Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 00 (2015) 1–28
Procedia Computer Science

Approximate analysis of biological systems by hybrid switching jump diffusion

Alessio Angius[a], Gianfranco Balbo[a], Marco Beccuti[a], Enrico Bibbona[b], Andras Horvath[a], Roberta Sirovich[b]

[a] *Università di Torino, Dipartimento di Informatica*
{*beccuti,angius,horvath,balbo*}*@di.unito.it*
[b] *Università di Torino, Dipartimento di Matematica*
{*roberta.sirovich,enrico.bibbona*}*@unito.it*

Abstract

In this paper we consider large state space continuous time Markov chains arising in the field of systems biology. For a class of such models, namely, for density dependent families of Markov chains that represent the interaction of large groups of identical objects, Kurtz has proposed two kinds of approximations. One is based on ordinary differential equations and provides a deterministic approximation, while the other uses a diffusion process with which the resulting approximation is stochastic. The computational cost of the deterministic approximation is significantly lower, but the diffusion approximation retains stochasticity and is able to reproduce relevant random features like variance, bimodality, and tail behavior that cannot be captured by a single deterministic quantity.

In a recent paper, for particular stochastic Petri net models, we proposed a jump diffusion approximation that aims at being applicable beyond the limits of Kurtz's diffusion approximation in order to cover the case when the process reaches the boundary with non-negligible probability. In this paper we generalize the method so that it can be applied to any density dependent Markov chains. Other limitations of the diffusion approximation in its original form are that it can provide inaccurate results when the number of objects in some groups is often or constantly low and that it can be applied only to pure density dependent Markov chains. In order to overcome these drawbacks, in this paper we propose to apply the jump-diffusion approximation only to those components of the model that are in density dependent form and are associated with high population levels. The remaining components are treated as discrete quantities. The resulting process is a hybrid switching jump diffusion, i.e., a diffusion with hybrid state space and jumps where the discrete state changes can be seen as switches that take the diffusion from a condition to another. We show that the stochastic differential equations that characterize this process can be derived automatically both from the description of the original Markov chains or starting from a higher level description language, like stochastic Petri nets. The proposed approach is illustrated on three models: one modeling the so called crazy clock reaction, one describing viral infection kinetics and the last considering transcription regulation.

Keywords: diffusion approximation, jump diffusion, stochastic differential equations with barriers.

1. Introduction

Stochastic modeling of the dynamics of biological systems gains in importance as more and more evidence is gathered that randomness plays an important role in many of these phenomena [1, 2]. In most cases, as in the pioneering works of Gillespie [3] and Kurtz [4], the stochastic process associated with the evolution of the biological system is a continuous time Markov chains (CTMC). In theory, CTMCs can be analyzed by well-established techniques [5] to characterize both their initial transient period and their long run behavior. In practice however the state space of the CTMC representation of a real phenomenon is often so large that an exact analytical treatment is not feasible.

One approach to the analysis of these models is simulation and, starting from [3], several simulation based techniques have been proposed. The main difficulty lies in the facts that because of the size of the state space many simulation runs are needed to characterize the system, and that often the interactions occur in significantly different time scales. Methods to overcome these difficulties were proposed in [6, 7, 8]. Approximate analytical techniques have also been considered. Some examples are the following. In [9] the authors propose a method that dynamically limits the state space to those states that are of non-negligible probability. Since the number of states can be huge even if not all states are considered, in [10, 11] approximate randomization methods have been proposed. Another natural approach is aggregation of states which can be done either by aggregating nearby states [12, 13] or by exploiting the idea of flow equivalence [14]. Techniques that are based on imposing a special dependency structure on the probabilities of the states were proposed in [15, 16].

An important alternative to the above approaches, initiated mainly by Kurtz, is based on constructing a simpler process to approximate the original CTMC when it models the interaction of large groups of identical objects (which can be members of species, or populations, or proteins, or enzymes, etc.). A key concept in these works is the so-called *density dependent* property. For density dependent CTMCs, as it was shown in [4], it is possible to derive a set of ordinary differential equations (ODE) that leads to a good deterministic approximation of the CTMC when the number of interacting objects is large. A stochastic approximation of density dependent processes using diffusion processes, characterized by stochastic differential equations (SDE), was proposed instead in [17]. The ODE based approximation can be strikingly poor when the number of interacting objects is not large enough to rule out variability and when the model involves particular random phenomenon - characterized by bi-modal distributions and/or switching behaviors - that are not possible to capture with a deterministic model. In these cases the diffusion based approximation could give better approximations, although it only works up to the first visit of the boundary of the state space. A recent review on the application of these techniques to model chemical reactions is given in [18].

In [19] we proposed a jump-diffusion diffusion approximation that aims at being applicable beyond the limit of Kurtz theory. Namely, since originally the approximation was defined only up to the first time when it reaches a boundary, we added in the approximating model an explicit description of the behavior at the boundaries. When a components attains a boundary, indeed, it stays there for a while and then it jumps back into the interior mimicking the behavior of the original Markov Chain. In this way, the approach is applicable to such systems where boundaries are reached with non-negligible probability as it happens, for example, in ecological models where there are species that can become temporarily extinct.

The scope of this paper is to refine and extend the jump-diffusion approximation further so that it can be applied to get a better approximation of more general Markov Chains. The motivation of this extension is twofold. First, is the case when the size of a subgroup of objects remains constantly low so that a generalized use of the diffusion approximation to the whole system would lead to inaccurate results. Second, is the situation in which the approximation is not applicable because the CTMC underlying the model does not belong to a density dependent family, even if it has a subset of components that interact in manners which enjoy the density dependent property. Our proposal is to approximate such Markov chains with a hybrid process, where our jump-diffusion approximation is applied to those components which correspond to groups with large number of members and which interact according to the property of density dependence. This category of components is referred to as *continuous* or *fluid* components and in the more usual Piecewise Deterministic Markov Processes (PDMP) approach they would be approximated by solving a system of switching ODEs. The remaining components, that we call *discrete*, are treated according to the mechanisms of the original CTMC. The process resulting from our new approach is a hybrid switching jump diffusion (HSJD) (cf. [20]) that uses jumps to handle both the discrete components and the behavior at the boundaries of the state space.

The above described *partial fluidization* is introduced starting from CTMCs. Yet, as models are usually defined in higher level languages, we show that the approximate jump diffusion process can be derived starting from stochastic Petri nets (SPN) as well.

