MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **71** (2014) 57-69

ISSN 0340 - 6253

Fast Probability Generating Function Method for Stochastic Chemical Reaction Networks

Pilwon Kim, Chang Hyeong Lee*

School of Technology Management and Department of Mathematical Sciences,

Ulsan National Institute of Science and Technology(UNIST),

Ulsan Metropolitan City 689-798, South Korea

pwkim@unist.ac.kr, chlee@unist.ac.kr

(Received June 25, 2013)

Abstract

Chemical master equations of the stochastic reaction network can be reformulated into a partial differential equation(PDE) of a probability generating function (PGF). Such PDEs are mostly hard to deal with due to variable coefficients and lack of proper boundary conditions. In this paper, we propose a way to reduce PGF-PDEs into a sparse linear system of coefficients of a power series solution. A power of such matrix gives a fast approximation of the solution. The process can be further accelerated by truncating high-order moments. The truncation also makes the method applicable to reaction networks with time-varying reaction rates. We show numerical accuracy of the method by simulating motivating biochemical examples including a viral infection model and G_2/M model.

1 Introduction

Stochastic chemical kinetics have been widely used for analyzing and simulating relatively small biochemical systems such as gene regulatory networks and protein folding models[1].

*Corresponding author

The governing equation of the stochastic model is the chemical master equation or Kolmogorov forward equation

$$\frac{\partial}{\partial t}p(\mathbf{n},t) = \sum_k a_k(\mathbf{n}-V_k)\cdot p(\mathbf{n}-V_k,t) - \sum_k a_k(\mathbf{n})\cdot p(\mathbf{n},t), \tag{1}$$

where $\mathbf{n}(t) = (n_1(t), n_2(t), \ldots, n_s(t))$, each $n_i(t)$ denotes the molecular number of ith species at time t, a_k is the propensity function or probability intensity for the kth reaction and V_k denotes the kth column of the stoichiometric matrix V of which (i,j)th entry is the change in the number of molecules of the ith species by the occurrence of the jth reaction [2]. Also, by identifying all possible states and transitions between states, one can write the chemical master equation by a linear ODE system

$$\frac{dp}{dt} = Kp, \tag{2}$$

where p denotes the vector of probability of all possible states and A denotes the rate matrix of transitions between all the states.

Since (1) and (2) are high dimensional systems in most real applications, it is very difficult or impossible to find the solution analytically or computationally. An alternative way of describing the stochastic dynamics is done by using the stochastic simulation algorithm(SSA)[3]. One realization of the SSA shows the trajectory of the time-evolution of the states. Since it is basically a Monte-Carlo type algorithm, one has to perform many realizations to obtain important statistical quantities such as mean and variance. Moreover, if a given system has relatively large number of molecules or some fast reactions, computational load for the SSA is usually very heavy, which makes the SSA less practical.

There have been many approaches to improve the SSA including Tau-leaping methods [4, 5, 6], reduction methods on the slow time scale[7, 8, 9, 10] and moment closure methods[11, 12, 13]. In the recent paper [14], the authors presented numerical schemes for finding the probability distribution and moments based on power series expansion and Padé approximation. The method successfully estimates the probability solution as well as the first two moments for biochemical examples.

In this paper, we present a novel approximation method called "fast probability generating function(FPGF) method" that improves the PGF method recently proposed in [14]. By using a reduction scheme, we convert the PGF-PDEs into a sparse linear system. This implies that symbolic computations of high order approximation are replaced with power computation of sparse matrices, and therefore long-term simulation of the reaction

network can be obtained more easily. We also introduce successive superimposition of the Taylor expansion. This is based on truncation of high order moments and repeated simplification of coefficient functions. In addition to keeping approximation order reasonably small, we can also simulate the networks with time-varying reaction rates.

An outline of the paper is as follows; In Section 2, we review the properties of PDEs that PGF should satisfy. Section 3 presents the reduction method for the PGF-PDEs. In what follows, we develop superimposition of Taylor expansion for the PGF methods. In Sections 3 and 4, we illustrate numerical accuracy of the method by simulating examples such as an enzyme-substrate reaction model, the viral infection network and the G_2/M transition model.

