SAÜ Fen Bil Der 20. Cilt, 3. Sayı, s. 617-625, 2016

Using conserved cycles in exact stochastic simulation algorithms

Derya Altıntan[1*]

01.08.2016 Geliş/Received, 02.09.2016 Kabul/Accepted

doi: 10.16984/saufenbilder.22901

ABSTRACT

Biochemical reaction systems involve many different species interacting via many different reaction channels. When the number of species and the abundance of species are so high, pure modeling approaches based on differential equations suffer from curse of dimensionality. If a system involves conserved cycles, abundances of some species can be obtained via algebraic relations which in turn will reduce the dimension of differential equations representing the dynamics of the system. In the present paper, we propose a numerical algorithm that uses Gauss-Jordan method to obtain conserved cycles in biochemical systems. We give this algorithm in Direct Method (DM), First Reaction Method (FRM) and Next Reaction Method (NRM) which obtain exact realizations of the state vector in stochastic modeling approach. We apply these three algorithms with/without using conservation relations to biochemical systems in different sizes and compare the computational costs of two different versions of each exact algorithm.

Keywords: conservation relations, gauss-jordan method, direct method, first reaction method, next reaction method

Korunumlu döngülerin stokastik simülasyon algoritmalarında kullanımı

ÖZ

Biyokimyasal reaksiyon sistemleri farklı reaksiyonlar aracılığıyla etkileşime giren birçok farklı türü içerir. Sistem içerisinde yer alan türlerin sayıları ve miktarları çok yüksek olduğunda, diferansiyel denklemlere dayanan saf modelleme yaklaşımları çok boyutluluktan muzdarip olurlar. Eğer bir sistem korunumlu döngüler içerirse, bazı türlerin miktarları cebirsel bağlantılar yoluyla elde edilebilir bu da sistemin dinamiklerini temsil eden diferansiyel denklemlerin boyutunu düşürür. Bu çalışmada, biyokimyasal reaksiyon sistemlerinde yer alan korunumlu döngüleri elde etmek için Gauss-Jordan metodunu kullanan bir nümerik algoritma öneriyoruz. Algoritmayı stokastik modelleme yaklaşımında konum vektörünün tam realizasyonlarını elde eden Direk Metod (DM), İlk Reaksiyon Metodu (FRM) ve Sonraki Reaksiyon Metodu (FRM) içerisinde verdik. Bu üç algoritmayı korunum bağıntılarını içerecek/içermecek şekilde farklı boyutlardaki biyokimyasal sistemlere uyguladık ve her tam lagoritmanın farklı iki versiyonunun hesaplama miktarları kıyasladık.

Anahtar Kelimeler: korunum bağıntıları, gauss-jordan metodu, direk metod, ilk reaksiyon metodu, sonraki reaksiyon metodu.

* Sorumlu Yazar / Corresponding Author
1 Selçuk Üniversitesi, Fen Fakültesi, Matematik Bölümü, Konya- altintan@selcuk.edu.tr, The author acknowledges support from Scientific and Technological Research Council of Turkey (TUBİTAK), Program no: 3501 Grant no. 115E252

1.INTRODUCTION

On microscopic level, biochemical networks may involve hundreds or thousands of species and also reaction channels. Many researchers from different disciplines have considered the problem of mathematical modeling of these systems [1]. The traditional approach based on the idea that the dynamics of biochemical reactions are continuous and deterministic uses Ordinary Differential Equations (ODEs) to model these systems. In cellular reactions, fluctuations in abundances of mRNA, DNA which are present in a very small copy numbers can lead to cell-to-cell variability. Since stochasticity and randomness in the nature of these processes can't be captured by ODEs, stochastic approach which uses Continuous Time Markov Chains (CTMCs) to model dynamics of these processes is proposed as an alternative [2], [3], [4]. In this modeling approach, the state vector of the process satisfies the Random Time Change Model (RTCM) [5] and the time evolution of the probability distribution of the system is defined by the Chemical Master Equation (CME) [6]. In most applications, researchers focus on the CME.

Differential equations used in both approaches suffer from the curse of dimensionality, because the dimension of corre-sponding ODEs and CMEs is proportional to the number of species and the copy numbers of species. Therefore, when the system of interest is large, it will be so difficult to obtain analytical solutions of these equations. As a result, dividing large systems into small subgroups to decrease the computational cost of analyzing the dynamics of original system is very popular among researchers. For example, in [7], authors used invariants and flow equivalent servers to transform complex biological models into simplified models which preserve the dynamics of the original model. In [8], two different approaches based on conservation relations and differences in the speed of reactions are proposed to simplify complex systems.