In some simpler cases, the proposed HSJD can be analyzed analytically by solving the Fokker-Planck partial differential equation of the process. Two such simple illustrative examples will be proposed. When the model is more complex, only simulation is feasible. We will describe a simple algorithm for the simulation and will report the results on some models in systems biology.

In the literature several flavors of hybrid models have been proposed and studied in the recent years. The PDMP class, which is strongly related to the class of HSJD processes we propose, has also been generalized in order to include the case in which the fluid component is a diffusion process. A comparison between our approach and those

proposed in the literature is deferred to Section 3.2. Petri nets with hybrid state spaces were introduced in [21] and generalized in [22]. The aim of these original proposals was both to handle systems in which the number of objects tend to become exceedingly large and to model intrinsically continuous quantities (like temperatures). Processes with hybrid state space are used also as a mean to analyze models in which not all sojourn times are exponentially distributed. In this context the continuous component of the state space is used to keep track of the age of the non-exponential durations. An important work in this direction is by Cox [23]. A recent book on hybrid switching diffusion is [20] which concentrates mainly on the mathematical theory for such processes.

This paper is organized as follows. Section 2 is devoted to provide the necessary background on density dependent families of CTMCs. In Section 3 we derive the stochastic differential equations that characterize the proposed hybrid switching jump diffusion approximation. Simple illustrative examples that can be handled analytically are shown in Section 4. Simulation based numerical illustration is provided in Section 5. Conclusions are drawn in Section 6. An Appendix contains some further material and some more details.

2. Background

This section is devoted to the necessary background on density dependent processes. We follow the exposition we provided in [19] adding different examples (described in 2.2 and in Appendix A) and omitting some details that are not strictly necessary to the scope of this paper. In particular, a notable difference with respect to [19] is that we do not require that the model is with bounded state space.

2.1. Density dependent CTMCs

Density dependent CTMCs often arise when a model describes the interaction of groups of identical objects. The term object is used intentionally to indicate that density dependent processes are present in many contexts. When modeling biological systems the objects are, for example, enzymes, proteins or members of populations. In networks of queues the objects are customers. Informally, the necessary condition for a model to be density dependent is that the rate of the interactions depends on the density of the population levels and not on the absolute population values.

Hereinafter we provide the definitions that are necessary for the rest of the paper. We will denote with $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$ and $\mathbb{N}$ the sets of real, rational, integer, and natural numbers, respectively. Given a positive constant, r, we will denote with $\mathbb{R}^r$ the r–dimensional cartesian product of the space $\mathbb{R}$. The letter u will be used to indicate the time index ranging continuously between $[0, +\infty)$ or $[0, T]$ when specified. The discrete states of a continuous time Markov chain will be denoted as k or h and range in the state space that is included in $\mathbb{Z}^r$. We will always consider the abstract probability space to be given as $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is a non empty set, $\mathcal{F}$ is a σ-algebra on Ω and $\mathbb{P}$ is the probability measure. Furthermore, $\mathbb{E}$ will denote the expectation with respect to $\mathbb{P}$. The formal definition of a family of density dependent CTMCs is the following [4].

Definition 1. *A family of Markov chains $X^{[N]}(u)$ with parameter N and with state space $E^{[N]} \subseteq \mathbb{Z}^r$, is called* density dependent *iff there exists a continuous non-zero function $f : \mathbb{R}^r \times \mathbb{Z}^r \to \mathbb{R}$ such that the instantaneous transition rate (intensity) from state k to state $k + l$ can be written as*

$$q^{[N]}_{k,k+l} = N f\left(\frac{k}{N}, l\right), \quad l \neq 0. \tag{1}$$

The indexing parameter of the family, N, has different meanings depending on the context. It can be the size of the considered area, the total number of considered objects, or the volume in which the interactions take place. The first argument of f in (1) is either the density associated with state k (if N is the area or the volume of the interactions), or the normalized state (if N is the total population size). The second argument is the vector that describes the effect of a transition on the state. The consequence of the fact that every member of the family shares the form given in (1) is twofold. First, in every CTMC of the family, the transitions have the same effect on the state. Second, given a state k, the intensities of the outgoing transitions, $q^{[N]}_{k,k+l}$, depend on k/N (and not on the state itself), and are proportional to the indexing parameter N. In the following we denote the set of possible state changes by C, i.e., $C = \{l \mid l \in \mathbb{Z}^r, l \neq 0, \exists k \in E^{[N]} : q^{[N]}_{k,k+l} \neq 0\}$, and the possible state changes from a given state k by C_k, i.e.,

$C_k = \{l \mid l \in \mathbb{Z}^r, l \neq 0, q^{[N]}_{k,k+l} \neq 0\}$. An example for a family of density dependent CTMCs is provided in Appendix A.1.

Definition 1 can be extended in order to include a larger class of models that can still be treated in the same framework. This leads to the definition of *near density dependence* where we require that the transition rates tend to the form given in (1), when the indexing parameter tends to infinity.

Definition 2. *A family of Markov chains $X^{[N]}(u)$ with parameter N and with state space $E^{[N]} \subseteq \mathbb{Z}^r$, is called* nearly density dependent *iff there exists a continuous non-zero function $f : \mathbb{R}^r \times \mathbb{Z}^r \to \mathbb{R}$ such that the instantaneous transition rate (intensity) from state k to state k + l can be written as*

$$q^{[N]}_{k,k+l} = N\left[f\left(\frac{k}{N}, l\right) + O\left(\frac{1}{N}\right)\right], \quad l \neq 0. \tag{2}$$

In the following subsection we provide an example of a density dependent process. A further example and a discussion of the properties of density dependence are provided in Appendix A.

