Stochastic chemical kinetics and application to enzyme kinetics

Alessandro Lo Cuoco, Sara Sponghi

- Department of Physics and Astronomy, Bologna -
- Physical methods of biology -

Abstract

This scientific report explores enzyme kinetics and the mathematical models used to study enzyme-substrate reactions. It covers fundamental concepts such as the mass action law, chemical master equation (CME), single-substrate enzyme-catalyzed reactions and the Goldbeter-Koshland switch. The report introduces practical tools, including Python scripts with graphical user interfaces and a C++ code library with Python bindings, to facilitate simulations and analysis. A key finding is that the total quasi-steady-state approximation (tQSSA) is a versatile and accurate approach in a wide range of scenarios, in contrast to the standard quasi-steady-state approximation (QSSA), which is limited to low-enzyme conditions. This is especially relevant since approximations offer significant speed advantages in both CME direct integration and stochastic simulation algorithms with respect to the exact formulation.

Contents

Chapter 1

Introduction and a brief history of chemical and enzyme kinetics

This project will review some of the methods to investigate chemical reactions, in particular with respect to simple applications to enzyme kinetics. Different methods arised from the 19th century until today, and we recognize, specifically, two main categories: models based on differential calculus and law of mass action, which is the classical formulation of chemical reaction (before matter were known to be quantized) and models based on chemical master equation, with stochastic realizations simulated thanks to Monte-Carlo methods (stochastic simulation algorithm, also known as Gillespie algorithm from the original paper in 1977[13]).

The first formulation of the law of mass action was given by Peter Waage and Cato Guldberg (1864)[1], who states that the speed of a chemical reaction is proportional to the quantity of each reacting substance. 1901 Nobel Prize in Chemistry Van 't Hoff studied chemical dynamics and its work enabled experimental determination of reaction rates from which rate laws and rate constants are derived. It is important to note that only elementary reactions follow the law of mass action; on the other hand, stepwise reactions (reactions involving one or more intermediate steps) give observational rate orders which do not match those of elementary reactions. From this knowledge, one can easily infer from experiments whether a known reaction is elementary or not. The rate equations predicted by law of mass action can be derived from first principles, starting from kinetic theory, using the theoretical framework of collision theory for ideal gases, which was proposed independently by Max Trautz[4] and William Lewis[5] in 1916 and in 1918, respectively.

In 1903, Victor Henri found that the enzyme reactions could be explained by assuming a binding interaction between the enzyme and the substrate[2]. A decade later, in 1913 Michaelis and Menten proposed a mathematical model of the reaction[3], for which they also proposed a rate law, which is nowadays known to be valid at low enzyme concentration. This rate law is

also known as the quasi-steady state approximation following the derivation and theoretical treatment by Briggs and Haldane (1925)[6].

Andrei Kolmogorov introduced in 1931 the equations which describe the time-evolution of stochastic jump processes (also known as continuous-time Markov chains): these equations are now known as the Kolmogorov equations, which arise as the differential form of the Chapman-Kolmogorov fundamental equation[7]. Setting the problem with a countable state space and an initial value, the Kolmogorov equations takes exactly the form of a so-called master equation, a term coined in 1940 by Nordsieck, Lamb and Uhlenbeck[8].

In the following years, Feller[9] and Doob[10] established that the time-to-the-next-jump is exponentially distributed and that the probability of the next event is proportional to the associated transition rate. One of the first applications of this method, in the 50s, was in the field of epidemics by Bartlett[11]. Only several years later, in 1977, the physicist Daniel Gillespie popularized the algorithm in the field of physical chemistry, for the resolution of chemical master equations[13], with important applications in biochemistry. This method had been subsequently called Gillespie algorithm but, due to pioneering work by Doob is also called Doob-Gillespie algorithm. It is also known by other descriptive names, like stochastic simulation algorithm, especially when referring to variants of the original method.

Chapter 2

Law of mass action and application to enzyme kinetics

Let's consider an elementary reaction such that

$$\sum_i m_i X_i \to \sum_i n_i X_i, \tag{2.1}$$

where in upper case are the chemical species while in lower case are the initial (reactants) and final (products) stoichiometric coefficients. The law of mass action says that, at constant temperature, in the continuous limit, the concentration rate of each chemical species is given by

$$\frac{d[X_i]}{dt} = k(n_i - m_i) \prod_j [X_j]^{m_j}. \tag{2.2}$$

where k is the reaction rate constant. In particular, the reaction rate is given by

$$v = k \prod_j [X_j]^{m_j}. \tag{2.3}$$

Non-elementary reactions can be usually expressed as a chain of elementary reactions (stepwise reaction).

2.1 Single-substrate enzyme-catalyzed reaction

A single-substrate enzyme-catalyzed reaction can be described by the following chain of reactions

$$E + S \underset{k_b}{\overset{k_f}{\rightleftarrows}} C \xrightarrow{k_{\text{cat}}} E + P \tag{2.4}$$

which correspond, due to the mass action law, to the following system of ordinary differential equations (ODEs):

$$\begin{aligned}
\frac{d[S]}{dt} &= -k_f[E][S] + k_b[C], \\
\frac{d[E]}{dt} &= -k_f[E][S] + k_b[C] + k_{\text{cat}}[C], \\
\frac{d[C]}{dt} &= k_f[E][S] - k_b[C] - k_{\text{cat}}[C], \\
\frac{d[P]}{dt} &= k_{\text{cat}}[C],
\end{aligned} \tag{2.5}$$

where $[S]$ is the substrate molar concentration, $[E]$ is the enzyme molar concentration, $[C]$ is the enzyme-substrate complex molar concentration and $[P]$ is the product molar concentration. From the conservation of the total enzyme concentration, we have $[E_T] := [E] + [C]$ such that $\frac{d}{dt}[E_T] = 0$. From the conservation of the total substrate and product concentration, we have $[S_T] := [S] + [C] + [P]$ such that $\frac{d}{dt}[S_T] = 0$. The existence of these constants implies that there are only two independent ODEs to solve, corresponding to two independent variables, for example $[C]$ and $[P]$:

$$\begin{aligned}
\frac{d[C]}{dt} &= k_f([E_T] - [C])([S_T] - [C] - [P]) - k_b[C] - k_{\text{cat}}[C], \\
\frac{d[P]}{dt} &= k_{\text{cat}}[C].
\end{aligned} \tag{2.6}$$

The number of independent variables can be further reduced to one using approximations.

2.1.1 Quasi-steady state approximation

The quasi-steady state approximation (QSSA, or sQSSA, meaning *standard QSSA*) is obtained under the assumption that $[C]$ does not change appreciably before $[S]$ varies appreciably[6] [26]. Mathematically speaking, we require that

$$\frac{d[C]}{dt} \approx 0, \tag{2.7}$$

from which we obtain a closed expression for the enzyme-substrate complex concentration $[C]$ as a function of the substrate concentration $[S]$:

$$[C] = \frac{[E_T][S]}{[S] + K_M}, \tag{2.8}$$

where $K_M = (k_b + k_{\text{cat}})/k_f$ is the Michaelis-Menten constant. Substituting in the products rate equation, we obtain

$$\frac{d[P]}{dt} = \frac{k_{\text{cat}}[E_T][S]}{[S] + K_M}, \tag{2.9}$$

which is the Michaelis-Menten rate law. Segel and Slemrod (1989) [16] showed that this approximation is valid for $[E] \ll [S] + [P] + K_M$, which also implies that $[C]$ is negligible with respect to the total substrate-product concentration: $[C] \ll [S] + [P]$. This enables us to make another approximation: $[S_T] \approx [S] + [P]$, so we do not need to know $[C]$ to compute the time evolution of concentration of substrates and products.

In many sources, it is also reported the catalytic efficiency (also called specificity constant) for a specific enzyme-substrate pair, given by the ratio k_{cat}/K_M.

2.1.2 Total quasi-steady state approximation

Introduced by Borghans et al. (1996) [17], this approximation is analogous to QSSA, with the difference that we perform a change of variable before applying the condition $d[C]/dt \approx 0$, i.e.:

$$[\hat{S}] = [S] + [C]. \tag{2.10}$$

Following the same steps as before, we obtain the following expression for $[C]$:

$$[C] = \frac{1}{2}\left([E_T] + [\hat{S}] + K_M - \sqrt{([E_T] + [\hat{S}] + K_M)^2 - 4[E_T][\hat{S}]}\right). \tag{2.11}$$

From the validity condition given by Tzafriri (2003) [20]:

$$\epsilon_{tQ}([S_T], [E_T]) = \frac{k_{\text{cat}}}{2k_f[S_T]}\left(\frac{[E_T] + [S_T] + K_M}{\sqrt{([E_T] + [\hat{S}] + K_M)^2 - 4[E_T][\hat{S}]}} - 1\right) \ll 1, \tag{2.12}$$

one can show that

$$\epsilon_{tQ}([S_T], [E_T]) \leq \epsilon_{tQ}(0, K_M) = \frac{k_{\text{cat}}}{4(k_b + k_{\text{cat}})}. \tag{2.13}$$

Since $\epsilon_{tQ} \leq \frac{k_{\text{cat}}}{4(k_b+k_{\text{cat}})} \leq \frac{1}{4}$ always holds, the condition $\epsilon_{tQ} \ll 1$ is not badly violated even in the worst case, so the range of validity is much wider for tQSSA than standard QSSA. Substituting in the products rate equation, we obtain

$$\frac{d[P]}{dt} = \frac{k_{\text{cat}}}{2}\left([E_T] + [\hat{S}] + K_M - \sqrt{([E_T] + [\hat{S}] + K_M)^2 - 4[E_T][\hat{S}]}\right), \tag{2.14}$$

which can also be rewritten so that it is more computationally stable for small values of $[\hat{S}]$:

$$\frac{d[P]}{dt} = \frac{2k_{\text{cat}}[E_T][\hat{S}]}{[E_T] + [\hat{S}] + K_M + \sqrt{([E_T] + [\hat{S}] + K_M)^2 - 4[E_T][\hat{S}]}}. \tag{2.15}$$

Note that $[S_T] = [\hat{S}] + [P]$ holds exactly.