To understand the dynamical behavior of a system, the stiochiometric matrix which represents the net change in the copy numbers of species produced by reactions has a crucial importance. Therefore, analysis of stiochiometric matrices gains popularity among researchers [8], [9], [10], [11]. Conserved cycles which can be extracted from the stiochiometric matrices can be used to simplify reaction systems. If the total amount of some species remains constant during the process, this means that there are conserved cycles in the system. For example, phosphorylation event in pathways is the result of Adenin Triphosphate (ATP) Adenosine Diphosphate (ADP) change, therefore, during such a phosphorylation process, the total amount of ATP and ADP equals to a constant value which can be obtained from their initial conditions [12]. Also, the number of genes in a gene expression never changes. As a result, if the number of ATP in the process is known at a specific time, then, the number of ADP at this specific time point can be obtained by subtracting the copy numbers of ATP from the total amount of ATP and ADP which is given initially or vice versa. Similarly, the number of passive genes at a specific time of the process can be obtained via the number of active genes at this time point or vice versa. There are different ways of obtaining conserved cycles in a process such as Haousholder based methods [13], Gauss-Jordan elimination methods [11], methods based on atomic transition networks of species [14]. We refer to [11] and the references therein for a review on the methods of obtaining conservation relations in a biochemical process of interest.

Conservation relations in biochemical systems are defined by algebraic equations. The number of conserved cycles in a biochemical system can be obtained from its stoichiometric matrix [11], [15], [16]. If we use ODEs to model a system with conserved cycles [4], [10], [12], using conservation relations transforms ODEs into Differential Algebraic Equa-tions (DAEs). Similarly, in case of using SDEs to model such systems, involving conservation relations transform SDEs into Stochastic Differential Algebraic Equations (SDAE) [17], [18].

There are two goals of the present paper. The first aim is to propose a MATLAB algorithm which obtains conserved cycles of a biochemical reaction system. This algorithm can also be involved in stochastic simulation algorithms which give exact realizations of a stochastic process whose probability function satisfies the CME. We mainly deal with exact algorithms which are Direct Method (DM), First Reaction Method (FRM), and Next Reaction Method (NRM). The second aim of the paper is to propose improvements of DM, FRM, and NRM algorithms which uses conserved cycles and to compare the computational costs of those new algorithms with original versions of them.

The organization of the paper is as follows: In Section II, we give a small summary on basics of stochastic modeling approach and DM, FRM, and NRM algorithms. Details of these algorithms can be found in the Appendix. Section III which can be considered as the main part of the paper is devoted to explain how the stoichiometric matrix can be used to obtain conserved cycles in the system. The MATLAB algorithm that finds conservation relations and the definition of comparison criteria are also given in this section. In Section IV, we implement the system to different biochemical systems with different sizes. Finally, Section VI concludes the

paper with a discussion on algorithms and gives an idea on future works of the subject.

2.BACKGROUND

Suppose that we have a well stirred system involving $N \in \mathbb{N}$ species $\{S_1, S_2, \ldots, S_N\}$ and M reaction channels $\{R_1, R_2, \ldots, R_M\}$.Let $X(t) = (X_1(t), X_2(t), \ldots, X_N(t)) \in \mathbb{N}_0^N$ be the state vector of the system at a time $t \geq 0$ where $X_j(t)$ is the number of molecules of S_j. A single occurrence of the $k-$th reaction channel R_k changes the system state from $X(t)$ to $X(t)+\nu_k$ Here, $\nu_k = (\nu_{1k}, \nu_{2k}, \ldots, \nu_{Nk}) \in \mathbb{Z}^N$ represents the stoichiometric vector whose $j-$th component ν_{jk} denotes the net effect of a single firing of R_k on the number of molecules of S_j. Given $X(t) = x$, the probability of an occurrence of reaction R_k in the time interval $[t, t+h)$ is determined by $\alpha_k(x)h$ for sufficiently small h. Here, $\alpha_k(x)$ is the product of the reaction rate constant and the number of distinct combinations of reactants of R_k and called as the *propensity function* [19].

If X is a Continuous Time Markov Chain (CTMC), a goal of stochastic modeling approach is to obtain the following probability function

$$p(t,x) = \mathbb{P}(X(t) = x \,|\, X(0) = x_0)$$

which satisfies the set of differential equations called Chemical Master Equation (CME) in the following form

$$\frac{\partial p(x,t)}{\partial t} = \sum_{j=1}^{M} \left[\alpha_j(x-\nu_j) p(x-\nu_j, t) - \alpha_j(x) p(x,t) \right] \quad (1)$$

The number of differential equations in this system is determined by the number of species and the number of molecules of species. If we have L number of molecules of each species, then, the number of differential equations in the corresponding CME is N^L. Therefore, it is very hard to solve CME analytically, except for mono molecular reaction systems [20]. As a result, simulation-based numerical methods that obtain exact realizations of the state vector of the system of interest are proposed. Stochastic Simulation Algorithms (SSAs) which are proposed by Gillespie can be considered as cornerstones of exact algorithms [21]. There are two different versions of SSAs, they are Direct Method (DM) and First Reaction Method (FRM). To improve the efficiency of FRM, Gibson and Bruck proposed the Next Reaction Method (NRM) [22]. DM and FRM differ from each other depending on the way of answering following two questions:

- What is the firing time of the next reaction, t^* ?
- What is the index of the firing reaction, μ ?

