



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#) 

Double Truncation Method for Simulation of Stochastic Chemical Reaction Networks



Soyeong Jeong

**Mathematical Science
Graduate School of UNIST**

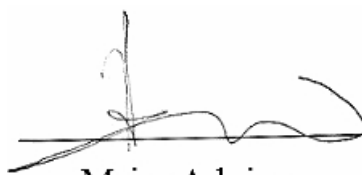
Double Truncation Method for Simulation of Stochastic Chemical Reaction Networks

A thesis
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Master of Science

Soyeong Jeong

05.02.2014

Approved by

A handwritten signature in black ink, appearing to be 'Pilwon Kim', written over a horizontal line.

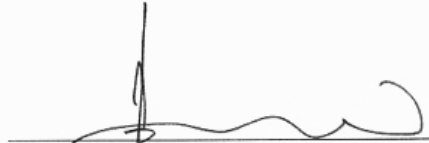
Major Advisor
Pilwon Kim

Double Truncation Method for Simulation of Stochastic Chemical Reaction Networks

Soyeong Jeong

This certifies that the thesis of Soyeong Jeong is approved.

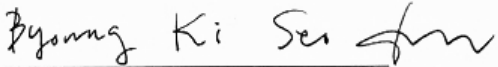
05.02.2014



Thesis Supervisor: Pilwon Kim



Chang Hyeong Lee: Thesis Committee Member #1



Byoung Ki Seo : Thesis Committee Member #2

Abstract

We develop Double Truncation Method(DTM) to understand the stochasticity of chemical reaction network on the mesoscopic scale. The Chemical Master Equation(CME) describes the probability distribution of the system accurately under the markov assumption. But solving CME is computationally heavy. It faces the curse of dimensionality because it considers reactions on every different states. To settle the issue, stochastic approach came out, typically Gillspie's stochastic simulation algorithm(SSA). SSA lifts the curse of dimensionality, but it needs too many realizations, which makes it less practical, in cases of the system consisting of large numbers of molecules or very different time scale reactions.

A recently developed Probability Generating Function(PGF) method supplements those weaknesses. It is a deterministic description and sparks the reactions in stead of considering those for all the states. By doing that, though it expresses the system efficiently, but it implements the symbolic computation and converges still slowly. So here we suggest DTM to speed up.

As suggested from the name, DTM has two truncations for time and for coefficients based on PGF method. We perform the first truncation for a short time and second truncation for small coefficients at each time step. First truncation or superimposition can be performed underlying the power series expansion on time. And next truncation can be conducted with the elimination of relatively small coefficient terms. Since the coefficient of PGF means the probability of a specific state, the sum of coefficient can be understood as an weight for the system. This observation enables us to ignore a great deal of small terms which do not affect the system significantly. The method is procedurally simple and powerful, especially for mesoscopic scale problems. It works well even for open systems, such as brusselator.

We apply the method to simulation of binding reactions, enzyme kinetics, transition model and brusselator and compare the results with those of SSA or matrix exponential.

Contents

List of Figures	vii
List of Tables	viii
I Introduction	1
II Chemical Master Equation	3
2.1 Backgrounds	3
2.2 Chemical Master Equation	5
2.3 Stochastic Simulations	6
2.3.1 Stochastic Simulation Algorithm	6
2.3.2 Tau-leaping Method	7
III Probability Generating Function Approach	8
3.1 Probability Generating Function	8
3.2 PGF-PDE	9
3.3 Probability Generating Function Method	9
IV Double Truncation Method	12
4.1 First Truncation	12
4.2 Second Truncation	13
V Simulation Results	15
5.1 Binding Reaction	15
5.2 Enzyme Kinetics	16
5.3 G_2/M Transition Model	17
5.4 Brusselator Model	18
VI Conclusion	21
References	22

List of Figures

Figure 5-1 The initial condition $\mathbf{n} = [20, 10, 0]$ and parameters $c_1 = 1, c_{11} = 0.1$ are assumed. In the figure of probability, each curve $p(n_1 = i)$ denotes the time-dependent probability solution that $n_1 = i, i = 10, 13, 15$, respectively. The terms which have a coefficient less than 10^{-8} are dropped out.	16
Figure 5-2 The initial condition $\mathbf{n} = [10, 20, 20, 0]$ and parameters $c_1 = 0.1, c_{11} = 1, c_2 = 0.5$ are assumed. In the figure of probability, each curve $p(n_2 = i)$ denotes the time-dependent probability solution that $n_2 = i, i = 5, 10, 17$ respectively. The terms which have a coefficient less than 10^{-7} are dropped out.	17
Figure 5-3 The initial condition $\mathbf{n} = [5, 4, 2, 3, 2, 3, 2, 0, 0, 0]$ and parameters $c_1 = c_4 = c_7 = c_{10} = 0.2, c_2 = c_5 = c_8 = c_{11} = 1, c_3 = c_6 = c_9 = c_{12} = 0.1$ are assumed. In the figure of probability, each curve $p(n_2 = i)$ denotes the time-dependent probability solution that $n_2 = i, i = 1, 3, 5$, respectively. The terms which have a coefficient less than 10^{-8} are dropped out.	19
Figure 5-4 The initial condition $\mathbf{n} = [0, 0]$ and parameters $c_1 = 1, c_2 = 0.0001, c_3 = c_4 = 0.1, a = 5, b = 1$ are assumed. In the figure of probability, each curve $p(n_2 = i)$ denotes the time-dependent probability solution that $n_2 = i, i = 30, 35, 40$, respectively. The terms which have a coefficient less than 10^{-8} are dropped out.	20

List of Tables

I

Introduction

We want to keep track of the probability distribution of the states in a chemical reaction network. The network simulates real world chemical system by formulating the reactant, reaction and product. Deterministic description with the information of molecules' position and velocity requires a heavy computational load in most cases. So one classify the the states according to the numbers of each type of molecules and consider the stochastic dynamics in the discrete state space.[1]

The probability of a state is determined based on the law of mass action, which is an underlying hypothesis from chemical kinetics.[2] It states that the speed of a chemical reaction is proportional to the amount of the reactants. The reaction speed is correspondence with the probability here. For a uni-molecular reaction, it is assumed that the probability is proportional to the number of the reacting substance. Physically, least condition above bimolecular reaction is a collision of reactants. So it is assumed that the probability is proportional to the number of encountering cases.[3]

The probability distribution can be understood as the solution of Chemical Master Equation(CME)[1, 3, 8, 10]. Under the markov assumption, CME is represented by

$$\frac{dp(\mathbf{n}, t)}{dt} = \sum (a_k(\mathbf{n} - V_k)p(\mathbf{n} - V_k, t) - a_k(\mathbf{n})p(\mathbf{n}, t))$$

where $p(\mathbf{n}, t)$ denotes the probability of the state $\mathbf{n} = (n_1, \dots, n_s)$ at time t , $a_k(\mathbf{n})$ denotes the propensity function for k -th reaction, V_k denotes the k -th column of V , and V means the stoichiometric matrix. However it is very difficult to solve CME analytically or numerically because of the curse of dimensionality. Since CME describes every transition among all the different states, its dimension is as high as the number of possible states, which is mostly large. There came out some methods lifting the curse of dimensionality, such as Gillespie's Stochastic Simulation Algorithm(SSA) or tau leaping method[3, 4]. Those methods are based on Monte-carlo type simulation. SSA is exact but time-consuming. It needs too many realizations in case

of a system entangled with the fast and slow reactions because the two independent random variables do not efficiently reflect the relation between reaction and time. To speed up, in tau leaping method, the time interval τ and the propensity function are fixed. The method has one random variable to determine the number of reactions following the Poisson distribution for $[t, t + \tau)$. It is faster than SSA, but still not enough. Because leap condition is limited to the fastest reaction, one cannot perform it as fast as one wishes.

