Stochastic Simulations

Autumn Semester 2025

Prof. Fabio Nobile *Assistants: Eliott Van Dieren and Riccardo Saporiti*

Project 7

submission deadline: 11 January 2026

Efficient Simulation of Stochastic Reaction Networks

1 Introduction and background

Stochastic reaction networks (SRNs) are continuous-time Markov chains intended to describe, from the kinetic point of view, the time-evolution of chemical systems in which molecules of different chemical species undergo a finite set of reaction channels.

A SRN is a stochastic process in continuous time, $\boldsymbol{X} : [0, T] \times \Omega \to \mathbb{Z}_+^d$, that describes the time evolution of a homogeneously-mixed chemically reacting system where d different species of molecules, $(S_1, S_2, ..., S_d)$, undergo a finite set of reaction channels $R_1, R_2, ..., R_J$. The $i-th$ coordinate of $\boldsymbol{X}$ is a non-negative integer that keeps track of the abundance of the $i-th$ species, S_i, at time t. The reaction channels are pairs, $R_j = (\boldsymbol{\nu}_j, a_j)$, $j = 1, ..., J$, where $\boldsymbol{\nu}_j \in \mathbb{Z}^d$ are known as stoichiometric vectors, and $a_j : \mathbb{Z}_+^d \to \mathbb{R}^+$ are known as propensity functions. More concretely, a SRN is a continuous-time Markov chain defined by the probabilities:

$$\begin{cases} P(\boldsymbol{X}(t + dt) = \boldsymbol{x} + \boldsymbol{\nu}_j \mid \boldsymbol{X}(t) = \boldsymbol{x}) = a_j(\boldsymbol{x})\, dt, \quad j = 1, 2, \ldots, J \\ P(\boldsymbol{X}(t + dt) = \boldsymbol{x} \mid \boldsymbol{X}(t) = \boldsymbol{x}) = 1 - \sum_{j=1}^{J} a_j(\boldsymbol{x})\, dt. \end{cases} \tag{1}$$

The set of equations (1) means that the probability of observing a jump of the process $\boldsymbol{X}$, from the state $\boldsymbol{x}$ to the state $\boldsymbol{x}+\boldsymbol{\nu}_j$, caused by the firing of the j-th reaction channel R_j, during the infinitesimal time interval, $(t, t + dt]$, is proportional to the length of the time interval dt, with $a_j(\boldsymbol{x})$ as the constant of proportionality. Furthermore due to the memory-less property of Markov processes, given that $\boldsymbol{X}$ is at state $\boldsymbol{x}$ at time t, the time to the next reaction is exponentially distributed with parameter $a_0(\boldsymbol{x}) = \sum_{j=1}^{J} a_j(\boldsymbol{x})$.

Another feature taken from the theory of stochastic chemical kinetics is the so called stochastic mass-action kinetics principle, which provides a mathematical model for the reaction channels $(R_j)_j$. The stochastic mass-action kinetics principle is usually represented by a diagram like:

$$\alpha_{j,1} S_1 + \cdots + \alpha_{j,d} S_d \xrightarrow{c_j} \beta_{j,1} S_1 + \cdots + \beta_{j,d} S_d, \tag{2}$$

implying that, when the j-th reaction channel, $R_j = (\boldsymbol{\nu}_j, a_j(\boldsymbol{x}))$, fires in the infinitesimal time interval $(t, t + dt]$, and the process $\boldsymbol{X}$ is at the state $\boldsymbol{x}$ at time t, the number of particles of the species S_i changes from x_i to $x_i - \alpha_{j,i} + \beta_{j,i}$. The vector $\boldsymbol{\nu}_j$ in (1) is then defined by

$$\boldsymbol{\nu}_j = (\beta_{j,1} - \alpha_{j,1}, ..., \beta_{j,d} - \alpha_{j,d}), \quad j = 1, ..., J, \tag{3}$$

and the propensity function a_j is given by

$$a_j(\boldsymbol{x}) = c_j \prod_{i=1}^{d} \frac{x_i!}{(x_i - \alpha_{j,i})!} \chi_{\{x_i \geq \alpha_{j,i}\}}, \tag{4}$$

where $c_j > 0$ and $\chi_{\{A\}}$ is the indicator function of the set A. The factor $\prod_{i=1}^{d} \chi_{\{x_i \geq \alpha_{j,i}\}}$ in (4) guarantees that the components of $\boldsymbol{x}$ remain non-negative at all times.

