Stochastic Simulation Algorithms for Chemical Reactions

Tae-Hyuk Ahn[*], **Yang Cao**[*], **and Layne T. Watson**[*+]
Departments of Computer Science[*] and Mathematics[+]
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061-0106

Abstract – *In various biochemical systems, discrete and stochastic approaches are more appropriate than continuous and deterministic approaches when the system has small numbers of molecules. Before the emergence of the stochastic simulation algorithm (SSA) by Gillespie, chemical master equations were used to stochastically model. Solving the master equations is often mathematically intractable. Therefore, it is reasonable to investigate the SSA in order to understand stochastic processes for biochemical systems. Ever since the SSA emerged, there have been many papers to improve the computational efficiency of the SSA. This paper explains and compares various stochastic simulation algorithms for chemical reactions, with particular attention to the Gillespie algorithm.*

Keywords: stochastic simulation algorithm(SSA), chemical reaction, tau-leaping, quasi-steady state, Goldbeter-Koshland switch

1. Introduction

Biochemistry is the study of chemical processes in living organisms, and overlaps with the disciplines of chemistry and molecular biology. A chemical reaction is a process that results in a change of chemical substances. The change is influenced by external factors such as temperature, density, and time. Chemical reactions usually yield some new products that are different from the initial substances. Current understanding of how and why chemical reactions occur is based on the atomic model of matter and quantum mechanics. There are many reaction types, but only a few simple reactions and the Goldbeter-Koshland switch will be considered here [1]. There are two important frameworks for modeling chemical reactions. The continuous and deterministic approach is adequate for understanding the average behavior of large numbers of molecules [2]. Deterministic modeling produces a system of first order ordinary differential equations (ODEs) with concentrations of chemical species as variables. These equations are called "reaction rate equations (RRE)" and the various species concentrations are produced by solving the RRE. The discrete and stochastic approach is proper for systems that contain small numbers of molecules. In living cells, small numbers of molecules react with each other and random behaviors (thermal noise) arise in the system. Therefore, stochastic simulation is an appropriate and accurate method for systems such as the cell cycle and gene regulatory networks [3].

One stochastic method solves the chemical master equation [4]. The chemical master equation (CME), the time-evolution equation, describes the probability density of species in chemical reactions. In spite of the advantage that the CME gives the time evolution of the probability distribution of all possible states directly, solving the CME is often mathematically intractable [5]. Another common stochastic approach is the stochastic simulation algorithm (SSA) using Monte Carlo methods to simulate the chemical processes defined by the CME ([5], [6]). The SSA is described in Section 2. The SSA is an exact stochastic method to simulate chemical systems, but the SSA is often slow because it simulates every reaction. Since the SSA emerged, there have been many attempts to improve the computational efficiency. Progress has been made to improve the time efficiency ([7], [8]), however, the core algorithm is the same. One remarkable attempt to improve the SSA is the tau-leaping method [9]. The tau-leaping method achieves increased computational efficiency by leaping over many fast reactions. The implicit tau-leaping method compensates for the tau-leaping method's difficulty with stiff systems [10]. Stiff systems are characterized by well separated fast and slow time scales in a dynamic system, the fastest of which is stable. Another algorithm for dealing with stiff systems is the slow-scale SSA [11]. Some approaches try to reduce time consumption with different assumptions such as QSSA [12] and tQSSA [13] for stiff systems. This paper compares computational efficiency and exactness between SSA, tau-leaping, implicit tau-leaping, QSSA, and tQSSA based on numerical experiments with simple chemical reactions and stiff systems.

2. Stochastic Simulation Algorithms

2.1 SSA

Suppose a biochemical system or pathway involves N molecular species $\{S_1, ..., S_N\}$. $X_i(t)$ denotes the number of molecules of species S_i at time t. People would like to study the evolution of the state vector $X(t) = (X_1(t), ..., X_N(t))$ given that the system was initially in the state vector $X(t_0)$. Suppose the system is composed of M reaction channels $\{R_1, ..., R_M\}$. In a constant volume Ω, assume that the system is well-stirred and in thermal equilibrium at some constant temperature. There are two important quantities in reaction channels R_j: the state change vector $v_j = (v_{1j}, ..., v_{Nj})$, and propensity function a_j. v_{ij} is defined as the change in the S_i molecules' population caused by one R_j reaction, and $a_j(x)dt$ gives the probability that one R_j reaction will occur in the next infinitesimal time interval $[t, t + dt)$.