2 Probability Generating Function

The probability generating function is defined as

$$G(\mathbf{z},t)=\sum_{\mathbf{n}=\mathbf{0}}^{\infty}\mathbf{z}^{\mathbf{n}}p(\mathbf{n},t), \tag{3}$$

where $\mathbf{z}=(z_1,\cdots,z_s)$, $z_i\in[-1,1]$, $\mathbf{n}=(n_1,n_2,\ldots,n_s)$ and

$$\mathbf{z}^{\mathbf{n}}\equiv z_1^{n_1}z_2^{n_2}\cdots z_s^{n_s}.$$

After differentiation of (3) with respect to t and application of the equation (1), one can derive a PDE

$$\frac{\partial G}{\partial t}=\sum_{|\mathbf{k}|\le m}H_{\mathbf{k}}(\mathbf{z},t)D^{\mathbf{k}}G. \tag{4}$$

Here $\mathbf{k}=(k_1,\cdots,k_s)$, $|\mathbf{k}|=k_1+\cdots+k_s$, m is the highest order of the reactions in the chemical system, and $D^{\mathbf{k}}G$ denotes any k-th order partial derivatives of G as

$$D^{\mathbf{k}}G=\frac{\partial^{k_1}}{\partial z_1^{k_1}}\frac{\partial^{k_2}}{\partial z_2^{k_2}}\cdots\frac{\partial^{k_s}}{\partial z_s^{k_s}}G.$$

One can see that the PDE (4) is linear and H_k is a polynomial function of degree at most order m which satisfies $H_k(\mathbf{z}=\mathbf{1},t)=0$ for $k=0,1,\cdots,m$.

The conditions of the PDE (4) can be found from the conditions on n[14]; The initial condition is

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

where $\mathbf{n}_0$ is the initial condition of $\mathbf{n}$, and also it is true that

$$G(\mathbf{z}=\mathbf{0},t)=p(\mathbf{n}=\mathbf{0},t),\ \text{and}\ G(\mathbf{z}=\mathbf{1},t)=\sum_{\mathbf{n}}p(\mathbf{n},t)=1. \tag{5}$$

Using the solution G of the PDE (4), one can find the important information for stochastic reaction network such as marginal probability, mean and the second moment [14]; If $N_i(t)$ denotes the number of molecules of ith species at time t, the probability that $N_i = \ell$ at time t is

$$p(N_i = \ell, t) = \frac{1}{\ell!} \frac{\partial^\ell G(\mathbf{z}, t)}{\partial z_i^\ell}\bigg|_{z_i=0, z_j=1, j\neq i}, \tag{6}$$

and the mean and second moment are obtained by the first and second derivatives of G

$$\frac{\partial}{\partial z_i} G(\mathbf{z} = \mathbf{1}, t) = E[N_i(t)] \tag{7}$$

$$\frac{\partial^2}{\partial z_i \partial z_j} G(\mathbf{z} = \mathbf{1}, t) = \begin{cases} E[N_i(t)N_j(t)] & \text{if } i \neq j \\ (E[N_i^2(t)] - E[N_i(t)]) & \text{if } i = j. \end{cases} \tag{8}$$

where $E[X(t)]$ denotes the expectation of $X(t)$.

Note that, in [14], slow convergence of the power series expansion like (7),(8) and (6) can be greatly improved by Padé approximation, in which the expansion is re-arranged into a ratio of two series expansions.

3 Fast Probability Generating Function Method

In the PGF method recently proposed in [14], $G(\mathbf{z}, t)$ is represented by a power expansion with respect to t as

$$G(\mathbf{z}, t) = \sum_{j=0}^{\infty} f_j(\mathbf{z}) t^j, \tag{9}$$

where $f_j, j = 0, 1, \ldots,$ are coefficient functions of $\mathbf{z}$. Note that the initial condition determines the first coefficient as $f_0(\mathbf{z}) = \mathbf{z}^{\mathbf{n}_0}$.

