Quasi Product Form Approximation for Markov Models of Reaction Networks

Alessio Angius[1], András Horváth[1], and Verena Wolf [2]

[1] Department of Computer Science, University of Torino, Torino, Italy
{angius,horvath}@di.unito.it
[2] Department of Computer Science, Saarland University, Saarbrücken, Germany
wolf@cs.uni-saarland.de

Abstract. In cell processes, such as gene regulation or cell differentiation, stochasticity often plays a crucial role. Quantitative analysis of stochastic models of the underlying chemical reaction network can be obstructed by the size of the state space which grows exponentially with the number of considered species. In a recent paper [1] we showed that the space complexity of the analysis can be drastically decreased by assuming that the transient probabilities of the model are in *product form*. This assumption, however, leads to approximations that are satisfactory only for a limited range of models. In this paper we relax the product form assumption by introducing the *quasi product form* assumption. This leads to an algorithm whose memory complexity is still reasonably low and provides a good approximation of the transient probabilities for a wide range of models. We discuss the characteristics of this algorithm and illustrate its application on several reaction networks.

1 Introduction

Most mathematical models assume that cell processes are deterministic [15]. In recent years, however, significant experimental evidence has shown that these processes involve stochastic fluctuation which is not captured by deterministic models. Some of the earliest works reporting on the role of stochasticity are: [3] where the authors show that a stochastic event has a crucial impact on mammalian cell differentiation; [2] where the authors state that "conventional deterministic kinetics cannot be used to predict statistics of regulatory systems that produce probabilistic outcomes"; and [8] where the importance of stochastic delays of initiation or interruptions of gene expression is revealed.

The first algorithm to analyse stochastic models of network of reactions was proposed by Gillespie who considered general chemical reaction systems [12,13]. The Gillespie algorithm provides a trajectory of the reaction network by a simulation whose underlying model is a discrete state, continuous time Markov chain (CTMC). This means that, at least in principle, the analysis of such networks can be carried out by constructing the infinitesimal generator matrix of the CTMC and computing its exponential [25]. In general, determining the exponential of a matrix can be problematic (see [22] where 19 different approaches

C. Priami et al. (Eds.): Trans. on Comput. Syst. Biol. XIV, LNBI 7625, pp. 26–52, 2012.

are listed and compared) but to matrices corresponding to a CTMC the numerically stable and efficient randomization (called also uniformization) approach can be applied [17,25]. However, even randomization can fail if the number of states of the CTMC is very large or infinite. And this is most often the case as each species adds one "dimension" to the state space which, consequently, grows exponentially with the number of species. This phenomenon is known as state space explosion and in this paper we propose a method to alleviate this problem by assuming that the transient probabilities can be approximated in a compact manner based on quasi product forms.

Related Work. A natural idea to circumvent the state space explosion problem is to develop approximate analysis techniques. One family of approximations is based on the relation of the trajectories of the CTMC and the trajectory determined by the deterministic, differential equation-based description of the system [18,19]. The simplest such approximation is the mean-field approach which provides a deterministic trajectory of the system behaviour. This deterministic trajectory can be seen as the approximate average behaviour of the model. Higher order moment closure techniques can provide an approximation not only for the mean but for higher order moments and joint moments as well [11,24].

Depending on the measure of interest, it may be necessary to maintain the state space of the model (for example, when calculating extinction probabilities). In this case, in order to decrease the state space, a straightforward approach is to bound the set of states that are considered [10]. This, as the system evolves, must be done in a dynamic manner in order to take into account at any transient time those states that have the largest probability. As the set of states to consider can remain huge, recently, faster and approximate randomization methods have been proposed in [21,27]. Also aggregation techniques can be used to face the state space explosion problem. Proposals in this directions are presented in [26,7] where nearby states are aggregated and in [6,9] where the concept of flow equivalence in applied.

Simulation remains the most widely used approach to analyse large or even infinite Markov chains. Since the state space is huge and the frequency with which transitions occur can be very high, not even simulation is easy to carry out. Beginning with [12], numerous papers proposed approaches to increase the efficiency of simulation of reaction systems. These approaches include explicit [14] and implicit [23] tau-leaping, which uses an approximation to consider many reactions in a single step, and the slow-scale stochastic simulation algorithm [5], which aims at facing stiffness of the dynamics of the model by distinguishing fast and slow reactions.

Contribution of the Paper. In [1] we proposed an approximation technique which operates on the original state space of the model (i.e., no reduction or aggregation steps are performed) and is based on the assumption that the transient probabilities of the model can be written in product form. This assumption leads to a highly compact description of the transient probabilities. Indeed, the space complexity of the computations grows only linearly with the number of

species while the growth is exponential when randomization is applied. In [1] we showed that the method can be applied to huge state spaces and we tested it on several reaction networks. It turned out that the method results in a good approximation if the reaction network resembles a network of M/M/∞ queues (i.e., queues with Poisson arrivals, exponential service time and infinite number of servers). This is because transient probabilities in such a network are in fact in product form [4]. On the other hand, for reaction networks in general the approximations can be rather poor.

In this paper we advance the technique proposed in [1] by relaxing the assumption that the transient probabilities are in product form. The relaxed assumption, which we call *quasi product form* assumption, leads to a computational method

- whose space complexity is lower than that of performing randomization,
- that results in a good approximation for a wide range of reaction networks.

The paper is organised as follows. In Section 2 the stochastic model associated with the network of reactions is described. In Section 3 we introduce the quasi product form approximation. An algorithm to implement the procedure based on the quasi product form assumption is discussed in Section 4. Application of the algorithm is illustrated in Section 5. In Section 6 a preliminary error validation approach is discussed. Conclusions are drawn in Section 7.

2 Stochastic Approach

We consider a system having a set of species $\mathcal{M} = \{S_1, S_2, ..., S_M\}$, interacting through N reactions:

$$\sum_{m=1}^{M} a_{nm} S_m \xrightarrow{\lambda_n} \sum_{m=1}^{M} b_{nm} S_m, \quad 1 \leq n \leq N. \tag{1}$$

The nth reaction uses up a_{nm} units of species S_m and produces b_{nm} units of it. Both a_{nm} and b_{nm} are non-negative integer values and will be organised into vectors as $a_n = (a_{n1}, ..., a_{nM})$ and $b_n = (b_{n1}, ..., b_{nM})$. We will denote by $c_{nm} = b_{nm} - a_{nm}$ the overall effect of reaction n on species S_m and the corresponding vector will be denoted by $c_n = (c_{n1}, ..., c_{nM})$. The speed of the nth reaction is determined by $\lambda_n \in \mathbb{R}^+$, also called reaction rate constant.

There are different approaches to associate a temporal behaviour with the reactions in (1). Here we focus on the well-established stochastic approach that associates a continuous time Markov chain (CTMC) $\{X(t), t \geq 0\}$ with the system [12]. The CTMC is discrete state, i.e., the quantity of a given species at any time t is given by an integer. Therefore, the state at time t is given by a vector of integers as $X(t) = (X_1(t), ..., X_M(t))$. In order to shorten the notation, in the rest of the paper we will omit the dependence on time and write X instead of $X(t)$. Reaction n is possible in a given state $x = (x_1, ..., x_M)$ if $x_m \geq a_{nm}, 1 \leq m \leq M$. We will apply the relation $\geq$ to vectors meaning that

$x \geq a_n$ if and only if $x_m \geq a_{nm}, 1 \leq m \leq M$. If reaction n is possible in state x then its transition rate, denoted by $\alpha_n(x)$, is given as

$$\alpha_n(x) = \lambda_n \prod_{m=1}^{M} \binom{x_m}{a_{nm}} \tag{2}$$

i.e., it depends on the reaction rate constant and the number of ways in which the involved molecules can react. The occurrence of reaction n changes the state of the CTMC from state x to state $x + c_n$. The probability that the system is in a given state x at time t, denoted by $Pr\{X = x\}$, satisfies the following well-known Chapman-Kolmogorov ordinary differential equation (ODE) (see, for example, [25])

$$\begin{aligned}\frac{dPr\{X = x\}}{dt} = &- Pr\{X = x\} \sum_{n:x \geq a_n} \alpha_n(x) + \\ &\sum_{n:x-c_n \geq a_n} Pr\{X = x - c_n\}\alpha_n(x - c_n)\end{aligned} \tag{3}$$

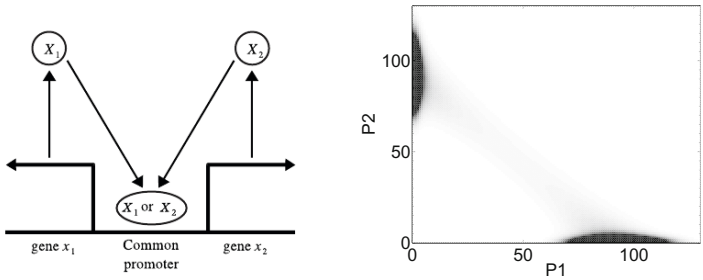
where the first term considers the transitions leaving state x while the second term the transitions leading to x. The ODE given in (3) is also known as the *chemical master equation*.

In the following we provide an example for a system of reactions. Throughout the paper, species of concrete examples will be denoted by symbols refering to the characteristics of the species. Instead, when reaction networks in general are considered we will use the general symbols introduced so far.

Example 1. We consider a gene regulatory network, called exclusive switch [20], that consists of two genes with overlapping promoter regions. Each of the two gene products, P_1 and P_2, inhibits the expression of the other if a molecule is bound to the promoter region of the DNA (called simply *Dna* in the following). In other words, if the *Dna* is bound to a molecule of P_1 (P_2) only molecules of type P_1 (P_2) can be produced, and if the *Dna* is free both types of proteins are produced. An illustration of the exclusive switch is depicted in Figure 1(a).