2.2 Goldbeter-Koshland switch

The Goldbeter-Koshland (GK) switch consists of a substrate-product pair S, S_P that is interconverted by two enzymes E, D:

$$\begin{aligned} E + S &\underset{k_{be}}{\overset{k_{fe}}{\rightleftarrows}} C \xrightarrow{k_e} E + S_P \\ D + S_P &\underset{k_{bd}}{\overset{k_{fd}}{\rightleftarrows}} C_P \xrightarrow{k_d} D + S \end{aligned} \tag{2.16}$$

These reactions are commonly used to describe phosphorylation and dephosphorylation processes, where S is phosphorylated by kinase E and S_P is dephosphorylated by phosphatase D. From the law of mass action we obtain these six differential equations:

$$\begin{aligned} \frac{d[S]}{dt} &= -k_{fe}[E][S] + k_{be}[C] + k_d[C_P], \\ \frac{d[S_P]}{dt} &= -k_{fd}[D][S_P] + k_{bd}[C_P] + k_e[C], \\ \frac{d[E]}{dt} &= -k_{fe}[E][S] + k_{be}[C] + k_e[C], \\ \frac{d[D]}{dt} &= -k_{fd}[D][S_P] + k_{bd}[C_P] + k_d[C_P], \\ \frac{d[C]}{dt} &= k_{fe}[E][S] - k_{be}[C] - k_e[C], \\ \frac{d[C_P]}{dt} &= k_{fd}[D][S_P] - k_{bd}[C_P] - k_d[C_P]. \end{aligned} \tag{2.17}$$

Like for the single-substrate case, we have conserved quantities: the total substrate concentration $[S_T] := [S] + [S_P] + [C] + [C_P]$, the total kinase concentration $[E_T] := [E] + [C]$ and the total phosphatase concentration $[D_T] := [D] + [C_P]$. Having these three conserved quantities, only three of the above differential equations are independent. For example, we can choose

$$\begin{aligned} \frac{d[S_P]}{dt} &= -k_{fd}([D_T] - [C_P])[S_P] + k_{bd}[C_P] + k_e[C], \\ \frac{d[C]}{dt} &= k_{fe}([E_T] - [C])([S_T] - [S_P] - [C] - [C_P]) - k_{be}[C] - k_e[C], \\ \frac{d[C_P]}{dt} &= k_{fd}([D_T] - [C_P])[S_P] - k_{bd}[C_P] - k_d[C_P]. \end{aligned} \tag{2.18}$$

2.2.1 Quasi-steady state approximation

The QSSA can be obtained by assuming

$$\frac{d[C]}{dt} \approx 0, \qquad \frac{d[C_P]}{dt} \approx 0, \tag{2.19}$$

from which, following the same steps as for the single-substrate case, we obtain

$$[C] = \frac{[E_T][S]}{K_{ME} + [S]}, \qquad [C_P] = \frac{[D_T][S_P]}{K_{MD} + [S_P]}, \tag{2.20}$$

where

$$K_{ME} = \frac{k_{be} + k_e}{k_{fe}}, \qquad K_{MD} = \frac{k_{bd} + k_d}{k_{fd}}. \tag{2.21}$$

Substituting into the equation for $d[S_P]/dt$:

$$\frac{d[S_P]}{dt} = \frac{k_e[E_T][S]}{K_{ME} + [S]} - \frac{k_d[D_T][S_P]}{K_{MD} + [S_P]}. \tag{2.22}$$

Note that $[S_T] \approx [S] + [S_P]$.

2.2.2 Total quasi-steady state approximation

The tQSSA is obtained by making two variable changes:

$$[\hat{S}] := [S] + [C], \qquad [\hat{S}_P] = [S_P] + [C_P]. \tag{2.23}$$

Then, applying the quasi-steady state condition as before, we obtain the following expression for the phosphorylated substrate concentration:

$$\begin{aligned}\frac{d[S_P]}{dt} = &\frac{2k_e[E_T][\hat{S}]}{[E_T] + [\hat{S}] + K_{ME} + \sqrt{\Big([E_T] + [\hat{S}] + K_{ME}\Big)^2 - 4[E_T][\hat{S}]}} - \\ &- \frac{2k_d[D_T][\hat{S}_P]}{[D_T] + [\hat{S}_P] + K_{MD} + \sqrt{\Big([D_T] + [\hat{S}_P] + K_{MD}\Big)^2 - 4[D_T][\hat{S}_P]}}.\end{aligned} \tag{2.24}$$

Note that $[S_T] = [\hat{S}] + [\hat{S}_P]$ holds exactly.

Chapter 3

Master equation and stochastic simulation algorithm

In the context of continuous-time Markov chains (CTMC), which is a kind of jump process where the state space of the system is countable but time is not, the Kolmogorov equations read (forward and backward, respectively)[7]

$$\begin{aligned} \frac{\partial P_{ij}}{\partial t}(s;t) &= \sum_k P_{ik}(s;t)A_{kj}(t), \\ \frac{\partial P_{ij}}{\partial s}(s;t) &= -\sum_k P_{kj}(s;t)A_{ik}(s), \end{aligned} \tag{3.1}$$

where $P_{ij}(s;t)$ is the probability of starting with the i-th state at the initial time s and ending up with the j-th state at the final time $t > s$, while $A(t)$ is the transition-rate matrix, which satisfies

$$A_{ij}(t) = \left[\frac{\partial P_{ij}}{\partial u}(t;u)\right]_{u=t}, \qquad A_{jk}(t) \geq 0, \quad j \neq k, \qquad \sum_k A_{jk}(t) = 0. \tag{3.2}$$

When fixing the initial time, e.g. $s = 0$, and the initial state, the Kolmogorov forward equations can be simplified by dropping the first index and the s variable:

$$\frac{dP_j}{dt}(t) = \sum_k P_k(t)A_{kj}(t). \tag{3.3}$$

This equation describes an initial-value problem and it is widely known as the master equation[8].

3.1 Chemical master equation

The chemical master equation (CME) for a set of reaction channels is given by[21]

$$\frac{\partial p(\mathbf{x},t)}{\partial t} = \sum_{n=1}^{N_r} a_n(\mathbf{x}-\nu_n)p(\mathbf{x}-\nu_n,t) - \sum_{n=1}^{N_r} a_n(\mathbf{x})p(\mathbf{x},t), \tag{3.4}$$

where $\mathbf{x} \in \mathbb{N}^{N_s}$ is the total number of molecules for each chemical species, N_r is the number of reaction channels and N_s is the number of chemical species. Furthermore, ν_n is the difference of population counts involved in a single reaction (i.e., the stoichiometric vector), $a_n(\mathbf{x})$ are the propensity functions (the transition rates for a particular reaction n). The propensity functions can be seen as the off-diagonal elements of the transition-rate matrix A_{ij} for a general master equation, while $-\sum_{n=1}^{N_r} a_n(\mathbf{x})$ is the diagonal element (from Eq. (3.2)).

Sunkara (2019) [25] found an equivalent formulation of the CME, where the probability of certain species counts $\mathbf{x} \in \mathbb{N}^{N_s}$ is substituted with the probability of certain reaction counts $\mathbf{r} \in \mathbb{N}^{N_r}$ from a certain initial state $\mathbf{x}_0$:

$$\frac{\partial p(\mathbf{r},t)}{\partial t} = \sum_{n=1}^{N_r} a_n(\mathbf{r}-\mathbf{1}_n)p(\mathbf{r}-\mathbf{1}_n,t) - \sum_{n=1}^{N_r} a_n(\mathbf{r})p(\mathbf{r},t), \tag{3.5}$$

where $\mathbf{1}_n$ is the Kronecker delta. This formulation may be more convenient since it leads to much simpler dynamics: in particular, since the reaction counts can only increase, this gives a simple feedforward propagation of probabilities for the associated Markov chain. We only need a map that from the reaction counts gives the species counts, since these latter variables are needed to compute the propensity functions:

$$\mathbf{x} = \mathbf{x}_0 + \sum_{n=1}^{N_r} \nu_n r_n. \tag{3.6}$$

Note that this map is not in general bijective. Given a solution to the reaction count CME $p(\mathbf{r},t)$, we obtain the probabilities associated to each state at time t through

$$p(\mathbf{x},t) = \sum_{\mathbf{r}\in\Gamma_{\mathbf{x}}} p(\mathbf{r},t), \tag{3.7}$$

where $\Gamma_{\mathbf{x}} := \left\{\mathbf{r}\middle|\mathbf{x} = \mathbf{x}_0 + \Sigma_{n=1}^{N_r}\nu_n r_n\right\}$ is the set of reaction counts that gives the same population counts $\mathbf{x}$. One problem with this formulation is that the reaction count itself is unbounded: for practical purposes, one can choose to limit the total number of reactions, as long as this choice does not affect the accuracy of the result (this choice will depend on the required length of the simulation).

Both these formulations can be casted into a system of ordinary differential equations with many different variables depending on the maximum allowed population numbers or reaction numbers. For small enough systems, these

equations can be simply solved using a standard integration method, for example a Runge-Kutta method. However, we are quickly caught by the curse of dimensionality for systems with many chemical species: a lot of computer memory may be required to perform a simulation in this case.

3.2 Stochastic simulation algorithm

An alternative way to solve the CME is through a Monte-Carlo method, i.e., performing several stochastic realizations, and then calculating the relevant statistics. The algorithm is based on the reaction probability density function [9] [10] [13] [21]:

$$\rho(\tau, j; \mathbf{x}, t) = a_j(\mathbf{x}) \exp\left(-\tau a_0(\mathbf{x})\right), \tag{3.8}$$

where $a_0(\mathbf{x}) = \Sigma_{n=1}^{N_r} a_n(\mathbf{x})$, while τ is the time between a reaction and the next one and j is the reaction channel index. When calculating the marginal probability density functions of the two variables, τ is shown to be an exponentially distributed random variable, while j is a statistically independent integer random variable with point probabilities $a_j(\mathbf{x})/a_0(\mathbf{x})$. Thus, the Monte-Carlo method is to draw two uniform (pseudo)random numbers $r_1, r_2 \in U(0,1)$ and then compute

$$\tau = -\frac{1}{a_0(\mathbf{x})} \ln r_1, \tag{3.9}$$

while j is given by the smallest integer satisfying

$$\sum_{n=1}^{j} a_n(\mathbf{x}) > r_2 a_0(\mathbf{x}) \tag{3.10}$$

and, finally, carry out the reaction by replacing

$$t \leftarrow t + \tau, \qquad \mathbf{x} \leftarrow \mathbf{x} + \nu_j. \tag{3.11}$$

Note that, since we require a large amount of realizations to obtain statistically significant results, stochastic simulation algorithm can be slower than direct integration of the CME for systems with few chemical species or small populations. In practice, this algorithm is very useful when solving the CME directly would be prohibitively expensive in terms of required memory or computational time.