Suppose the reaction rates in the network do not change with time and therefore the coefficients of the PDE do not involve t explicitly. This implies $H_{\mathbf{k}}(\mathbf{z}, t) = H_{\mathbf{k}}(\mathbf{z})$ in (4). Plugging (9) into the both sides of (4) and comparing the power series leads to a recursive relation between $f_{j+1}(\mathbf{z})$ and $f_j(\mathbf{z})$, $j = 0, 1, \cdots$. One can derive coefficient functions $f_1(\mathbf{z}), f_2(\mathbf{z}), \cdots, f_J(\mathbf{z})$ from $f_1(\mathbf{z}) = \mathbf{z}^{\mathbf{n}_0}$ and then construct an approximation $\sum_{j=0}^{J} f_j(\mathbf{z}) t^j$ for sufficiently large J. Since the initial function $f_0(\mathbf{z})$ and the coefficient of the PDE (3) are polynomials, all $f_j(\mathbf{z})$ generated are polynomials as well. However, even though symbolic computation of polynomials is generally not expensive in computer algebra tools like Mathematica, the number of terms to handle grows fast, which makes the computation slow down as the approximation order j increases.

Now, we present the Fast PGF method, which accelerates the above computational process by turning the system into a sparse linear system. Let us set

$$f_j(\mathbf{z}) = \sum_{|\mathbf{n}| \le J + |\mathbf{n_0}|} f_{j,\mathbf{n}}\ \mathbf{z}^{\mathbf{n}}. \tag{10}$$

The number of the coefficients needed in the Jth order approximation is

$$N = \sum_{i=0}^{J+|\mathbf{n_0}|} \binom{i+s-1}{s-1}. \tag{11}$$

As the recursive formula for f_j can be rephrased in terms of $f_{j,\mathbf{n}}$ and the corresponding relations are linear, a "PGF linear system" is obtained as

$$F_j = \frac{1}{j} M F_{j-1} \tag{12}$$

where F_j is a vector of which components are $f_{j,\mathbf{n}}, |\mathbf{n}| \le J$ and M is an $N \times N$ matrix. M is a large but sparse matrix, since the formula is combination of at most m times of differentiation followed by product of $H_{\mathbf{k}}$ which is also a polynomial of degree m.

The idea of the fast approximation is that the matrix M can be reduced into a much sparser matrix, by deriving a reduced recursive form for specific statistical information of interest rather than dealing with the original matrix for general approximation. For example, computing ith moment requires to find $G_i \equiv \dfrac{\partial G}{\partial z_i}$.

$$\frac{\partial G_i}{\partial t} = \sum_{|\mathbf{k}| \le m} \left(\frac{\partial H_{\mathbf{k}}}{\partial z_i} D^{\mathbf{k}} G + H_{\mathbf{k}} D^{\mathbf{k}} G_i \right). \tag{13}$$

One important observation is that the part of the recursive formula generated from the second term $H_{\mathbf{k}} D^{\mathbf{k}} G_i$ does not contribute the final computation of ith moment, since they vanish when plugged $\mathbf{z} = \mathbf{1}$. This implies that the recursive form for ith moment can be obtained not necessarily from (13) but from the reduced equation

$$\frac{\partial G_i}{\partial t} = \sum_{|\mathbf{k}| \le m} \frac{\partial H_{\mathbf{k}}}{\partial z_i} D^{\mathbf{k}} G. \tag{14}$$

Since the coefficient function $\dfrac{\partial H_{\mathbf{k}}}{\partial z_i}$ is of order at most $m-1$, this reduction makes the resulted matrix M much sparser in practical computation.

Turning the symbolic computations into computation of power of reduced sparse matrices makes a higher-order approximation possible. This immediately means that the method can be applied to stiff systems such as the viral infection model in the next section. We call the method presented in this section fast probability generating function(FPGF) method.