The SSA simulates every reaction event ([5], [6]). With $X(t) = x$, $p(\tau, j|x, t)d\tau$ is defined as the probability that the next reaction in the system will occur in the infinitesimal time interval $[t+\tau, t+\tau+d\tau)$, and will be an R_j reaction. By letting $a_0(x) \equiv \sum_{j=1}^{M} a_j(x)$, the equation

$$p(\tau, j|x, t) = a_j(x) \exp(-a_0(x)\tau),$$

can be obtained. A Monte Carlo method is used to generate τ and j. On each step of the SSA, two random numbers r_1 and r_2 are generated from the uniform (0,1) distribution. From probability theory, the time for the next reaction to occur is given by $t + \tau$, where

$$\tau = \frac{1}{a_0(\mathbf{x})} \ln(\frac{1}{r_1}).$$

The next reaction index j is given by the smallest integer satisfying

$$\sum_{j'=1}^{j} a_{j'}(x) > r_2 a_0(x).$$

After τ and j are obtained, the system states are updated by $X(t+\tau) := x+v_j$, and the time is updated by $t := t + \tau$. This simulation iteration proceeds until the time t reaches the final time.

2.2 Explicit Tau-leaping

The SSA is an exact stochastic method for chemical reactions, however, it is very slow for many practical systems because the SSA simulates one reaction at a time. One approximate simulation approach is *tau-leaping* [9]. The basic idea of the tau-leaping method is that many reactions can be simulated at each step with a preselected time τ. The tau-leaping method requires that the selected τ must be small enough to satisfy the "leap condition": The expected state change induced by the leap must be sufficiently small that propensity functions remain nearly constant during the time step τ.

$K_j(\tau; x, t)$ is defined as the number of times, given $X(t) = x$, that reaction channel R_j will fire in the time interval $[t, t + \tau)$ where $j = 1, \ldots, M$. If $X(t) = x$, then the state can be updated by

$$X(t + \tau) = x + \sum_{j=1}^{M} K_j(\tau; x, t)v_j.$$

$K_j(\tau; x, t)$ is modelled by a Poisson random variate. The explicit tau-leaping method assumes

$$K_j(\tau; x, t) = P_j(a_j(x)\tau),$$

where P_j is a Poisson random variate with mean and variance $a_j(x)\tau$.

In order to select the largest value of τ that satisfies the leap condition, the Jacobian matrices for the propensity functions are used ([9], [14]). One new approach is to select τ such that relative changes in the propensity functions are bounded [15]. This new τ selection procedure is faster and more accurate than previous methods. Therefore, the explicit tau-leaping method proceeds as follows. Select a τ that satisfies the leap condition. Generate the Poisson random variables for each reaction and adjust the leap time by $t := t + \tau$ and the states by $X(t + \tau) := x + \sum_{j=1}^{M} K_j(\tau; x, t)v_j$. This simulation iteration also proceeds until the time t reaches the final time t_f.

2.3 Implicit Tau-leaping

The implicit tau-leaping method addresses the shortcomings of the explicit tau-leaping method when the systems are stiff [10]. Stiff systems are characterized by well separated fast and slow time scales in a dynamic system, with the fast mode being stable. In a stiff system, solutions by the explicit tau-leaping method are unstable unless the time stepsize τ is kept smaller than the smallest (fastest) time scale in the system [10].

The tau-leaping method in Section 2.2 is an explicit method because the propensity functions a_j are evaluated at the current known state x. Therefore, the future state $X(t + \tau)$ is an explicit function of $X(t)$, and the states can be updated by the equation,

$$X^{et}(t + \tau) = x + \sum_{j=1}^{M} v_j P_j(a_j(x)\tau),$$

where the superscript “et” stands for explicit method. The implicit method is described by

$$X^{it}(t+\tau) =$$

$$x + \sum_{j=1}^{M} v_j \Big[\tau a_j\big(X^{it}(t+\tau)\big) + P_j(a_j(x)\tau) - \tau a_j(x)\Big],$$

where the superscript “it” stands for implicit method. The implicit equation is solved by Newton’s method, and the floating point state $X^{it}(t+\tau)$ is rounded to the nearest integer values.