2.2. Mass action chemical kinetics

Chemical reaction models with rates that follow the law of mass action are (nearly) density dependent families of Markov Chains. As an example let us consider the following chemical reactions

$$A + B \xrightarrow{\nu_1} 2A, \quad 2A \xrightarrow{\nu_2} A + B, \quad 2A + B \xrightarrow{\nu_3} A + B$$

and assume that the intensities of the reactions follow the stochastic law of mass action. This means that the intensity of a reaction is proportional to the number of distinct ways the molecules can form its input. Further, it is inversely proportional to V^{n-1} where V is the volume and n is the number of molecules that form the input of the reaction. The reason for this is that the bigger the volume the less probable that the molecules on the left hand side of the reaction collide. Accordingly the intensities are

$$q_{(i,j),(i+1,j-1)} = \nu_1 \frac{ij}{V}, \quad q_{(i,j),(i-1,j+1)} = \nu_2 \frac{i(i-1)}{2V}, \quad q_{(i,j),(i-1,j)} = \nu_3 \frac{i(i-1)j}{2V^2}$$

where we assumed that the state is described by a pair (i, j) with i providing the number of molecules of A and j the number of molecules of B. The above intensities can be rewritten as

$$q_{(i,j),(i+1,j-1)} = V\nu_1\left(\frac{i}{V}\right)\left(\frac{j}{V}\right), \; q_{(i,j),(i-1,j+1)} = V\left(\frac{\nu_2}{2}\left(\frac{i}{V}\right)^2 - \frac{\nu_2}{2V}\left(\frac{i}{V}\right)\right), \; q_{(i,j),(i-1,j)} = V\left(\frac{\nu_3}{2}\left(\frac{i}{V}\right)^2\left(\frac{j}{V}\right) - \frac{\nu_3}{2V}\left(\frac{i}{V}\right)\left(\frac{j}{V}\right)\right)$$

where the first intensity is in exact density dependent form while the other two contain additional terms in the order of $O(1/V)$ once i and j are fixed. It turns out that this difference does not preclude the use of the approximation framework we consider in this paper. For this reason near density dependence is introduced in Definition 2.

2.3. Approximations of Density dependent CTMCs

Let us turn our attention now to two approximations of density dependent CTMCs, the limiting deterministic process introduced in [4] and the sequence of diffusion processes introduced in [17]. Both of these approximations rely on processes with continuous state space and thus fall into the category of "fluid" approximations. The first approximation employs a set of ordinary differential equations (ODEs) with one equation per group, i.e., it provides a deterministic approximation of the stochastic behavior of the system. It was shown in [4] that, if the number of interacting objects tends to infinity, there exists a formal relation between the approximation provided by the ODEs and the original process. In practice, for a given finite number of interacting objects, the approximation is usually seen as a mean to provide the approximate expected number of objects for each group. The second approximation employs instead stochastic differential equations (SDEs) and is referred to as the diffusion approximation of the process. As it was shown in [17], also in case of the diffusion approximation there exists a formal relation between the original process and the approximation, but, in contrast to the deterministic one, this relation holds for any number of interacting objects and not only in the limiting case. A crucial difference between the two approximations is that

the deterministic one provides only the approximate mean behaviour of the random variables of interest while the diffusion approximation leads to their approximate joint distribution.

In order to introduce the approximations we need to define the family of normalized CTMCs given as $Z^{[N]}(u) = \frac{X^{[N]}(u)}{N}$ with state space $S^{[N]}$, which is also referred to as the density process. Note that using the normalized CTMCs the state spaces of all the members of a density dependent family are brought to the same scale, and thus become comparable.

Based on the properties summarized in Appendix A.2, the following result was shown by Kurtz in [4]. Given a nearly density dependent family of CTMCs $X^{[N]}(u)$, if the limit of the initial conditions tends to z_0, i.e.,

$$\lim_{N\to\infty} Z^{[N]}(0) = \lim_{N\to\infty} \frac{X^{[N]}(0)}{N} = z_0$$

and the function

$$F(y) = \sum_{l\in C} l f(y, l)$$

satisfies some relatively mild conditions, then the density process $Z^{[N]}(u)$ converges to a deterministic function $z(u)$. The function $z(u)$ solves the system of ODEs [1]

$$\begin{aligned} dz(u) &= F(z(u))du, \\ z(0) &= z_0. \end{aligned} \tag{3}$$

The convergence is in the following sense: for every $\delta > 0$ we have

$$\lim_{N\to\infty} \mathbb{P}\left\{\sup_{u\le T} \left|Z^{[N]}(u) - z(u)\right| > \delta\right\} = 0. \tag{4}$$

where T is the upper limit of the considered finite time horizon.

The function $z(u)$ is usually interpreted as the asymptotic mean. The difference $Z^{[N]}(u) - z(u)$ can be seen instead as the "noisy" part of $Z^{[N]}(u)$. It was shown in [4] that for $N \to \infty$ the density process $Z^{[N]}(u)$ flattens out at its mean value and that the magnitude of the noise around the mean is[2]

$$Z^{[N]}(u) - z(u) = O\left(\frac{1}{\sqrt{N}}\right). \tag{5}$$

In practice, the convergence given in eq. (4) is often used in case of a finite N to approximate the stochastic process $X^{[N]}(u) = N * Z^{[N]}(u)$ with the deterministic function $x^{[N]}(u) = N * z(u)$. This approximation disregards the noise term which is in the order $N * O(1/\sqrt{N}) = O(\sqrt{N})$ that is small compared to the order of the mean (that is N), but not in absolute terms. Moreover, it ignores every detail of the probability distribution of $X^{[N]}(u)$ except for its mean. It is easy to see that there are cases, e.g. multimodal or highly variable distributions, where the mean gives little information about the actual location of the probability mass, cf. [24].

Let us stress that the convergence holds only if $\lim_{N\to\infty} Z^{[N]}(0) = \lim_{N\to\infty} \frac{X^{[N]}(0)}{N} = z_0$, meaning that the corresponding sequence of initial conditions $X^{[N]}(0)$ needs to grow linearly with N. This implies that if $X^{[N]}(u)$ is multivariate then each non-zero entry of the vector describing the initial state of the process has to grow with the same rate.

An approximation of a density dependent family $X^{[N]}$ which preserves its stochastic nature and has a better order of convergence was proposed in [17, 25]. In order to introduce this approximation, let us denote by $H(S^{[N]})$ the convex envelope of the state space of the density process $Z^{[N]}$ which can be seen as the potential state space of the continuous approximation of the discrete process.

[1] Equation (3) is equivalent to the form $\frac{dz(u)}{du} = F(z(u))$. We have chosen the "differential" form written in (3) to be consistent with the notation that will be introduced for the stochastic differential equations.

[2] When used to compare a pair of stochastic processes $A(u)$ and $B(u)$, the O notation has the following precise meaning: $A(u) - B(u) = O(g(N))$, for some function $g(N)$ that is infinitesimal when $N \to \infty$, if and only if there exists an almost surely finite random variable Γ_T with finite moments of any order, such that $\sup_{u\le T}|A(u) - B(u)| \le \Gamma_T g(N)$.