3.1 Application 1: Enzyme Kinetics

As the first application, we consider a fundamental chemical kinetic model, the enzyme-substrate reaction system

$$E \ + \ S \underset{c_{-1}}{\overset{c_1}{\rightleftarrows}} ES \overset{c_2}{\to} \ E + \ P, \tag{15}$$

where E, S, ES and P denote enzyme,substrate, enzyme-substrate complex and product, respectively and c_1, c_{-1} and c_2 are probability constants for reactions. If we denote the molecular numbers of E, S, ES and P by n_1, n_2, n_3 and n_4, respectively, the governing equation of the stochastic enzyme-substrate system is obtained as

$$\begin{aligned} & \frac{\partial p(n_1,n_2,n_3,n_4,t)}{\partial t} \\ = \ & c_1(n_1+1)(n_2+1)p(n_1+1,n_2+1,n_3-1,n_4,t) \\ + \ & c_{-1}(n_3+1)p(n_1-1,n_2-1,n_3+1,n_4,t) + c_2(n_3+1)p(n_1-1,n_2,n_3+1,n_4-1,t) \\ - \ & (c_1n_1n_2 + c_{-1}n_3 + c_2n_3)p(n_1,n_2,n_3,n_4,t). \end{aligned} \tag{16}$$

One can see that there are two conservation quantities in the system and thus we can remove two of variables, say n_3, n_4[14]. After the removal, we can derive a PDE of $G(z_1, z_2, t)$ from the master equation

$$\begin{aligned} G_t \ &= \ c_1(1-z_1z_2)G_{12} + \Big((c_{-1}+c_2)z_1 - c_{-1}z_1^2z_2 - c_2z_1^2\Big)G_1 \\ &+ \ A\Big(c_{-1}z_1z_2 + c_2z_1 - (c_{-1}+c_2)\Big)G, \end{aligned}$$

where A is the conserved quantity $n_1 + n_3$.

To illustrate how the method work in a matrix system, let us take a simple case with $n_0 = n_1 = 1$ and $J = 1$. Then the corresponding matrix M in (12) is

$$\begin{bmatrix} \alpha B_0 & c_{-1}A & 0 & 0 & c_2A & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha B_1 & 0 & B_1c_{-1} & 0 & 0 & 0 & B_1c_2 & 0 & 0 \\ 0 & 0 & \alpha B_0 & 0 & c_{-1}A & 0 & 0 & 0 & c_2A & 0 \\ 0 & 0 & 0 & \alpha B_2 & 0 & 0 & B_2c_{-1} & 0 & 0 & 0 \\ c_1 & 0 & 0 & 0 & B_1 - \alpha c_1 & 0 & 0 & B_1c_{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha B_0 & 0 & 0 & c_{-1}A & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha B_3 & 0 & 0 & 0 \\ 0 & 2c_1 & 0 & 0 & 0 & 0 & 0 & B_2 - 2\alpha c_1 & 0 & 0 \\ 0 & 0 & 2c_1 & 0 & 0 & 0 & 0 & 0 & B_1 - 2\alpha c_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha B_0 \end{bmatrix}$$

where $B_j = -A + j$, $j = 0, 1, 2, 3$ and $\alpha = c_2 + c_{-1}$. One can keep generating $F_j \in \mathbf{R^{10}}$ in (12), whose components represent the coefficients of $1, z_1, z_2, z_1^2, z_1z_2, z_2^2, z_1^3, z_1^2z_2, z_1z_2^2, z_2^3$,

respectively. This leads to a construction of $f_j(z)$ and eventually

$$G(\mathbf{z}, t) \approx \sum_{j=0}^{J} f_j(\mathbf{z}) t^j.$$

Figure 1 shows comparison of the exact solution of the master equation (16) and the approximate solution obtained from the FPGF method.