Developing fast simulation algorithms for SRN is the goal of this project. At the core of them is the following random time change representation

$$\boldsymbol{X}(t) = \boldsymbol{x}_0 + \sum_{j=1}^{J} \boldsymbol{\nu}_j Y_j \left(\int_0^t a_j(\boldsymbol{X}(s)) ds \right), \tag{5}$$

where $Y_j : \mathbb{R}^+ \times \Omega \to \mathbb{N}$ are *independent unit-rate* Poisson processes. For more details regarding SRNs, see [1] and [2, Chapter 1].

2 Stochastic Simulation Algorithm (SSA)

In this project, we consider the Michaelis–Menten reactions, which involve four species ($d = 4$), namely a substrate S_1, an enzyme S_2, a complex S_3 and a product S_4; and three reactions ($J = 3$)

$$\begin{cases} S_1 + S_2 \xrightarrow{c_1} S_3, \\ S_3 \xrightarrow{c_2} S_1 + S_2, \\ S_3 \xrightarrow{c_3} S_4 + S_2. \end{cases} \tag{6}$$

Overall, this system describes how the substrate is converted into the product with the help of an enzyme.

Consider now the quantity $p(\tau, j|\boldsymbol{x}, t)$, defined as follows: given $\boldsymbol{X}(t) = \boldsymbol{x}$, $p(\tau, j|\boldsymbol{x}, t)dt$ is the probability that the next reaction will be the $j - th$ reaction and will occur in the time interval $[t + \tau, t + \tau + dt)$. One can derive the following expression

$$p(\tau, j|\boldsymbol{x}, t) = \frac{a_j(\boldsymbol{x})}{a_{sum}(\boldsymbol{x})} a_{sum}(\boldsymbol{x}) e^{-a_{sum}(\boldsymbol{x})\tau} = a_j(\boldsymbol{x}) e^{-a_{sum}(\boldsymbol{x})\tau}, \tag{7}$$

where $a_{sum}(\boldsymbol{x}) = \sum_{k=1}^{J} a_k(\boldsymbol{x})$. Formally, $p(\tau, j|\boldsymbol{x}, t)$ is the joint distribution of two random variables (one discrete and one continuous) and (7) shows that it can be written as the product of two distributions

- $\{a_j(\boldsymbol{x})/a_{sum}(\boldsymbol{x})\}_{j=1}^{J}$ are probabilities defining the *next reaction index*.
- $a_{sum}(\boldsymbol{x}) e^{-a_{sum}(\boldsymbol{x})\tau}$ is the probability density function of a continuous random variable with *exponential distribution* that determines the *time until the next reaction*.

Thanks to this last comment we can simulate independently a reaction index and a reaction time. Using these considerations, the Stochastic Simulation Algorithm (SSA) is given in Algorithm 1.

Algorithm 1 Stochastic Simulation Algorithm

1: Given $\boldsymbol{X}(0)$, set $t = 0$.
2: Evaluate $\{a_k(\boldsymbol{X}(t))\}_{k=1}^{J}$ and $a_{sum}(\boldsymbol{X}(t)) = \sum_{k=1}^{J} a_k(\boldsymbol{X}(t))$.
3: Sample $I \in \{1, 2, ..., J\}$ with probability mass function $\mathbb{P}(I = j) = \frac{a_j(\boldsymbol{X}(t))}{\sum_{l=1}^{J} a_l(\boldsymbol{X}(t))}$, which is the probability that the j^{th} reaction happens.
4: Sample the time of the next reaction: $\tau \sim \text{Exp}\,(a_{sum}(\boldsymbol{X}(t)))$, where $a_{sum}(\boldsymbol{X}(t))$ is the rate parameter of the exponential random variable.
5: $\boldsymbol{X}(t + \tau) = \boldsymbol{X}(t) + \boldsymbol{\nu}_I$, and update t to $t + \tau$.

2.1 Goals for this section

1. Using (3) and (4), derive the stoichiometric vectors and the propensity functions associated to the Michaelis-Menten SRN defined in (6).

2. Consider the following *rate constants*: $c_1 = 0.0017$, $c_2 = 10^{-4}$, $c_3 = 0.1$ and the following initial conditions $X_1(0) = 312$, $X_2(0) = 125$, $X_3(0) = X_4(0) = 0$. Using the SSA algorithm, simulate the reactions until time exceeds $T = 50$. Plot few realizations of the evolution in time of each state $X_i(t)$ and comment your results.