3.3 Single-substrate enzyme-catalyzed reaction

The propensity functions for a single-substrate enzyme-catalyzed reaction are associated to each reaction channel:

$$E + S \xrightarrow{\kappa_f} C, \qquad C \xrightarrow{\kappa_b} E + S, \qquad C \xrightarrow{\kappa_{\text{cat}}} E + P. \tag{3.12}$$

So, the three propensity functions are given by

$$a_f(E,S) = \kappa_f ES, \qquad a_b(C) = \kappa_b C, \qquad a_{\text{cat}}(C) = \kappa_{\text{cat}} C, \tag{3.13}$$

where E, S, C, P refer to the numbers of molecules for the different chemical species rather than their concentrations.

The relation between a population number X and the relative molar concentration $[X]$ is

$$[X] = \frac{X}{N_A \Omega}, \tag{3.14}$$

where N_A is the Avogadro constant and Ω is the total volume of the system. The constants κ_i are related to the constants k_i used in the ODE treatment of the reaction rates[23]:

$$\kappa_f = \frac{k_f}{N_A \Omega}, \qquad \kappa_b = k_b, \qquad \kappa_{\text{cat}} = k_{\text{cat}}. \tag{3.15}$$

Remembering that we have only two independent variables thanks to the conservation of $S_T = S + C + P$ and $E_T = E + T$, the CME gives

$$\begin{aligned} \frac{\partial p(C,P,t)}{\partial t} &= a_f(C-1,P)p(C-1,P,t) + a_b(C+1)p(C+1,P,t) + \\ &+ a_{\text{cat}}(C+1)p(C+1,P-1,t) - a_0(C,P)p(C,P,t), \end{aligned} \tag{3.16}$$

where $a_f(C,P) = \kappa_f(E_T - C)(S_T - C - P)$ and $a_0(C,P) = a_f(C,P) + a_b(C) + a_{\text{cat}}(C)$. The Gillespie algorithm for this system is directly obtained from the propensity functions. Note that we can perform the tQSSA in the stochastic formulation as in the deterministic one by using only one propensity function $a(\hat{S})$[26], i.e.:

$$a(\hat{S}) = \frac{2\kappa_{\text{cat}} E_T \hat{S}}{E_T + \hat{S} + \kappa_M + \sqrt{(E_T + \hat{S} + \kappa_M)^2 - 4E_T\hat{S}}}, \tag{3.17}$$

where $\hat{S} = S + C = S_T - P$ and $\kappa_M = \frac{\kappa_b + \kappa_{\text{cat}}}{\kappa_f}$, related to the usual Michaelis-Menten constant through

$$\kappa_M = K_M N_A \Omega. \tag{3.18}$$

This can be viewed as having only one monomolecular reaction described by

$$\hat{S} \to P. \tag{3.19}$$

3.4 Goldbeter-Koshland switch

For the stochastic formulation of the GK switch, we recognize six propensity functions related to the six reaction channels

$$\begin{aligned} E + S &\xrightarrow{\kappa_{fe}} C, \qquad C \xrightarrow{\kappa_{be}} E + S, \qquad C \xrightarrow{\kappa_e} E + S_P, \\ D + S_P &\xrightarrow{\kappa_{fd}} C_P, \qquad C_P \xrightarrow{\kappa_{bd}} D + S_P, \qquad C_P \xrightarrow{\kappa_d} D + S. \end{aligned} \tag{3.20}$$

Choosing the coordinates S_P, C, C_P as the independent variables, the propensity functions are given by

$$\begin{aligned} a_{fe}(S_P, C, C_P) &= \kappa_{fe}(E_T - C)(S_T - S_P - C - C_P), \\ a_{fd}(S_P, C_P) &= \kappa_{fd}(D_T - C_P)S_P, \\ a_{be}(C) = \kappa_{be}C, &\qquad a_e(C) = \kappa_e C, \\ a_{bd}(C_P) = \kappa_{bd}C_P, &\qquad a_d(C_P) = \kappa_d C_P. \end{aligned}$$

For stochastic tQSSA, we have only two reactions corresponding to

$$\hat{S} \rightleftarrows \hat{S}_P, \tag{3.21}$$

for which the propensity functions are given by

$$\begin{aligned} a_e(\hat{S}) &= \frac{2k_e E_T \hat{S}}{E_T + \hat{S} + K_{ME} + \sqrt{\left(E_T + \hat{S} + K_{ME}\right)^2 - 4E_T\hat{S}}}, \\ a_d(\hat{S}_P) &= \frac{2k_d D_T \hat{S}_P}{D_T + \hat{S}_P + K_{MD} + \sqrt{\left(D_T + \hat{S}_P + K_{MD}\right)^2 - 4D_T\hat{S}_P}}, \end{aligned} \tag{3.22}$$

with only one independent variable since $\hat{S}, \hat{S}_P$ are related by $\hat{S} = S_T - \hat{S}_P$.

Chapter 4

The code

The code has been written in Python and C++. The Python scripts offer a user-friendly graphical user interface (GUI) for investigating enzyme-substrate reactions, allowing users to compare simulation results across different mathematical models. They have the following dependencies:

- NumPy for numerical operations.
- SciPy for numerical integration (odeint, which uses LSODA from the Fortran library Odepack[15], an implementation of a method by Petzold[14]) and steady-state computation using a root-finding algorithm (fsolve, which uses HYBRD and HYBRJ implmentations from the Fortran library Minpack of the Powell's hybrid method[12]).
- Matplotlib for visualization of graphs and animations.
- Tkinter for GUI elements, like buttons and text fields.
- Custom modules "gillespie" and "cme" written in C++ and made available as Python libraries thanks to Pybind11.

The Python scripts also require ffmpeg to be installed to run animations.

Key functionalities include extracting user-supplied parameters like total enzyme and substrate counts and rate constants via the "Run Simulation" button. The simulation is performed using the Gillespie algorithm or integrating the CME for the exact formulation, the quasi-steady-state approximation (QSSA) and the total quasi-steady-state approximation (tQSSA). After the simulation, the collected data is displayed in various forms, including variation of population numbers/concentrations over time or with respect to the variation of a parameter, probabilities of a given state, statistical averages and standard deviations, probability densities for completion times, and probability densities for steady-state phosphorylated substrate count. These outputs are generated for both the cme and gillespie simulation methods.

4.1 C++ code and Python bindings

This code part has been written as a library in C++20 with Python bindings using Pybind11[1]. The repository is structured in the following way:

- `experiments`: directory containing example code using the library and experiments for this project.
- `include/sck`: directory containing the C++ header files to include in implementation files.
  - `cme.hpp`: it includes classes for generic CME equation integration and applications to enzyme kinetics (single-substrate enzyme-catalyzed reaction and Goldbeter-Koshland switch).
  - `gillespie.hpp`: it includes classes for generic Gillespie algorithm and applications to enzyme kinetics (single-substrate enzyme-catalyzed reaction and Goldbeter-Koshland switch).
  - `runge_kutta.hpp`: explicit Runge-Kutta methods used for integration of the CME equation.
  - `tensor.hpp`: classes, aliases and data structures for vectors, matrices and tensors with some helper functions.
- `pybind`: directory containing C++ implementation files that binds the code inside the 'include' directory. It also contains the Windows dynamic-link libraries which can be directly imported in Python scripts (on Linux, one would need to recompile them).
  - `cme_pybind.cpp`: Python bindings for CME.
  - `gillespie_pybind.cpp`: Python bindings for Gillespie algorithm.
  - `runge_kutta_pybind.cpp`: Python bindings for Runge-Kutta methods.
- `tests`: Tests that verify the correctness of the implementation.

4.1.1 Dependencies

The C++ header files have no dependencies other than a C++20-complying compiler. To compile the Python bindings, Pybind11 must be installed on the same machine. The resulting Python libraries can be easily imported from Python and will require NumPy 1.7.0 or any later version.

[1] `https://github.com/locuoco/stochastic-chemical-kinetics`

4.1.2 Example

A simple example code using the `gillespie` module in Python is the following:

```
import matplotlib.pyplot as plt

import gillespie

g = gillespie.single_substrate(kf=10, kb=9, kcat=1,
  ET=10, ST=9)

x, t = g.simulate()

plt.plot(t, x[:,g.species.C],
  drawstyle='steps-post', label='C')
plt.plot(t, x[:,g.species.P],
  drawstyle='steps-post', label='P')

plt.xlabel('Time')
plt.ylabel('Population')
plt.legend()

plt.show()

```

This simple snippet creates an object `g` for a single-substrate enzyme-catalyzed reaction system to be simulated using Gillespie algorithm with the specified parameters. The `simulate` method without arguments performs a simulation until completion (when all substrates are transformed into products) and returns the vector of states `x` and times `t` associated to each reaction event. `x` is a 2-order array where the first dimension matches the length of `t`, while the second one matches the number of independent variables. To select the desired species, the `species` enumeration is employed in the second dimension of the array. Matplotlib is used to plot the results.

4.2 GUI Description

The GUI window constists of a label asking the user to select an experiment to start the simulation[2]. Below the label there is a dropdown menu presenting a list of available experiments, where the user can choose a specific simulation to run. Directly beneath the dropdown menu, a button labeled "Start simulation" is placed. When the user clicks this button, a new window corresponding to the chosen experiment to perform is opened.

Hereafter the list of available options for the user:

- GUI-ode-single-substrate

[2] `https://github.com/pupak250194/Physical-Methods-of-Biology---Enzyme-Reactions`

- GUI-ode-gk-switch
- GUI-ode-gk-switch-steady-state
- GUI-cme-averages
- GUI-cme-completion-times
- GUI-cme-evolution
- GUI-cme-stationary
- GUI-gillespie-averages
- GUI-gillespie-completion-times
- GUI-gillespie-evolution
- GUI-gillespie-stationary

The menu selector also offers GUIs to compare the results for both the Gillespie and the CME methods for the tQSSA model - see Ch. ***Results and comparisons*** for further explanation.