The model involves five species, namely *Dna*, $Dna.P_1$, $Dna.P_2$, P_1, P_2 where the "dot" means that the *Dna* is bound to $P1$ (P_2). Thus, a state x is a vector of five non-negative integers, $(x_1, x_2, x_3, x_4, x_5)$, with the species ordered as above. The species interact through ten reactions:

- $Dna \longrightarrow Dna + P_1$ models production of P_1 in case of free promoter region with $a_1 = (1, 0, 0, 0, 0)$, $b_1 = (1, 0, 0, 1, 0)$, $c_1 = (0, 0, 0, 1, 0)$ and transition rate $\alpha_1(x) = \lambda_1 \cdot x_1$,
- $Dna \longrightarrow Dna + P_2$ models production of P_2 in case of free promoter region with $a_2 = (1, 0, 0, 0, 0)$, $b_2 = (1, 0, 0, 0, 1)$, $c_2 = (0, 0, 0, 0, 1)$ and transition rate $\alpha_2(x) = \lambda_2 \cdot x_1$,
- $P_1 \longrightarrow \emptyset$ describes the degradation of P_1 with $c_3 = (0, 0, 0, -1, 0)$ and $\alpha_3(x) = \lambda_3 \cdot x_4$,



(a) Illustration of the interactions (adapted from [20]).

(b) Bistable protein distribution.

Fig. 1. The exclusive switch

- $P_2 \longrightarrow \emptyset$ describes the degradation of P_2 with $c_4 = (0, 0, 0, 0, -1)$ and $\alpha_4(x) = \lambda_4 \cdot x_5$,
- $Dna + P_1 \longrightarrow Dna.P_1$ represents the binding of P_1 with $c_5 = (-1, 1, 0, -1, 0)$, and $\alpha_5(x) = \lambda_5 \cdot x_1 \cdot x_4$,
- $Dna + P_2 \longrightarrow Dna.P_2$ represents the binding of P_2 with $c_6 = (-1, 0, 1, 0, -1)$, and $\alpha_6(x) = \lambda_6 \cdot x_1 \cdot x_5$,
- $Dna.P_1 \longrightarrow Dna + P_1$ corresponds to the unbinding of P_1 with $c_7 = -c_5$ and $\alpha_7(x) = \lambda_7 \cdot x_2$,
- $Dna.P_2 \longrightarrow Dna + P_2$ corresponds to the unbinding of P_2 with $c_8 = -c_6$ and $\alpha_8(x) = \lambda_8 \cdot x_3$,
- $Dna.P_1 \longrightarrow Dna.P_1 + P_1$ models the production of P_1 when the promoter is occupied with $c_9 = c_1$ and $\alpha_9(x) = \lambda_9 \cdot x_2$,
- $Dna.P_2 \longrightarrow Dna.P_2 + P_2$ models the production of P_2 when the promoter is occupied with $c_{10} = c_2$ and $\alpha_{10}(x) = \lambda_{10} \cdot x_3$.

The initial state of the system is $(1, 0, 0, 0, 0)$. Due to the overlap of the promoters we have that only one molecule of the species Dna, $Dna.P_1$, $Dna.P_2$ can be present at a time leading to the invariant

$$Dna + Dna.P_1 + Dna.P_2 = 1$$

i.e., the possible values for (x_1, x_2, x_3) are $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$. Note that if the binding to the promoter is likely and the unbinding is rare then the distribution of P_1 and P_2 can become bistable as it is depicted in Figure 1(b). This happens in this setting because each gene can "monopolize" the promoter region increasing its population while molecules of the other population can only degrade.

Finally, in Figure 2 we provide the diagram representing the outgoing transitions of a generic state of the CTMC of the exclusive switch model. As all state variables must be non-negative, the transitions depicted in the figure are either possible or not depending on the actual value of $(x_1, x_2, x_3, x_4, x_5)$. Note that the state space is infinite due to the unboundedness of P_1 and P_2.

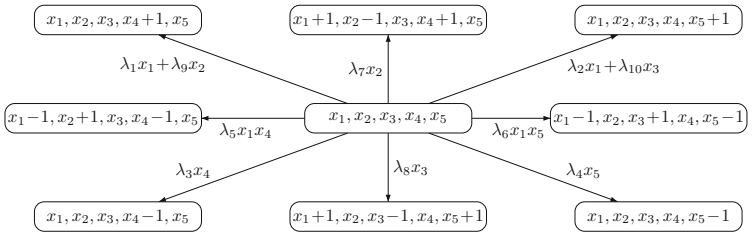


Fig. 2. Exclusive switch: Markov chain

The aim of the paper is to provide a memory efficient, approximate technique to analyse the transient behaviour of reaction networks. In case of the exclusive switch model, this means that we aim to approximate the transient probabilities

$$Pr\{Dna = x_1, Dna.P_1 = x_2, Dna.P_2 = x_3, P_1 = x_4, P_2 = x_5\}$$

i.e., the probability that the state of the system at time t is $(x_1, x_2, x_3, x_4, x_5)$.

3 Quasi Product Form Approximation

In [1] we presented an approximate analysis method for stochastic reaction networks which is based on the assumption that the transient probabilities are in product form, i.e.,

$$Pr\{X = x\} = \prod_{i=1}^{M} Pr\{X_i = x_i\} \tag{4}$$

This assumption leads to an algorithm for the computation of the transient probabilities whose space complexity is much lower than that of computing the transient probabilities by the classical and widely used randomization approach (see, for example, [25]). Since the transient probabilities in a network of M/M/∞ queues are in product form [4], the product form assumption leads to exact results for these networks. In [1] we showed that the approximation is satisfactory if the model resembles a network of M/M/∞ queues but can give imprecise results in other cases. In this paper we propose a more relaxed assumption that leads to a good approximation for a wider range of models. In particular, we will assume that there exist sets of species whose conditional probabilities depend only on a set of other species and not on all the rest of the species. For example, if we assume that the conditional probabilities of species 1 and 2 depend only on species 3, 4 and 5 then we can write

$$Pr\{X_1 = x_1, X_2 = x_2 \mid X_3 = x_3, X_4 = x_4, ..., X_M = x_M\} =$$
$$Pr\{X_1 = x_1, X_2 = x_2 \mid X_3 = x_3, X_4 = x_4, X_5 = x_5\}$$

A set of assumptions like the one above allows us to decompose the probability $Pr\{X_1 = x_1, X_2 = x_2, ..., X_M = x_M\}$ into a product. As this product is not in

the classical product form given in (4), we will refer to it as *quasi product form* and in the following we provide its formal description.

The quasi product form decomposition of the transient probabilities is conveniently described by a directed forest, denoted by $\mathcal{F}$. The set of the nodes of the forest is denoted by $\mathcal{V}$ and a given node, $v \in \mathcal{V}$, represents a subset of the species. The index set of the species represented by node v is denoted by $I(v)$. The set $\mathcal{V}$ must be such that it provides a partitioning of the set of species, i.e., $\cup_{v\in\mathcal{V}} I(v) = \{1, 2, ..., M\}$ and $\forall v_1, v_2 \in \mathcal{V}, v_1 \neq v_2 : I(v_1) \cap I(v_2) = \emptyset$. The set of edges of the forest, denoted by $\mathcal{E}$, provides the assumed dependency structure of the transient probabilities. Specifically, if $e = (u, v) \in \mathcal{E}$ then the conditional probability of the species in v *depends* on those species that are present in u. The set of species present in the predecessors of v will be denoted by $P(v)$, i.e., $P(v) = \cup_{u:(u,v)\in\mathcal{E}} I(u)$. The conditional probability of the species in $I(v)$ is independent of those species that are not present in $P(v)$, i.e.,

$$Pr\{\wedge_{i\in I(v)}(X_i = x_i) \mid \wedge_{j\in\{1,2,...,M\}\setminus I(v)}(X_j = x_j)\} = Pr\{\wedge_{i\in I(v)}(X_i = x_i) \mid \wedge_{j\in P(v)}(X_j = x_j)\}$$

where $\wedge$ denotes conjunction. By considering every node of the tree, the probability of a given state of the system, $(x_1, ..., x_M)$, can be written as

$$\begin{aligned} Pr\{\wedge_{i\in\{1,2,...,M\}}(X_i = x_i)\} = \\ \prod_{v\in\mathcal{V}} Pr\{\wedge_{i\in I(v)}(X_i = x_i) \mid \wedge_{j\in P(v)}(X_j = x_j)\} = \\ \prod_{v\in\mathcal{V}} \frac{Pr\{\wedge_{i\in Q(v)}(X_i = x_i)\}}{Pr\{\wedge_{j\in P(v)}(X_j = x_j)\}} \end{aligned} \tag{5}$$

where we applied the notation $Q(v) = I(v) \cup P(v)$. In the following we give two examples for the forest $\mathcal{F}$.