On the other hand, we focus on the mesoscopic scale systems because huge system is relatively easy to predict[3]. In a macroscopic view, deterministic Reaction Rate Equation(RRE) approximates to CME. Because the time to the next reaction is short, the discrete stochastic process becomes continuous. This enables us to describe the system as a set of coupled ODEs with a few variables and get the results easily. So we consider mesoscopic scale system. Even for the mesoscopic system, it is not easy to get the results with the introduced conventional methods. Recently developed Probability Generating Function(PGF) method[6, 7, 9] can make up for the weak points. The method is based on a deterministic approach and triggers reactions at every infinitesimal time step without considering all transitions on every different state unlike CME. Though it entails symbolic computations, PGF expresses n state vectors with n terms, whereas CME needs $n \times n$ terms. In addition, PGF is procedurally simple and gives us an intuitive point of view on the system. Introducing PGF, one can convert CME into a partial differential equation(PDE), say, PGF-PDE. Otherwise, PGF-PDE can be derived from the chemical equation easily. Because of the lack of the boundary conditions, we cannot use the conventional schemes like the finite difference method. In stead, we find a semi-analytic solution using power series expansion. Since it converges too slowly, we suggest the Double Truncation Method(DTM).

DTM has two truncations based on the physical meaning of PGF for time and small coefficients. The first truncation can be performed based on power series expansion. The truncation order means the maximum number of reactions during the given time interval. For the second truncation, we remove the small coefficient terms because those imply rare events. Since the coefficient means the probability of the state from the definition of PGF, the sum of small coefficient can be seen as an weight for the system. Under the markov assumption, the reactions are independent, so it does not affect the system significantly when it is very small.

An outline is as follows; Section 2 gives a mathematical background of a CME. In section 3, the details of PGF method and derivation of PGF-PDE are written. In section 4, we present the DTM here. In section 5, we show the simulation results. We apply the method to simulation of binding reactions, enzyme kinetics, G_2/M transition, and brusselator and compare the results with those of SSA or matrix exponential. Section 6 is a conclusion.

II

Chemical Master Equation

Chemical master equation(CME) is the governing equation for probability distribution of the states in a chemical reaction system. With the distribution, the interesting stochastic figures can be easily computed including mean and variance. There are some assumptions for CME.[1, 8, 10]

2.1 Backgrounds

It is assumed that the molecules are uniformly distributed at a fixed temperature in a control volume. There are A different types of molecules z_1, \dots, z_A , and these chemical species may participate in B different types of reactions R_1, \dots, R_B .

The standard of the classification for the states is the number of chemical species. Each element means the number of each type of molecules. A state vector is denoted by \mathbf{n} , where $\mathbf{n} = (n_1, \dots, n_A)$. The i -th element n_i is the number of z_i . With the spatial information such as all molecules' position and velocity, one can take the deterministic description. But in most cases, a deterministic approach is computationally heavy and very expensive. To simplify the situation, we classify the states and follow their probability. This stochastic description gives us a valid result as long as the system is spatially homogeneous.

To describe the state-change, we introduce a stoichiometric matrix V . It considers relative quantities of the state vectors after reactions. The matrix is defined as a column-wise sense. After k -th reaction, the state \mathbf{n} changes into $\mathbf{n} + V_k$. (j, k) th element of V means the change of the number of z_j after k -th reaction. Sign convention is positive for products, and negative for reactants.

We describe the chemical reaction system under the markov assumption. A stochastic process N is a collection of random variables $\{N(t), t \in T\}$. $N(t)$ denotes the state space and $N = \{\mathbf{n}_0, \mathbf{n}_1, \dots, \mathbf{n}\}$. T is the (time) index set of the stochastic process and $T = \{t_0, t_1, \dots, t\}$. If the conditional probability distribution of future states is determined depending on the current

state, that is,

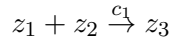
$$p(N(t) = \mathbf{n}_i | N(t_{i-1}) = \mathbf{n}_{i-1}, \dots, N(t_0) = \mathbf{n}_0) = p(N(t) = \mathbf{n}_i | N(t_{i-1}) = \mathbf{n}_{i-1})$$

for all $t \in T$ and all $\mathbf{n} \in N$, then we say the stochastic process is under the markov assumption.

We denote $p(N(t) = \mathbf{n}_i)$ as the probability that the state is \mathbf{n}_i at time t .

Propensity function, denoted by $a_j(\mathbf{n})$ for a state \mathbf{n} , determines the probability in the system. The probability of reaction j taking place in the infinitesimal time dt is given by $a_j(\mathbf{n})dt$. In 1864, Peter Waage and Cato Guldberg discovered the law of mass kinetics that the speed of a chemical reaction is depending on the numbers of the reactants. The law is the basis of chemical kinetics and gives us a strong physical assumption. The reaction speed corresponds to the probability for a reaction. So the propensity function for uni-molecular reaction is proportional to the number of reactant. For the above bimolecular reaction is proportional to the number of cases of a collision. Because two molecules react when they encounter. The reaction constant is a inherent value and means that every encounter does not come to a reaction.

Let us take an example.



The propensity function for this reaction is given by

$$a_1(\mathbf{n}) = c_1 n_1 n_2$$

where c_1 is a reaction constant.

Propensity function for reaction k is denoted $a_k(\mathbf{n})$ for a state \mathbf{n} a time t . Then the probability is given by

$$P(\mathbf{n} | R_k) = a_k(\mathbf{n} - V_k)dt, \quad 0 \leq k \leq B.$$

where $P(\mathbf{n} | R_k)$ means the probability that n occurs, when R_k happens. By the law of total probability

$$P(\mathbf{n}) = \sum_{j=0}^B P(\mathbf{n} | R_k)P(R_k),$$

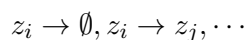
then we find that

$$\frac{p(x, t + dt) - p(x, t)}{dt} = \sum_{j=1}^B (a_k(\mathbf{n} - V_k)p(\mathbf{n} - V_k, t) - a_k(\mathbf{n})p(\mathbf{n}, t)).$$

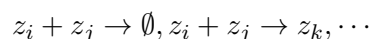
By letting dt approach to 0, we derive the CME as following.

$$\frac{dp(\mathbf{n}, t)}{dt} = \sum_{j=1}^B (a_k(\mathbf{n} - V_k)p(\mathbf{n} - V_k, t) - a_k(\mathbf{n})p(\mathbf{n}, t)).$$

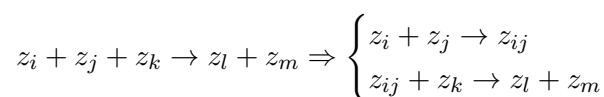
All types of reaction can be separated by a series of elementary reactions. Elementary reaction is an unimolecular or bimolecular reaction. Unimolecular.