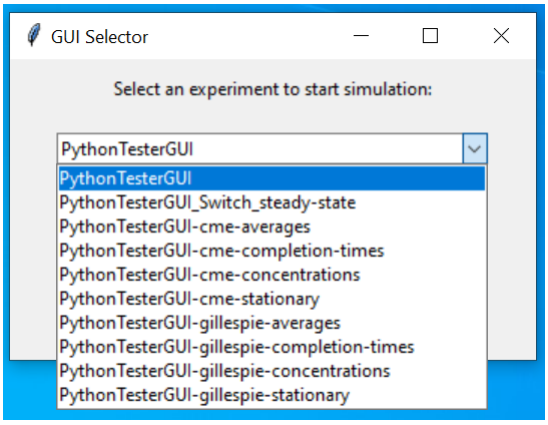


Figure 4.1: Dropdown menu of the GUI selector.

4.2.1 GUI-ode-single-substrate

The script implements a graphical user interface (GUI) for the resolution of the set of ODEs associated to the single-substrate enzyme-catalyzed reaction system (Eq. 2.6). The ODEs are integrated using `odeint` from `scipy.integrate`. The GUI presents entry fields for the user to set parameters like total enzyme concentration (ET), total substrate concentration (ST), initial concentrations of enzyme-substrate complex and products (C0 and P0), and rate constants (kf, kb, kcat). The user can specify the desired

simulation time through an input field. Matplotlib is used to generate plots for the exact formulation, the QSSA, and tQSSA (see Fig. 4.2). The GUI allows for immediate comparison of the different approximations with different parameters and their impact on the concentration profiles.

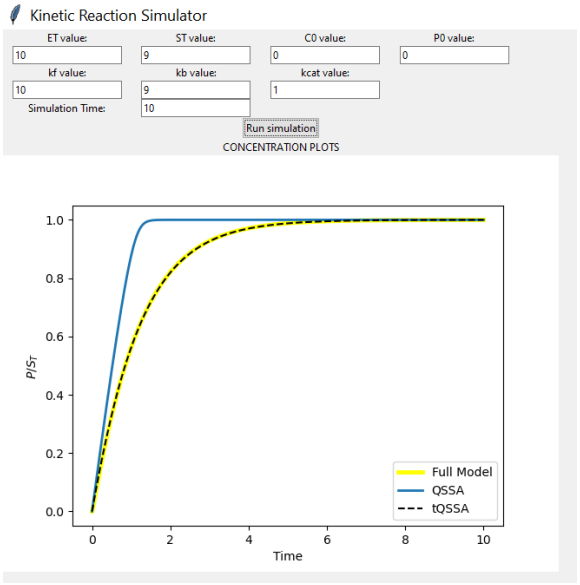


Figure 4.2: GUI-ode-single-substrate. The figure shows the simulation results after pressing the "Start Simulation" button. The output shows the ratio of products population (P) to total substrate (ST) over time. The exact formulation is depicted in yellow, the QSSA in default color and tQSSA as a dashed line in black. The figure shows the results of a simulation with default parameters.

4.2.2 GUI-ode-gk-switch

This script implements a GUI for the resolution of the set of ODEs associated to the Goldbeter-Koshland switch (Eq. 2.18). Analogously as before, the ODEs are integrated using `scipy.integrate.odeint`.

The user can set various parameters, like total kinase concentration (ET), total phosphatase concentration (DT), total substrate concentration (ST) and rate constants (kfe, kbe, ke, kfd, kbd, kd). Plots are generated for the exact formulation, the QSSA and the tQSSA using Matplotlib.

4.2.3 GUI-ode-gk-switch-steady-state

This script provides a GUI along with the plot of the steady-state for the Goldbeter-Koshland switch (Eq. 2.18) as a function of the total kinase concentration. The steady-state is found using a root-finding algorithm

offered by `fsolve` from `scripy.optimize` applied to the ODEs.

The GUI provides entry fields for users to set various parameters, including total kinase concentration (ET), total phosphatase concentration (DT), total substrate concentration (ST) and rate constants (kfe, kbe, ke, kfd, kbd, kd). The exact formulation, the QSSA, and the tQSSA all predict the same switch behavior at low phosphatase concentration. However, at high phosphatase concentrations, the QSSA wrongly continues to give the same prediction, while the tQSSA agrees with the exact formulation, for which the switch behavior is absent.

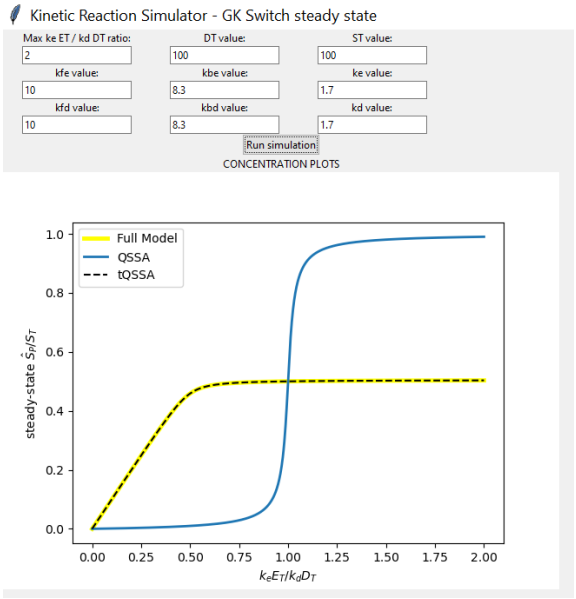


Figure 4.3: GUI-ode-gk-switch-steady-state. The figure shows the simulation results after pressing the "Start Simulation" button. The output shows the ratio of product (SP) to total substrate (ST) as a function of k_eE_T/k_dD_T. The figure shows the results of a simulation with default parameters.

4.2.4 GUI-cme-averages

This script calculates the average and standard deviation of products as a function of time, starting from the integration of the CME for the single-substrate enzyme-catalyzed reaction (see Eq. 3.16). Indeed, from the solution of the CME $p(C, P, t)$ (for the exact formulation), one can calculate the n-th moment of the products at time t in the following way:

$$P^{(n)}(t) = \sum_{C=0}^{E_T} \sum_{P=0}^{S_T} P^n p(C, P, t), \tag{4.1}$$

where $P^{(1)}(t)$ is the average and $\sqrt{P^{(2)}(t) - P^{(1)}(t)^2}$ is the standard deviation of P at time t.

The script makes use of the "cme" module. Like for the other scripts, the GUI includes input fields for parameters such as total enzyme population, total substrate population and rate constants. Users can freely adjust these parameters, directly observing their influence on the chemical system.

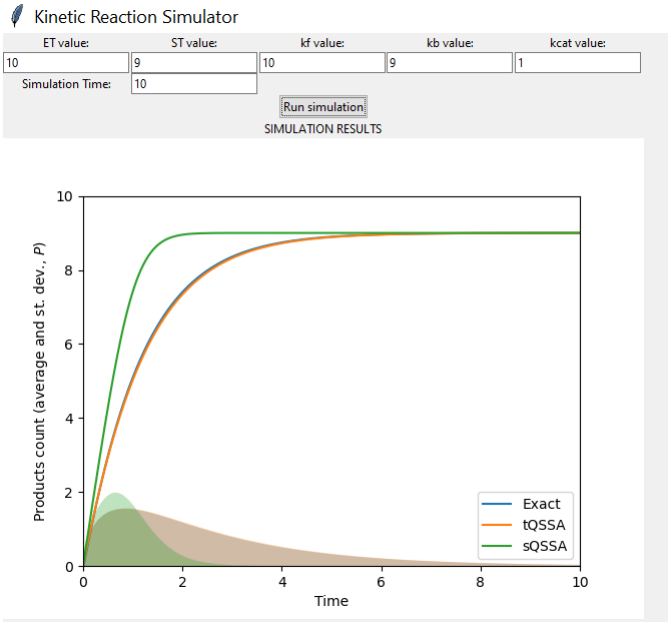


Figure 4.4: GUI-cme-averages. The figure shows the simulation results after pressing the "Start Simulation" button. The output shows the averages and the standard deviation of products as a function of time. The figure shows the results of a simulation with default parameters.

4.2.5 GUI-cme-completion-times

This script calculates the completion times from the integration of the CME for the single-substrate enzyme-catalyzed reaction (Eq. 3.16). While this kind of reaction requires an infinite time in the continuous limit using the law of mass action, the reaction will always require a finite amount of time in the stochastic formulation. From the solution of the CME $p(C, P, t)$ (for the exact formulation), the frequency of completed reactions at time t is

$$\sum_{C=0}^{E_T} p(C, S_T, t) = p(0, S_T, t), \tag{4.2}$$

where the last equality holds if $p(C, P, 0) = 0$ for $C > 0$. The results are put in a histogram.

The script makes use of the "cme" module. Along with the GUI, the average completion time and its standard deviation are displayed on the terminal screen.

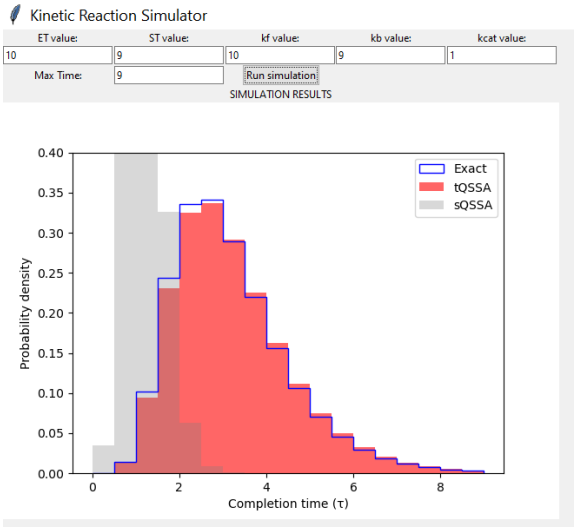


Figure 4.5: GUI-cme-completion-times. The figure shows the simulation results after pressing the "Start Simulation" button. The simulation results show the histogram of the probability density of completion times for the exact formulation, the QSSA and the tQSSA. The averages and the standard deviation are shown on terminal. The figure shows the results of a simulation with default parameters.

```
Running simulation with parameters: ST=9, ET=10, kf=10.0, kb=9.0, kcat=1.0
Exact 3.172990894326048 +/- 1.3289053298076534
tQSSA 3.2322134338058515 +/- 1.3499847976471138
sQSSA 1.1829265917541494 +/- 0.4267018304684226
```

Figure 4.6: GUI-cme-completion-times. The average completion time and the standard deviation are displayed on terminal.