2.4 QSSA

The explicit and implicit tau-leaping methods achieve increased computational efficiency by attempting to leap over many fast reactions. The quasi-steady state approximation (QSSA) [12], however, improves the efficiency with a steady state assumption. Assume that the species are separated into slow reacting species and fast reacting species. In deterministic kinetics, the net rate of formation is approximately equal to zero when the fast reacting species are in a quasi-steady state. Consider simple common enzyme kinetic reactions (Michaelis-Menten kinetics). For substrate S, enzyme E, and product P, the Michaelis-Menten reaction is

$$E + S \underset{k_{-1}}{\overset{k_1}{\Longleftrightarrow}} E:S \xrightarrow{k_2} P + E,$$

where k_1, k_{-1}, and k_2 are the rate constants. E:S is the enzyme-substrate complex after the combination of substrate and enzyme. The rate equations corresponding to this reaction are

$$\frac{d[S]}{dt} = -k_1[S][E] + k_{-1}[E:S],$$

$$\frac{d[E:S]}{dt} = -\frac{d[E]}{dt} = -(k_{-1}+k_2)[E:S] + k_1[S][E],$$

$$\frac{d[P]}{dt} = k_2[E:S],$$

where $[X]$ denotes the concentration of the species X. Assume that total enzyme concentration $E_T = [E] + [E:S]$ and $[S] \gg [E:S]$. From the assumption, the characteristic time scale of $[S]$ is very slow in comparison to that of $[E:S]$, and $[E:S]$ reaches steady state quickly. From the quasi-steady state assumption, the rate equation for $[E:S]$ is approximated by

$$\frac{d[E:S]}{dt} = 0.$$

Mathematically, it is possible to obtain a single rate equation,

$$\frac{d[S]}{dt} = -k_2[E:S] = -\frac{k_2 E_T[S]}{K_m + [S]},$$

where $K_m = (k_{-1} + k_2)/k_1$.

Similarly, the quasi-steady state assumption can be applied to the stochastic formulation. The QSSA in stochastic kinetics implies that the net rate of change for the conditional probability distribution of the fast reacting species is equal to zero. For the above Michaelis-Menten kinetics, the separate master equation for the enzyme-substrate complex is

$$\frac{dP([E:S] \mid [\hat{S}]; t)}{dt} = 0,$$

where $[\hat{S}] = [S] + [E:S]$. From the above equation, one can easily derive the reduced system

$$\hat{S} \longrightarrow P,$$

with the propensity function

$$a(s) = \frac{k_2 E_T[\hat{S}]}{K_m + [\hat{S}]}.$$

Finally, the SSA is applied to this reduced system, and shows improved computational efficiency compared with using SSA on the original system.

2.5 tQSSA

In Section 2.4, the quasi-steady state approximation (QSSA) eliminates the fastest reacting variable under some assumptions. In Michaelis-Menten kinetics, the necessary condition for the QSSA is $S_0 \gg E_T$, where S_0 is the initial substrate concentration, and E_T is the total enzyme concentration. In a protein interaction network (PIN), however, the enzymes and substrates often swap their roles [16]. Therefore, the QSSA condition will not be true for such a PIN. Borghans et al. proposed that the proper slow timescale variable is $[\hat{S}] = [S] + [E:S]$ instead of $[S]$ [17]. In terms of this variable, the deterministic equations are

$$E_T = [E] + [E:S] = \text{constant},$$

$$[E:S]^2 - \big(E_T + K_M + [\hat{S}]\big)[E:S] + E_T[\hat{S}] = 0,$$

$$\frac{d[\hat{S}]}{dt} = -k_2[E:S].$$

This is called the total quasi-steady state approximation (tQSSA). To derive the equations for the stochastic simulation under the tQSSA, reduce the system to

$$\hat{S} \longrightarrow P$$

with the propensity function

$$a(s) = k_2[E:S],$$

where

$$[E:S] =$$

$$\frac{\big(E_T + K_m + [\hat{S}]\big) - \sqrt{\big(E_T + K_m + [\hat{S}]\big)^2 - 4E_T[\hat{S}]}}{2}.$$