3. Estimate with Monte Carlo the quantity of interest $Z = \mathbb{E}[X_4(T)]$. Fix a threshold for the half-width of a confidence interval at level $1 - \alpha$ and iteratively increase the sample size N until you achieve the desired tolerance. Report and comment your results.

4. Consider the *rate constants*: $c_1 = 0.001$, $c_2 = 0.005$, $c_3 = 0.01$, the initial state $X_1(0) = 100$, $X_2(0) = 100$, $X_3(0) = X_4(0) = 0$ and final time $T = 1$. Using a crude Monte Carlo estimator, compute the following quantity of interest $Z = \mathbb{E}[g(\boldsymbol{X}(T))] = \mathbb{E}[\chi_{\{X_3(T)>22\}}]$. Based on different sample sizes give an estimation of the variance associated with your estimator. What do you observe?

5. Consider the implementation of an Importance Sampling (IS) strategy to approximate $Z = \mathbb{E}[g(\boldsymbol{X}(T))] = \mathbb{E}[\chi_{\{X_3(T)>22\}}]$. The importance distribution is constructed by modifying the rate constants c_1, c_2 and c_3 to $\tilde{c}_1$, $\tilde{c}_2$ and $\tilde{c}_3$. Therefore, the associated propensity functions are modified as well. Choose heuristic values for the new rate constants. Justify your choice and quantify the variance reduction obtained.

3 Approximated simulation algorithms

3.1 Tau-Leaping

If the SRN involves many molecules or some fast reactions the SSA requires to simulate many events in the time interval $[0, T]$ of interest. An alternative approach is to fix a time step of length τ and fire all the reactions that would take place in the interval $[t, t + \tau]$. The propensity functions a_j are evaluated on the state $\boldsymbol{X}(t)$ and are not updated for the whole time interval. Using this approximation in (5) leads to the *tau-leaping* method:

$$\boldsymbol{X}(t + \tau) = \boldsymbol{X}(t) + \sum_{j=1}^{J} \boldsymbol{\nu}_j \mathcal{P}_j(a_j(\boldsymbol{X}(t))\tau), \tag{8}$$

here $\mathcal{P}_j(a_j(\boldsymbol{X}(t))\tau)$ counts the number of times the $j-th$ reaction fired in $[t, t+\tau)$. If we suppose that $a_j(\boldsymbol{X}(t))$ stays constant in $[t, t+\tau)$, then $\mathcal{P}_j(a_j(\boldsymbol{X}(t))\tau)$ has a Poisson distribution with parameter $a_j(\boldsymbol{X}(t))\tau$. Then, the *tau-leaping* algorithm can be implemented according to Algorithm 2.

Algorithm 2 Tau Leaping

1: Given $\boldsymbol{X}(0)$, set $t = 0$.
2: For $j = 1, ..., J$, sample independently $p_j \sim \mathcal{P}_j(a_j(\boldsymbol{X}(t))\tau)$
3: $\boldsymbol{X}(t+\tau) = \max(\boldsymbol{0}, \boldsymbol{X}(t) + \sum_{j=1}^{J} \boldsymbol{\nu}_j p_j)$, and update t to $t+\tau$.

3.1.1 Goals for this subsection

1. Verify that when the rates $a_j(\boldsymbol{X}(t))$ are kept constant in the interval $[t, t+\tau]$, then the random variables p_j, $j = 1, ..., J$ in step 2 of Algorithm 2 are independent and each p_j has a Poisson distribution with parameter $a_j(\boldsymbol{X}(t))\tau$.

2. Implement the *tau-leaping* algorithm according to Algorithm (2) and simulate the Michaelis–Menten SRN until time $T = 50$. Use the same initial condition and rates as in point 2 of section 2. Consider a time discretization $\tau = 0.2$. Comment your results.

3. Estimate $Z_{\Delta t} = \mathbb{E}[X_4(T)]$ with a Monte Carlo estimator. The subscript '$_{\Delta t}$' emphasizes the bias introduced by the time discretization. Compare your results with those you obtained in point 3 of section 2, in which the estimation error is only affected by the sample size.