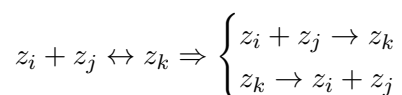
Bimolecular.



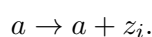
For example, let us consider the trimolecular reaction.



Let us consider the reversible reaction.



If there exists a reaction with catalyst a , the catalyst can be the reactant and the product at the same time like following.



If there is no molecule, the reaction never happens. Under this observation, we focus on the elementary reaction.

2.2 Chemical Master Equation

Under the markov assumption, the Chemical Master Equation(CME) is a linear coupled system of ODEs as follows.

$$\frac{dp(\mathbf{n}, t)}{dt} = \sum (a_k(\mathbf{n} - V_k)p(\mathbf{n} - V_k, t) - a_k(\mathbf{n})p(\mathbf{n}, t))$$

CME can be simply expressed as

$$\frac{dp(\mathbf{n}, t)}{dt} = Kp(\mathbf{n}, t).$$

For closed reaction systems, the probability function $p(\mathbf{n}, t)$ has an analytic solution,

$$p(\mathbf{n}, t) = e^{Kt}p(0).$$

But in most cases, though it is mesoscopic scale, CME is very difficult to solve because of the curse of dimensionality. To avoid that, there have been some attempts based on the stochastic simulations.

2.3 Stochastic Simulations

2.3.1 Stochastic Simulation Algorithm

Gillespie's Stochastic Simulation Algorithm(SSA) is not suffering from the high dimensionality, but still it has some problems. To explain SSA, we need two different probability quantities, which denote $P_0(\tau | x, t)$, and $p(\tau, j | x, t)$ where x denotes state and t time.[3, 4]

First, $P_0(\tau | x, t)$ is the probability that no reaction occurs over $[t, t + \tau)$. So from the definition of propensity function,

$$P_0(\tau + \delta\tau | x, t) = P_0(\tau | x, t) \left(1 - \sum_{k=1}^B a_k(x) d\tau\right).$$

Then,

$$\frac{P_0(\tau + \delta\tau | x, t) - P_0(\tau | x, t)}{d\tau} = -a_{sum}(x)P_0(\tau, x, t)$$

where $a_{sum}(x) := \sum_{k=1}^B a_k(x)$. So we can solve this ODE, and its solution is

$$P_0(\tau, x, t) = e^{-a_{sum}(x)\tau}.$$

Second, $p(\tau, j | x, t)$ is the joint probability that the next reaction will be the j th reaction and will occur in the time interval $[t + \tau, t + \tau + d\tau)$. Then using the definitions of P_0 and propensity function,

$$p(\tau, j | x, t) d\tau = P_0(\tau | x, t) a_j(x) d\tau.$$

So if we substitute it with above solution, we get

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

It can be rewritten as

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

The joint probability density function $p(\tau, j | x, t)$ is the product of two individual density functions. The first term $a_j(x)/a_{sum}(x)$ relates to the next reaction index. And the second term $a_{sum}(x)e^{-a_{sum}(x)\tau}$ corresponds to the time until next reaction. It suggests that the time random variable has an continuous exponential distribution. So from two uniformly distributed

random numbers, we take

$$\tau = \frac{1}{a_0(x)} \ln\left(\frac{1}{r_1}\right)$$

$$j = \text{the smallest integer satisfying } \sum_{j'=1}^j a_{j'}(x) > r_2 a_0(x)$$

. The algorithm is as follows. Evaluate every propensity function for a state $\mathbf{x}(t)$ and their sum. Generate two random numbers to determine the next reaction j and the time to the next reaction τ . Finally, update the state vectors along with j th reaction, and t to $t + \tau$. Iterate this from the beginning. Though SSA is exact, but in cases of the system consisting of very different time scale reactions or large numbers of molecules, it converges too slowly. To speed up, tau-leaping method came out.

2.3.2 Tau-leaping Method

Tau-leaping method improved the speed comparing with SSA but still not fast enough. If the system has many molecules, the sum of propensity function is large. So the reaction time τ rarely affects the system. So we can set the leap condition τ satisfying propensity function has no big change for $[t, t + \tau)$. The number of reactions follows the Poisson distribution ($\mathbb{P}(P = i) = e^{-\lambda} \frac{\lambda^i}{i!}, i = 0, 1, 2, \dots$).

The algorithm is following. From a Poisson random variable $\mathbf{p}_j(m_j, \tau)$ with mean and variance $m_j \tau$, we determine the number of reaction for time $[t, t + \tau)$. Then the state vector is changed as

$$\mathbf{x}(t + \tau) = \mathbf{x}(t) + \sum_{j=1}^B \nu_j \mathbf{p}_j(a_j(\mathbf{x}(t)), \tau).$$

The time is updated t to $t + \tau$. Iterate from the beginning.

The method is approximate so it is sometimes much slower than SSA because of the assumption. Leap condition τ is fitted to the fastest reaction and large number of molecules is needed. So for mesoscopic scale systems, it is often unsuitable.

III

Probability Generating Function Approach

Probability Generating Function(PGF) method gains the upper hand for a high-dimensional computation comparing with CME. PGF is structurally simple because the probability distribution is nothing but the coefficients of the terms. Therefore it is easy to deal with. But it needs the symbolic computations. So it converges too slowly.

3.1 Probability Generating Function

Probability Generating Function(PGF) is an efficient function to describe the probability distribution for a discrete state space.[6] For $\mathbf{z} = (z_1, \dots, z_A)$ and $\mathbf{n} = (n_1, \dots, n_A)$, let us denote

$$\mathbf{z}^{\mathbf{n}} = z_1^{n_1} z_2^{n_2} \dots z_A^{n_A}.$$

It means that the number of z_i is n_i respectively. We call \mathbf{n} as a state vector. Then a PGF is defined as

$$G(\mathbf{z}, t) = \sum_{\mathbf{n}=\mathbf{0}}^{\infty} \mathbf{z}^{\mathbf{n}} p(\mathbf{n}, t)$$

where $z_i \in [-1, 1]$. $p(\mathbf{n}, t)$ is the probability that the state being \mathbf{n} at time t .