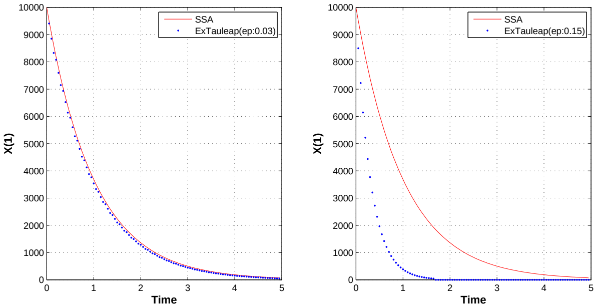

Fig. 1. The SSA simulation (solid lines) and the explicit tau-leaping simulation (dotted lines) for the irreversible isomerization. The error control parameter ϵ is 0.03 (left) and 0.15 (right).

Table 1. The number of runs and elapsed CPU time (sec) with the SSA and explicit tau-leaping method, where $t_f = 5$, $c_1 = 1$, $X_1 = 10^4$, and $\epsilon = 0.03$.

Number of runs	1000	5000	10000	50000
SSA	48.74	240.40	487.76	2433.28
Explicit Tau-leaping	2.73	14.21	28.37	143.29

Finally, the SSA is applied to this reduced system, and shows improved computational efficiency compared with using SSA on the original system. Moreover, the tQSSA overcomes the modelling shortcomings of the QSSA.

3. Numerical Experiments

The irreversible isomerization system and the Goldbeter-Koshland switch will be used here to compare various stochastic algorithms. The irreversible isomerization will show computational efficiency of the explicit tau-leaping method compared with the SSA on a nonstiff system. The GK switch is a suitable stiff model to compare various stochastic algorithms. All programs are implemented in Fortran 95 and run on a Linux system with a dual core 3.00 GHz CPU and 2 GB of memory.

3.1 Irreversible Isomerization

The first application is the simplest chemical reaction, the irreversible isomerization

$$S_1 \xrightarrow{c_1} 0.$$

The initial parameters for this system are reaction rate constant $c_1 = 1$, and 10^4 S_1 molecules at time 0. Figure 1 shows the SSA and explicit tau-leaping results. The final time is $t_f = 5$ and the error control parameters are $\epsilon = 0.03$ (left) and $\epsilon = 0.15$ (right). In order to compare the exactness, 5000 runs are used for both methods, with Fig. 1 showing mean and mean ± one standard deviation (three lines, visually identical). Figure 1 shows that increasing ϵ, while making tau-leaping faster, affects the accuracy. The exact SSA method requires 9925 steps to reach the final time ($t_f = 5$). The tau-leaping method requires 167 (34) leap steps for $\epsilon = 0.03$ ($\epsilon = 0.15$). Table 1 compares the computational efficiencies of the explicit tau-leaping method and the SSA. Explicit tau-leaping is about 17 times faster than the SSA with an appropriate approximation for this model.

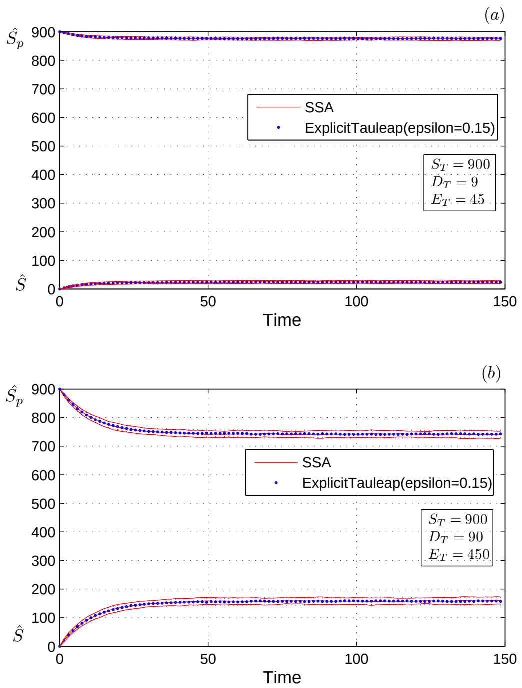

Fig. 2. The SSA simulation (solid lines) and the explicit tau-leaping simulation (dotted lines) for the GK switch. State space (S_T, D_T, E_T) is (a):(900, 9, 45) and (b):(900, 90, 450).