In [17, 25] it has been shown that there exists an open set $E \subset H(S^{[N]})$ (i.e., a subspace that does not contain the boundaries of $H(S^{[N]})$) in which the density process $Z^{[N]}$ can be approximated by a diffusion $Y^{[N]}(u)$ with state space E. The diffusion $Y^{[N]}(u)$ is characterized by the system of SDEs

$$dY^{[N]}(u) = F(Y^{[N]}(u))du + \sum_{l \in C} \frac{l}{\sqrt{N}} \sqrt{f(Y^{[N]}(u), l)}\, dW_l(u) \tag{6}$$

where the $\{W_l(u)\}$ are independent standard one-dimensional Brownian motions and f is given in eq. (1). The approximation holds up to the first time $Y^{[N]}(u)$ leaves E.

The structure of eq. (6) is the following: the first term is the same that appears in eq. (3), while the second term represents the contribution of the noise and is responsible for the stochastic nature of the approximating process $Y^{[N]}$. A further relation between eq. (6) and eq. (3) can be obtained by considering that the stochastic part of the equation is proportional to $1/\sqrt{N}$, meaning that as $N \to \infty$, this term becomes negligible and $Y_\infty(u)$ solves the same ODE written in eq. (3). Let us remark that the construction of such noise is not based on an ad hoc assumption, but is derived formally from the original Markov chain.

A rigorous mathematical treatment of SDEs can be found in [26, 27] where Itō calculus is introduced. In the physical literature the notation $\frac{dW(u)}{du} = \xi(u)$ is often used even if Brownian motion is nowhere differentiable and $\xi(u)$ is called a *gaussian white noise*. This SDE approach that goes back to the already cited [17, 25] has been applied in many contexts, e.g., it is used under the name of *Langevin equations* to model chemical reactions in [28]. Let us recall that chemical reaction models are nearly density dependent.

As for the relation of the diffusion approximation and the original density process, in [17] it has been proven that, for any finite N, we have

$$Z^{[N]}(u) - Y^{[N]}(u) = O\left(\frac{\log N}{N}\right) \tag{7}$$

which, compared to eq. (5), is a better convergence rate. Thus, the process $N * Y^{[N]}(u)$ approximates the CTMC $X^{[N]}(u)$ with an error of order $\log N$ which is much lower than the $\sqrt{N}$ of the deterministic fluid approximation.

Finally, let us stress that the approximation is valid only up to the first exit time from the open set E. For many applications the natural state space is bounded and closed and the process may reach the boundary of E in a finite time τ with non-negligible probability. In such cases, since the approximating process $Y^{[N]}(u)$ is no longer defined for any $u \geq \tau$, the diffusion approximation is not applicable. To overcome this limitation suitable boundary conditions must be set and this problem, that was considered neither in [17] nor in [25], will be tackled in Section 3.1.

In Appendix A.3 it is shown that density dependent families of CTMCs (including models of chemical kinetics) often arise from Stochastic Petri Nets (SPNs) models and we describe how to translate the general theory here proposed into the language of SPNs.

3. Jump-diffusion approximations and the hybrid switching extention

In many biological systems of practical interest the applicability of the fluid approximations we have introduced in Section 2 is limited by several factors: (a) the presence of boundaries in the state space that are visited with non-negligible probability, (b) the explicit violation of the density dependent property, (c) the low population levels that make the fluid approximation inaccurate, and (d) the presence of "slow components" that renders the fluid approximation ineffective. The aim of this section is to introduce a new model that relies on hybrid switching jump diffusion (HSJD) processes and that is able to give a reliable approximation of Markov models for reaction networks even in the presence of these factors.

3.1. From CTMCs to jump diffusion processes

At the boundary of the state space, indeed, some of the state changes that are possible in the interior are not enabled any longer (e.g. an enzymatic reaction cannot occur if there are no enzyme molecules available). If the Markov chain spends some time at the boundary, it may happen that special behaviors, which do not occur or occur rarely in the interior, become significant and cause the appearance of different modes in the joint probability distribution. Such

multi-modal behavior cannot be captured by a deterministic approximation which at most can keep track of the mean value of the marginal distributions of each component (cf. [29]). Not even the diffusion approximation given in eq. (6) solves the problem since it is defined only in the interior of the state space and when the boundary is reached its validity ceases unless suitable boundary conditions that are not specified in the original literature are imposed. In [19] the problem was solved in the context of density dependent Markov models that are described by SPNs whose transitions fire with rates proportional to their enabling degrees (i.e., depend on the number of tokens in their input places) and whose places are all covered by P-invariants (i.e., the state space is bounded; for more details on the SPN formalism see [30]). In particular, in [19] a jump diffusion approximation has been introduced that we now recast in the more general setting of nearly density dependent models.

Let us consider a nearly density dependent Markov chain $X^{[N]}(u)$ and its normalized version $Z^{[N]}(u)$. $Z^{[N]}$ has state space $S^{[N]} \subseteq \mathbb{Q}^r$ and it is such that the possible values of some of its components can be bounded by a minimal and/or by a maximal value. Such minimal and maximal values are present, for example, in a chemical reaction model where the number of molecules of a given chemical compound cannot become negative, or in a population model, where any given sub–population can neither become negative nor can exceed the total population size.

The instantaneous transition rates of the normalized Markov chain $Z^{[N]}$ will be denoted by $p^{[N]}_{x,x+l/N}, x \in \mathbb{Q}^r, l \in \mathbb{Z}^r$. We will use the same notation for the transition rates extended in the natural way to the state space of the diffusion approximation where $x \in \mathbb{R}^r$. Let us recall that the natural state space of the diffusion approximation $Y^{[N]}(u)$ is the convex hull of the state space of the Markov chain, $H(S^{[N]}) \subset \mathbb{R}^r$. Notice however that $Y^{[N]}$ is defined only in the interior of such space and up to the first time the process visit its boundary. In order to extend the diffusion approximation to include the boundary of the state space, we introduce a new approximating process $\tilde{Y}^{[N]}(u)$ which behaves like the diffusion approximation $Y^{[N]}$ in the interior of the state space and that, when the boundary is reached, mimics the jump behavior of the original Markov chain. We describe the process $\tilde{Y}^{[N]}(u)$ more formally in the following paragraphs.