Given $X(t) = x$, the DM considers t^* as a random variable distributed according to an exponential distribution with mean $\alpha_0(x) = \sum_{j=1}^{M} \alpha_j(x)$ and μ is a vector distributed according to an integer density function $\alpha_\mu(x)/\alpha_0(x)$. Differently than DM, FRM produces different $t_j^*, j = 1,2,\ldots,M$ values for each reaction according to exponential distribution with mean $\alpha_j(x)$. Chooses t^* as the smallest of t_j^* values and obtain μ such that $t^* = t_\mu^*$. The NRM finds t^* and μ values by using the same strategy with FRM. To decrease the computational cost of the FRM, it constructs two data structures namely *dependency graph* J and *indexed priority queue* Q. Dependency graph shows that propensity functions are changed when a given reaction fired. Naturally, when a single reaction fired only updating propensities whose values are changed instead of updating all propensities decreases the computational cost of the algorithm. Indexed priority queue is an another data structure that saves the time increments of each reaction to reuse them in case of need. Algorithms of these methods can be seen in Appendix.

3.CONSERVATION RELATIONS AND NEW ALGORITHMS

In pure modeling approaches such as traditional deterministic approach based on ODEs or stochastic approach based on CME, the number of differential equations directly proportional to the number of species in the biochemical process under consideration. As a result, when the number of species in the system is so high, the number of differential equations in the corresponding ODE or CME is also high. If there exist conserved cycles in the system, abundance of some species can be found via algebraic relations which in turn will reduce the number of differential equations in the corresponding differential equation set. As mentioned before, the meaning of having conserved cycles in biochemical processes is that the total sum of abundance of some species doesn't change during the time interval under consideration. If we consider a gene expression,

the number of gene doesn't change or if we consider an enzyme-substrate system such as Michaelis Menten kinetics, the total amount of enzyme and enzyme-substrate complex is constant during the time of process [4], [12].

Existence of conserved cycles in a biochemical system can be understood from the stoichiometric matrix, S. For a reaction system with N species and M reactions, $S \in \mathbb{Z}^{N\times M}$ whose $k-$th column is the stiochiometric vector of the reaction R_k, ν_k, $k = 1,2,\ldots,M$. If there are conserved cycles in the system, this means that S is not full rank, i.e., $W = Rank\{S\} < N$ [11], [15], [16]. Although there are different ways of obtaining conservation relations from S, algorithm proposed in the present study is based on the Gauss-Jordan method. We refer to [11], for more details on Gauss Jordan method and also details of different methods for obtaining conserved cycles from S. Let $\widetilde{S} \in \mathbb{N}^{W\times M}$ be the row reduced echelon form of S. $\widetilde{S}$ can be obtained by multiplying S with a matrix $U \in \mathbb{R}^{N\times N}$ which is the product of elementary matrices such that $US = \begin{bmatrix} \widetilde{S} \\ 0 \end{bmatrix}$.

If we partition U into two different matrices $\widetilde{U} \in \mathbb{N}^{W\times M}$, $\overline{U} \in \mathbb{N}^{(N-W)\times M}$ such that

$$\widetilde{U}S = \widetilde{S}, \quad \overline{U}S = 0.$$

Here, $\overline{U}$ is called as the conservation matrix. The matrix, U, also partitions the state vector, X, into two different parts as follows

$$\widetilde{U}X = \widetilde{X}, \quad \overline{U}X = \overline{X} = C$$

where C denotes the conservation constant vector. For the rest of the paper, $\widetilde{X}$, $\overline{X}$ will be referred as independent and dependent variables, respectively. By using conservation relations, we can separate the system into two parts. If we use differential equations to model the system, the time derivative of the independent variables, $\widetilde{X}$, are represented by differential equations while the abundances of some species are represented by the algebraic relations given in $\overline{X}$. Each component of $\overline{X}$ corresponds a conserved cycle in the system and the value of conservation vector can be found from the initial state vector, $X(0)$ by using the equation $\overline{U}X(0) = C$.

Given the principled way of obtaining $\widetilde{X}$, $\overline{X}$ values, we must obtain the original state vector X. To achieve this goal, we will use the properties of U. It is trivial that U has an inverse U^{-1}. Then, if we partition U^{-1} into two matrices which are $\widetilde{U}^{-1} \in \mathbb{R}^{N\times W}$, $\overline{U}^{-1} \in \mathbb{R}^{N\times(N-W)}$, then we can easily obtain

$$X = \widetilde{U}^{-1}\widetilde{X} + \overline{U}^{-1}C.$$

Now, based on Gauss Jordan method we can give the following two algorithms.