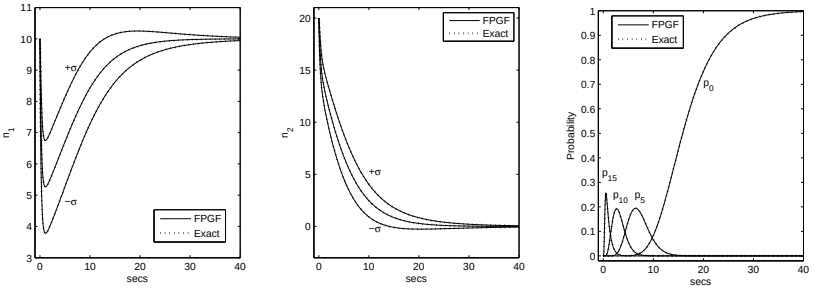

Figure 1: Comparison of mean, standard deviation and marginal probability for the enzyme-substrate model. In the three figures, we assume $n_1(0) = 10, n_2(0) = 20$, $A = 10$ and parameters $c_1 = 0.1s^{-1}, c_{-1} = 1s^{-1}, c_3 = 0.5s^{-1}$. The two left figures show the comparison of exact values and approximate values obtained from the FPGF method for mean and mean $\pm$ standard deviations($\pm\sigma$) of n_1 and n_2, respectively and the right figure shows the comparison of the exact probability and the approximate probability obtained from the FPGF method. In the figure, p_i denotes the probability that the number n_2 of the substrate is i. The exact probability, mean and variance are obtained by rewriting Equation (16) into the linear ODE system $\frac{dp}{dt} = Kp$ under the given initial condition and then solving it by MATLAB.

3.2 Application 2: Viral Infection model in E.coli

We consider an interesting biochemical model. An intracellular viral infection model in E.coli [15, 16] is described by the following six reactions;

$$\begin{aligned}
R_1 &: G \xrightarrow{c_1} T + G \\
R_2 &: T \xrightarrow{c_2} \phi \\
R_3 &: T \xrightarrow{c_3} T + G \\
R_4 &: G + P \xrightarrow{c_4} \phi \\
R_5 &: T \xrightarrow{c_5} P + T \\
R_6 &: P \xrightarrow{c_6} \phi,
\end{aligned}$$

where we denote genome, template and viral protein by G, T and P, respectively. The reaction parameters are given as the table

Parameter	Value
c_1	0.025day^{-1}
c_2	0.25day^{-1}
c_3	1.0day^{-1}
c_4	$11.25 \times 10^{-6}\text{day}^{-1}$
c_5	1000day^{-1}
c_6	1.9985day^{-1}

To apply the FPGF method, we first derive the PDE for the probability generating function $G_(z_1, z_2, z_3, t)$ as

$$G_t = c_1 z_1(z_2-1)G_1 + G_2\Big(c_2-(c_2+c_3+c_5)z_2+c_5 z_2 z_3+c_3 z_1 z_2\Big)+c_6(1-z_3)G_3+c_4(1-z_1 z_3)G_{13},$$

where $G_i = \frac{\partial G}{\partial z_i}, i = 1, 2, 3$ and $G_{13} = \frac{\partial^2 G}{\partial z_1 \partial z_3}$.

It is known that the model has two fast reactions R_5 and R_6, which makes computations by the SSA very expensive and intensive[17]. Since simulation of this model by the SSA requires a huge amount of time, we use an approximate stochastic simulation algorithm(ASSA) done by [15] and compare the results of the FPGF to those of the ASSA in Figure 2.

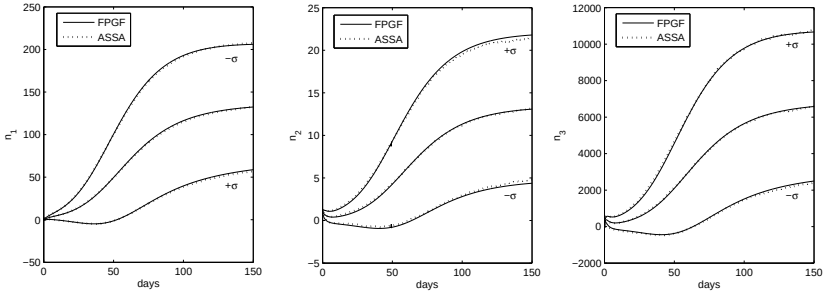

Figure 2: Comparison of mean and standard deviation for the viral infection model. The initial condition $n_1(0) = 0, n_2(0) = 1$,and $n_3(0) = 0$ are assumed[15]. The three figures show the comparison of mean and mean $\pm$ standard deviations($\pm\sigma$) of n_1, n_2 and n_3 obtained from the ASSA and the FPGF. The results from the ASSA are based on 50,000 realizations.

Table 1 compares CPU times taken by the FPGF and the ASSA and shows that the FPGF is by far faster than the ASSA.