In order to identify those components that are actually at their minimal or maximal values, we define a map $B : H(S^{[N]}) \to \mathcal{P}(\{1, \cdots, r\})$, from the state space of the diffusion to the power set of the set of indexes, which, given a state x, provides the set of indexes of the extremal components. Hence, $B(\tilde{Y}^{[N]}(u))$ is the set of indexes of the components that are on the boundary at time u [3]. Let us notice that if no components of the actual state $\tilde{Y}^{[N]}(u)$ attain the boundaries, then $B(\tilde{Y}^{[N]}(u)) = \varnothing$ and $\tilde{Y}^{[N]}(u)$ has the same behavior as the diffusion $Y^{[N]}(u)$. As soon as $B(\tilde{Y}^{[N]}(u))$ becomes non-empty, the set of possible state changes $C_x = \{l \in \mathbb{Z}^r : l \neq 0,\ p^{[N]}_{x,x+l} \neq 0\}$ is split dynamically (depending on the current state $x = \tilde{Y}^{[N]}(u)$) into two subsets, $\overset{\star}{C}_x$ and $\overset{\circ}{C}_x$. The former contains those state changes that either modify a component which is extremal or which are such that the corresponding rate depends on an extremal component, i.e.

$$\overset{\star}{C}_x = \left\{ l : \exists\, i \in B(x) \text{ such that either } l_i \neq 0 \text{ or } \frac{\partial p^{[N]}_{x,x+l/N}}{\partial x_i} \neq 0 \right\},$$

the latter is the complement set, i.e. $\overset{\circ}{C}_x = C_x - \overset{\star}{C}_x$.

As long as the state changes included in $\overset{\star}{C}_{\tilde{Y}^{[N]}(u)}$ do not occur, the subsystem made of the components with indexes not included in $B(\tilde{Y}^{[N]}(u))$ can still be approximated by diffusion which are analogous to eq. (6) except that the sums are restricted to the state changes in $\overset{\circ}{C}_{\tilde{Y}^{[N]}(u)}$.

The events included in $\overset{\star}{C}_{\tilde{Y}^{[N]}(u)}$ encode the behavior at the boundary. We keep them discrete and we treat them as a jump process which is responsible for all the events of the type "the i-th component leaves the boundary". The amplitudes and the intensities of the jumps are formally taken from the original CTMC and depend on the entire state of the process, i.e., on all its components, no matter whether they are at the boundary or not. The approximating jump

[3] Let us suppose to have a process with two components, X_t and Y_t, both bounded between 0 and 1. The map B applied to the vector (X_t, Y_t) returns the set of the indexes of the components at the boundary. If at time $t = 7$ the process is in $(X_7 = 0.34, Y_7 = 1)$, the map B returns $\{2\}$, meaning that the second component, namely Y_t, lies on one of the boundaries.

diffusion $\tilde{Y}^{[N]}(u)$ which embodies both the fluid evolution and the discrete events solves the following system of SDEs

$$d\tilde{Y}^{[N]}(u) = \sum_{l \in \mathring{C}_{\tilde{Y}^{[N]}(u)}} l f\left(\tilde{Y}^{[N]}(u), l\right) du + \frac{1}{\sqrt{N}} \sum_{l \in \mathring{C}_{\tilde{Y}^{[N]}(u)}} l \sqrt{f\left(\tilde{Y}^{[N]}(u), l\right)} dW_l(u) + \sum_{l \in \overset{\star}{C}_{\tilde{Y}^{[N]}(u)}} \frac{l}{N} dM_l^{[N]}(u) \tag{8}$$

where $M_l^{[N]}(u)$ is the counting process that describes how many events with state change l occurred in the time interval $(0, u]$ and whose intensity is given by

$$\mu_l(\tilde{Y}^{[N]}(u^-)) = p^{[N]}_{\tilde{Y}^{[N]}(u^-), \tilde{Y}^{[N]}(u^-) + \frac{l}{N}}$$

which depends on the actual state of the process $\tilde{Y}^{[N]}$ right before the jump[4]. Let us recall that a counting process is a stochastic process with positive, integer and increasing values. It is the natural model for the number of outcomes in a system over time. As the diffusion given in (6), also (8) has its proper mathematical definition in its integral form, where the integrals with respect to the Brownian motions W_l and with respect to the counting processes $M_l^{[N]}$ have to be defined in the general Itô theory of integration with respect to semimartingales [26, 27].

Equation (8) is a system of equations, one for every component of the process. This might seem contradictory with our previous description according to which only the components not included in $B(\tilde{Y}^{[N]}(u))$ are fluidized. Let us however remark that the fluid increments in the first sum of (8) do not affect the component at the boundary since if $l \in \mathring{C}_x$ then the ith entry of l is 0 for any $i \in B(x)$. On the other hand, a component that is not at the boundary at a given time is moved by the continuous compounding of the fluid increments that sums up with the effect of the jumps.

3.2. From CTMCs to hybrid switching diffusions with jumps

Many real systems violate the basic assumption that justifies the fluid approximation, i.e., that events are very frequent and cause very small state changes. An example is a Markov chain in which some of the transition rates have the nearly density dependent form of (2) and others do not. It is reasonable to "fluidize" the system partially, making the effect of the density dependent state changes continuous, but keeping the other state changes as discrete events. Another example is a Markov model whose components have values ranging in different scales. For example, when some components of the system model resources with limited availability. Even if the rates are all nearly density dependent, it is natural to consider a family of initial conditions where the components not related to the resources grow large with the indexing parameter of the family, while those modeling resources are kept fixed. Accordingly, we propose to fluidize only those components whose increase is not in contrast with the modeling assumptions. Moreover, if the transition rates follow the law of mass action given in eq. (A.6), the reactions that involve reactants of which there are only a few are typically slower than those that involve the abundant species, introducing different time scales in the reaction rates. A better approximation is achieved if the components with small values and the reactions with law rates are kept discrete, while the rest is fluidized.

Different hybrid approaches of this kind have been adopted in the literature and we list a few here for comparison. A special mention has to be done to Piecewise Deterministic Markov Processes (PDMPs, cf. [31] for the mathematical theory and [32, 33, 34, 35, 36, 37] for applications in computer science and biology) that, as we shall see, are the ODE based counterpart to our HSJD approach. Even hybrid approaches based on diffusion have already appeared in the literature, mainly using switching diffusions (see [20] for the mathematical theory and [38, 39, 40, 33] for applications) but, as far as we know, no treatment of the behavior at the boundaries has been included into such models. Other general and popular approaches to hybrid systems where the fluid components are evolving according to a diffusion have been studied in [41, 42, 33] under the name of (Generalized) Stochastic Hybrid Models. Such models have a different origin (aircraft dynamics) and have not been introduced as approximations of Markov chains. In their dynamics, reaching of the boundaries of the state place has a role, but it is a different one. Moreover, the partitioning of the state space between fluid and discrete components is static, independent of the system evolution. The specific contribution of this paper is to adopt the jump diffusion paradigm for the partially fluidized system that allows the fluid components to visit the boundaries and to jump back in the interior of the state space in almost the same way as the original chain does.