Algorithm 1: Algorithm that obtains conserved cycles of a given biochemical reaction system

Input : The stoichiometric matrix S, rank of matrix W

Output: Independent and dependent variables

1. $T = [S\ eye(N)]$ (Obtain augmented matrix $T \equiv [S, I] \in \mathbb{Z}^{M\times 2N}$ where $I \in \mathbb{N}^{N\times N}$ is the identity matrix)
2. $\widetilde{T} = rref(T)$ (Obtain row reduced echelon form of T which is represented by $\widetilde{T}$)
3. $U = \widetilde{T}(:, M+1:2N)$ (Obtain U from $\widetilde{T}$ by taking last $(2N-M-1)$ columns of $\widetilde{T}$)
4. $\overline{U} = U(1:W,:)$, $\widetilde{U} = U(W+1:M,:)$ (Construct $\overline{U}$, $\widetilde{U}$)
5. $U^{-1} = inv(U)$ (Obtain inverse of U which is U^{-1})
6. $\widetilde{U}^{-1} = U(:,1:W)$, $\overline{U}^{-1} = U(:,W+1:N)$ (Obtain inverse of $\widetilde{U}^{-1}$ which is $\overline{U}^{-1}$)
7. $\widetilde{X} = \widetilde{U}X$, $\overline{X} = \overline{U}X$ (Obtain $\widetilde{X}$, $\overline{X}$)

This main algorithm can be used in any simulation based numerical method which obtains realizations of biochemical reactions for stochastic modeling. The following algorithm is the summary of any algorithm that uses Algorithm 1.

Algorithm 2: Summary of algorithms that uses Algorithm 1.

Input : The stoichiometric vectors $\nu_j, j = 1,2,\ldots,M, \widetilde{U}, \widetilde{X}, \widetilde{U}^{-1}, \overline{U}, \overline{X}, \overline{U}^{-1}$

Output: The state of the system and the time of the process.

1. Set $t = 0$, $X = x_0$.
2. Obtain t_μ^* depending on the algorithm under consideration.
3. Define a new stiochiometric vector $\nu_j^{new} = U\nu_j, j = 1,2,\ldots,M.$
4. Update $\widetilde{X} = \widetilde{X} + \nu_j^{new}[1:W]$.
5. Obtain original state vector $X = \widetilde{U}^{-1}\widetilde{X} + \overline{U}^{-1}C$.
6. Update t depending on the type of the algorithm under consideration.
7. Compute the propensities of functions.
8. Go to Step 2.

In this study, we compare computational costs of three exact algorithms DM, FRM and NRM with/without using conservation relations. All algorithms are written in MATLAB programming language. This comparison will be based on three quantities which are CPU time refers to the time used by MATLAB for the algorithm from the time it was started and average search depth, average weighted degree. Average search depth and average weighted degree are defined in [23] as follows.

Average Search Depth S : This term denotes the average number of operations needed to obtain the index of the next reaction. It is computed as follows $S = \sum_{j=1}^{M} jk_j \Big/ \sum_{j=1}^{M} k_j$, where M is the number of reactions in the system, k_j is the total number of occurrence of reaction R_j in the simulation time of interest.

Average Weighted Degree D This quantity can be considered as a kind of measure of the dependency graph. It shows the average number of reactions whose propensities changed by an occurrence of a given reaction. It is computed by $\mathcal{D} = \sum_{j=1}^{M} d_j k_j \Big/ \sum_{j=1}^{M} k_j$ where d_j represents the total number of reactions whose propensities are changed by a single firing of reaction R_j which can be obtained from dependency graph J.

4.APPLICATIONS

In this section of the present study, we will apply the proposed algorithms into Gene-expression model, Michaelis- Menten kinetics and Phosphorelay system which involves one, two, three conserved cycles, respectively.

4.1. Gene Expression

Gene expression is a very well known process which begins with activation of genes to produce a specific protein. Active gene, $\mathrm{Gene}_{\mathrm{on}}$, forms mRNA by transcription step and mRNA produces protein via translation. Then, produced protein and mRNA are consumed. The state vector of the system is $\mathrm{X(t)} = (\mathrm{Gene}_{\mathrm{on}}, \mathrm{Gene}_{\mathrm{off}}, \mathrm{mRNA}, \mathrm{Protein})^T$. Reactions in the system, propensity functions and stoichiometric vectors can be seen in Table 1. Copy numbers of $\mathrm{Gene}_{\mathrm{on}}$, mRNA, $\mathrm{Protein}$ produced by the DM are shown in Figure 1. We have used and as follows

Table 1. Reactions, propensity functions and stoichiometric vector for gene expression model.

Reaction	Propensity Function	Stoichiometric vector
$R_1 : \mathrm{Gene}_{\mathrm{on}} \xrightarrow{c_1} \mathrm{Gene}_{\mathrm{off}}$	$\alpha_1(x) = c_1 x_1$	$\nu_1 = (-1,1,0,0)^T$
$R_2 : \mathrm{Gene}_{\mathrm{off}} \xrightarrow{c_2} \mathrm{Gene}_{\mathrm{on}}$	$\alpha_2(x) = c_2 x_2$	$\nu_2 = (1,-1,0,0)^T$
$R_3 : \mathrm{Gene}_{\mathrm{on}} \xrightarrow{c_3} \mathrm{Gene}_{\mathrm{on}} + \mathrm{mRNA}$	$\alpha_3(x) = c_3 x_1$	$\nu_3 = (0,0,1,0)^T$
$R_4 : \mathrm{mRNA} \xrightarrow{c_4} \mathrm{mRNA} + \mathrm{Protein}$	$\alpha_4(x) = c_4 x_3$	$\nu_4 = (0,0,0,1)^T$
$R_5 : \mathrm{mRNA} \xrightarrow{c_5} \emptyset$	$\alpha_5(x) = c_5 x_3$	$\nu_5 = (0,0,-1,0)^T$
$R_6 : \mathrm{Protein} \xrightarrow{c_6} \emptyset$	$\alpha_6(x) = c_6 x_4$	$\nu_6 = (0,0,0,-1)^T$

$$S = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{pmatrix}, \quad U = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Since $Rank\{S\} = 3$, we have only one conserved cycle which is $\mathrm{Gene}_{\mathrm{on}} + \mathrm{Gene}_{\mathrm{off}} = C$. Then, independent and dependent variables have the form