	FPGF	ASSA
CPU Time	2.0131×10^3	3.6289×10^4

Table 1: Comparison of CPU times of FPGF and ASSA in Application 2

4 Superimposition of Truncated Expansions

Even with the reduction method suggested in the previous section, a high order approximation generally requires large consumption of computational resources. This is especially true for the reaction networks involving many species. Another issue about the PGF method is dealing with the networks with time-varying reaction rates. If $H_{\mathbf{k}}$ varies with time, the recursive formula from Equation (9) may generate non-polynomial terms and the PGF method loses its computational benefit.

These two issues with the PGF method can be worked out by successive superimposition of Taylor expansion based on truncation of higher order moment. Let h be a time step. Consider successive functions $G^{[r]}(\mathbf{z}, t), r = 0, 1, \cdots$, which approximate $G(\mathbf{z}, t)$ on $rh \le t \le (r+1)h, r = 0, 1, \cdots$, respectively. They should satisfy

$$G^{[r]} = \sum_{j=0}^{J} f_j^{[r]}(\mathbf{z}) t^j, \tag{17}$$

$$G^{[r+1]}(\mathbf{z}, (r+1)h) \approx G^{[r]}(\mathbf{z}, (r+1)h). \tag{18}$$

Note that $G^{[r]}$ is Taylor-expanded at $t = rh$ and its value at $t = (r+1)h$ is carried over to $G^{[r+1]}$ to be used for the next Taylor expansion. Our motivation of this successive superimpositions is to keep approximation order J low in favor of computational efficiency while carrying over essential informations like first several moments to a next function over the following time interval.

The main concern in this approach is how to choose a way to simplify $G^{[r]}(\mathbf{z}, (r+1)h)$ into $G^{[r+1]}(\mathbf{z}, (r+1)h)$ in (18) so that the corresponding computation in the next step is easy enough. Here, we use a projection to a low degree polynomial with high order moments zero. In [11], it is shown that the truncating higher order moments to zeros in the approximation yields convergent numerical solutions. Let us define T_J be a set of polynomials $p(\mathbf{z}) : \mathbf{R}^s \to \mathbf{R}$ of degree J or less whose Jth moments are all zeros. Consider a projection $p(\mathbf{z}) = \mathrm{proj}_J g(\mathbf{z})$ for an analytic function $g(\mathbf{z})$, such that $p(\mathbf{z}) \in T_J$ agrees with $f(\mathbf{z})$ upto $(J-1)$th order in the Taylor expansion. For example, if we want

to truncate the third moment and keep the approximation order as low as $J = 3$ at each time step, then the following relations should be used to determines a projection $G^{[r+1]} = \text{proj}_3 G^{[r]}$: At $\mathbf{z} = 0$, we have

$$\begin{aligned} G^{[r+1]} &= G^{[r]} \\ G_i^{[r+1]} &= G_i^{[r]} \\ G_{ij}^{[r+1]} &= G_{ij}^{[r]} \end{aligned}$$

and, for the third order moments,

$$\begin{aligned} G_{iii}^{[r+1]} &= (G_i^{[r]} - 1)(3G_{ii}^{[r]} - G_i^{[r]}(2G_i^{[r]} - 1)) \\ G_{iij}^{[r+1]} &= (2G_i^{[r]} - 1)(G_{ij}^{[r]} - G_i^{[r]}G_j^{[r]}) + G_{ii}^{[r]}G_j^{[r]}, \quad i \neq j \\ G_{ijk}^{[r+1]} &= G_i^{[r]}G_{jk}^{[r]} + G_j^{[r]}G_{ik}^{[r]} + G_k^{[r]}G_{ij}^{[r]} - 2G_i^{[r]}G_j^{[r]}G_k^{[r]}, \quad i \neq j \neq k \neq i \end{aligned}$$

where i, j and $k = 1, \cdots, s$ denote distinct directions for partial derivatives. That is, all the derivatives of $G^{[r+1]}$ of second order or less with respect to $\mathbf{z}$ are the same as those of $G^{[r]}$ and the third order derivatives are set to be zero. One can see the corresponding method does not require computation of higher order coefficients f_i and therefore the involved computation is much faster than the original PGF method. Another notable benefit of the method is that it enables us to deal with a system with time-varying reaction rates, since we can reasonably assume such reaction coefficients are fixed during the short time interval h.