Example 2. For the exclusive switch the involved species, Dna, $Dna.P_1$, $Dna.P_2$, P_1 and P_2, can be partioned into three nodes, v_1, v_2 and v_3, such that node v_1 is associated with the species $Dna, Dna.P_1$ and $Dna.P_2$, node v_2 is associated with P_1 and node v_3 with P_2. The forest, composed of a single tree, is depicted in Figure 3. This leads to the following decomposition of the transient probabilities

$$\begin{aligned} Pr\{Dna = x_1, Dna.P_1 = x_2, Dna.P_2 = x_3, P_1 = x_4, P_2 = x_5\} = \\ Pr\{Dna = x_1, Dna.P_1 = x_2, Dna.P_2 = x_3\} \times \\ Pr\{P_1 = x_4 \mid Dna = x_1, Dna.P_1 = x_2, Dna.P_2 = x_3\} \times \\ Pr\{P_2 = x_5 \mid Dna = x_1, Dna.P_1 = x_2, Dna.P_2 = x_3\} \end{aligned}$$

Example 3. The assumption of complete product form would be expressed by a forest with M nodes, $v_1, ..., v_M$, such that $I(v_i) = \{i\}$, and an empty set of arcs, $\mathcal{E} = \emptyset$. With this forest the probabilities are in the form given in (4).

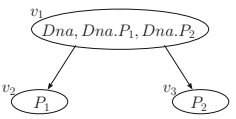


Fig. 3. The forest representing the assumed quasi product form structure for the exclusive switch

In order to compute the transient probabilities based on the quasi product form assumption expressed by the forest $\mathcal{F}$, we need the quantities appearing in (5). Since $P(v) \subseteq Q(v)$, the quantities in the denominator can be computed simply by appropriate summing of the quantities in the numerator. The quantities in the numerator can instead be computed by the differential equations provided by the following theorem.

Theorem 1. *If the transient probabilities satisfy the quasi product form decomposition expressed by the forest $\mathcal{F}$, then the following differential equation holds for all nodes $v \in \mathcal{V}$ and every possible values of $x_i, i \in Q(v)$:*

$$\frac{dPr\{\wedge_{i\in Q(v)}(X_i = x_i)\}}{dt} =$$

$$\sum_{\substack{(y_1,\ldots,y_M):\\ k\in Q(v), y_k = x_k}} \left(- \prod_{v\in\mathcal{V}} \frac{Pr\{\wedge_{i\in Q(v)}(X_i = y_i)\}}{Pr\{\wedge_{j\in P(v)}(X_j = y_j)\}} \sum_{n: y\geq a_n} \lambda_n \prod_{m=1}^{M} \binom{y_m}{a_{nm}} + \right.$$

$$\left. \sum_{n: y-c_n\geq a_n} \prod_{v\in\mathcal{V}} \frac{Pr\{\wedge_{i\in Q(v)}(X_i = y_i - c_{ni})\}}{Pr\{\wedge_{j\in P(v)}(X_j = y_j - c_{nj})\}} \lambda_n \prod_{m=1}^{M} \binom{y_m - c_{nm}}{a_{nm}} \right)$$

Proof. It is easy to see that we have

$$\frac{dPr\{\wedge_{i\in Q(v)}(X_i = x_i)\}}{dt} = \frac{d}{dt} \sum_{\substack{(y_1,\ldots,y_M):\\ k\in Q(v), y_k = x_k}} Pr\{\wedge_{i\in\{1,\ldots,M\}}(X_i = y_i)\} \quad (6)$$

where the order of the derivative and the summation can be exchanged. By applying the Chapman-Kolmogorov equations given in (3) and the quasi product form assumption given in (5) the theorem follows.

Note that on the right-hand side of the equation in Theorem 1, due to the presence of the binomial coefficients, we have quantities that are proportional to conditional factorial joint moments of the quantities of subsets of species.

In the following example we apply Theorem 1 to all nodes of the forest introduced in Example 2 and depicted in Figure 3 in order to provide the necessary differential equations for the exclusive switch.

Example 4. According to the partitioning given in Example 2, the species indicated by $Q(v_1)$ are Dna, $Dna.P_1$ and $Dna.P_2$ and the set of species given by $P(v_1)$ is the empty set. The possible values, (x_1, x_2, x_3), for these three species

are $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$. Let us first consider the case when the Dna is free, i.e., $(x_1,x_2,x_3)=(1,0,0)$. By following (6) and applying the Chapman-Kolmogorov equations we have

$$\begin{aligned}
&\frac{dPr\{Dna=1,Dna.P_1=0,Dna.P_2=0\}}{dt} = \sum_{x_4,x_5}\Bigg(\\
&-Pr\{Dna=1,Dna.P_1=0,Dna.P_2=0,P_1=x_4,P_2=x_5\}(\lambda_5 x_4+\lambda_6 x_5)+ \\
&\lambda_7 Pr\{Dna=0,Dna.P_1=1,Dna.P_2=0,P_1=x_4-1,P_2=x_5\}+ \\
&\lambda_8 Pr\{Dna=0,Dna.P_1=0,Dna.P_2=1,P_1=x_4,P_2=x_5-1\}\Bigg) \qquad (7)
\end{aligned}$$

where the first term in the summation of the right-hand side corresponds to binding of the Dna to P_1 (with speed λ_5) or P_2 (with speed λ_6) and the second and third term describes the unbinding of P_1 (λ_7) and P_2 (λ_8). By applying the quasi product form assumption given in Example 2, the right-hand side of (7) becomes

$$\begin{aligned}
&-\Bigg(\lambda_5 \sum_{x_4} x_4 Pr\{Dna=1,Dna.P_1=0,Dna.P_2=0,P_1=x_4\}+ \\
&\lambda_6 \sum_{x_5} x_5 Pr\{Dna=1,Dna.P_1=0,Dna.P_2=0,P_2=x_5\}\Bigg)+ \\
&\lambda_7 Pr\{Dna=0,Dna.P_1=1,Dna.P_2=0\}+ \\
&\lambda_8 Pr\{Dna=0,Dna.P_1=0,Dna.P_2=1\}
\end{aligned}$$

Note that the first (second) term of the above quantity is proportional to the expected amount of P_1 (P_2) given that the system is in a state with $Dna=1, Dna.P_1=0, Dna.P_2=0$. By similar reasoning, for $(x_1,x_2,x_3)=(0,1,0)$, i.e, when the Dna is bound to P_1, we get

$$\begin{aligned}
&\frac{dPr\{Dna=0,Dna.P_1=1,Dna.P_2=0\}}{dt} = \\
&-\lambda_7 Pr\{Dna=0,Dna.P_1=1,Dna.P_2=0\}+ \\
&\lambda_5 \sum_{x_4} x_4 Pr\{Dna=1,Dna.P_1=0,Dna.P_2=0,P_1=x_4\}
\end{aligned}$$

For the case when the Dna is bound to P_2, i.e., for $(x_1,x_2,x_3)=(0,0,1)$, we get the counterpart of the above expression as

$$\begin{aligned}
&\frac{dPr\{Dna=0,Dna.P_1=0,Dna.P_2=1\}}{dt} = \\
&-\lambda_8 Pr\{Dna=0,Dna.P_1=0,Dna.P_2=1\}+ \\
&\lambda_6 \sum_{x_5} x_5 Pr\{Dna=1,Dna.P_1=0,Dna.P_2=0,P_2=x_5\}
\end{aligned}$$

Now we turn our attention to node v_2 (Figure 3). The species indicated by $Q(v_2)$ are Dna, $Dna.P_1$, $Dna.P_2$ and P_1 while the species given by $P(v_2)$ are Dna, $Dna.P_1$ and $Dna.P_2$. We have to consider all possible values for all four species given by $Q(v_2)$. We first consider the case when we have free Dna (consequently, no $Dna.P_1$ and $Dna.P_2$), i.e., $(x_1, x_2, x_3) = (1, 0, 0)$, and a generic amount, x_4, of P_1. By following Theorem 1 we get

$$\begin{aligned}
&\frac{dPr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0, P_1 = x_4\}}{dt} = \\
&- \lambda_1 Pr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0, P_1 = x_4\} + \\
&\lambda_1 Pr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0, P_1 = x_4 - 1\} - \\
&\lambda_3 x_4 Pr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0, P_1 = x_4\} + \\
&\lambda_3 (x_4 + 1) Pr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0, P_1 = x_4 + 1\} - \\
&\lambda_5 x_4 Pr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0, P_1 = x_4\} - \\
&\lambda_6 \sum_{x_5} \frac{x_5 Pr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0, P_2 = x_5\}}{Pr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0\}} \times \\
&\qquad Pr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0, P_1 = x_4\} + \\
&\lambda_7 Pr\{Dna = 0, Dna.P_1 = 1, Dna.P_2 = 0, P_1 = x_4 - 1\} + \\
&\lambda_8 Pr\{Dna = 0, Dna.P_1 = 0, Dna.P_2 = 1, P_1 = x_4\}
\end{aligned} \tag{8}$$

where the terms on the right-hand side correspond to, respectively: outgoing probability by production of P_1 (with free Dna); incoming probability by production of P_1 (with free Dna); outgoing probability by degradation of P_1; incoming probability by degradation of P_1; binding of Dna with P_1; binding of Dna with P_2; unbinding of Dna with P_1; and unbinding of Dna with P_2. It is worth to note that the effect of the quasi product form assumption is that the term corresponding to the binding of Dna with P_2 is determined by the conditional expected value of P_2 given that the Dna is free. Indeed the summation in that term is equal to

$$E\{P_2 \mid Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0\}$$

which corresponds to the expected value of P_2 conditioned by the state of the promoter region. Next we consider the situation that the Dna is bound to P_1, i.e., $(x_1, x_2, x_3) = (0, 1, 0)$, and a generic amount, x_4, of P_1. Theorem 1 leads to

$$\begin{aligned}
&\frac{dPr\{Dna = 0, Dna.P_1 = 1, Dna.P_2 = 0, P_1 = x_4\}}{dt} = \\
&- \lambda_3 x_4 Pr\{Dna = 0, Dna.P_1 = 1, Dna.P_2 = 0, P_1 = x_4\} + \\
&\lambda_3 (x_4 + 1) Pr\{Dna = 0, Dna.P_1 = 1, Dna.P_2 = 0, P_1 = x_4 + 1\} - \\
&\lambda_9 Pr\{Dna = 0, Dna.P_1 = 1, Dna.P_2 = 0, P_1 = x_4\} + \\
&\lambda_9 Pr\{Dna = 0, Dna.P_1 = 1, Dna.P_2 = 0, P_1 = x_4 - 1\} - \\
&\lambda_7 Pr\{Dna = 0, Dna.P_1 = 1, Dna.P_2 = 0, P_1 = x_4\} + \\
&\lambda_5 (x_4 + 1) Pr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0, P_1 = x_4 + 1\}
\end{aligned}$$

where the terms on the right-hand side correspond to, respectively: outgoing probability by degradation of P_1; incoming probability by degradation of P_1; outgoing probability by production of P_1 (with bound Dna); incoming probability by production of P_1 (with bound Dna); unbinding of P_1; and binding of P_1.