[4] A function f evaluated in u^- is defined as the left-sided limit $f(u^-) = \lim_{x \to u^-} f(x)$.

For the sake of clarity, we will first introduce the process disregarding the behavior at the boundary. The machinery that allows the visit of the boundaries will be added later on.

Let us hence introduce a partial fluidization of a family of r-dimentional Markov chains $X^{[N]}$. The formal requirements on $X^{[N]}$ for the approximation to hold will be made clear after having introduced the necessary notation. Intuitively, we need that a subsystem of the original model containing a subset of its components and a subset of its possible state changes is nearly density dependent. This subsystem is fluidized and all the rest of the system (the "problematic" part) is left discrete.

We assume that we are given the partition of the set of indexes $\{1,\dots,r\}$ defined by two disjoint subsets: F, with cardinality n, and D, with cardinality m such that $n+m=r$, and $F\cup D=\{1,\dots,r\}$. The components $X_i^{[N]}, i\in F$, will be fluidized and their initial condition will be proportional to the indexing parameter N. The initial condition for the components $X_i^{[N]}, i\in D$, will remain instead constant as N increases. The partition of the components is supposed to be explicitly given as part of the model itself. The process we aim to approximate is the partially scaled Markov chain $Z^{[N]}$ in which only the components in F are divided by N (and will be approximated by a diffusion) while the components in D are kept as they are. Consequently, $Z_i^{[N]}=X_i^{[N]}$ for all $i\in D$ and $Z_j^{[N]}=\frac{X_j^{[N]}}{N}$ for all $j\in F$. Let us denote the state space of $Z_j^{[N]}$ by $S'^{[N]}$. A transition of the chain $X^{[N]}$ causes a state change from state k to state $k+l$ with a rate $q_{k,k+l}^{[N]}$. The corresponding transition of the partially scaled chain $Z^{[N]}$ moves the chain from state x to state $x+l'$ with rate $p_{x,x+l'}^{[N]}=q_{k,k+l}^{[N]}$ where the state change l' is such that $l'_i=l_i$ for all $i\in D$ and $l'_j=\frac{l_j}{N}$ for all $j\in F$.

The partition of the components requires to partition the possible state changes $C=\{l\in\mathbb{Z}^r:\ l\neq 0 \text{ and } \exists\ x: p_{x,x+l'}^{[N]}\neq 0\}$ into two disjoint sets C^F and C^D, which distinguish the fluid events that happen continuously by diffusion and the discrete ones that happen by jumps. This partitioning is made on the basis of the following criteria. Those transitions that modify the discrete components necessarily cause discrete increments and hence belong to C^D. The events whose rates depend on a discrete component are also treated as discrete because their rates could be too slow and the number of involved object could be too few to justify a continuous approximation. The remaining events, which are with rates in the nearly density dependent form, are in C^F. Formally we have

$$C^F=\left\{l\in C:\ l_i=0 \text{ and } \frac{\partial p_{x,x+l'}^{[N]}}{\partial x_i}=0,\ \forall i\in D, x\in S'^{[N]}\right\}.$$

Let us remark that the condition $l_i=0,\forall i\in D$, is strictly necessary for preserving the discreteness of the components in D while the condition regarding the transition rates is introduced mainly to increase the quality of the approximation and might be dropped in the case one needs to simulate the model faster (but obviously with less accuracy). The remaining state changes are included in $C^D=C-C^F$. Let us stress that the above partition of the state changes is static, i.e., it does not depend on the current state of the process.

We approximate the partially scaled process $Z^{[N]}$ with the process $\Upsilon^{[N]}$ whose components evolve with hybrid switching jump diffusion dynamics described by the following equations:

$$\begin{aligned} d\Upsilon_j^{[N]}(u) &= \sum_{l\in C^F} l_j f\left(\Upsilon^{[N]}(u),l\right)du + \sum_{l\in C^F}\frac{l_j}{\sqrt{N}}\sqrt{f\left(\Upsilon^{[N]}(u),l\right)}dW_l(u) + \sum_{l\in C^D}\frac{l_j}{N}\,dJ_l^{[N]}(u), \qquad j\in F,\\ d\Upsilon_i^{[N]}(u) &= \sum_{l\in C^D} l_i\,dJ_l^{[N]}(u), \qquad i\in D \end{aligned} \tag{9}$$

where $J_l^{[N]}(u)$ is the process counting the events that causes a partially scaled state change l' (corresponding to a state change l in the original chain) which occurred in the time interval $(0,u]$ and whose intensity is given by

$$\nu_l(\Upsilon^{[N]}(u^-)) = p^{[N]}_{\Upsilon^{[N]}(u^-),\Upsilon^{[N]}(u^-)+l'} \tag{10}$$

In order to include the behavior on the boundary of the state space for the components in F, we proceed as explained in the previous section. The set C^F is split dynamically (depending on the current state of the process x) into two subsets $\overset{\star}{C}{}_x^F$ and $\overset{\circ}{C}{}_x^F$ defined as

$$\overset{\star}{C}{}_x^F=\left\{l\in C^F:\ \exists\ i\in B(x)\cap F \text{ such that either } l_i\neq 0 \text{ or } \frac{\partial p_{x,x+l'}^{[N]}}{\partial x_i}\neq 0\right\},$$

and $\mathring{C}^F_x = C^F - \overset{\star}{C}{}^F_x$, respectively. Finally, the hybrid switching jump diffusion process $\tilde{\Upsilon}^{[N]}(u)$ accounting both for the hybrid definition of the model and for the boundary conditions is given as

$$
\begin{aligned}
d\tilde{\Upsilon}^{[N]}_j(u) &= \sum_{l \in \mathring{C}^F_{\tilde{\Upsilon}^{[N]}(u)}} l_j f\left(\tilde{\Upsilon}^{[N]}(u), l\right) du + \sum_{l \in \mathring{C}^F_{\tilde{\Upsilon}^{[N]}(u)}} \frac{l_j}{\sqrt{N}} \sqrt{f\left(\tilde{\Upsilon}^{[N]}(u), l\right)} dW_l(u) + \sum_{l \in C^D \cup \overset{\star}{C}{}^F_{\tilde{\Upsilon}^{[N]}(u)}} \frac{l_j}{N}\, dJ^{[N]}_l(u), \qquad j \in F, \\
d\tilde{\Upsilon}^{[N]}_i(u) &= \sum_{l \in C^D} l_i\, dJ^{[N]}_l(u), \qquad i \in D,
\end{aligned} \tag{11}
$$

where, again, $J^{[N]}_l(u)$ is the process counting the events that cause a partially scaled state change l' (corresponding to a state change l in the original chain) which occurred in the time interval $(0, u]$ and whose intensity is given by

$$
\nu_l(\Upsilon^{[N]}(u^-)) = p^{[N]}_{\Upsilon^{[N]}(u^-), \Upsilon^{[N]}(u^-)+l'}. \tag{12}
$$