4.1 Application 3: G_2/M Transition Model

We consider the G_2/M transition network in the eukaryotic cell cycle as follows[14, 18];

$$X + Y_p \underset{c_2}{\overset{c_1}{\rightleftarrows}} C_x \xrightarrow{c_3} X_p + Y_p, \quad E_1 + X_p \underset{c_5}{\overset{c_4}{\rightleftarrows}} C_x^e \xrightarrow{c_6} X + E_1$$

$$X_p + Y \underset{c_8}{\overset{c_7}{\rightleftarrows}} C_y \xrightarrow{c_9} X_p + Y_p, \quad E_2 + Y_p \underset{c_{11}}{\overset{c_{10}}{\rightleftarrows}} C_y^e \xrightarrow{c_{12}} Y + E_2.$$

We denote the number of molecules of $X_p, Y_p, X, Y, E_1, E_2, C_x, C_x^e, C_y, C_y^e$ by $n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9, n_{10}$, respectively and

Using the notations

$$G_i = \frac{\partial G}{\partial z_i}, i = 1, 2, \ldots, 10, \quad G_{ij} = \frac{\partial^2 G}{\partial z_i \partial z_j}, i, j = 1, 2, \ldots, 10,$$

we obtain a PDE of the moment generating function $G(\mathbf{z}, t)$ [14]

$$\begin{aligned} G_t &= c_1 G_{23}(z_7 - z_2 z_3) + c_2 G_7(z_2 z_3 - z_7) + c_3 G_7(z_1 z_2 - z_7) \\ &+ c_4 G_{15}(z_8 - z_1 z_5) + c_5 G_8(z_1 z_5 - z_8) + c_6 G_8(z_3 z_5 - z_8) \\ &+ c_7 G_{14}(z_9 - z_1 z_4) + c_8 G_9(z_1 z_4 - z_9) + c_9 G_9(z_1 z_2 - z_9) \\ &+ c_{10} G_{26}(z_{10} - z_2 z_6) + c_{11} G_{10}(z_2 z_6 - z_{10}) + c_{12} G_{10}(z_4 z_6 - z_{10}). \end{aligned}$$

In Figure 3, we compare the simulation results obtained from the SSA and the FPGF method. Here, the stochastic reaction parameter values $c_1 = c_4 = c_7 = c_{10} = 1\ s^{-1}$, $c_2 = c_5 = c_8 = c_{11} = 10\ s^{-1}$, $c_3 = c_6 = c_9 = c_{12} = 100\ s^{-1}$ are used. The system is stiff and requires as a small time step as $h = 0.001$ for the FPGF and as many realizations as 500,000 times for the SSA, respectively, to attain the results without fluctuations.

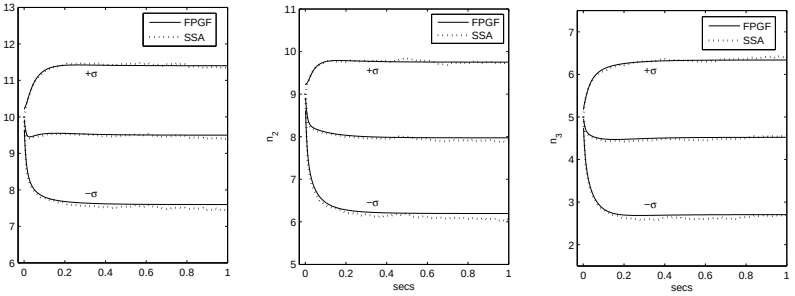

Figure 3: Comparison of mean and standard deviation for the G_2/M transition model. We assume the initial condition $n_1(0) = 10, n_2(0) = 9, n_3(0) = 5, n_4(0) = 4, n_5(0) = 4, n_6(0) = 5, n_7(0) = n_8(0) = n_9(0) = n_{10}(0) = 0$. The three figures show the comparison of mean and mean $\pm$ standard deviations($\pm\sigma$) of n_1, n_2 and n_3 obtained from the SSA and the FPGF.

Computational time comparison in Table 2 between FPGF and SSA shows how much improvement is achieved by superimposition of truncated expansions.