The last situation to consider for what concerns v_2 is when the Dna is bound to P_2, i.e., $(x_1, x_2, x_3) = (0, 0, 1)$, and a generic amount, x_4, of P_1. We get

$$\begin{aligned}
&\frac{dPr\{Dna = 0, Dna.P_1 = 0, Dna.P_2 = 1, P_1 = x_4\}}{dt} = \\
&- \lambda_3 x_4 Pr\{Dna = 0, Dna.P_1 = 0, Dna.P_2 = 1, P_1 = x_4\} + \\
&\lambda_3 (x_4 + 1) Pr\{Dna = 0, Dna.P_1 = 0, Dna.P_2 = 1, P_1 = x_4 + 1\} - \\
&\lambda_8 Pr\{Dna = 0, Dna.P_1 = 0, Dna.P_2 = 1, P_1 = x_4\} + \\
&\lambda_6 \sum_{x_5} \frac{x_5 Pr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0, P_2 = x_5\}}{Pr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0\}} \times \\
&\qquad Pr\{Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0, P_1 = x_4\}
\end{aligned}$$

where on the right-hand side the terms are, respectively: outgoing probability by degradation of P_1; incoming probability by degradation of P_1; unbinding of P_2; and binding with P_2. As before, the effect of the quasi product form assumption is that the speed of the binding of the Dna with P_2 is proportional to the conditional expected amount of P_2.

The last node of the forest (Figure 3), v_3, leads to the counterpart of the expressions reported above for node v_2.

4 Algorithm

In this section, we provide a sketch of the implementation of the algorithm that follows from the quasi product form assumption. We focus on the representation of the system of ODEs and do it in such a way that it can be used in common ODE solvers.

As described in the previous section, the computation of the transient probabilities based on the quasi product form assumption requires the quantities involved in (5) and, in particular, it needs the probabilities $Pr\{\wedge_{i \in Q(v)}(X_i = x_i)\}$ since they allow the computation of any marginal probability refering to a subset of the species in $Q(v)$. Thus, the collection of all the marginal distributions representing the sets $Q(v), v \in \mathcal{V}$, is enough to carry out the computations. Nevertheless, since it can happen that there exist v_1 and v_2 such that $Q(v_1) \subset Q(v_2)$, considering all nodes in $\mathcal{V}$ can lead to a redundant set of ODEs. This happens in case of the exclusive switch where v_1 has only outgoing arcs and, consequently, $Q(v_1)$ is contained both in $Q(v_2)$ and $Q(v_3)$. The overhead caused by this redundancy can be either negligible or non-negligible depending on the applied quasi product form assumption. In Table 1, from line 1 to 10, we propose a simple way to eliminate the redundancy by computing the minimal set of marginal distributions (stored in the variable *Marg*). The algorithm consists of two nested

loops which collect (in the variable Q) the species representing the dependencies of a node (including the species in the node itself) and construct a new marginal distribution only if the node has incoming arcs or if the node does not have outgoing arcs at all (in order to guarantee the presence of those species that are completely independent from the others). The object representing the new marginal distribution itself is instantiated in line 9 and added to the set of marginals collected in *Marg*.

Table 1. Algorithm: Preprocessing for the quasi product form approximation

```
0  Preprocessing() begin
   // Makes the marginal distribution set
1    Marg := ∅;
2    forall v ∈ V do
3      Q := ∅;
4      forall i such that (i, v) ∈ E do
5        Q := Q ∪ I(i);
6      end
7      if Q ≠ ∅ ∨ (Q == ∅ ∧ ∄(v, i) ∈ E) then
8        Q := Q ∪ I(v);
9        Marg := Marg ∪ marginal.init(Q);
10   end
     ......
11   forall (m, m') such that m, m' ∈ Marg do
12     Int = m.Q ∩ m'.Q;
13     forall q ∈ m'.Q do
14       if Int ≠ ∅ then m.conditions.insert(q, Int);
15       m.marginals.insert(q, m');
       end
16     forall q ∈ m.Q do
17       if Int ≠ ∅ then m'.conditions.insert(q, Int);
18       m'.marginals.insert(q, m);
19     end
20   end
21 end
```

In the following we concentrate on the so-called evaluation step, i.e., the computation of the derivatives that are necessary to perform the numerical integration of the ODEs. This step requires to represent the marginal distributions and the following variables are necessary in order to carry out the computations:

- Q: set containing the indexes of the species composing the marginal distribution,
- *states*: list of all possible values that the quantities of the species present in Q can assume,
- *reac*: list of those reactions that can move probability mass over the states of the marginal distribution,
- *conditions*: data structure that, given an index of a species, returns the indexes of those species in Q that condition it,
- *marginals*: data structure that, given an index of a species, determines from which marginal distributions its conditional moments (in most cases its conditional expectation) have to be computed.

Table 2. Algorithm: Data structure describing a marginal distribution

```
0  data struct marginal begin
1    Q; // the indexes of the species composing the marginal
2    states; //states of the marginal distribution
3    reacs; // list of reactions able to modify the species in Q
4    conditions; // data structure containing the indexes of the conditoned species
5    marginals; // data structure containing the indexes of the marginals
                   in which the other species can be found
     ......
6    init(Q) begin
7      this.Q := Q
8      reacs := {r|∃j ∈ Q, c_{r,j} > 0};
9    end
10 end
```

A partial implementation of a data structure with the above variables is reported in Table 2 where we detailed the function *init* used in line 9 of Table 1.

Note that, since several reactions can have a null impact on a marginal distribution, it is worth to store *reac* in order to save time during the evaluation step. The representation of *conditions* and *marginals* is trivial since they can be expressed through simple matrices. Despite this, for sake of completeness, in lines $11-21$ of Table 1 we provide the pseudo-code to initialize these data structures starting from the list of the marginal distributions. The function *insert* used in this part of the algorithm inserts a new association in the data structure *condition* in such a way that the species q represents the research key and the second argument the set of species to retrieve.

Finally, in Table 3 we provide the algorithm for the evaluation step where *s.prob, s.der*, and *s.succ(r)* refer, respectively, to the probabilty of a state at time t, the derivative of the probability of a state at time t and the state reached by the occurrence of reaction r in the state s. The algorithm is similar to the one used to handle the classical chemical master equation (see lines 17 and 18). The main differences are in lines 8-16 where we make a distinction between the species involved in the marginal distribution and the others that will be considered through their conditional moments. In lines 9 and 10 the algorithm retrieves that subset of species belonging to m which affects the ith species, and stores the current values of these species according to the current state s in *condition*. Subsequently, the algorithm checks if the corresponding conditional moment has been already computed (line 11). If yes, it uses the quantity (line 14), otherwise it is computed using the simple algorithm provided in Table 4 and stored (lines 12 and 13). The conditional moment is then used to update the rate of the reaction. Once the rate is computed, it is used to determine the derivative of the probability of both state s and its successor (lines 17 and 18).

The number of conditional moments stored in the data structure *condexp* is strongly related to the model and the applied quasi product form decomposition. For example, for the exclusive switch with the decomposition presented in Figure 3, only two values need to be stored: $E\{P_1 \mid Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0\}$ and $E\{P_2 \mid Dna = 1, Dna.P_1 = 0, Dna.P_2 = 0\}$. Moreover, considering that each marginal uses only a subset of the reactions and they probably

Table 3. Algorithm: Procedure describing the evaluation step

```
0  Eval() begin
1    condexp := ∅
2    forall m ∈ Marg do
3      forall s ∈ m.states do
4        forall r ∈ m.reac do
5          rate = λ_r;
6          forall i ∈ M do
7            if i ∈ m.Q then rate = rate * binom(s_i, a_{r,i});
8            else if a_{r,i} ≠ 0 then
9              C := m.conditions.get(i);
10             condition := {(j, s_j)|∀j ∈ C};
11           if condexp.notexists(i, condition) then
12             tempexp = ComputeMom(i, m, condition);
13             condexp := condexp ∪ tempexp;
14           else tempexp = condexp.extract(i, condition);
15           rate = rate * tempexp;
16         end
17         s.der = s.der − rate * s.prob;
18         s.succ(r).der = s.succ(r).der + rate * s.prob;
19       end
20     end
21   end
22 end
```

Table 4. Algorithm: Procedure to compute the conditional probabilities

```
0  ComputeMom(i,m,condition) begin
1    prob = 0;
2    exp =0;
3    m' := m.marginals.get(i);
4    forall s ∈ m'.states do
5      exp = exp + s.prob * s_i;
6      if ∀(i, j) ∈ condition s_i = j ∨ condition == ∅ then prob = prob + s.prob;
7    end
8    if prob == 0 then return 0;
9    else return exp / prob;
10 end
```

use a limited set of species not belonging to the set Q, it is likely that the number of moments that need to be stored is low and in general negligible with regards to the number of states. There can be however situations in which many conditional moments must be computed and the same one is applied many times for a sequence of states. For this reason, we suggest to store these quantities in a cache from which recently calculated entries can be retrieved.