3.2 Goldbeter-Koshland switch

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

$$D + S_p \underset{k_{-1d}}{\overset{k_{1d}}{\rightleftharpoons}} D:S_p \xrightarrow{k_{2d}} S + D,$$

$$E + S \underset{k_{-1e}}{\overset{k_{1e}}{\rightleftharpoons}} E:S \xrightarrow{k_{2e}} S_p + E.$$

The parameter values are $S_T = 900$, $k_{1d} = 0.05555\text{min}^{-1}$, $k_{-1d} = 0.83\text{min}^{-1}$, $k_{2d} = 0.17\text{min}^{-1}$, $k_{1e} = 0.05\text{min}^{-1}$, $k_{-1e} = 0.8\text{min}^{-1}$, and $k_{2e} = 0.1\text{min}^{-1}$. In order to observe how the relationship between S_T and E_T affects the results, the two

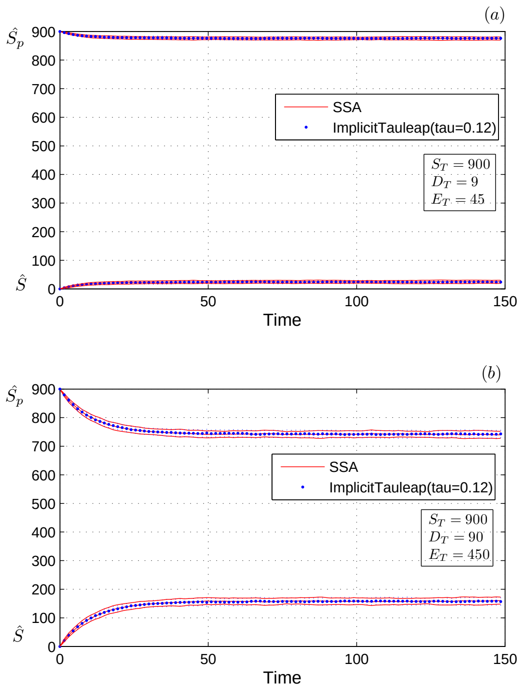

Fig. 3. The SSA simulation (solid lines) and the implicit tau-leaping simulation (dotted lines) for the GK switch. State space (S_T, D_T, E_T) is (a):(900, 9, 45) and (b):(900, 90, 450).

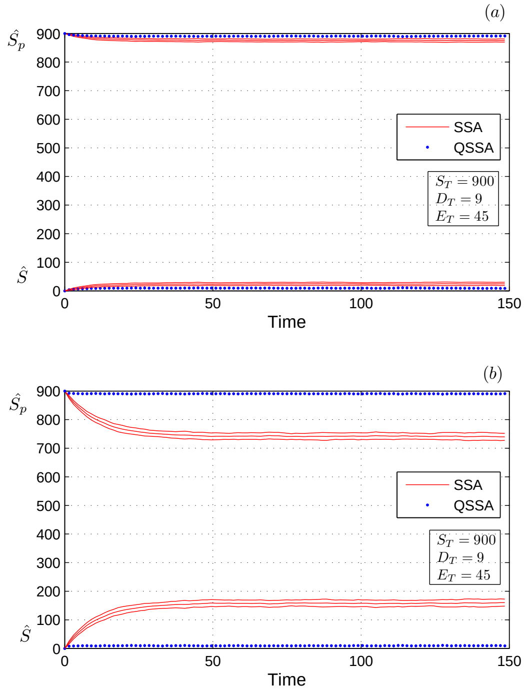

Fig. 4. The SSA simulation (solid lines) and the QSSA simulation (dotted lines) for the GK switch. State space (S_T, D_T, E_T) is (a):(900, 9, 45) and (b):(900, 90, 450).