For convenience,

$$G_{i_1, \dots, i_k} = \frac{\partial}{\partial z_{i_k}} \dots \frac{\partial}{\partial z_{i_1}} G.$$

From a PGF, we can get mean, variance, probability as follows.

$$M_i(t) = G_i(\mathbf{z} = 1, t)$$

$$V_i(t) = G_{ii}(\mathbf{z} = 1, t) + G_i(\mathbf{z} = 1, t) - [G_i(\mathbf{z} = 1, t)]^2$$

$$P_i(k, t) = \frac{1}{k!} \frac{\partial^k G(\mathbf{z}, t)}{\partial z_i^k} \Big|_{z_i=0, z_j=1, j \neq i}$$

where $P_i(k, t)$ denotes the marginal probability of i at time t .

3.2 PGF-PDE

Let $G(\mathbf{z}, t)$ solve the partial differential equation.

$$G_t = F(z_1, \dots, z_A, G, G_i, \dots)$$

On the left hand side, there is the derivative of G with respect to time. Because we want to see the progression of probability of molecular state vectors with time. On the right hand side, there is a function holding a mechanism describing the production and the annihilation of molecules and change of the probability. In order to find the probability of each state vector, we recall the propensity function. By differentiation with respect to the reactants, we can derive the number of reactants and make the propensity function. With differentiation and multiplication, we also control the state vector on the same principle of the stoichiometric matrix. Let us take some examples.

Unimolecular reaction.

$$z_1 \xrightarrow{c_1} z_2, \quad a_1(\mathbf{n}) = c_1 n_1 \longrightarrow F = c_1 G_1(z_2 - z_1)$$

Bimolecular reaction.

$$z_1 + z_2 \xrightarrow{c_2} z_3, \quad a_2(\mathbf{n}) = c_2 n_1 n_2 \longrightarrow F = c_2 G_{12}(z_3 - z_1 z_2)$$

Dimerization reaction.

$$z_1 + z_1 \xrightarrow{c_3} z_2, \quad a_3(\mathbf{n}) = \frac{c_3}{2} n_1 (n_1 - 1) \longrightarrow F = \frac{c_3}{2} G_{11}(z_2 - z_1^2)$$

Other types of reaction such as trimolecular, or reversible reaction can be described by a series of elemental reactions as we wrote in section 2.1.

3.3 Probability Generating Function Method

We put the initial condition as

$$G(\mathbf{z}, t = 0) = \mathbf{z}_0^{\mathbf{n}}$$

3.3 Probability Generating Function Method

where \mathbf{n}_0 denotes the numbers of each type of molecules at $t = 0$. From physical observation, there exists only one state at the beginning.[6, 7, 9]

The boundary conditions are

$$G(\mathbf{z} = \mathbf{1}, t) = 1 \text{ and } G(\mathbf{z} = \mathbf{0}, t) = 0.$$

This boundary condition is trivial and does not give us any clue to solve the given PDE. That makes it more difficult to find an analytic solution. By taking a semi-analytic approach based on the power series and Pade approximation, we can deal with it.

From the definition of PGF and the fact that all the events are independent, we can compute the probability depending on the time. So PGF is described by

$$\begin{aligned} G(\mathbf{z}, t) &= \sum_{\mathbf{n}=\mathbf{0}}^{\infty} p_{\mathbf{n}}(t) \mathbf{z}^{\mathbf{n}} \\ &= \sum_{n=0}^{\infty} f_n(\mathbf{z}) t^n \end{aligned}$$

where $f_n, n = 0, 1, \dots$, are polynomials of \mathbf{z} . By putting the initial condition $f_0(\mathbf{z}) = \mathbf{z}^{\mathbf{n}_0}$ into the PGF-PDE, we can derive f_n recursively. Since those are all polynomials and computed explicitly, they give us many computational benefits. However, it converges too slowly.

To make it faster, Pade approximation is an effective way. For the formal form of power series, Pade approximation is

$$\sum_{k=0}^{\infty} c_k t^k = \frac{\sum_{k=0}^L a_k t^k}{\sum_{k=0}^M b_k t^k} + O(t^{L+M+1}).$$

That is,

$$\left(\sum_{k=0}^{\infty} c_k t^k \right) \left(\sum_{k=0}^M b_k t^k \right) = \sum_{k=0}^L a_k t^k + O(t^{L+M+1}).$$

By the method of undetermined coefficients and setting $b_0 = 1$, we find

$$\begin{aligned} a_0 &= c_0 \\ a_1 &= c_1 + c_0 b_1 \\ a_2 &= c_2 + b_1 c_1 + b_2 c_0 \\ &\vdots \\ a_L &= c_L + \sum_{i=1}^{\min(L,M)} b_i c_{L-i} \end{aligned}$$

Also, for t^{L+1}, \dots, t^{L+M} , we find a system

$$\begin{bmatrix} c_{L-M+1} & c_{L-M+2} & \cdots & c_L \\ c_{L-M+2} & c_{L-M+3} & \cdots & c_{L+1} \\ \vdots & \vdots & \vdots & \vdots \\ c_L & c_{L+1} & \cdots & c_{L+M-1} \end{bmatrix} \begin{bmatrix} b_M \\ b_{M-1} \\ \vdots \\ b_1 \end{bmatrix} = - \begin{bmatrix} c_{L+1} \\ c_{L+2} \\ \vdots \\ c_{L+M} \end{bmatrix}$$

where $c_k = 0$ for negative k . Thus, we can find the coefficient of Pade approximation. But it takes a time to find Pade approximant and the error cannot be predicted rigorously.

IV

Double Truncation Method

We suggest Double Truncation Method(DTM) for an efficient computation by taking advantages of PGF method. There are two truncations for time and small coefficients based on the physical observation. The first truncation for order N during time dt means the maximum number of reactions is N for the time. The second truncation for small coefficients means the elimination of rare events in the system. The details are following.

4.1 First Truncation

We implement the first truncation on time. We find $f_1(\mathbf{z}), f_2(\mathbf{z}), \dots, f_n(\mathbf{z})$ iteratively by putting this into PGF-PDE. By doing that, we find $G(\mathbf{z}, t)$. Truncation on time makes the equation shorter and it reduces the computation load a lot. By Taylor series, we expand $G(\mathbf{z}, t)$ on time as

$$G(\mathbf{z}, t) = \sum_{n=0}^{\infty} f_n(\mathbf{z})t^n.$$

With first few terms, it works well as long as the time step is infinitesimally small. So we solve PGF-PDE by successive superimposition of truncated Taylor expansion.

$$G^1(\mathbf{z}, t) = \sum_{n=0}^N t^n f_n^1(\mathbf{z}, t), \quad G^1(\mathbf{z}, 0) = \mathbf{z}^{\mathbf{n}}$$

$$G^2(\mathbf{z}, t) = \sum_{n=0}^N t^n f_n^2(\mathbf{z}, t), \quad G^2(\mathbf{z}, h) = G^1(\mathbf{z}, h)$$

$$G^3(\mathbf{z}, t) = \sum_{n=0}^N t^n f_n^3(\mathbf{z}, t), \quad G^3(\mathbf{z}, 2h) = G^2(\mathbf{z}, 2h)$$

⋮

The time interval h and the order N represent that the maximum number of reactions for $[t, t + h)$ is N in a chemical system.