4.2.6 GUI-cme-evolution

The GUI is built to provide functionalities to dynamically visualize the progress of the reaction by integrating the CME for the single-substrate enzyme-catalyzed reaction (Eq. 3.16) in real-time. Users can specify parameters such as total enzyme concentration (ET), total substrate concentration (ST), and rate constants (kf, kb, kcat) to observe the time-evolution of the reaction. The GUI displays a grid representing the state space of the reaction, with the x-axis being the product count (P) and the y-axis being the enzyme-substrate complex count (C). The color-coded grid serves as a probability

map, where brighter colors indicate higher probabilities.

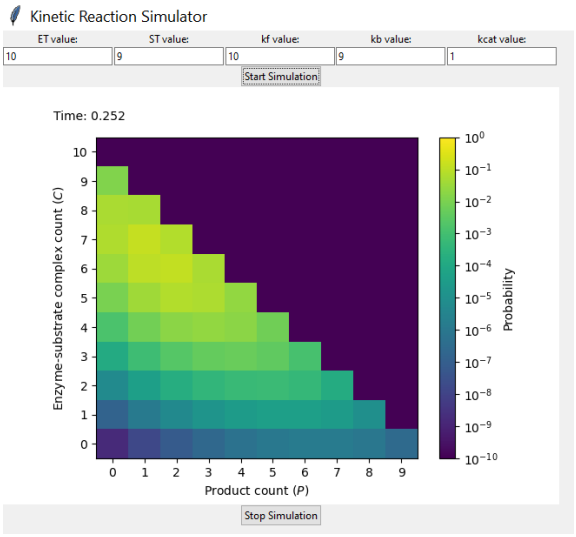


Figure 4.7: GUI-cme-evolution. The figure shows the simulation results after pressing the "Start Simulation" button. The output is an animation of the probability map evolution of the product and the enzyme-substrate complex count. The figure shows the results of a simulation with default parameters.

4.2.7 GUI-cme-stationary

This script calculates the stationary distribution for the stochastic Goldbeter-Koshland switch by integrating the CME until relaxation is achieved starting from an arbitrary initial state. The pre-set time for the relaxation of the system may need to be manually tuned to avoid spurious results for certain initial conditions. At the end of the simulation, the marginal probabilities for the distribution of the phosphorylated substrate count are collected and put into a histogram.

Users input key parameters through the GUI, including total kinase, total phosphatase, total substrate, and the rate constants, to define the reaction conditions. The GUI displays the steady-state phosphorylated substrate count ($\hat{S}_P$) on the x-axis, and the probability density on the y-axis. The resulting steady-state phosphorylated substrate count distributions are plotted in histograms and the information about the averages and the standard deviation are displayed on terminal. The result for the exact formulation is not given due to excessive computational time required.

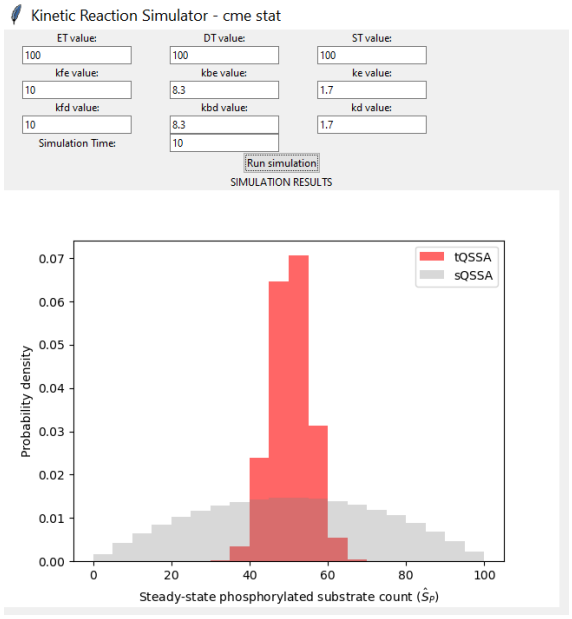


Figure 4.8: GUI-cme-stationary. The figure shows the simulation results after pressing the "Start Simulation" button. The output shows the steady-state phosphorylated substrate count distributions. The figure shows the results of a simulation with default parameters.

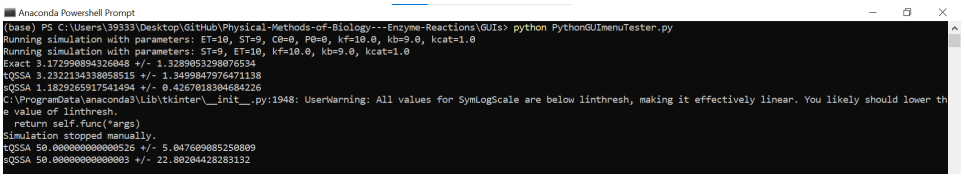


Figure 4.9: GUI-cme-stationary. The figure information about the averages and the standard deviation are displayed on terminal.

4.2.8 GUI-gillespie-averages

The script calculates the average and standard deviations of products for the single-substrate enzyme-catalyzed reaction using stochastic realizations from Gillespie algorithm, analogously to what has been done with CME direct integration. To do so, we perform 5000 Gillespie simulations. Since stochastic realizations are not synchronized with each other in time, one has to convert the simulation data into a histogram. For each stochastic realization, from the Gillespie algorithm evolution of products we take the finite difference and fill a histogram with products time-evolution data points with weights fixed by the finite difference. The original data is then approximated by taking the cumulative sum of the calculated histogram. By computing the statistical

moments of the histograms, one can then easily take the average and the standard deviation associated to each bin. The final result is conveniently displayed in a regular plot.

The program takes user-defined parameters through the GUI, including total enzyme concentration (ET), total substrate concentration (ST), and rate constants (kf, kb, kcat). The results are visualized within the GUI, showing average product counts and their standard deviations over time.

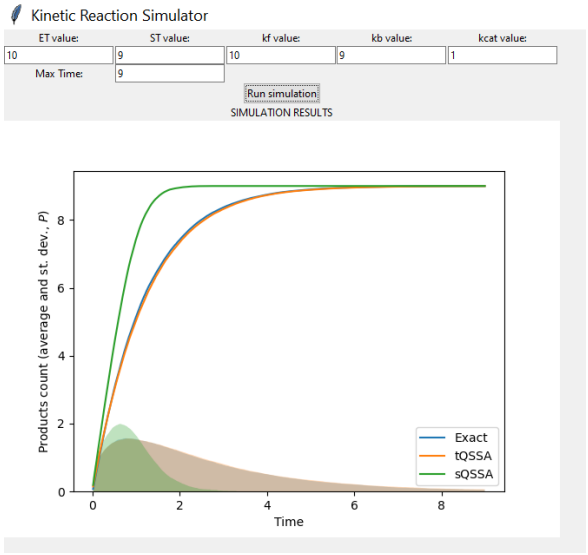


Figure 4.10: GUI-gillespie-averages. The figure shows the simulation results after pressing the "Start Simulation" button. The output shows the averages and the standard deviation of products as a function of time. The figure shows the results of a simulation with default parameters.

4.2.9 GUI-gillespie-completion-times

This script calculates the completion times for the single-substrate enzyme-catalyzed reaction using Gillespie algorithm. This experiment is a natural choice to test this algorithm due to ease of implementation. When the reaction is completed, no further reaction events are possible: the sum of all propensity functions is zero. This latter condition is easily checked to interrupt automatically the current simulation when the reaction has been completed. In this case, the completion time for a single stochastic realization is simply the time when the last reaction event occurrs. 20,000 simulations for each model (exact, QSSA and tQSSA) are performed, and the completion times are collected inside a histogram.

The GUI displays the histograms of completion times for the exact formulation, the QSSA and the tQSSA, allowing users to visually compare the performance of different approximations.

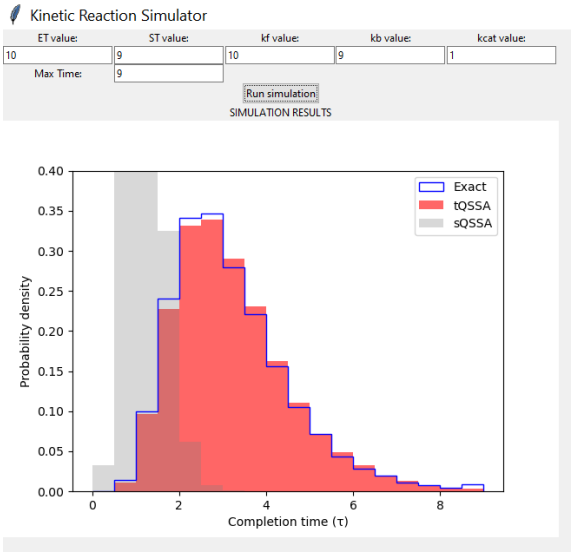


Figure 4.11: GUI-gillespie-completion-times. The figure shows the simulation results after pressing the "Start Simulation" button. The output shows the distribution of the completion times for each model, whereas the mean and the standard deviation of completion times are displayed on terminal. The figure shows the results of a simulation with default parameters.

```
Exact 3.183652265743762 +/- 1.3453322367329283
tQSSA 3.218269165393131 +/- 1.3262402640780226
sQSSA 1.1802223307202684 +/- 0.3991807389414913
```

Figure 4.12: GUI-gillespie-completion-times. The average completion time and the standard deviation are displayed on terminal.

4.2.10 GUI-gillespie-evolution

The script plots a simple stochastic realization of a single-substrate enzyme-catalyzed reaction using Gillespie algorithm.

The GUI interface allows for users to input critical parameters, including total enzyme concentration (ET), total substrate concentration (ST), and rate constants (kf, kb, kcat). A plot of the reaction is displayed, where individual reaction events are clearly visible as sudden "jumps". Specifically, it shows the variation of the populations of the enzyme-substrate complex (C) and the product (P) as a function of time.

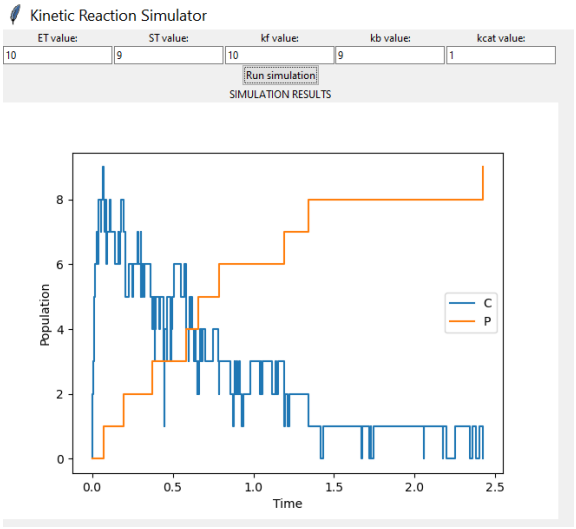


Figure 4.13: GUI-gillespie-evolution. The figure shows the simulation results after pressing the "Start Simulation" button. The output shows the enzyme-substrate complex (C) and the product (P) population variations as a function of time. The figure shows the results of a simulation with default parameters.