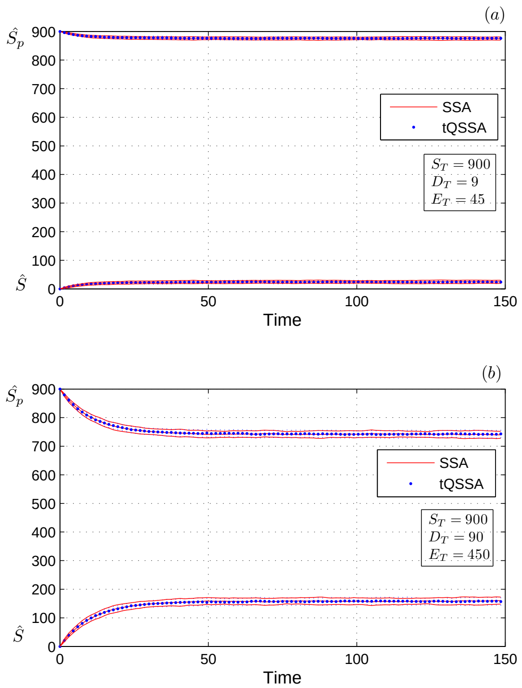

Fig. 5. The SSA simulation (solid lines) and the tQSSA simulation (dotted lines) for the GK switch. State space (S_T, D_T, E_T) is (a):(900, 9, 45) and (b):(900, 90, 450).

Table 2. The number of runs and elapsed CPU time (sec) for the SSA, explicit tau-leaping, implicit tau-leaping, QSSA, and tQSSA algorithms.

Number of runs	1000	5000	10000	50000
SSA	44.57	216.77	434.10	2175.45
Explicit Tau-leaping	4.75	18.23	46.57	181.31
Implicit Tau-leaping	18.43	89.13	181.29	887.44
QSSA	2.63	13.23	26.33	132.47
tQSSA	2.67	13.33	26.79	133.68

cases $(D_T, E_T) = (9, 45)$ and $(D_T, E_T) = (90, 450)$ are compared. $\hat{S}_p$ and $\hat{S}$ are defined by $\hat{S}_p = S_p + D : S_p$ and $\hat{S} = S + E : S$.

Figures 2 and 3 show the mean and mean ± standard deviation trajectories of $\hat{S}_p$ and $\hat{S}$ for the explicit and implicit tau-leaping approximation algorithms and the SSA. When $\hat{S}_p$ reaches steady state, the standard deviations of the SSA are approximately 4.3 (a) and 13.2 (b). In the explicit method, the error parameter value ($\epsilon = 0.15$) was chosen so that the standard deviation matched that of the SSA. The fixed step value ($\tau = 0.12$) was chosen similarly for the implicit method. Table 2 shows that the explicit method is faster than the implicit method for the same accuracy, due to the cost of the Newton iteration in the implicit method.

Figures 4 and 5 show the mean and mean ± one standard deviation trajectories of $\hat{S}_p$ and $\hat{S}$ for the QSSA and tQSSA algorithms. Figure 4(b) shows that the results with the QSSA are totally different from those for the SSA. From the explanation in Section 2.4, the QSSA is that $S_T \gg E_T$. If $S_T \approx E_T$ or $S_T \ll E_T$, then the results from the QSSA algorithm are not reliable. Figure 4(a) shows that if $S_T \gg E_T$, then the results of the QSSA algorithm are similar to those from the SSA. In contrast with Figure 4, Figure 5 shows that the tQSSA algorithm works when $S_T \approx E_T$ or $S_T \ll E_T$.

In terms of CPU time, Table 2 shows that the QSSA and tQSSA algorithms are the fastest approximate algorithms. The QSSA and tQSSA algorithms are almost 20 times faster than the SSA. The explicit and implicit tau-leaping approximations also have improved computational time over the SSA.