$\widetilde{\mathrm{X}}(\mathrm{t}) = (\mathrm{Gene}_{\mathrm{on}}, \mathrm{mRNA}, \mathrm{Protein})^T, \overline{\mathrm{X}} = \mathrm{Gene}_{\mathrm{on}} + \mathrm{Gene}_{\mathrm{off}}$.

In our application, we have $X(0)=(0,1,20,80)^T$, which gives $\text{Gene}_{\text{on}}+\text{Gene}_{\text{off}}=1$ and $c_1=0.1s^{-1}, c_2=0.1s^{-1}$, $c_3=0.5s^{-1}, c_4=2s^{-1}$, $c_5=0.025s^{-1}$ and.The system is simulated in time interval $t\in[0,150]s$. Comparison of algorithms with/ without using conserved cycles can be seen in Table 4.

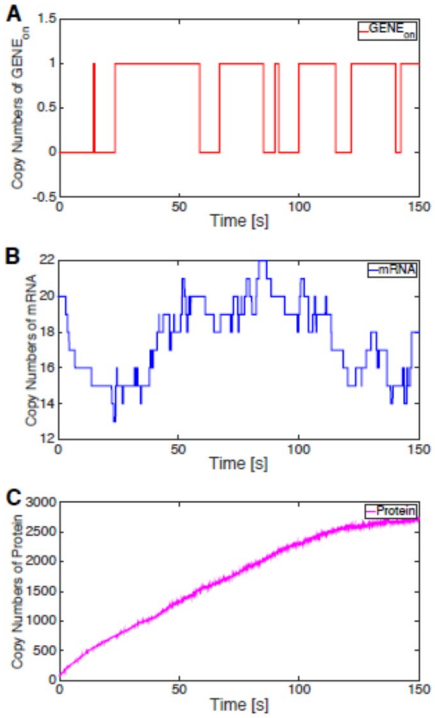


Fig 1. Copy numbers of Gene_{on} (A), mRNA (B), Protein (C) in gene expression model given in Table I. In our illustration, we use $X(0)=(0,1,20,80)^T$, $c_1=0.1s^{-1}$, $c_2=0.1s^{-1}, c_3=0.5s^{-1}$, $c_4=2s^{-1}$, $c_5=0.025s^{-1}$ and $c_6=0.01s^{-1}$ and obtain realizations via DM.

4.2. Michaelis-Menten Kinetics

Michaleis-Menten kinetics is a fundamental enzymatic mechanism that is valid for reactions involving only a single substrate. The enzyme (E) binds to the substrate (S) to form a enzyme-substrate (ES) which in turn forms protein (P). The state vector of the system in our model is $X(t)=(E,S,ES,P)^T$. Reactions, propensity functions and also stoichiometric vectors of the system can be seen in the following Table 2.

Table 2. Reactions, propensity functions and stoichiometric vectors for the Michaelis-Menten Kinetics.

Reaction	Propensity Function	Stoichiometric vector
$R_1: E+S \xrightarrow{c_1} ES$	$\alpha_1(x)=c_1x_1x_2$	$\nu_1=(-1,-1,1,0)^T$
$R_2: ES \xrightarrow{c_2} E+S$	$\alpha_2(x)=c_2x_3$	$\nu_2=(1,1,-1,0)^T$
$R_3: ES \xrightarrow{c_3} E+P$	$\alpha_3(x)=c_3x_3$	$\nu_3=(1,0,-1,1)^T$

The stiochiometric matrix S and U has the form

$$S=\begin{pmatrix}-1&1&1\\-1&1&0\\1&-1&-1\\0&0&1\end{pmatrix},\quad U=\begin{pmatrix}0&0&1&1\\0&0&0&1\\1&0&1&0\\0&1&1&1\end{pmatrix}$$

The rank of S is 2. Then, we have two conserved cycles

$$E+ES=C_1, S+ES+P=C_2.$$

Our new independent and dependent variables have the form

$$\tilde{X}=(ES+P,P)^T, \overline{X}=(E+ES,E+ES+P)^T.$$

In our application, we have $X(0)=(65,695,5,0)^T$ which in turn will produce $C=(70,700)^T$, reaction rate constants are $c_1=0.02, c_2=0.1, c_3=0.5$ and the system is simulated in $t\in[0,30]$. Realizations of the system generated by DM with these values can be seen in the left panel of Figure 2. Computational costs of two different versions of each algorithm can be seen in the Table 4.