4.2.11 GUI-gillespie-stationary

This script will calculate the stationary distribution of products for the Goldbeter-Koshland switch using Gillespie algorithm. In this case the stationary distribution can be approximated by collecting all states during the evolution of the system inside a histogram (and weighting by time permanence of states). The simulation time, which can be set manually by the user, must be long enough for the system to relax to the stationary distribution and to obtain statistically significant results.

The GUI allows users to input critical parameters, including total kinase (ET), total phosphatase (DT), total substrate (ST) and the rate constants (kfe, kbe, ke, kfd, kbd, kd). The plot visualizes the probability density of the distribution of phosphorylated substrate count ($\hat{S}_P$). The three usual models are used, i.e., the exact formulation, the QSSA, and the tQSSA.

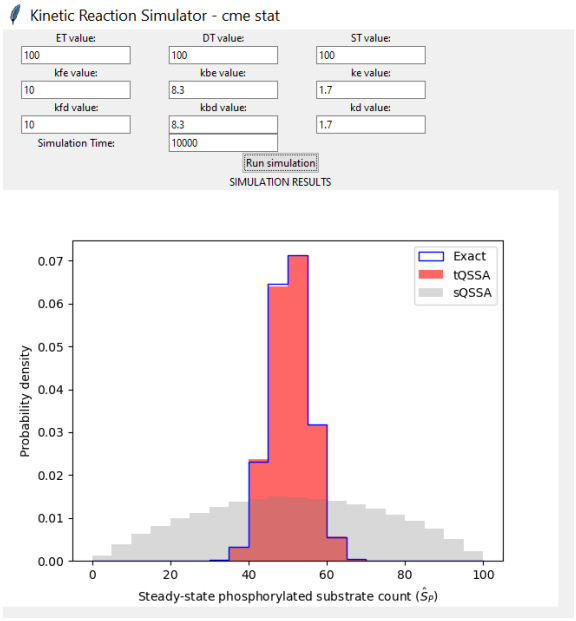


Figure 4.14: GUI-gillespie-stationary. The figure shows the simulation results after pressing the "Start Simulation" button. The output shows the steady-state phosphorylated substrate count distributions. The figure shows the results of a simulation with default parameters.

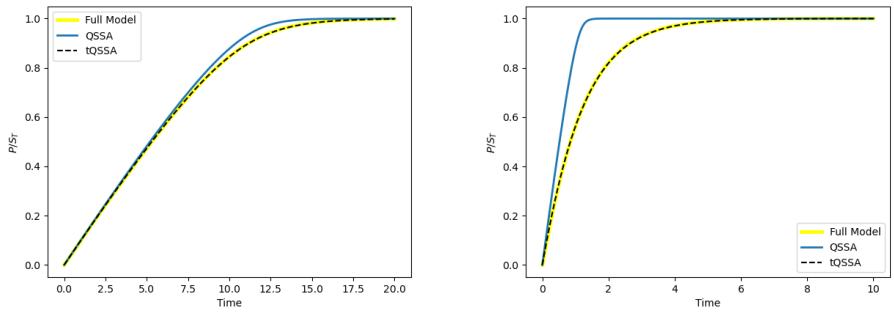
```
Exact 50.0569419943058 +/- 5.006073621400597
tQSSA 50.05538472059586 +/- 5.061009211558477
sQSSA 50.68915311192979 +/- 22.67582286696985
```

Figure 4.15: GUI-gillespie-stationary. The average and the standard deviation of steady-state phosphorylated substrate count are displayed on terminal.

Chapter 5

Results & comparisons

In this chapter we will show that the Michaelis-Menten (MM) rate law (i.e., the QSSA) is a useful tool for interpreting enzyme-catalyzed reactions where the substrate concentrations (or populations) are much higher than the enzymes', in both deterministic and stochastic formulations. The MM rate law, however, will lead to inaccurate and unrealistic dynamics when applied outside of its intended range. The total quasi-steady state approximation (tQSSA) is a straightforward correction that can be used for higher enzyme concentrations (or populations). In the following pages we will compare the computational experiments results for the single-substrate enzyme-catalyzed reaction and the Goldbeter-Koshland switch at different levels of enzyme concentrations or populations, and using different models like the exact formulation, the QSSA and the tQSSA.

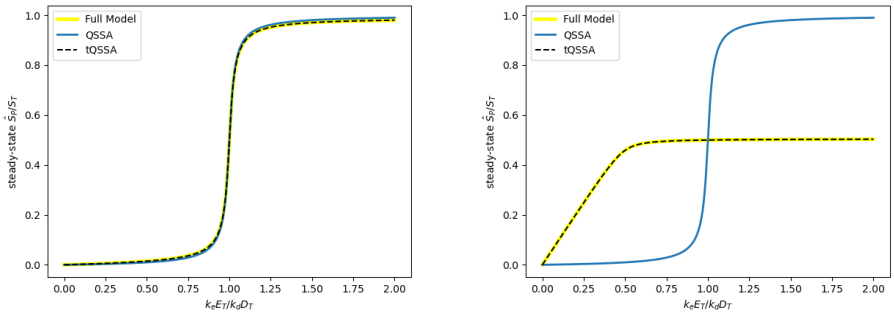


(a) Low enzyme concentration: $[E_T] = 1$.

(b) High enzyme concentration: $[E_T] = 10$.

Figure 5.1: Time evolution of products for the single-substrate experiment using law of mass action. Simulation Parameters: $k_f = 10$, $k_b = 9$, $k_{\text{cat}} = 1$ and $[S_T] = 9$. The initial values for C and P were both set to 0.

In Fig. 5.1 we reported the plots for the various approximations of the single-substrate enzyme-catalyzed reaction system at low and high enzyme concentrations using the law of mass action. At low enzyme concentration, i.e. $[E_T] \ll [S_T]$, the QSSA model is trustworthy and it is a good approximation, as predicted by the validity conditions by Segel and Slemrod [16]. On the other hand, if $[E_T]$ is high, the results of the QSSA prediction greatly differs from the exact formulation. On the other hand, the time evolution is precisely reproduced by the tQSSA model regardless of the total enzyme concentration value. As $[E_T]$ increases the QSSA loses accuracy in comparison to the tQSSA. This result agrees with the one by Kim and Tyson [26] (Fig. 1d-e).



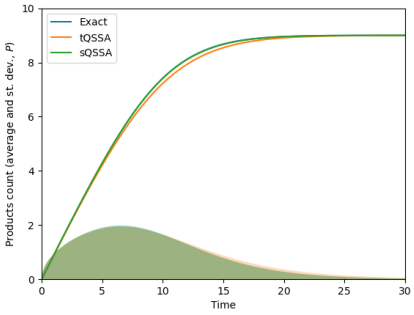
(a) Low phosphatase concentration: $[D_T] = 1$.

(b) High phosphatase concentration: $[D_T] = 100$.

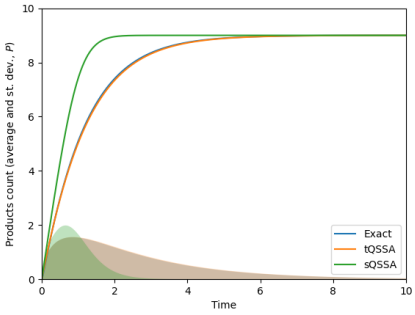
Figure 5.2: Steady-state phosphorylated substrate concentration $[\hat{S}_P]$ as a function of the total kinase concentration $[E_T]$ for the GK switch using law of mass action. Simulation Parameters: $k_{fe} = 10$, $k_{be} = 8.3$, $k_e = 1.7$, $k_{fd} = 10$, $k_{bd} = 8.3$, $k_d = 1.7$ and $[S_T] = 100$.

A similar analysis can be done for the Goldbeter-Koshland switch in Fig. 5.2. The name of this system comes from the notable behavior of the steady-state phosphorylated substrate $[\hat{S}_P]$ as a function of the total kinase concentration $[E_T]$, which approximates a sigmoid function for low total phosphatase $[D_T]$, like a switch that it is activated at $[E_T] > k_d[D_T]/k_e$. This is a very well-known example of ultrasensitivity in a biological system. Both QSSA and tQSSA perform well in this $[D_T] \ll [S_T]$ limit. However, standard QSSA wrongly continues to predict the same switch-like behavior also at high values of $[D_T]$. But, in this case, the system no longer exhibits an ultrasensitivity property: on the other hand, the tQSSA gives the correct michaelian curve. This result agrees with the one by Kim and Tyson [26] (Fig. 2b,d).

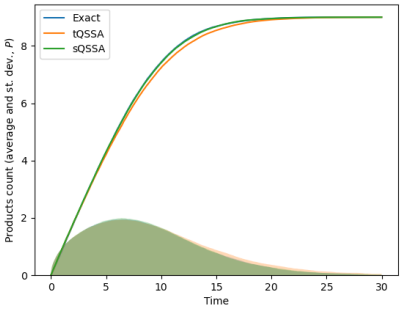
We will now proceed our analysis with the stochastic formulation. In Fig. 5.3, we show how the average and the standard deviation of products change during a single-substrate reaction, using both CME direct integration and



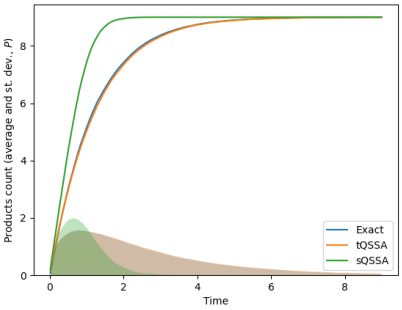
(a) CME direct integration simulation, low enzyme population, $E_T = 1$.



(b) CME direct integration simulation, high enzyme population, $E_T = 10$.