After r superimpositions, for time $t \in [(r - 1)h, rh]$,

$$G^r(\mathbf{z}, t) = \sum_{\mathbf{n}} \mathbf{z}^{\mathbf{n}} p^r(\mathbf{n}, t)$$

where $p^r(\mathbf{n}, t)$ means the probability for the state being \mathbf{n} at time t .

4.2 Second Truncation

We implement the second truncation on small coefficients. We define the carry-over functional F mapping $G^r(\mathbf{z}, t)$ into

$$\sum_{\mathbf{n}} \mathbf{z}^{\mathbf{n}} I_{\epsilon}(p^r(\mathbf{n}, t))$$

where I_{ϵ} is a function defined as

$$I_{\epsilon}(x) = \begin{cases} x & \text{if } x > \epsilon \\ 0 & \text{if } x \leq \epsilon. \end{cases}$$

Then,

$$G^r(\mathbf{z}, t) \leq F(G^r(\mathbf{z}, t)) + \epsilon m_{\epsilon, r}$$

where $m_{\epsilon, r}$ is the number of eliminated terms depending on ϵ and r . It is obvious from the construction that $m_{\epsilon, r}$ approaches zero as ϵ goes to zero. Since the coefficient means the probability of the state at time t , the second truncation guarantees the consistency for the state probabilities. It is physically hard to occur the rare events. We see the probability as an weight for the system. Since all the events are independent, the probability of a path-dependent state becomes smaller. In fact, the probability distribution follows a multinomial distribution.[5] So the sum of eliminated probabilities never exceeds itself over time. Hence, we ignore them for efficiency.

We can find an uniform error bound on time for the mean and variance as long as the systems being dealt with are closed because the number of each type of molecules is bounded. Even in cases of open ones, one can derive an error bound for the mean and variance as follows.

$$Error_m(t) = |g_i(\mathbf{z} = 1, t)| \leq m_{\epsilon} \epsilon \max(n_i)$$

$$Error_v(t) = |g_{ii}(\mathbf{z} = 1, t) + g_i(\mathbf{z} = 1, t) - [g_i(\mathbf{z} = 1, t)]^2| \leq m_{\epsilon} \epsilon (2 \max(n_i)^2 + \max(n_i))$$

where $g = G(\mathbf{z}, t) - F(G(\mathbf{z}, t))$ and n_i is the i -th element of n , which belongs to the possible state space. For a given ϵ , $m_{\epsilon} = \sum_{i=0}^r m_{\epsilon, i}$, where r is the number of superimpositions. So

we can find the error bound as long as we know the maximum number of interesting molecules after k reactions.

First, we define the reaction vector $\mathbf{R} = (R_1, \dots, R_B)$, s -th element denotes the number of s -th reaction of k reactions. All the elements in \mathbf{n}_k are nonnegative where \mathbf{n}_k denotes a state vector after k reactions. Then

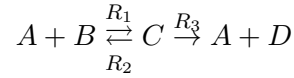
$$\mathbf{n}_k = \mathbf{n}_0 + V \cdot \mathbf{R}$$

where \mathbf{n}_0 is the initial condition and V is a stoichiometric matrix. We find the maximum number of elements for all \mathbf{R} satisfying that

$$\sum_{s=1}^B R_s \leq k.$$

In this way we determine the maximum value of \mathbf{n} .

Let us take an example.



There are four kinds of molecules A,B,C,D and three reactions R_1, R_2, R_3 . And $\mathbf{n}_0 = (10, 10, 20, 5)$.

Then

$$\begin{pmatrix} 10 \\ 10 \\ 20 \\ 5 \end{pmatrix} + \begin{pmatrix} -1 & 1 & 1 \\ -1 & 1 & 0 \\ 1 & -1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix} = \begin{pmatrix} n_{k,1} \\ n_{k,2} \\ n_{k,3} \\ n_{k,4} \end{pmatrix}$$

We want to find the maximum number of the first element $n_{k,1} = 10 - R_1 + R_2 + R_3$, where $n_{k,2}, n_{k,3}, n_{k,4} \geq 0$. The conditions are following.

$$\begin{cases} n_{k,2} = 10 - R_1 + R_2 \geq 0 \\ n_{k,3} = 20 + R_1 - R_2 - R_3 \geq 0 \\ n_{k,4} = 5 + R_3 \geq 0 \end{cases}$$

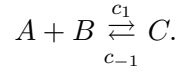
Then $20 \geq -R_1 + R_2 + R_3$. So the maximum number of A, $\max(n_{k,1})$ is 30 after k reactions. In the same way, we can find $\max(n_{k,i})$ for $i = 2, 3, 4$. For a given ϵ , we get the m_ϵ computationally. Therefore we may reasonably conclude the error bound for the second truncation as above as long as the first truncation does not have a decisive effect on the system.

V

Simulation Results

5.1 Binding Reaction

We consider a binding/unbinding reaction consisting of two elemental reactions.



The state vector is $\mathbf{n} = (n_1, n_2, n_3)$ and n_1, n_2, n_3 denotes the numbers of molecules of species A,B,C respectively.

The stoichiometric matrix is

$$V = \begin{pmatrix} -1 & 1 \\ -1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Then CME is

$$\frac{dp(\mathbf{n}, t)}{dt} = c_1(n_1 + 1)(n_2 + 1)p(\mathbf{n} + V_1, t) + c_{-1}(n_3 + 1)p(\mathbf{n} + V_2, t) - (c_1 n_1 n_2 + c_{-1} n_3)p(\mathbf{n}, t)$$

where V_k means the k-th column of V.

The following PDE can be derived from the reaction formula simply as

$$G_t = c_1(z_3 - z_1 z_2)G_{12} + c_{-1}(z_1 z_2 - z_3)G_3.$$

We find the initial condition and boundary condition as following.

$$\text{Initial condition : } G(\mathbf{z}, t = 0) = z_1^{n_1} z_2^{n_2} z_3^{n_3}$$

$$\text{Boundary condition : } G(\mathbf{z} = 1, t) = 1 \text{ and } G(\mathbf{z} = 0, t) = 0.$$

The derived PDE with boundary conditions is converted to the CME. We can easily check by using the method of undetermined coefficients.