Since for the examples used in this paper the necessary conditional joint factorial moments are simply conditional expectations, we presented the algorithms considering expectations only. The generalization to joint moments is straightforward but would lead to cumbersome notation in the algorithms in Table 3 and 4.

The last consideration is about the "cut" of the states having negligible probability mass. This technique is based on a treshold under which the states are not considered during the integration step. Consequently, the overall computational

time can be significantly reduced. Furthermore, in case of unbounded state spaces, it allows us not to define the bound *a priori*. The use of this technique in combination with the quasi product form approach is feasible and effective but, since its explanation in details is out of the scope of the paper, the reader is referred to [16].

5 Numerical Illustrations

In this section we show numerical results obtained using the quasi product form assumption. We apply the approximation to two models with various settings of the parameters. For all the cases, we compare the results obtained by the proposed quasi product form approximation with the exact behaviour computed on the original CTMC of the model. In order to provide a visual comparison of the original behaviours and the approximated values, we provide in figures the expectations, the variances and the marginal distributions of those species that better represent the dynamics of the models. The algorithm based on the quasi product form assumption has been implemented in JAVA using the odeToJava package[1] to solve the system of ODEs. All the experiments have been performed on a Intel Centrino Dual Core with 4Gb of RAM.

5.1 Exclusive Switch Model

As anticipated before, if the unbinding of the promoter is unlikely, the exclusive switch model behaves in a bistable way because either of the two proteins, P_1 and P_2, can monopolise the promoter region of the *Dna* and obstruct consequently the growth of the other. In this situation, the amounts of the two proteins are inversely correlated in such a way that a high number of molecules of P_1 corresponds to low quantities of P_2 and viceversa. Intuitively, this fact seems to indicate that the quasi product form assumption represented in Figure 3 leads to imprecise approximation because it does not consider directly the joint distribution of P_1 and P_2. Nevertheless, as it will be illustrated by the presented numerical results, the negative correlation between the two proteins and the associated bistable marginal distributions can be captured, in an indirect manner, by the state of the promoter.

The approximations will be carried out with two sets of parameters, as reported in Table 5. The first set is symmetric, i.e., the two proteins have the same probability to monopolise the promoter region. With the second set P_2 has an advantage over P_1.

As mentioned earlier, the initial state is $x = (1, 0, 0, 0, 0)$. In Figure 4 the marginal protein distribution is depicted for three time points (because of the symmetric settings the probabilities are identical for P_1 and P_2). One can note

[1] Available at `http://www.netlib.org/ode/` and developed by M. Patterson and R. J. Spiteri.

Table 5. Exclusive switch : The two sets of parameters used to perform the tests

# Set	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}
1	0.5	0.5	0.005	0.005	0.01	0.01	0.005	0.005	0.5	0.5
2	1.0	2.0	0.1	0.1	0.01	0.01	0.005	0.005	1.0	2.0

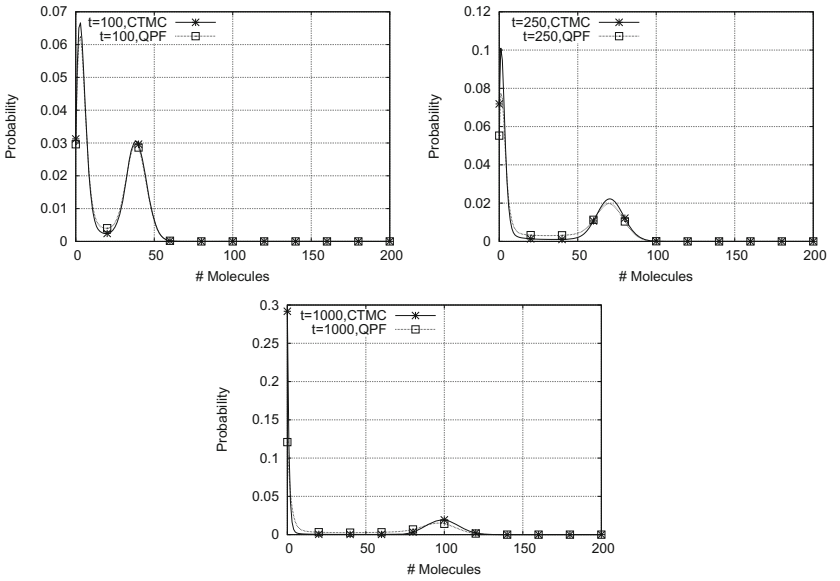


Fig. 4. Exclusive switch: marginal distributions of P_1 (P_2) at time $t = 100, 250$ and 1000 using the symmetric set of parameters

that already after 100 time units, the protein distribution gets split in two parts forming a bistable distribution. The quasi product form approach is able to catch precisely the shape of this distribution. As time elapses the bistability gets more marked. At time $t = 250$ the approximation still provides a good picture of the behaviour of the model but the numerical values are not as precise as for smaller values of t. In steady state, which can be observed at $t = 1000$, the quasi product form assumption captures well the bistability but gives a quite inaccurate approximation of the lower probabilities (those less than 10^{-3}) and of the probability of having zero of one of the two proteins.

In Figure 5 we show the mean and the variance of the protein quantity. The approximate mean is very accurate for all time points while the variance is underestimated. The fact that the variances are less accurate is not surprising. In fact, by applying the quasi product form assumption, distributions are often "substituted" by their mean values during the calculations. (One such example is the summation in (8)). The overall effect of this is that the approximate variance is lower than the exact one.

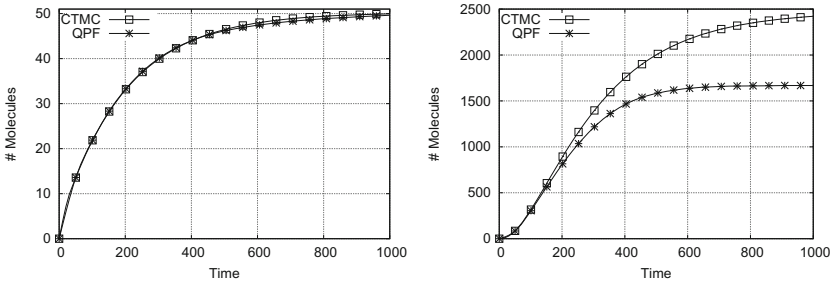


Fig. 5. Exclusive switch: The expectation and the variance of the quantity of P_1 (P_2) as function of the time with the symmetric set of parameters

Table 6. Exclusive switch : Probability of having the *Dna* free, bound to P_1 and bound to P_2 after 1000 time units

Parameter set	Method	free *Dna*	*Dna* bound to P_1	*Dna* bound to P_2
1	**CTMC**	0.004956	0.497521	0.497521
1	**QPF**	0.009597	0.495201	0.495201
2	**CTMC**	0.023392	0.293140	0.657854
2	**QPF**	0.024851	0.284785	0.690362

In Table 6 we provide the probabilities of having the *Dna* promoter region free, bound to P_1 and bound to P_2 after 1000 time units. The approximation captures the fact that the promoter region is free with low probability but the numerical value is almost twice larger than the real value.

In order to show that the quasi product form approximation does not take advantage of the symmetry of the previous setting, we provide now the results for the asymmetric set of parameters. As shown in Table 5, in this case the production of P_2 is two times faster than that of P_1. Figure 6 depicts the distribution of the two proteins after 50 and 100 time units (with this parameter set steady state is almost reached at $t = 100$). One can see that the quasi product form approximation provides a very precise view of the protein distributions. Consequently, the expectations and the variances (Figure 7) and the probabilities of the three promoter regions (Table 6) are reproduced accurately as well.

The numerical integration of the ODEs associated with quasi product form assumption required, in the worst case, less than 20 seconds whereas the solution of the CTMC through uniformisation took several minutes. Considering space complexity, if the considered maximum for the protein quantities is p_{max}, then the quasi product form assumption leads to $3 \times 2 \times (1 + p_{max})$ equations while the number of states in the original CTMC is $3 \times (1 + p_{max})^2$.