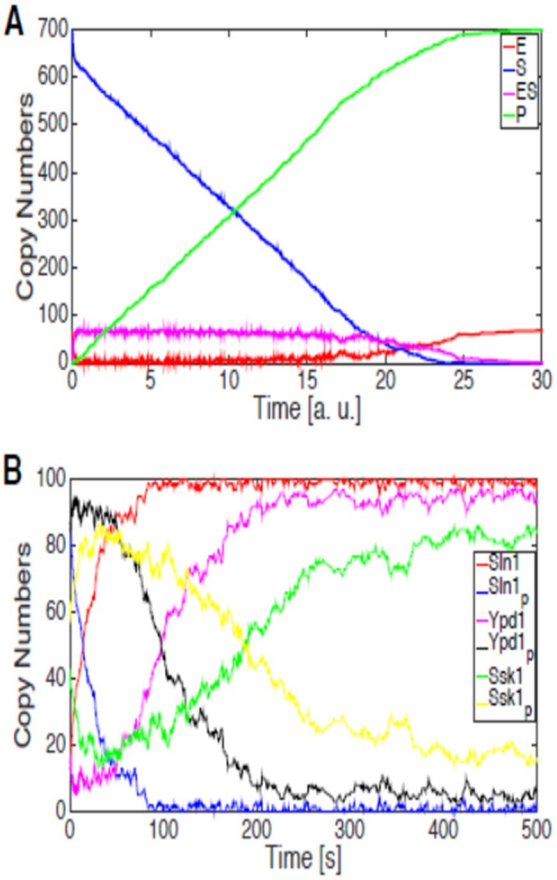


Fig. 2. (A) Copy numbers of species in Michaelis-Menten model which is obtained by using $X(0)=(65,695,5,0)^T$ and reaction constants $c_1=0.02, c_2=0.1, c_3=0.5$ (B) Dynamics of species of Phosphorelay System with $X(0)=(15,85,13,87,40,60)^T$ $c_1=0.004s^{-1}$, $c_2=0.005 molec^{-1}s^{-1}$, $c_3=0.001$ $molec^{-1}s^{-1}$, $c_4=0.02s^{-1}$

4.3. Phosphorelay System

Our third biochemical reaction system is phosphorelay system which keeps HOG signalling pathway inactive when external osmolarity of the system is normal [12]. The reaction system involves phosphorylation and dephosphorylation of three components $Sln1$, $Ypd1$, $Ssk1$. Phosphorylated $Sln1$, $Sln1_p$, binds $Ypd1$ to phosphorylate $Ypd1$, $Ypd1_p$, and $Ypd1_p$ binds $Ssk1$ to form $Ssk1_p$. The state vector of the system is $X(t)=(Sln1, Sln1_p, Ypd1, Ypd1_p, Ssk1, Ssk1_P)^T$. The change of the abundances of species can be seen in the right panel of Figure 2. Details of the model can be seen in Table 3.

Table 3. Reactions, propensity functions and stoichiometric vectors for the Phosphorelay System.

Reaction	Propensity Function	Stoichiometric vector
$R_1: Sln1 \xrightarrow{c_1} Sln1_p$	$\alpha_1(x)=c_1x_1x$	$\nu_1=(-1,1,0,0,0,0)$
$R_2: Sln1_p + Ypd1 \xrightarrow{c_2} Ypd1_p + Sl$	$\alpha_2(x)=c_2x_2x_3$	$\nu_2=(1,-1,-1,1,0,0)^T$
$R_3: Ypd1_p + Ssk1 \xrightarrow{c_3} Ssk1_p + Y$	$\alpha_3(x)=c_3x_4x_5$	$\nu_3=(0,0,1,-1,-1,1)^T$
$R_4: Ssk1_p \xrightarrow{c_4} Ssk1$	$\alpha_4(x)=c_4x_6$	$\nu_4=(0,0,0,0,1,-1)^T$

The stiochiometric matrix S and U has the form

$$S=\begin{pmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}, \quad U=\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Then, we have

$$\tilde{X}=(Sln1_p+Ypd1_p+Ssk1_p, Ypd1_p+Ssk1_p, Ssk1_p)^T$$
$$\bar{X}=(Sln1+Sln1_p, Ypd1+Ypd1_p, Ssk1+Ssk1_p)^T.$$

In our application, we have $X(0)=(15,85,13,87,40,60)^T$ which gives the conservation constant vector as $C=(100,100,100)^T$ and reaction constants $c_1=0.004s^{-1}, c_2=0.005 molec^{-1}s^{-1}, c_3=0.001$ $molec^{-1}s^{-1}$, $c_4=0.02s^{-1}$. We simulate the system in $t\in[0,500]s$.

Table 4. Comparison of the CPU algorithms. For both versions of each algorithm, we have used the same random numbers.