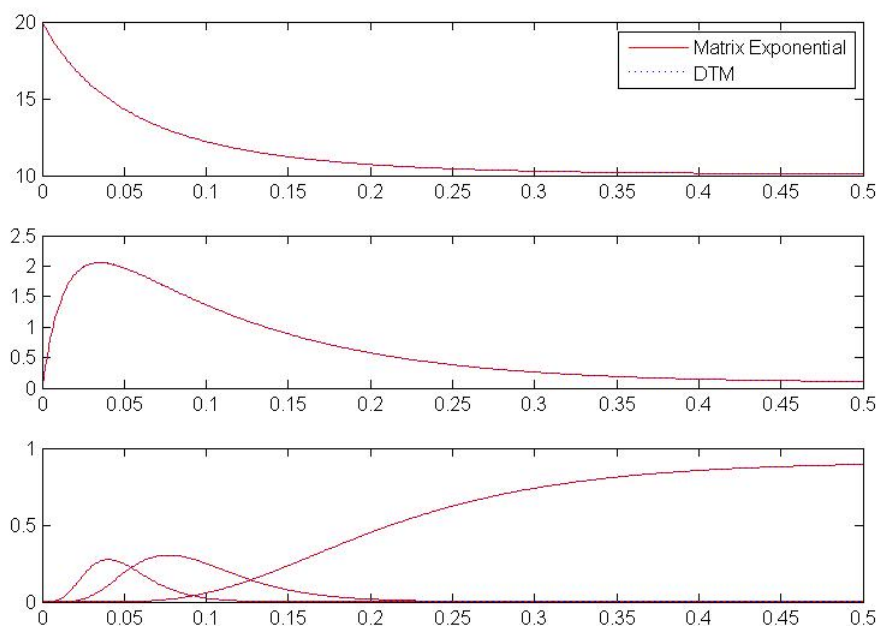
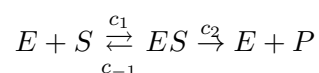


Figure 5-1: The initial condition $\mathbf{n} = [20, 10, 0]$ and parameters $c_1 = 1, c_{11} = 0.1$ are assumed. In the figure of probability, each curve $p(n_1 = i)$ denotes the time-dependent probability solution that $n_1 = i, i = 10, 13, 15$, respectively. The terms which have a coefficient less than 10^{-8} are dropped out.

5.2 Enzyme Kinetics

The enzyme kinetics is one of the most important biochemical reactions. We consider an enzyme-substrate system.



where E,S,ES,P denotes enzyme, substrate, enzyme-substrate complex and product, respectively. The state vector $\mathbf{n} = (n_1, n_2, n_3, n_4)$ lies in same order.

The stoichiometric matrix is

$$V = \begin{pmatrix} -1 & 1 & 1 \\ -1 & 1 & 0 \\ 1 & -1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

Then CME is

$$\begin{aligned} \frac{dp(\mathbf{n}, t)}{dt} &= c_1(n_1 + 1)(n_2 + 1)p(\mathbf{n} + V_1, t) + c_{-1}(n_3 + 1)p(\mathbf{n} + V_2, t) \\ &\quad + c_2(n_3 + 1)p(\mathbf{n} + V_3, t) - (c_1 n_1 n_2 + c_{-1} n_3 + c_2 n_3)p(\mathbf{n}, t) \end{aligned}$$

where V_k means the k -th column of V .

The following PDE can be derived from the reaction formula simply as

$$G_t = c_1(z_3 - z_1z_2)G_{12} + c_{-1}(z_1z_2 - z_3)G_3 + c_2(z_1z_4 - z_3)G_3$$

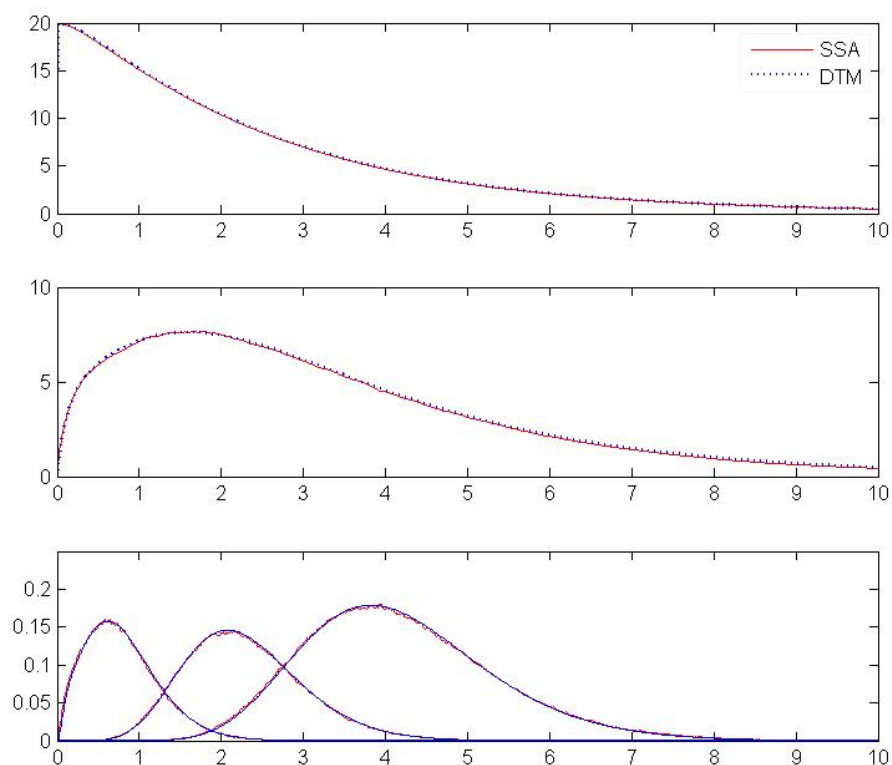
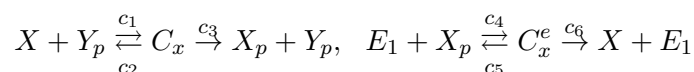
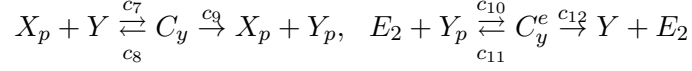


Figure 5-2: The initial condition $\mathbf{n} = [10, 20, 20, 0]$ and parameters $c_1 = 0.1, c_{11} = 1, c_2 = 0.5$ are assumed. In the figure of probability, each curve $p(n_2 = i)$ denotes the time-dependent probability solution that $n_2 = i, i = 5, 10, 17$ respectively. The terms which have a coefficient less than 10^{-7} are dropped out.

5.3 G_2/M Transition Model

G_2/M transition model describes the cell cycle in all the eukaryotic organisms. Understanding the regulators of the G_2 -to-mitosis phase transition is very important. Because malfunctioning of regulators leads to a chromosomal mutation, such as cancer. The process of G_2/M network is given by the reaction scheme





The state vector $\mathbf{n} = (n_1, n_2, \dots, n_9, n_{10})$ lies in order that $X_p, Y_p, X, Y, E_1, E_2, C_x, C_x^e, C_y, C_y^e$.