	FPGF	SSA
CPU Time	1.9425×10^2	2.3253×10^3

Table 2: Comparison of CPU times of FPGF and SSA in Application 3

However, in addition to less computational time, another benefit of the FPGF is significant memory saving. While the SSA often needs excessive amount of memory in

iterative computation, the FPGF hardly causes any severe memory issue since if has a similar algorithmic structure to standard numerical schemes for one dimensional PDEs.

5 Conclusion

In this work, we presented a fast approximation method that improves the probability generating function(PGF) method for computation of the solutions of stochastic reaction networks. Advantage of the PGF method is, rather than struggling with the chemical master equation of extremely high order, one can handle a partial differential equation of low order. However, it still requires intense computation for high order approximation. The FPGF method proposed in this paper turns the PGF-PDEs into a sparse linear system and replaces symbolic computation with power computation of such matrices. The computation can be even further accelerated by successive superimposition of the Taylor expansion. Truncating high order moments and simplifying coefficient functions repeatedly, one can find long-term approximation of statistical information while keeping approximation order reasonably small. This truncation method also enables us to simulate the networks with time-varying reaction rates to which the original PGF method cannot apply. We expect that the FPGF method proposed in the paper will be used for a fast and accurate approximation for computation of real and complex stochastic chemical reaction networks.

Acknowledgments: The first author(P. Kim) acknowledges that this work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0023486). The corresponding author(C.H. Lee) was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2010-0024849).

References

[1] C. V. Rao, D. M. Wolf, A. P. Adams, Control, exploitation and tolerance of intracellular noise, *Nature* **420** (2002) 231–237.

[2] D. T. Gillespie, A rigorous derivation of the chemical master equation, *Physica A* **188** (1992) 404–425.

[3] D. T. Gillespie, Exact simulation of coupled chemical reactions, *J. Phys. Chem.* **81** (1977) 2340–2361.

[4] D. T. Gillespie, Approximate accelerated stochastic simulation of chemically reacting systems, *J. Chem. Phys.* **115** (2001) 1716–1733.

[5] Y. Cao, D. T. Gillespie, L. R. Petzold, Efficient step size selection for the tau–leaping simulation method, *J. Chem. Phys.* **124** (2006) 044109.

[6] D. F. Anderson, Incorporating postleap checks in tau–leaping, *J. Chem. Phys.* **128** (2008) 054103.

[7] Y. Cao, D. T. Gillespie, L. R. Petzold, The slow–scale stochastic simulation algorithm, *J. Chem. Phys.* **122** (2005) 014116.

[8] B. Munsky, M. Khammash, The finite state projection algorithm for the solution of the chemical master equation, *J. Chem. Phys.* **124** (2006) 044104.

[9] C. H. Lee, R. Lui, A reduction method for multiple time scale stochastic reaction networks, *J. Math. Chem.* **46** (2009) 1292–1321.

[10] C. H. Lee, R. Lui, A reduction method for multiple time scale stochastic reaction networks with non–unique equilibrium probability, *J. Math. Chem.* **47** (2010) 750–770.

[11] C. H. Lee, K. Kim, P. Kim, A moment closure method for stochastic reaction networks, *J. Chem. Phys.* **130** (2009) 134107.

[12] P. Milner, C. S. Gillespie, D. J. Wilkinson, Moment closure approximations for stochastic kinetic models with rational rate laws, *Math. Biosci.* **231** (2011) 99–104.

[13] C. H. Lee, A moment closure method for stochastic reaction networks with general kinetics, *MATCH Commun. Math. Comput. Chem.* **70** (2013) 785–800.

[14] P. Kim, C. H. Lee, A probability generating function method for stochastic reaction networks, *J. Chem. Phys.* **136** (2012) 234108.

[15] J. Goutsias, Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems, *J. Chem. Phys.* **122** (2005) 184102.

[16] J. S. R. Srivastava, L. You, J. Yin, Stochastic versus deterministic modeling of intracellular viral kinetics. *J. Theor. Biol.* **218** (2002) 309–321.

[17] E. L. Haseltine, J. B. Rawlings, Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics, *J. Chem. Phys.* **117** (2002) 6959–6969.

[18] A. Kumar, K. Josić, Reduced models of networks of coupled enzymatic reactions, *J. Theor. Biol.* **278** (2011) 87–106.