System	Algorithm	Conditions	CPU	Average Search Dept S	Average Weighted Degree D
Gene Expression	DM	Original	0.9600	4.6537	1.0159
	DM	Conserved	0.9500	4.6506	1.0205
	FRM	Original	0.9800	4.6983	1.0200
	FRM	Conserved	1.1400	4.6586	1.0174
	NRM	Original	0.4200	4.7999	1.0072
	NRM	Conserved	0.3800	4.8319	1.0092
Michaelis-Menten Kinetics	DM	Original	0.2100	1.9223	3
	DM	Conserved	0.2900	1.9210	3
	FRM	Original	0.2400	1.9210	3
	FRM	Conserved	0.2800	1.9111	3
	NRM	Original	0.1400	1.9360	3
	NRM	Conserved	0.1000	1.8827	3
Phosphorelay Saytem	DM	Original	0.1700	2.7841	2.5161
	DM	Conserved	0.2000	2.7656	2.4587
	FRM	Original	0.2700	2.8054	2.5159
	FRM	Conserved	0.1800	2.7867	2.5200
	NRM	Original	0.1400	2.8556	2.4623
	NRM	Conserved	0.1700	2.8982	2.4587

5. APPENDIX

The steps of Direct Method, First Reaction Method and Next Reaction Method can be seen as follows.

Algorithm 3: Direct Method.

1. Set $t=0$, the initial state $X=x_0$, set stoichiometric vector $\nu_j, j=1,2,\ldots,M.$
2. Compute the propensity function $\alpha_j(X)$ and calculate $\alpha_0(X)=\sum_{j=1}^{M}\alpha_j(X)$
3. Draw two random numbers r_1, r_2 from U (0,1) which represents the set of uniformly distributed numbers on [0,1].
4. Calculate the firing time of the next reaction $t^*=\frac{1}{\alpha_0(X)}\log(\frac{1}{r_1}).$
5. Calculate the firing time of the next reaction μ such that $\sum_{k=1}^{\mu-1}\alpha_k(X)<r_2\alpha_0(X)\le\sum_{k=1}^{\mu}\alpha_k(X)$
6. Update $t\to t+t^*, X\to X+\nu_\mu.$
7. Go to Step 2.

Algorithm 4: First Reaction Method.

1. Set $t=0$, the initial state $X=x_0$, set stoichiometric vector $\nu_j, j=1,2,\ldots,M.$
2. Compute the propensity function $\alpha_j(X)$ and calculate $\alpha_0(X)=\sum_{j=1}^{M}\alpha_j(X)$
3. Draw M random numbers $r_1, r_2, \ldots, r_M$ from U (0,1) .
4. Compute $t_j^*=\frac{1}{\alpha_j(X)}\log(\frac{1}{r_j}), j=1,2,\ldots,M.$
5. Obtain
   $t^* \equiv$ the smallest of $\{t_1^*, t_2^*, \ldots, t_M^*\}$,
   $\mu \equiv$ the index of the smallest of $\{t_1^*, t_2^*, \ldots, t_M^*\}$.

   Update $t\to t+t^*, X\to X+\nu_\mu.$
6. Go to Step 2.

Algorithm 5: Next Reaction Method.

1. Set $t=0$, the initial state $X=x_0$, set stoichiometric vector $\nu_j, j=1,2,\ldots,M$, generate dependency graph *J.*
2. Compute the propensity function $\alpha_j(X)$ and calculate $\alpha_0(X)=\sum_{j=1}^{M}\alpha_j(X)$
3. Draw M random numbers $r_1, r_2, \ldots, r_M$ from U (0,1) .
4. Compute $t_j^*=\frac{1}{\alpha_j(X)}\log(\frac{1}{r_j}), j=1,2,\ldots,M.$
5. Store t_j^* values in the indexed priority queue Q.
6. Obtain t^* such that it is the smallest value stored in the indexed priority queue Q.
7. Obtain the index of the next reaction μ such that $t^*=t_\mu^*.$
8. Update $t\to t+t^*, X\to X+\nu_\mu.$
9. Based on the dependency graph *J,* for each reaction R_k whose propensity is affected from the occurrence of R_μ
   - Update $\alpha_{k,new}(X)=\alpha_k(X), k\neq\mu.$
   - $t_\mu^*=t+(\frac{\alpha_{k,old}(X)}{\alpha_{k,new}(X)})(t_k^*-t), k\neq\mu.$
   - If $k=\mu, t_\mu^*=t+\frac{1}{\alpha_\mu(X)}\log(\frac{1}{r}), r\in$ U (0,1)

- Replace the old t_k^* values in Q with the new values.