The stoichiometric matrix is

$$V = \begin{pmatrix} 0 & 0 & 1 & -1 & 1 & 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 \end{pmatrix}$$

Then CME is

$$\begin{aligned} \frac{dp(\mathbf{n}, t)}{dt} = & c_1(n_2 + 1)(n_3 + 1)p(\mathbf{n} + V_1, t) + c_2(n_7 + 1)p(\mathbf{n} + V_2, t) \\ & + c_3(n_7 + 1)p(\mathbf{n} + V_3, t) + c_4(n_1 + 1)(n_5 + 1)p(\mathbf{n} + V_4, t) \\ & + c_5(n_8 + 1)p(\mathbf{n} + V_5, t) + c_6(n_8 + 1)p(\mathbf{n} + V_6, t) \\ & + c_7(n_1 + 1)(n_4 + 1)p(\mathbf{n} + V_7, t) + c_8(n_9 + 1)p(\mathbf{n} + V_8, t) \\ & + c_9(n_9 + 1)p(\mathbf{n} + V_9, t) + c_{10}(n_2 + 1)(n_6 + 1)p(\mathbf{n} + V_{10}, t) \\ & + c_{11}(n_{10} + 1)p(\mathbf{n} + V_{11}, t) + c_{12}(n_{10} + 1)p(\mathbf{n} + V_{12}, t) \\ & - [c_1 n_2 n_3 + (c_2 + c_3)n_7 + c_4 n_1 n_5 + (c_5 + c_6)n_8 \\ & + c_7 n_1 n_4 + (c_8 + c_9)n_9 + c_{10} n_2 n_6 + (c_{11} + c_{12})n_{10}]p(\mathbf{n}, t) \end{aligned}$$

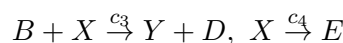
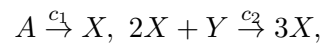
where V_k means the k-th column of V .

The following PDE can be derived from the reaction formula simply as

$$\begin{aligned} G_t = & c_1(z_7 - z_2 z_3)G_{23} + c_2(z_2 z_3 - z_7)G_7 + c_3(z_1 z_2 - z_7)G_7 + c_4(z_8 - z_1 z_5)G_{15} \\ & + c_5(z_1 z_5 - z_8)G_8 + c_6(z_3 z_5 - z_8)G_8 + c_7(z_7 - z_2 z_3)G_{23} + c_8(z_2 z_3 - z_7)G_7 \\ & + c_9(z_1 z_2 - z_7)G_7 + c_{10}(z_8 - z_1 z_5)G_{15} + c_{11}(z_1 z_5 - z_8)G_8 + c_{12}(z_4 z_6 - z_{10})G_{10} \end{aligned}$$

5.4 Brusselator Model

We consider the open model, Brusselator.



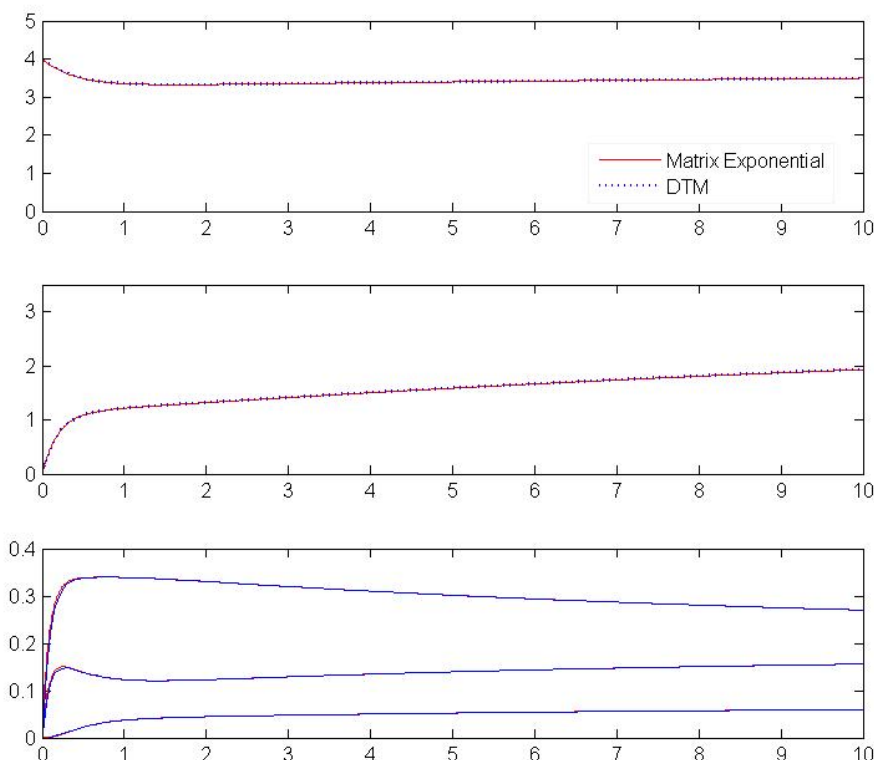


Figure 5-3: The initial condition $\mathbf{n} = [5, 4, 2, 3, 2, 3, 2, 0, 0, 0]$ and parameters $c_1 = c_4 = c_7 = c_{10} = 0.2, c_2 = c_5 = c_8 = c_{11} = 1, c_3 = c_6 = c_9 = c_{12} = 0.1$ are assumed. In the figure of probability, each curve $p(n_2 = i)$ denotes the time-dependent probability solution that $n_2 = i, i = 1, 3, 5$, respectively. The terms which have a coefficient less than 10^{-8} are dropped out.

Brusselator model system is hard to handle, because it is a type of autocatalytic reaction. Hence it has infinitely many reachable states. It can be understood as an infinite dimensional ODE system. A,B are catalysts and D,E are products, which does not affect the stochastic dynamics. So we see two variables X,Y. The state vector $\mathbf{n} = (n_1, n_2)$ lies in order X and Y.

The stoichiometric matrix is

$$V = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 0 & -1 & 1 & 0 \end{pmatrix}$$

where V_k means the k-th column of V.

The following PDE can be derived simply as

$$G_t = c_1 a(z_1 - 1)G + \frac{c_2}{2} z_1^2 (z_1 - z_2)G_{112} + c_3 b(z_2 - z_1)G_1 + c_4 (1 - z_1)G_1$$

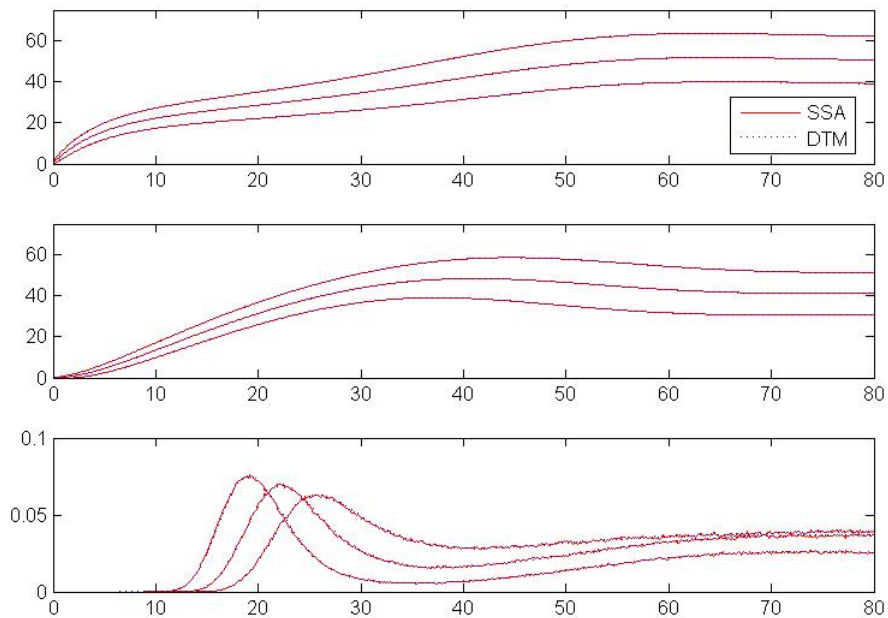


Figure 5-4: The initial condition $\mathbf{n} = [0, 0]$ and parameters $c_1 = 1, c_2 = 0.0001, c_3 = c_4 = 0.1, a = 5, b = 1$ are assumed. In the figure of probability, each curve $p(n_2 = i)$ denotes the time-dependent probability solution that $n_2 = i, i = 30, 35, 40$, respectively. The terms which have a coefficient less than 10^{-8} are dropped out.