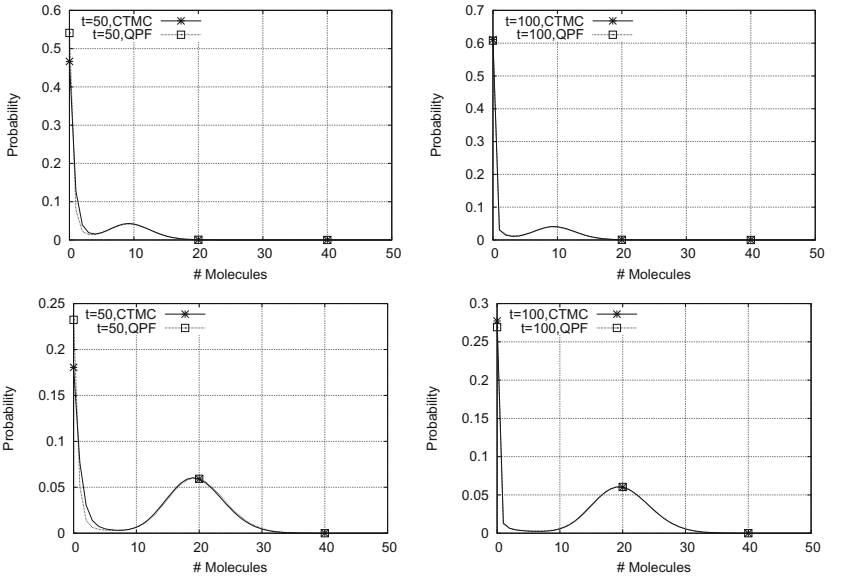


Fig. 6. Exclusive switch: Marginal distributions of P_1 and P_2 at time $t = 50, 100$ using the asymmetric set of parameters

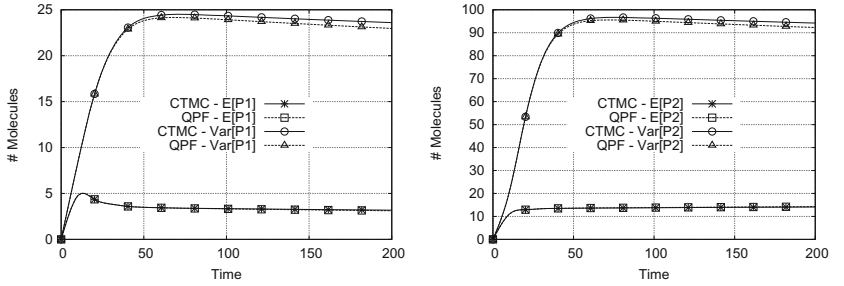


Fig. 7. Exclusive switch: The expectation and the variance of the quantity of P_1 and P_2 as function of the time with the asymmetric set of parameters

5.2 Multi-attractor Model

As a second example, in order to test the quasi product form approximation with a more complex model, we propose a part of the multi-attractor model considered by Zhou et al. [28] describing the interactions among three genes, namely, *Pax*, *Mafa*, and *Delta*. Each gene has a corresponding protein that is able to bind itself to promoter regions on the *Dna*. The graph representing the possible bindings of the genes is depicted in Figure 8 where edges with solid lines correspond to binding without inhibition whereas the dotted ones indicate the inhibitions. Considering all possible bindings, the model involves 13 species. The first 10 of these can assume only boolean values and represent all the possible

states of the three promoter regions. The last three, instead, describe the number of molecules of proteins present in the system.

The reactions are listed in Table 7 where we denote with the suffix *Prot* the proteins and use the suffix *Dna* with reference to promoter regions. The "dot" has the same meaning as in case of the exclusive switch model.

As in the case of the exclusive switch, the overlap of the common promoters leads to some invariants:

$$\begin{aligned} MafaDna + MafaDna.PaxProt + MafaDna.MafaProt + MafaDna.DeltaProt = \\ DeltaDna + DeltaDna.PaxProt + DeltaDna.MafaProt + DeltaDna.DeltaProt = \\ PaxDna + PaxDna.DeltaProt = 1 \end{aligned}$$

Accordingly, the production of the proteins is modulated in $2 \times 4 \times 4 = 32$ different ways corresponding to all the possible combinations of the states in which promoter regions can be.

The state space of the underlying CTMC is infinite and the number of states having a non negligible probability mass blows up over three dimensions. In this situation, if the parameters are not such that the protein quantities remain low, any analytical solution of the CTMC is unfeasible by using common techniques whereas the analysis through the quasi product form assumption remains possible.

The quasi product form assumption we propose is similar to the one used in case of the exclusive switch. It is described by a forest of 4 nodes in such a way that node v_1 is associated with all the species representing the promoter regions, and nodes v_2, v_3 and v_4 correspond to *PaxProt*, *MafaProt*, and *DeltaProt*, respectively. As depicted in Figure 8, the forest has three edges, $v_1 \to v_2$, $v_1 \to v_3$ and $v_1 \to v_4$, indicating that the dependences among the proteins is taken into account, in an indirect manner, through the state of the promoter regions. This implies that the resulting system of ODEs has one equation for each protein, for every possible protein quantity and every possible state of the promoter region. Consequently, if the considered maximal protein quantity is p_{max} for every protein then the number of equations is $3 \times (p_{max} + 1) \times 32$. This is much less than the number of states in the original CTMC which, considering the same range of protein levels, equals $(p_{max} + 1)^3 \times 32$.

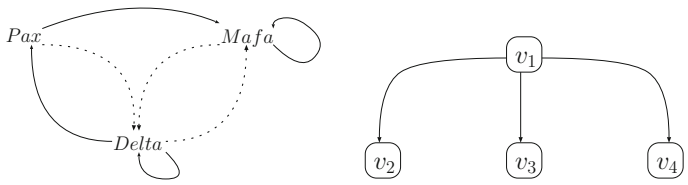


Fig. 8. The Multi-attractor model: The graph representing the interactions among the genes (left) and the forest describing the quasi product form (right)

Table 7. Multi-attractor: The reaction system

Reactions		
$PaxDna$	$\longrightarrow$	$PaxDna + PaxProt$
$PaxProt$	$\longrightarrow$	$\emptyset$
$PaxDna + DeltaProt$	$\longrightarrow$	$PaxDna.DeltaProt$
$PaxDna.DeltaProt$	$\longrightarrow$	$PaxDna + DeltaProt$
$MafaDna$	$\longrightarrow$	$MafaDna + MafaProt$
$MafaProt$	$\longrightarrow$	$\emptyset$
$MafaDna + PaxProt$	$\longrightarrow$	$MafaDna.PaxProt$
$MafaDna.PaxProt$	$\longrightarrow$	$MafaDna + PaxProt$
$MafaDna.PaxProt$	$\longrightarrow$	$MafaDna.PaxProt + MafaProt$
$MafaDna + MafaProt$	$\longrightarrow$	$MafaDna.MafaProt$
$MafaDna.MafaProt$	$\longrightarrow$	$MafaDna + MafaProt$
$MafaDna.MafaProt$	$\longrightarrow$	$MafaDna.MafaProt + MafaProt$
$MafaDna + DeltaProt$	$\longrightarrow$	$MafaDna.DeltaProt$
$MafaDna.DeltaProt$	$\longrightarrow$	$MafaDna + DeltaProt$
$DeltaDna$	$\longrightarrow$	$DeltaDna + DeltaProt$
$DeltaProt$	$\longrightarrow$	$\emptyset$
$DeltaDna + PaxProt$	$\longrightarrow$	$DeltaDna.PaxProt$
$DeltaDna.PaxProt$	$\longrightarrow$	$DeltaDna + DeltaProt$
$DeltaDna.PaxProt$	$\longrightarrow$	$DeltaDna.PaxProt + DeltaProt$
$DeltaDna + MafaProt$	$\longrightarrow$	$DeltaDna.MafaProt$
$DeltaDna.MafaProt$	$\longrightarrow$	$DeltaDna + MafaProt$
$DeltaDna + DeltaProt$	$\longrightarrow$	$DeltaDna.DeltaProt$
$DeltaDna.DeltaProt$	$\longrightarrow$	$DeltaDna + DeltaProt$
$DeltaDna.DeltaProt$	$\longrightarrow$	$DeltaDna.DeltaProt + DeltaProt$

We test the quasi product form approach on this model with three sets of parameters as reported in Table 8. Since the state space of the original model is large, we compare the results of the quasi product form approach with statistics obtained through the Monte Carlo simulation of the original CTMC. The initial state of the model is such that all the promoter regions are free and no proteins are present in the system.

Due to the low propensity of the binding reactions compared to the other rates, the first set of parameters represents the most desirable situation to apply the proposed quasi product form assumption. This is because bindings are the only reactions for which, due to the assumption, distributions are considered through their mean values. If these reactions are much less frequent than the others then the distribution of a protein is barely influenced by another one and the quasi product form assumption is plausible. Figures 9 and 10 reflect this situation showing a perfect match between the results obtained through the quasi product form approximation and the simulations of the original CTMC.

Table 8. Multi-attractor model: The three sets of parameters used to perform the tests where d refers to degradation reactions, b and u correspond to binding and unbinding reactions, respectively, and p to production reactions

Parameter set	d	b	u	p
1	0.1	0.01	1.0	5.0
2	0.1	0.01	0.001	5.0
3	0.1	1.0	1.0	5.0

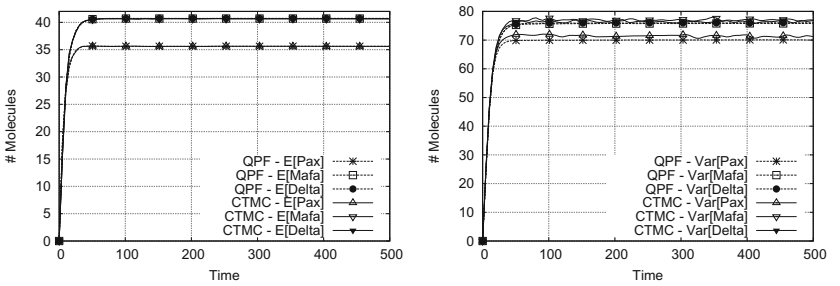


Fig. 9. Multi-attractor model: Expectations (left) and variances (right) of the three proteins with $d = 0.1, b = 0.01, u = 1, p = 5$