10. Go to Step 2.

6. DISCUSSION

In the present paper, we give an algorithm to obtain conservation relations in a biochemical system by using Gauss- Jordan method given in [11]. The algorithm can be used to convert differential equations representing the dynamics of biochemical reaction networks into differential algebraic forms. In [18], the algorithm is used to convert SDEs to SDAEs and involved in computer based simulation algorithm of jump diffusion approximation. In this paper, we used proposed algorithm to improve DM, FRM, NRM such that these new versions of algorithms can obtain independent variables, conservation constants and original state vectors whose definitions can be found in Section III. Computational costs of those algorithms in case of involving/ not involving conservation relations are compared based on different criteria. All algorithms are written in MATLAB version R2014b and applied to three different reaction systems. For all algorithms initial random numbers are kept fixed, other random numbers are drawn during the simulation. It can be seen in Table IV, there is no appreciable change in CPU values, average search depth, average weighted degree for two different versions of DM, NRM, FRM. The CPU time differences between the algorithms are the result of time increments which are obtained by using random numbers generated during the process. It must be taken into account the fact that transforming the independent variables into original state vectors is an important factor which increases the computational cost of it. Fixing all random variables to see the net effect of this step in the computational cost will also be studied. Another open question of this study is how conservation cycles can be involved in original CME. In other words, conserved cycles reduces ODEs into DAEs, SDEs ito SDAEs. So, what is the correspondence of DAEs, SDAEs when CME is used to model the system. We hope this paper will be an initial step for researchers from different areas studying on systems involving conserved cycles.

REFERENCES

[1] A. Kremling, J. Saez-Rodriguez, Systems biologyan engineering perspective, Journal of Biotechnology 129 (2007) 329–351.

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

[3] N. G. v. Kampen, Stochastic processes in physics and chemistry, Amsterdam; New York : North-Holland ; New York : sole distributors for the USA and Canada, Elsevier North-Holland, 1981.

[4] D. Wilkinson, Stochastic modelling for systems biology, Chapman & Hall/CRC mathematical and computational biology series, Boca Raton, FL : Taylor & Francis, 2006.

[5] D. Anderson, T. Kurtz, Continuous time markov chain models for chemical reaction networks, in: H. Koeppl, G. Setti, M. d. Bernardo, D. Densmore (Eds.), Design and Analysis of Biomolecular Circuits, Springer-Verlag, 2011.

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

[7] F. Cordero, A. Horv´ath, D. Manini, L. Napione, M. De Pierro, S. Pavan, A. Picco, A. Veglio, M. Sereno, F. Bussolino, G. Balbo, Simplification of a complex signal transduction model using invariants and flow equivalent servers, Theoretical Computer Science 412 (2011), 6036–6057.

[8] T. R. Maarleveld, R. A. Khandelwal, B. G. Olivier, B. Teusink, F. J. Bruggeman, Basic concepts and principles of stoichiometric modeling of metabolic networks, Biotechnol. J. 8 (2013) 997–1008.

[9] D. W. Schryer, M. Vendelin, P. Peterson, Symbolic flux analysis for genome-scale metabolic networks, BMC Systems Biology 5 (81).

[10] R. Winkler, Stochastic differential algebraic equations in transient noise analysis, in: Scientific Computing in Electrical Engineering, Mathematics in Industry, Springer, 2004, pp. 151–158.

[11] H. Sauro, B. Ingalls, Conservation analysis in biochemical networks computational issues for software writers, Biophysical Chemistry 109 (2004) 1–15.

[12] E. Klipp, W. Liebermeister, C. Wierling, A. Kowald, H. Lehrach, R. Herwig, Systems Biology: A Textbook, WILEY - VCH Verlag GmbH & Co. KGaA, 2009.

[13] R. R. Vallabhajosyula, V. Chickarmane, H. M. Sauro, Conservation analysis of large biochemical networks, Bioinformatics 22 (3) (2006) 346–353.

[14] H. S. Haraldsdottir, R. M. T. Fleming, Identification of conserved moieties in metabolic networks by graph theoretical analysis of atom transition networks, arXiv (2016). arXiv:arXiv:1605.05371.

[15] S. Schuster, T. Hofer, Determining all extreme semi-positive conservation relations in chemical reaction systems: A test criterion for conservativity, J. Chem. Soc. Faraday Trans. 87 (16) (1991) 2561–2566.

[16] A. Cornish-Bowden, J.-H. Hofmeyr, The role of stoichiometric analysis in studies of metabolism: An example, Journal of Theoretical Biology 216 (2002) 179–191.

[17] R. Winkler, Stochastic differential algebraic equations of index 1 and applications in circuit simulation, Journal of Computational and Applied Mathematics 157 (2003) 477–505.

[18] A. Ganguly, D. Altıntan, H. Koeppl, Jump-diffusion approximation of stochastic reaction dynamics: Error bounds and algorithms, Multiscale Model. Simul. 13 (4) (2015) 1390–1419.

[19] D. Gillespie, The chemical langevin equation, Journal of Chemical Physics 113 (1) (2000) 297–306.

[20] T. Jahnke, W. Huisinga, Solving the chemical master equation for monomolecular reaction systems analytically, J. Math. Biol. 54 (2007) 1–26.

[21] D. T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, J. Comput. Phys. 22 (1976) 403–434.

[22] M. Gibson, J. Bruck, Efficient exact stochastic simulation of chemical systems with many species and many channels, The Journal of Physical Chemistry A 104 (9) (2000) 1876–1889.

[23] Y. Cao, H. Li, L. Petzold, Efficient formulation of the stochastic simulation algorithm for chemically reacting systems, J. Chem. Phys 121 (9) (2004) 4059–4067.