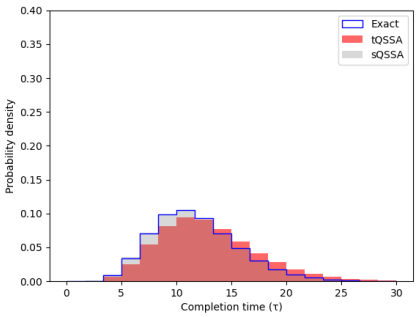
(c) Gillespie algorithm simulation, low enzyme population, $E_T = 1$.



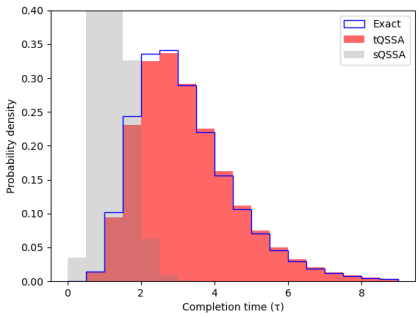
(d) Gillespie algorithm simulation, high enzyme population, $E_T = 10$.

Figure 5.3: Average and standard deviation of the product count as a function of time for a single-substrate reaction using CME direct integration (integration step: $\Delta t = 10^{-4}$) and Gillespie algorithm (5000 Monte-Carlo simulations). Simulation Parameters: $\kappa_f = 10$, $\kappa_b = 9$, $\kappa_{\text{cat}} = 1$ and $S_T = 9$ (same as in Fig. 5.1 with $N_A\Omega = 1$).

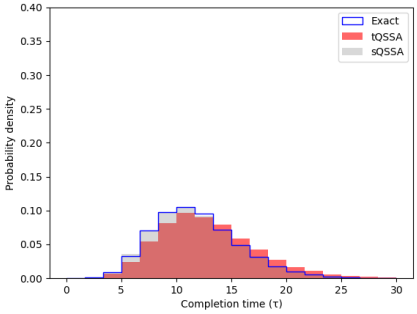
stochastic simulation algorithm. Differently from what we observe in Fig. 5.1, at low enzyme population E_T (Fig. 5.3a and 5.3c) the QSSA gives an overall better prediction of the exact formulation than tQSSA (with average lines and standard deviation curves almost completely overlapping), although tQSSA still gives a reasonably accurate result. This difference in behaviors of QSSA with respect to tQSSA also suggests that even the first-order statistics may not be replicated using the deterministic law of mass action. At high enzyme population (Fig. 5.3b and 5.3d), tQSSA gives the best prediction for both average and standard deviation. The qualitative behavior of the standard deviation in all cases has a simple explanation: since both the initial and the final states are completely determined, the standard deviation



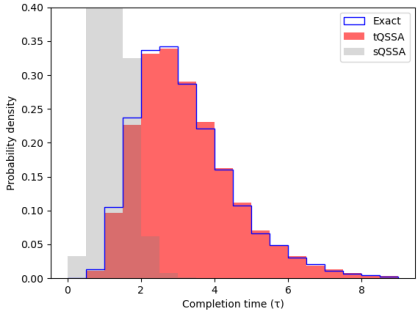
(a) CME direct integration simulation, low enzyme population, $E_T = 1$.



(b) CME direct integration simulation, high enzyme population, $E_T = 10$.



(c) Gillespie algorithm simulation, low enzyme population, $E_T = 1$.



(d) Gillespie algorithm simulation, high enzyme population, $E_T = 10$.

Figure 5.4: Probability density of the completion times for a single-substrate reaction using CME direct integration ($\Delta t = 10^{-4}$) and Gillespie algorithm (20,000 Monte-Carlo simulations). Simulation parameters as in Fig. 5.3.

start at zero and tends to zero as $t \to \infty$ (because the completion time is unbounded), while it reaches a maximum during evolution. From the plots, the maximum standard deviation appears to be reached more or less at the time where half of the initial substrate is converted into products, on average. This may make sense since at this point ($P = S_T/2$) the state is the furthest from its boundaries (i.e., $P = 0$ and $P = S_T$).

In Fig. 5.4, we reported the distribution of the completion times probability density for a single-substrate reaction using stochastic methods. Analogously as before, at low E_T values (Fig. 5.4a and 5.4c), QSSA closely follows the exact formulation, with tQSSA giving a slightly worse approximation. As E_T increases, the histograms of the probability density of the tQSSA and the exact formulation mostly overlap, while QSSA completely fails at giving a

	$E_T = 1$		$E_T = 10$	
	CME	Gillespie	CME	Gillespie
Exact	11.8 ± 4.0	11.8 ± 3.9	3.17 ± 1.33	3.19 ± 1.35
tQSSA	12.9 ± 4.4	12.9 ± 4.4	3.23 ± 1.35	3.23 ± 1.36
sQSSA	11.8 ± 4.0	11.8 ± 4.0	1.18 ± 0.43	1.18 ± 0.40

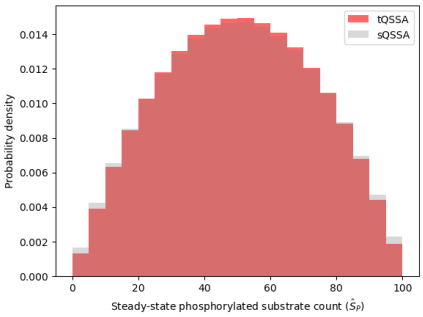
Table 5.1: Completion times mean and standard deviation.

reasonable prediction of completion times. Our comparison between tQSSA and the exact model at high enzyme population agrees with the Gillespie simulation done by Kim and Tyson [26] (Fig. 4a).

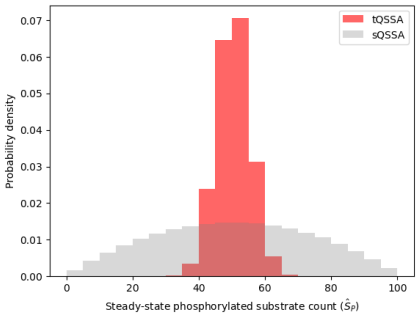
Next, we analyze the stationary distribution for the Goldbeter-Koshland switch. Differently from the continuous-limit case, in the stochastic formulation we have a stationary distribution of states rather than a single steady state. From the graph, we directly observe the effect of ultrasensitivity at low kinase and phosphatase populations (Fig. 5.5a and 5.5c) at the "activation" point $E_T = D_T$: the distribution is really spread out across all possible states in this case. For high $E_T = D_T$ (Fig. 5.5b and 5.5d), the different approximations give different results: QSSA wrongly gives the identical distribution while the tQSSA provide a much more localized distribution around the peak at $\hat{S}_P = S_T/2$, in agreement with the exact formulation. Note that in Fig. 5.5 we did not report the CME direct interagration results for the exact formulation due to excessive computational time required. Indeed, the CME approach for this system has $O(E_T D_T S_T)$ computational complexity for the exact formulation, but only $O(S_T)$ for QSSA and tQSSA. We used a longer relaxation time for the low-enzyme case: since there are just two enzymes at work in this case, it is expected that the system will require a much longer time to relax. For the Gillespie simulations, we had to use a very long simulation time for time-averaging in order to obtain statistically significant results. For the low-enzyme case, we found that Gillespie algorithm is a bad choice for small systems with long ergodic time like this one[1]. On the other hand, it appears to be much more convenient in the high-enzyme case. In all cases, the usage of the tQSSA accelerates the computation with respect to the exact formulation with negligible loss of accuracy. The slight asymmetry visible in the high-enzyme case is due to bin aliasing rather than noise: for example, the $\hat{S}_P$ count for which the probability is maximum ($S_T/2$) is contained in the right-central bin rather than the left one. We chose this bins subdivision to better compare the result with the one by Kim and Tyson [26] (Fig. 4c-d).

Finally, in Fig. 5.6 we used data from previous simulations for the tQSSA to better compare the results from Gillespie and CME. We observe that, in all cases, the results from CME and Gillespie are hardly distinguishable, as

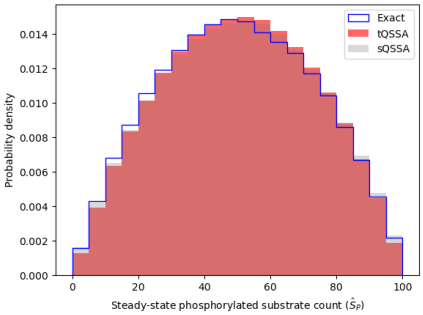
[1] In this case, $E_T = D_T = 1$, it is expected that the system has a much longer ergodic time, due to the lower number of degrees of freedom.



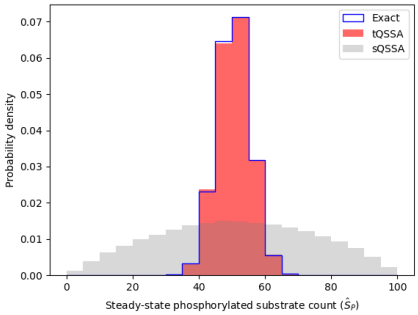
(a) Low enzymes populations: $E_T = 1$, $D_T = 1$. Integration (relaxation) time = 1000.



(b) High enzymes populations: $E_T = 100$, $D_T = 100$. Integration (relaxation) time = 100.



(c) Low enzymes populations: $E_T = 1$, $D_T = 1$. Simulation (averaging) time = 10^7.



(d) High enzymes populations: $E_T = 100$, $D_T = 100$. Simulation (averaging) time = 10,000.

Figure 5.5: Stationary distribution of $\hat{S}_P$ in the GK switch using CME direct integration and Gillespie algorithm. Simulation Parameters: $\kappa_{fe} = 10$, $\kappa_{be} = 8.3$, $\kappa_e = 1.7$, $\kappa_{fd} = 10$, $\kappa_{bd} = 8.3$, $\kappa_d = 1.7$ and $S_T = 100$ (same as in Fig. 5.2 with $N_A\Omega = 1$).

long as enough Monte-Carlo simulations are performed for the stochastic simulation algorithm. When using tQSSA, both CME and Gillespie are fast enough. tQSSA can indeed be used to accelerate not only CME integration by dimensionality reduction, but also Gillespie simulations by removing the need to calculate the reactions associated to the formation and dissociation of the enzyme-substrate complex.