The resulting process is a *hybrid switching jump diffusion* in which the jumps account both for the jumps of intrinsically discrete components and for the jumps out of the boundaries of the fluid components. Let us remark that *hybrid* jump diffusions are not a generalization of jump diffusion processes. On the contrary they can be seen as a special case, since the discrete component can simply be described as a jump-diffusion process that does only have jumps and no diffusion.

Let us note that in the limit when $N \to \infty$ the noise in the first equation of (11) vanishes and the model becomes a PDMP in which the discrete components modulates the ODE and the fluid levels have an influence on the intensity (12) of the jumps. The mathematical theory of hybrid switching diffusions together with the special case of PDMP can be found in the recent monograph [20] where also some hybrid switching jump-diffusions are briefly sketched in the appendix.

3.3. Simulation of the proposed hybrid switching jump diffusion

From (11) an Euler scheme based, approximate simulation algorithm can be derived directly. Assume that the state of the process is known at time u and that we aim to simulate the process in a short time interval. The length of this interval is Δt if no discrete state change occurs in $[u, u+\Delta t]$ and it is shorter otherwise. First, we check if a discrete state change occurs in the time interval $[u, u+\Delta t]$. For this purpose, we generate an exponential random variable whose rate, ν, is equal to the sum of the intensities of the involved counting processes, i.e., $\nu = \sum_{l \in C^D \cup \overset{\star}{C}{}^F_{\tilde{\Upsilon}^{[N]}(u)}} \nu_l(\Upsilon^{[N]}(u))$. Let us denote this random variable by r. If $r > \Delta t$ then no discrete state change occurs in the considered interval. If $r \le \Delta t$ then we have to simulate which counting process generates the event. This is done by generating a discrete random variable according to a discrete probability distribution with mass function $m_l = \nu_l(\Upsilon^{[N]}(u))/\nu$ with $l \in C^D \cup \overset{\star}{C}{}^F_{\tilde{\Upsilon}^{[N]}(u)}$. We denote the resulting random variable by k. In order to simulate the diffusion we need to generate $|\mathring{C}^F_{\tilde{\Upsilon}^{[N]}(u)}|$ random variables according to the standard normal distribution. Let us denote these random variables by W'_l with $l \in \mathring{C}^F_{\tilde{\Upsilon}^{[N]}(u)}$. Having generated the above random numbers the state of the process is updated first according to

$$
\begin{aligned}
\tilde{\Upsilon}^{[N]}_j(u + \min(r, \Delta t)) =& \tilde{\Upsilon}^{[N]}_j(u) + \sum_{l \in \mathring{C}^F_{\tilde{\Upsilon}^{[N]}(u)}} l_j f\left(\tilde{\Upsilon}^{[N]}(u), l\right) \min(r, \Delta t) + \sum_{l \in \mathring{C}^F_{\tilde{\Upsilon}^{[N]}(u)}} \frac{l_j}{\sqrt{N}} \sqrt{f\left(\tilde{\Upsilon}^{[N]}(u), l\right)} W'_l \sqrt{\min(r, \Delta t)} + \\
& I(r \le \Delta t) \frac{l_k}{N} \qquad j \in F \\
\tilde{\Upsilon}^{[N]}_i(u + \min(r, \Delta t)) =& \Upsilon^{[N]}_i(u) + I(r \le \Delta t) l_k \qquad i \in D
\end{aligned}
$$

where I is the indicator function. Due to the presence of the diffusion, it is possible that the previous update results in a state in which some components are smaller than their possible minimal or larger than their possible maximal values. If a component is smaller (greater) than its minimal (maximal) value then we set it to its minimal (maximal) value. In other words, we simulate the process in $[u, u + \Delta t]$ as if it was unbounded and subsequently we test whether the process left its valid state space. If Δt is sufficiently small then the above simulation procedure results in traces that reflect properly the behavior of the original process. The choice of Δt can be made on the basis of the actual

state of the process by taking into account the maximum among the drifts. We found that for our examples $\Delta t = \max\left\{l_j f\left(\tilde{\Upsilon}^{[N]}(u), l\right) \text{ with } l \in \mathring{C}^F_{\tilde{\Upsilon}^{[N]}(u)}\right\}^{-1}/100$ is a reasonable compromise between execution time and precision. A more implementation oriented description of the simulation algorithm is provided in Appendix B.

Note that with this simulation approach the borders are reached with delay. For instance, if a component is positive at time u and it is updated to a negative value at time $u + \Delta t$, then we assume that the process reached level 0 at time $u + \Delta t$. It can also happen that we miss the fact that the process reached a border. Indeed there is a positive probability that an unbounded diffusion process is positive at time u and at time $u+\Delta t$ and it is negative somewhere inside $[u, u+\Delta]$. With sufficiently small Δt this causes insignificant imprecision. A more precise treatment of the borders, with which the time when the process reaches the border is estimated more precisely, is possible and we plan to experiment with it in the future.

Let us remark that the simulation algorithm described above is similar to the hybrid stochastic simulation methods based on dynamic partitioning (e.g., [**? ?**]). The main difference is that, while these hybrid simulation methods were derived heuristically through clever simulation speed ups, our simulation algorithm arises by applying a simple Euler discretization scheme to the SDEs (11) that describe the approximating HSJD model.

4. Analytic calculation

In the previous section we proposed a stochastic process, namely a hybrid switching jump diffusion (HSJD), to approximate a class of CTMCs. In the case when the system contains a single fluid component, the process (11) reduces to the *elementary return process* studied by Feller in[5] [43, 44]. In this section we illustrate the possibility of calculating the distribution of the involved quantities by numerical integration of the Fokker-Plank equations there introduced. To this end, we first analyze a simple model without switching and then extend it with a switch.