4. Conclusions and Future Work

The simulation results reported here, while limited, show important characteristics of each approximation algorithm. The explicit tau-leaping method improves computational efficiency, compared to the SSA, for nonstiff systems, but can be unstable on stiff systems. The implicit tau-leaping method is stable, but much slower than the explicit method. The tQSSA algorithm produces excellent agreement with the SSA and is more efficient by an order of magnitude. All of these approaches are ultimately based on the SSA, and remain impractical for realistic PINs (e.g., the cell cycle). In fairness, for the problems here, solving the CME directly is more efficient than using any SSA variant; the scalability of both these direct CME solution algorithms and these SSA variants is unknown. Future work will analyze and compare different approaches, find new methods to improve the SSA, and contribute to the stochastic simulation software package StochKit.

REFERENCES

[1] A. Goldbeter and D. E. Koshland. "An amplified sensitivity arising from covalent modification in biological systems"; *Proc. Natl. Acad. Sci.*, **78**, pp. 6840–6844, 1981.

[2] E. O. Voit. "Computational analysis of biochemical systems", Cambridge University Press, 2000.

[3] H. H. McAdams and A. Arkin. "Stochastic mechanisms in gene expression"; *Proc. Natl. Acad. Sci.*, **94**, pp. 814–819, 1997.

[4] I. Oppenheim, K. E. Shuler, and G. H. Weiss. "Stochastic processes in chemical physics: the master equation", MIT Press, 1977.

[5] D. T. Gillespie. "Exact Stochastic Simulation of Coupled Chemical Reactions"; *Journal of Physical Chemistry*, **81**, pp. 2340–2361, 1977.

[6] D. T. Gillespie. "A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions"; *Journal of Computational Physics*, **22**, pp. 403–434, 1976.

[7] M. A. Gibson and J. Bruck. "Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels"; *Journal of Physical Chemistry*, **104**, pp. 1876–1889, 2000.

[8] Y. Cao, H. Li and L. R. Petzold. "Efficient Formulation of the Stochastic Simulation Algorithm for Chemically Reacting Systems"; *Journal of Chemical Physics*, **121**, pp. 4059–4067, 2004.

[9] D. T. Gillespie. "Approximate accelerated stochastic simulation of chemically reacting systems"; *Journal of Chemical Physics*, Journal of Chemical Physics. **115**, pp. 1716–1733, 2001

[10] M. Rathinam, L. R. Petzold, Y. Cao and D. T. Gillespie. "Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method"; *Journal of Chemical Physics*, **119**, pp. 12784–12794, 2003.

[11] Y. Cao, D. T. Gillespie and L. R. Petzold. "The slow-scale stochastic simulation algorithm"; *Journal of Chemical Physics*, **122**, pp. 014116, 2005.

[12] C. V. Rao and A. P. Arkin. "Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm"; *Journal of Chemical Physics*, **118**, pp. 4999–5010, 2003.

[13] A. C. Cilibetro, F. Capuani and J. J. Tyson. "Modeling Networks of Coupled Enzymatic Reactions Using the Total Quasi-Steady State Approximation"; *PLoS Computational Biology*, **3**, pp. 463–472, 2007.

[14] D. T. Gillespie and L. R. Petzold. "Improved leap-size selection for accelerated stochastic simulation"; *Journal of Chemical Physics*, **119**, pp. 8229–8234, 2003.

[15] Y. Cao, D. T. Gillespie and L. R. Petzold. "Efficient Stepsize Selection for the Tau-Leaping Simulation"; *Journal of Chemical Physics*, **124**, pp. 144109, 2006.

[16] N. Blüthgen and H. Herzel. "How robust are switches in intracellular signaling cascades?"; *Journal of Theoretical Biology*, **225**, pp. 293–300, 2003.

[17] J. A. M. Borghans, R. J. de Bore and L. A. Segel. "Extending the quasi-steady state approximation by changing variables"; *Bulletin of Mathematical Biology*, **58**, pp. 43–63, 1996.

[18] A. R. Tzafriri and E. R. Edelman. "The total quasi-steady-state approximation is valid for reversible enzyme kinetics"; *Journal of Theoretical Biology*, **226**, pp. 303–313, 2004.