	$E_T = 1$		$E_T = 10$	
	CME	Gillespie	CME	Gillespie
Exact	-	49.6 ± 22.7	-	50.0 ± 5.0
tQSSA	50.0 ± 22.4	50.1 ± 22.4	50.0 ± 5.0	50.1 ± 5.1
sQSSA	50.0 ± 22.8	50.1 ± 22.8	50.0 ± 22.8	50.7 ± 22.9

Table 5.2: Phosphorylated substrate count $\hat{S}_P$ mean and standard deviation.

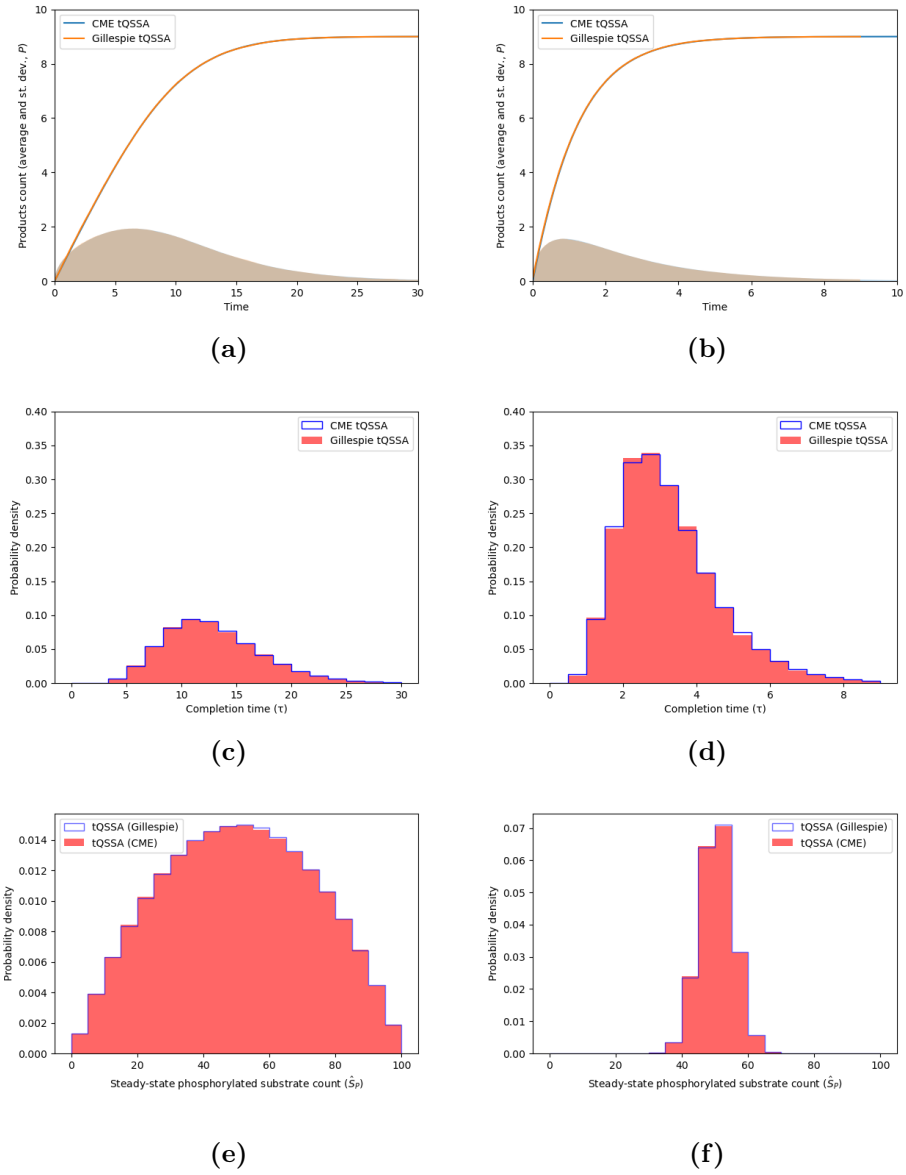


Figure 5.6: Comparison between the Gillespie and the CME algorithm. Simulation parameters as in previous experiments.

Chapter 6

Conclusion

This scientific report offers a brief exploration of enzyme kinetics, covering various mathematical models and computational methods for the study of enzyme-substrate reactions. It aims to provide an overview along with a comparative analysis of the different results.

The report delves into fundamental mathematical models, from the mass action law and CME to Michaelis-Menten kinetics and the Goldbeter-Koshland switch. It also discusses the practical application of approximations, such as the quasi-steady-state approximation (QSSA) and the total quasi-steady-state approximation (tQSSA), outlining their respective domains of applicability and limitations.

Practical tools, including Python scripts with graphical user interfaces, C++ code with Python bindings, are introduced to enable simulations and analysis of enzyme kinetics. These resources provided a dynamic and versatile means to experiment with and compare different models and methodologies.

The tQSSA has proven to be a reliable approximation applicable in a wide range of scenarios without significant loss of accuracy, in contrast to standard QSSA, which is accurate only in the low-enzyme limit. TQSSA yields precise forecasts of enzyme kinetics in the context of law of mass action and CME: in particular, in the latter case, statistics match those from the exact formulation. Additionally, we also observed that approximations are significantly computationally faster both in CME direct integration and stochastic simulation algorithm.

Further research is needed to investigate the computational time required for both CME direct integration and Gillespie algorithm. Determining the most convenient method is not trivial, since it may depend both on the size of the system and its "ergodicity": depending on the case, the computation of the statistical averages might require very long simulations using Gillespie algorithm[1]. Nonetheless, we expect that, apart from specific degenerate

[1] "In practice, one may find an apparent 'breaking of ergodicity' even for systems which are ergodic, if the 'time' over which the averaging is extended is not long enough, i.e., less

cases, problems in higher dimensions will be more tractable using stochastic simulation algorithm.

than the so-called 'ergodic time' τ_e"[19].

Bibliography

[1] C. M. Guldberg and P. Waage (1864). *Studies Concerning Affinity*, Forhandlinger i Videnskabs-Selskabet i Christiania, 35.

[2] V. Henri (1903). *Lois Générales de l'Action des Diastases*, Paris: Hermann.

[3] L. Michaelis, M. L. Menten (1913). *Die Kinetik der Invertinwirkung*, Biochem Z. 49: 333–369.

[4] M. Trautz (1916). *Das Gesetz der Reaktionsgeschwindigkeit und der Gleichgewichte in Gasen. Bestätigung der Additivität von Cv - 3/2 R. Neue Bestimmung der Integrationskonstanten und der Molekül durchmesser*, Zeitschrift für anorganische und allgemeine Chemie, Volume 96, Issue 1, Pages 1–28.

[5] W. C. M. Lewis (1918). *XLI.—Studies in catalysis. Part IX. The calculation in absolute measure of velocity constants and equilibrium constants in gaseous systems*, J. Chem. Soc., Trans., 113, 471-492.

[6] G. E. Briggs, J. B. S. Haldane (1925). *A Note on the Kinetics of Enzyme Action*, Biochem J. 19(2): 338–339.

[7] A. Kolmogorov (1931). *Über die analytischen Methoden in der Wahrscheinlichkeitsrechnung*, Mathematische Annalen. 104: 415–458.

[8] A. Nordsieck, W. E. Lamb Jr., G. E. Uhlenbeck (1940). *On the theory of cosmic-ray showers I the furry model and the fluctuation problem*, Physica 7(4): 344-360.

[9] W. Feller (1940). *On the Integro-Differential Equations of Purely Discontinuous Markoff Processes*, Transactions of the American Mathematical Society 48 (3): 4885–15.

[10] J. L. Doob (1945). *Markoff chains – Denumerable case*, Transactions of the American Mathematical Society. 58 (3): 455–473.

[11] M. S. Bartlett (1953). *Stochastic Processes or the Statistics of Change*, Journal of the Royal Statistical Society, Series C. 2 (1): 44–64.

[12] M. J. D. Powell (1970), *A Hybrid Method for Nonlinear Equations. Numerical Methods for Nonlinear Algebraic Equations*, P. Rabinowitz, editor. Gordon and Breach.

[13] D. T. Gillespie (1977). *Statistical Exact Stochastic Simulation of Coupled Chemical Reactions*, Berlin/New York: Springer/Academic Press.

[14] L. Petzold (1983). *Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations*, SIAM Journal on Scientific and Statistical Computing, Vol. 4, No. 1, pp. 136-148.

[15] A. C. Hindmarsh (1983). *ODEPACK, A Systematized Collection of ODE Solvers*, IMACS Transactions on Scientific Computation, Vol 1., pp. 55-64.

[16] L. A. Segel, M. Slemrod (1989). *The quasi-steady-state assumption: a case study in perturbation*, SIAM Rev. 31(3):446-477.

[17] J. Borghans, R. de Boer, L. Segel (1996). *Extending the quasi-steady state approximation by changing variables*, Bull. Math. Biol. 58: 43-63.

[18] R. A. Copeland (2000). *ENZYMES: A Practical Introduction to Structure, Mechanism, and Data Analysis*, second edition, Wiley-VCH.

[19] K. Binder, D. W. Heermann (2002). *Monte Carlo Simulation in Statistical Physics: An Introduction*, Springer Series in Solid-State Sciences, Springer, Berlin.

[20] A. R. Tzafriri (2003). *Michaelis-Menten kinetics at high enzyme concentrations*, Bull. Math. Biol. 65(6):1111-1129.

[21] D. T. Gillespie (2007). *Stochastic Simulation of Chemical Kinetics*, Annu. Rev. Phys. Chem. 58:35-55.

[22] M. G. Pedersen, A. M. Bersani, E. Bersani, G. Cortese (2008). *The total quasi-steady-state approximation for complex enzyme reactions*, Mathematics and Computers in Simulation 79: 1010-1019.

[23] Y. Cao, D. C. Samuels (2009). *Discrete Stochastic Simulation Methods for Chemically Reacting Systems*, Methods Enzymol. 454: 115-140.

[24] K. R. Sanft, D. T. Gillespie, L. R. Petzold (2011). *Legitimacy of the stochastic Michaelis-Menten approximation*, IET Syst. Biol., Vol. 5, Iss. 1, pp. 58-69.

[25] V. Sunkara (2019). *On the Properties of the Reaction Counts Chemical Master Equation*, Entropy 21(6), 607.

[26] J. K. Kim, J. J. Tyson (2020). *Misuse of the Michaelis-Menten rate law for protein interaction networks and its remedy*, PLoS Computational Biology 16(10).

[27] T.-H. Ahn, Y. Cao, L. T. Watson. *Stochastic Simulation Algorithms for Chemical Reactions*, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.