   Fix a threshold $1-\alpha$ for a confidence interval and iteratively increase the sample size N until you achieve the desired tolerance. Compare then, for the same threshold $1-\alpha$, the confidence intervals you obtained by the SSA algorithm and the explicit *tau-leaping*. Test different values of $\tau \geq 0.2$ and different sample sizes. What can you conclude?

4. In order to increase the time-stepsize of the *tau-leap* algorithm, one can think at an implicit discretization. To do this we add and subtract from (8) the term $\sum_{j=1}^{J} \mathbb{E}[p_j]\boldsymbol{\nu}_j$. The *implicit tau-leap* method reads

   (a) Solve with Newton (or use a built in function) $\boldsymbol{y} = \boldsymbol{X}(t) + \tau \sum_{j=1}^{J} a_j(\boldsymbol{y})\boldsymbol{\nu}_j$

   (b) Simulate: $\boldsymbol{X}(t+\tau) = \boldsymbol{X}(t) + \tau \sum_{j=1}^{J} \boldsymbol{\nu}_j \mathcal{P}_j(a_j(\boldsymbol{y})\tau)$ (observe that $\boldsymbol{X}(t+\tau) \in \mathbb{Z}_+^d$).

   Repeat the previous point with the implicit *tau-leaping* scheme. Which algorithm provides a more reliable estimation?

3.2 Chemical Langevin Equation

Suppose now that τ is chosen in such a way that $a_j(\boldsymbol{X}(t))\tau$ is large for all $j = 1, ..., J$. The central limit theorem guarantees that a Poisson random variable with large mean is well approximated by a Normal random variable with the same mean and variance. Replacing $\mathcal{P}_j(a_j(\boldsymbol{X}(t))\tau)$ with $a_j(\boldsymbol{X}(t))\tau + \sqrt{a_j(\boldsymbol{X}(t))\tau} Z_j$, where the $Z_j \sim \mathcal{N}(0,1)$ we arrive at the relation

$$\boldsymbol{Y}(t+\tau) = \boldsymbol{Y}(t) + \tau \sum_{j=1}^{J} \nu_j a_j(\boldsymbol{Y}(t)) + \sqrt{\tau} \sum_{j=1}^{J} \nu_j \sqrt{a_j(\boldsymbol{Y}(t))} Z_j, \tag{9}$$

where $\boldsymbol{Y}$ replaces what before we called $\boldsymbol{X}$ to emphasize that, using the central limit theorem, we switched from integer to real numbers in the representation of the number of molecules of each species. Equation (9) corresponds to the Euler-Maruyama discretization of the Stochastic Differential Equation (SDE)

$$d\boldsymbol{Y}(t) = \sum_{j=1}^{J} \nu_j a_j(\boldsymbol{Y}(t))dt + \sum_{j=1}^{J} \nu_j \sqrt{a_j(\boldsymbol{Y}(t))} dW^j(t), \tag{10}$$

where W^j are independent scalar Brownian Motions. This SDE is named Chemical-Langevin-Equation.

3.2.1 Goals for this subsection

1. For each chemical species of the Michaelis–Menten SRN derive the explicit form of the associated Chemical-Langevin-Equation. That is, by replacing the stoichiometric coefficients in (10) and the propensity functions with your computations of section 2, point 1, express the coefficients of the SDE as a function of the state variables $Y_j(t)$ and the reaction rates c_j.

2. Apply the Euler-Maruyama scheme to simulate equation (10) under the form derived in the previous point. Set the same initial condition $\boldsymbol{Y}(0) = \boldsymbol{X}(0)$ as in section 2, point 2, and use a time discretization step $\tau = 0.2$. Plot the evolution in time of each state $Y_i(t)$ and comment on your results.

   Hint: you may need to take the absolute value of the state vector $\boldsymbol{Y}(t)$ since it is not guaranteed to remain nonnegative.

3. Repeat point 3 of section 3.1. Compare the confidence interval you derive with a high-fidelity simulation based on the SSA algorithm. Finally, fix, a priori, the computational cost of your simulation in terms of sample size and time-step size. Compare the $tau-leaping$ and the Chemical-Langevin algorithms. Which of the two is more accurate?

References

[1] Daniel T Gillespie, *A general method for numerically simulating the stochastic time evolution of coupled chemical reactions*, Journal of Computational Physics **22** (1976), no. 4, 403–434.

[2] Alvaro Moraes, *Simulation and statistical inference of stochastic reaction networks with applications to epidemic models*, Ph.D. thesis, KAUST Research Repository, 2015.