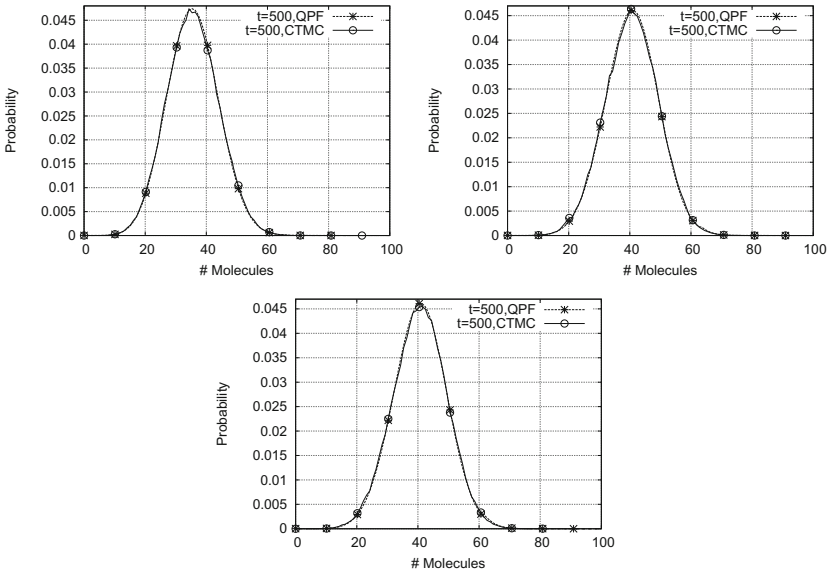


Fig. 10. Multi-attractor model: Marginal probabilities of the *PaxProt* (left), *MafaProt* (right) and *DeltaProt* (below) with $d = 0.1, b = 0.01, u = 1, p = 5$

The second set of parameters is able to generate strong correlations among the distribution of the proteins (similarly, to those present in the exclusive switch model). This is achieved by setting the unbinding constants to a much lower value (see Table 8) which implies that, even if they are rare, bindings will eventually occur and proteins can monopolise the promoter. Despite the fact that this setting is less favourable for the quasi product form assumption, the approximation, as it can be seen in Figure 11, catches both the expectations and the variances of the three proteins. Moreover, as shown in Figure 12, also the marginal distributions of the proteins are captured precisely. Note that all three

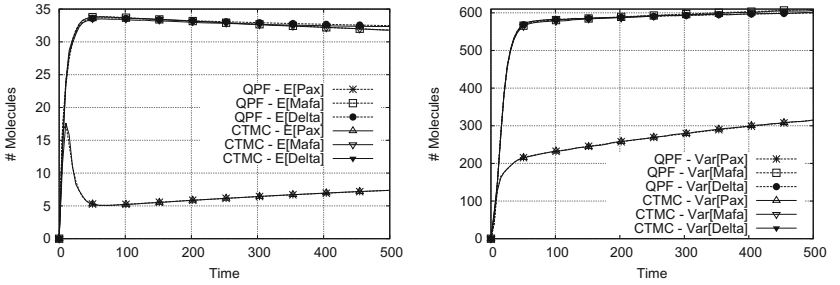


Fig. 11. Multi-attractor model: Expectations (left) and variances (right) of the three proteins with $d = 0.1, b = 0.01, u = 0.001, p = 5$

proteins have bistable distributions but this is hard to see in case of *PaxProt* because this protein is at level 0 with high probability. The goodness of the approximation is evident from the curves representing the marginal distributions (Figure 12). Both the probability mass at zero and the rarer event around 50 are precisely reconstructed by the proposed approximation.

As last example, we provide a case in which the quasi product form approximation is not able to provide a good estimation of the probability distributions. In order to challenge the quasi product form assumption, we choose a set of parameters in which the binding and the unbinding reactions are extremely

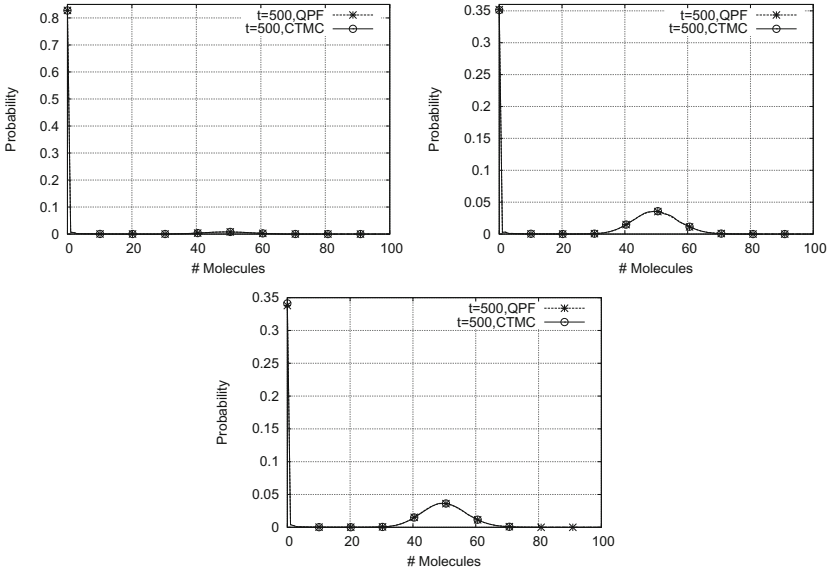


Fig. 12. Multi-attractor model: Marginal probabilities of the *PaxProt* (left), *MafaProt* (right) and *DeltaProt* (below) with $d = 0.1, b = 0.01, u = 0.001, p = 5$

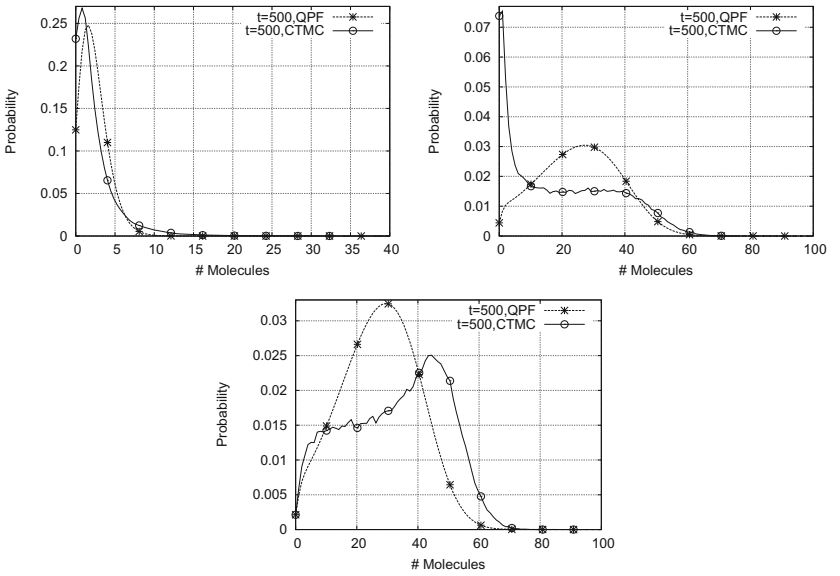


Fig. 13. Multi-attractor model: Marginal probabilities of *PaxProt* (left), *MafaProt* (right) and *DeltaProt* (below) with $d = 0.1, b = 1, u = 1, p = 5$

frequent. Since in our approximation the marginal distributions "communicate" only through conditional expectations, we expect that the computations give a result which is similar to the original in average but is not able to catch the effects of the fluctuations given by the frequent bindings and unbindings. Figure 13 depicts the marginal distributions of the proteins and their approximations. In case of *PaxProt* the approximation is reasonble, while for *MafaProt* and *DeltaProt* the irregular shapes are not captured well. Nevertheless, we point out that, even if the peculiarities of the distributions are not captured (e.g., the peak near zero for *MafaProt*), the approximated distributions provide a good picture of the support of the original distributions. Finally, in Figure 14, it is possible to observe the expectations and the variances of the three proteins. The slopes of the original curves are preserved by the approximation and the error over the trajectories is reasonable.

The computation of the quasi product form required about an hour and a half for the first and the third set of parameters whereas the second requires about 30 minutes. By setting the treshold of the probabilities to 10^{-6}, an integration step of the ODEs has considered, on average, 1000 states distributed over the three marginal probabilities. The corresponding original, three-dimensional state space of the proteins contains about 3×10^7 states. Common, exact analysis techniques cannot handle such amount of states using conventional hardware.

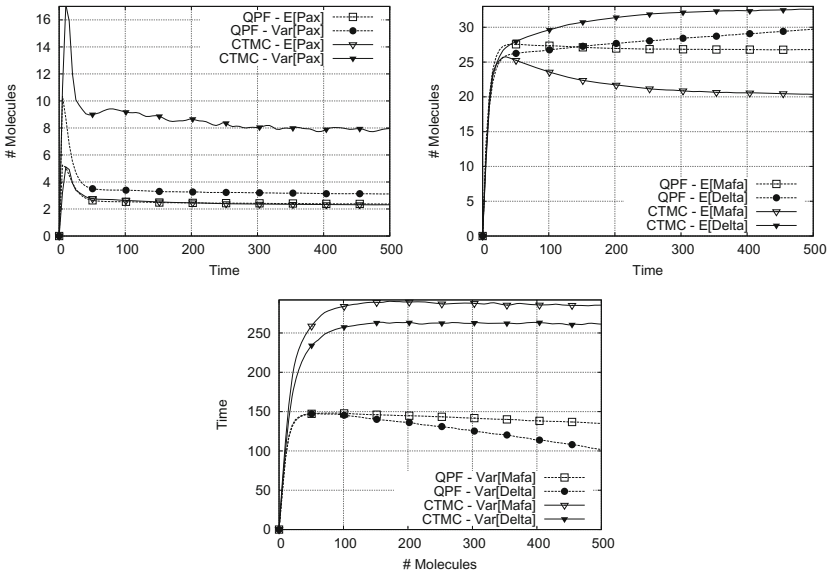


Fig. 14. Multi-attractor model: Expectations and variances of the three proteins with $d = 0.1, b = 1, u = 1, p = 5$

6 Error Evaluation

A thorough error analysis or error bounding of the calculations based on quasi product forms is out of scope for this paper and is left as future work. We present, however, a preliminary approach to validate the results obtained by the quasi product form assumption. This approach provides a first quick evaluation of the goodness of the results and can point out where and how much the quasi product form deviates from the original behavior of the system under study.

