that there is a parallel pressure gradient at the enzyme surface [11]. An overview of the system is shown in figure E.3.

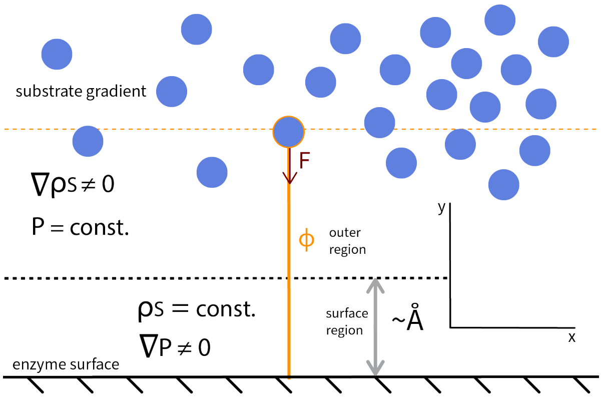


Figure E.3: The effect of the potential on the substrate gradient near the surface is a constant accumulation, or no substrate at all. This causes a pressure gradient near the surface.

We can now write down Stokes' equation in the direction parallel to the enzyme:

$$\eta \Delta v_x = \frac{\partial \mathbf{p}}{\partial x}, \tag{E.16}$$

where the body force is now zero, since the interaction of the substrate with the enzyme is perpendicular to the enzyme's surface. Since the surface layer is only several Angströms thick, there is a big difference in scale between the parallel and the perpendicular surface layer. An approximation like in lubrication theory can thus be applied to equation (E.17) [17]:

$$\frac{\partial v_x}{\partial y} = -\frac{k_B T}{\eta}\left(1 - e^{-\frac{\phi}{k_B T}}\right)\frac{\partial \rho_s}{\partial x}. \tag{E.17}$$

From this, the asymptotic average value of the velocity can be determined to be:

$$v_x|_{y=\infty} = \frac{k_b T}{\eta}\left[\int_0^\infty y\left(e^{-\phi(y)/k_b T} - 1\right) dy\right]\frac{\partial \rho_s}{\partial x}, \tag{E.18}$$

where we take the integral to $y = \infty$ due to the fact that the surface region is very thin. This integral is subject to a no-slip condition or something more lenient at $r = 0$ depending on the medium and a no stress condition at $r = \infty$ [6].

F

Order Parameter

The order parameter is defined in a similar way to how it is introduced by Alsenafi et al. [4, 3]. In this model it is only dependent on the enzyme molecules. Therefore:

$$O(t) = \left(\frac{1}{2N}\right) \sum_{(x,y)\in G} \sum_{(\tilde{x},\tilde{y})\sim(x,y)} e(x,y)e((\tilde{x},\tilde{y})), \tag{F.1}$$

where G is the lattice that it is defined on. For the calculation of its upper and lowe limit we once again refer to Alsenafi et al. [4, 3].