VI

Conclusion

In this work, we suggest an efficient method to simulate the chemical reaction network using Probability Generating Function(PGF) method. From conventional methods, such as Chemical Master Equation(CME), Stochastic Simulation Algorithms(SSA) to recently developed PGF method, those methods are less practical in some cases. CME is hard to handle because it suffers from the curse of dimensionality. Though SSA settles the dimensional problem, it is not suitable for the stiff system and too slow. Whereas PGF method alleviates the dimensional problem but it still suffers from it.

So we suggest Double Truncation Method(DTM). With two truncations for time and for small coefficients, we can get the probability distribution much faster. Moreover, DTM is numerically simple and physically reasonable. We also clarify the derivation of PGF-PDE from the reaction formula easily.

We expect that it would be one of important future works to find the optimized error bound for every specific reaction, for the mean and the variance. As of now, an error measurement for even conventional methods is an intricate problem, not in a heuristic approach. Unlike other methods, DTM has some advantages because the sum of eliminated coefficients can be expressed as an order of a threshold ϵ . So more sophisticated analysis for the error measurement would be a good future work.

References

- [1] Gillespie, D. T. (1992). *A rigorous derivation of the chemical master equation*. Physica A: Statistical Mechanics and its Applications 188(1-3): 404-425. [1](#), [3](#)
- [2] E.W. Lund (1965). *Guldberg and Waage and the Law of Mass Action*. Journal of Chemical Education 42: 548-550. [1](#)
- [3] Higham, D. J. (2008). *Modeling and simulating chemical reactions*. SIAM Review 50(2): 347-368. [1](#), [2](#), [6](#)
- [4] Gillespie, D. T. (1977). *Exact stochastic simulation of coupled chemical reactions*. Journal of Physical Chemistry 81(25): 2340-2361. [1](#), [6](#)
- [5] Jahnke, T. and W. Huisinga (2007). *Solving the chemical master equation for monomolecular reaction systems analytically*. Journal of Mathematical Biology 54(1): 1-26. [13](#)
- [6] Kim, P. and C. H. Lee (2012). *A probability generating function method for stochastic reaction networks*. Journal of Chemical Physics 136(23). [2](#), [8](#), [10](#)
- [7] Lee, C. H., et al. (2009). *A moment closure method for stochastic reaction networks*. The Journal of Chemical Physics 130(13): -. [2](#), [10](#)
- [8] Van Kampen, N. G. (1992). *Stochastic processes in physics and chemistry*. Access Online via Elsevier. [1](#), [3](#)
- [9] Kim, P. and C. H. Lee (2014). *Fast Probability Generating Function Method for Stochastic Chemical Reaction Networks*. MATCH Communications in Mathematical and in Computer Chemistry 71(1): 57-69. [2](#), [10](#)
- [10] Sunkara, Vikram (2013). *Analysis and Numerics of the Chemical Master Equation*. Ph. D. thesis, Australian National University. [1](#), [3](#)
- [11] ILIE, S., et al. (2009). *Numerical solution of stochastic models of biochemical kinetics*. Canadian Applied Mathematics Quarterly 17(3): 523-554.
- [12] Adalsteinsson, D., et al. (2004). *Biochemical network stochastic simulator (BioNetS): software for stochastic modeling of biochemical networks*. BMC bioinformatics 5(1): 24.

-
- [13] Gibson, M. A. and J. Bruck (2000). *Efficient exact stochastic simulation of chemical systems with many species and many channels*. The journal of physical chemistry A 104(9): 1876-1889.
- [14] Gillespie, D. T. (1976). *A general method for numerically simulating the stochastic time evolution of coupled chemical reactions*. Journal of computational physics 22(4): 403-434.
- [15] Gillespie, D. T. (2001). *Approximate accelerated stochastic simulation of chemically reacting systems*. The Journal of Chemical Physics 115(4): 1716-1733.
- [16] Gillespie, D. T. (2007). *Stochastic simulation of chemical kinetics*. Annu. Rev. Phys. Chem. 58: 35-55.
- [17] Li, H. and L. R. Petzold (2007). *Stochastic simulation of biochemical systems on the graphics processing unit*. Santa Barbara: Department of Computer Science, University of California.
- [18] Lotka, A. J. (1920). *Analytical note on certain rhythmic relations in organic systems*. Proceedings of the National Academy of Sciences of the United States of America 6(7): 410.
- [19] Rao, C. V. and A. P. Arkin (2003). *Stochastic chemical kinetics and the quasi-steady-state assumption: application to the Gillespie algorithm*. The Journal of Chemical Physics 118(11): 4999-5010.
- [20] Rathinam, M., et al. (2003). *Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method*. The Journal of Chemical Physics 119(24): 12784-12794.
- [21] Zhang, J. (2009). *Numerical Methods for the Chemical Master Equation*. Ph. D. thesis, Virginia Polytechnic Institute and State University.

Acknowledgements

먼저 이 논문을 쓰기까지 가장 많은 도움을 주신 김필원 교수님께 감사를 드립니다. 제가 기초 과목인 선형대수를 공부할 때부터 연구 주제를 선정하고, 논문을 쓰는 과정까지 저를 지도해주시고, 힘들 때마다 저를 격려해주셔서 감사합니다. 나중에 제가 누군가를 가르칠 수 있는 상황이 온다면 교수님을 떠올리면서 가르쳐 주고 싶습니다.

제가 질문을 할 때마다 최선을 다해서 대답해 주시고, 또 좋은 질문을 제게 해주시는 권봉석 교수님께 감사합니다.

졸업 논문 심사 위원으로서 많은 조언을 해주셨던 이창형 교수님과 서병기 교수님께도 감사의 인사를 전하고 싶습니다.

보고싶은 신상묵 교수님께도 감사를 드립니다.

항상 배려하고, 양보해주고, 가장 오랜 시간을 함께 생활하면서 동고동락하는 우리 연구실 선후배 동료들, 선이 언니, 문성환 박사님, My roommate Vinuselvi, 지금은 떨어져있는 나의 친한 친구들에게도 감사의 인사를 하고 싶습니다.

마지막으로 언제나 나를 응원해주는 우리 가족에게 감사합니다. 자랑스러운 우리 부모님, 언니, 남동생에게 고마움과 사랑을 전합니다.