Let us assume that the transient probabilities have been computed up to time t under the quasi product form assumption. The probability of a given state, $Pr\{X = x\}$, can be then calculated by (5). We can compute the derivative $dPr\{X = x\}/dt$ under the quasi steady state assumption by applying the formulas provided in (5) and in Theorem 1. Let us denote this quantity by $p'_{\mathrm{QPF}}(t, x)$. The same derivative can be computed considering the behavior of the original CTMC based on (3), i.e., *without* assuming quasi product form. In other words, we use the quasi product form assumption to compute the probabilities up to time t and then calculate how much these probabilities would be moved by the original CTMC in an infinitesimal interval. The resulting derivative will be denoted by $p'_{\mathrm{CTMC}}(t, x)$. The difference of the two derivatives can be used to

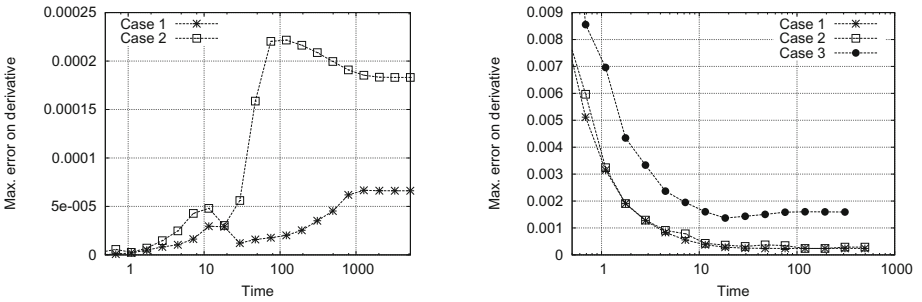


Fig. 15. Error measure for the exclusive switch (left) and for the multi-attractor model (right) for the different sets of parameters

quantify how much the quasi product form assumption deviates from the original behavior. In particular, we use the quantity

$$\max_x \left| p'_{\mathrm{QPF}}(t,x) - p'_{\mathrm{CTMC}}(t,x) \right| \tag{9}$$

i.e., the maximum of the absolute value of the differences, to quantify the error introduced by the quasi product form approximation at time t.

In Figure 15 we depicted the above error measure for the exclusive switch and for the multi-attractor model using the different parameter sets introduced in Section 5. For the exclusive switch, in case of both parameter sets, the error is low all along the calculations and it stabilizes as the process reaches steady state. For the second parameter set, the error is somewhat higher and it reflects the fact that in this case the original probabilities are captured with somewhat less precision (see Table 6). For the multi-attractor model the error is higher and the difference between the well-approximated cases (first and second set of parameters) and the poorly approximated case (third set of parameters) is reflected by the error measure.

In general, a positive trait of the measure in (9) is that it does not require to calculate the transient behavior of the original Markov chain, and thus it can be calculated in a memory efficient manner. A negative trait is that it does not lend itself to error bounding (it would lead to highly untight bounds).

As for future development, the proposed measure can be used to identify that subset of states where the quasi product form assumption results in high error. Indeed, we plan to develop an extension of the algorithm where only a part of the state space is assumed to be in quasi product form and this part is chosen dynamically during the transient interval.

7 Conclusions

In this paper we proposed an approximate solution technique for the analysis of Markov models of reaction networks. The technique is based on the assumption that the transient probabilities can be decomposed into a product. This

product, which we call *quasi product form*, is a relaxed version of the classical product form widely used in analysing the steady state of queueing networks. We presented several numerical examples for which the quasi product form approximation provides satisfactory precision. In case of these examples the choice of the applied quasi product form was natural. An automatic identification of the appropriate quasi product form decomposition is out of scope for this paper and will be studied in the future. One idea in this direction is the use of moment closure techniques to quickly explore the correlations of the involved species and decompose the probabilities accordingly. Also numerical integration techniques for the ODEs resulting from the proposed approach will have to be developed in order to speed up the computations.

References

1. Angius, A., Horváth, A.: Product form approximation of transient probabilities in stochastic reaction networks. Electronic Notes on Theoretical Computer Science 277, 3–14 (2011)
2. Arkin, A., Ross, J., McAdams, H.H.: Stochastic kinetic analysis of the developmental pathway bifurcation in phage lambda-infected escehrichia coli cells. Genetics 149(4), 1633–1648 (1998)
3. Bennett, D.C.: Differentiation in mouse melanoma cells: initial reversibility and an in-off stochastic model. Cell 34(2), 445–453 (1983)
4. Boucherie, R.J., Taylor, P.: Transient product form distributions in queueing networks. Discrete Event Dynamic Systems: Theory and Applications 3, 375–396 (1993)
5. Cao, Y., Gillespie, D.T., Petzold, L.R.: The slow-scale stochastic simulation algorithm. J. Chem. Phys. 122(1) (2005)
6. Chandy, K.M., Herzog, U., Woo, L.S.: Parametric analysis of queueing networks. IBM Journal of R. & D. 19(1), 36–42 (1975)
7. Ciocchetta, F., Degasperi, A., Hillston, J., Calder, M.: Some investigations concerning the CTMC and the ode model derived from bio-pepa. Electron. Notes Theor. Comput. Sci. 229, 145–163 (2009)
8. Cook, D.L., Gerber, A.N., Tapscott, S.J.: Modeling stochastic gene expression: implications for haploinsufficiency. Proc. Natl. Acad. Sci. USA 95(26), 15641–15646 (1998)
9. Cordero, F., Horváth, A., Manini, D., Napione, L., Pierro, M.D., Pavan, S., Picco, A., Veglio, A., Sereno, M., Bussolino, F., Balbo, G.: Simplification of a complex signal transduction model using invariants and flow equivalent servers. Theor. Comput. Sci. 412(43), 6036–6057 (2011)
10. Dayar, T., Mikeev, L., Wolf, V.: On the numerical analysis of stochastic Lotka-Volterra models. In: Proc. of the Workshop on Computer Aspects of Numerical Algorithms (CANA 2010), pp. 289–296 (2010)
11. Engblom, S.: Computing the moments of high dimensional solutions of the master equation. Appl. Math. Comput. 180, 498–515 (2006)
12. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. 81(25), 2340–2361 (1977)
13. Gillespie, D.T.: A rigorous derivation of the chemical master equation. Physica A 188(1), 404–425 (1992)

14. Gillespie, D.T.: Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys. 115, 1716–1733 (2001)
15. Hasty, J., McMillen, D., Isaacs, F., Collins, J.J.: Computational studies of gene regulatory networks: in numero molecular biology. Nature Reviews Genetics 2(4), 268–279 (2001)
16. Henzinger, T.A., Mikeev, L., Mateescu, M., Wolf, V.: Hybrid numerical solution of the chemical master equation. In: CMSB, pp. 55–65 (2010)
17. Jensen, A.: Markoff chains as an aid in the study of Markoff processes. Skandinavisk Aktuarietidskrift 36, 87–91 (1953)
18. Kurtz, T.G.: Solutions of ordinary differential equations as limits of pure jump Markov processes. Journal of Applied Probability 1(7), 49–58 (1970)
19. Kurtz, T.G.: The Relationship between Stochastic and Deterministic Models for Chemical Reactions. J. Chem. Phys. 57(7), 2976–2978 (1972)
20. Loinger, A., Lipshtat, A., Balaban, N.Q., Biham, O.: Stochastic simulations of genetic switch systems. Phys. Rev. E 75, 021904 (2007), http://link.aps.org/doi/10.1103/PhysRevE.75.021904
21. Mateescu, M., Wolf, V., Didier, F., Henzinger, T.A.: Fast adaptive uniformisation of the chemical master equation. IET Systems Biology 4(6), 441–452 (2010)
22. Moler, C., Loan, C.V.: Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. SIAM Review 45(1), 3–49 (2003)
23. Rathinam, M., Petzold, L.R., Cao, Y., Gillespie, D.T.: Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. J. Chem. Phys. 119(24), 12784–12794 (2003)
24. Singh, A., Hespanha, J.P.: Moment closure techniques for stochastic models in population biology. In: American Control Conference, pp. 4730–4735 (2006)
25. Stewart, W.J.: Introduction to the Numerical Solution of Markov Chains. Princeton University Press (1995)
26. Zhang, J., Watson, L.T., Cao, Y.: Adaptive aggregation method for the chemical master equation. Int. J. of Computational Biology and Drug Design 2(2), 134–148 (2009)
27. Zhang, J., Watson, L.T., Cao, Y.: A modified uniformization method for the solution of the chemical master equation. Computers & Mathematics with Applications 59(1), 573–584 (2010)
28. Zhou, J.X., Brusch, L., Huang, S.: Predicting pancreas cell fate decisions and reprogramming with a hierarchical multi-attractor model. PLoS ONE 6(3) 6(3), 16 (2011), http://dx.plos.org/10.1371/journal.pone.0014752