Let us consider the so called *crazy clock reaction* (cf. [45] and references quoted therein). It is an autocatalytic system composed of two reactions, $A \to B$ and $A + B \to 2B$, and initial state $(N, 0)$ corresponding to N molecules of type A and 0 molecules of type B. The intensities associated with the two transitions in state (i, j) are $\lambda_1 i$ and $\lambda_2 ij/N$, respectively. The second reaction is usually much faster since it has a quadratic rate, but it cannot occur until the first molecule of B has been produced by means of the first reaction.

As both reactions transform one A into one B, we have the invariant $i + j = N$ and hence a single variable is sufficient to describe a state. In the following we will use the number of As as state descriptor. In the associated CTMC the intensity of the possible transitions is $q_{(i),(i-1)} = \lambda_1 i + \lambda_2 i(N - i)/N$. Accordingly, the function f required by Definition 1 is

$$f(x, l) = \begin{cases} \lambda_1 x + \lambda_2 x(1 - x) & \text{if } l = -1 \\ 0 & \text{otherwise} \end{cases}$$

We evaluate the model with $\lambda_1 = 3, \lambda_2 = 6000$ and $N = 1000$. Note that after normalization the variable $x = \frac{i}{N}$ is between 0 and 1.

Since there is no switch in the model, the HSJD approximation can be derived from (8), assuming the following form

$$\begin{aligned} d\tilde{Y}^{[N]}(u) = -I\left(0 < \tilde{Y}^{[N]}(u) < 1\right)\left(f\left(\tilde{Y}^{[N]}(u), -1\right)du + \frac{1}{\sqrt{N}}\sqrt{f\left(\tilde{Y}^{[N]}(u), -1\right)}dW_{-1}(u)\right) \\ -I\left(\tilde{Y}^{[N]}(u) = 1 \text{ or } \tilde{Y}^{[N]}(u) = 0\right)\frac{1}{N}dM^{[N]}(u) \end{aligned} \tag{13}$$

where I is the indicator function and the intensity of the counting process, $M^{[N]}(u)$, is $Nf(\tilde{Y}^{[N]}(u), -1)$, that is $N\lambda_1$ when $\tilde{Y}^{[N]}(u) = 1$ and 0 when $\tilde{Y}^{[N]}(u) = 0$. In (13) the first term on the right hand side accounts for the drift and for the diffusion which is "switched off" when the process is at the upper boundary. The second term is the jump process that moves the process away from the upper boundary. The initial condition is $\tilde{Y}^{[N]}(0) = 1$. Let us remark again that at the lower boundary $f(0, -1) = 0$ and hence the intensity of the jumps is zero so that the lower boundary is absorbing.

[5]The elementary return process is described in details in [43], but be aware of the numerous misprints in the formulae of that paper. A less readable, but misprint free version is contained in [44].

The probability density function (pdf) of the quantity of As at time u is mixed, it has a probability mass at the boundaries (at $x = 0$ and at $x = 1$) and it is continuous elsewhere. The continuous part will be denoted by $\pi(u, x)$, i.e.,

$$\pi(u, x) = \frac{\partial}{\partial x}\mathbb{P}\left\{\tilde{Y}^{[N]}(u) \leq x | \tilde{Y}^{[N]}(0) = 1\right\} \text{ for } 0 < x < 1$$

We will refer to the probability masses as $\pi_0(u)$ and $\pi_1(u)$, i.e.,

$$\pi_0(u) = \mathbb{P}\left\{\tilde{Y}^{[N]}(u) = 0 | \tilde{Y}^{[N]}(0) = 1\right\}, \quad \pi_1(u) = \mathbb{P}\left\{\tilde{Y}^{[N]}(u) = 1 | \tilde{Y}^{[N]}(0) = 1\right\},$$

Furthermore, we will denote by $\pi(u, 0^+)$ and $\pi(u, 1^-)$ the limit values of the continuous density at 0 and 1, respectively.

The time evolution of $\pi(u, x)$ is described by a Fokker-Planck partial differential equation (PDE) equipped with the suitable boundary conditions [43]. We have

$$\frac{\partial}{\partial u}\pi(u, x) = \frac{\partial}{\partial x}\left(f(x, -1)\,\pi(u, x)\right) + \frac{\partial^2}{\partial x^2}\left(\frac{f(x, -1)}{2N}\pi(u, x)\right) + \delta\left(1 - \frac{1}{N}\right)Nf(1, -1)\pi_1(u) \qquad \text{for } 0 < x < N \tag{14}$$

where $\delta(y)$ denotes a Dirac delta distribution centered at y. In (14) on the right hand side, the first term corresponds to the drift of the process, the second to the diffusion coefficient (and these two are standard in Fokker-Planck equations) while the third describes the way the process "jumps away" from the upper boundary. This last term contains a Dirac distribution because the level reached after the jump has a deterministic distribution. It takes into account also the intensity of the corresponding rate of the original CTMC, $Nf(N, -1)$, and depends on the probability mass at $x = 1$ as well. The boundary condition at the upper boundary is given by

$$\frac{\partial}{\partial u}\pi_1(u) = -f(1, -1)\,\pi(u, 1^-) - \frac{\partial}{\partial x}\left(\frac{f(x, -1)}{2N}\pi(u, x)\right)\Bigg|_{x=1^-} - Nf(1, -1)\pi_1(u)$$

where the first two terms provide the rate of net probability flux toward the upper boundary at time t and the third term is the flux exiting from the boundary due to the jumps. At the lower boundary we have $f(0, -1) = 0$ which means that both the drift and the diffusion coefficient are 0. Nevertheless, thanks to the diffusion the lower boundary is reachable. This boundary is absorbing and the probability mass can be calculated simply by

$$\pi_0(u) = 1 - \pi_1(u) - \int_0^1 \pi(u, x)dx$$

The initial condition is

$$\pi_0(u) = 0 \qquad \pi_1(u) = 1 \qquad \pi(0, x) = 0, \text{ for every } 0 < x < N$$

which means that the initial level is 1 with probability 1 (note that we deal with the normalized process). Since both boundaries are regular (cf. [43]), to single out the correct solution we need to impose the following further boundary conditions

$$\pi(u, 1^-) = 0 \qquad \text{and} \qquad \lim_{x \to 0^+} x^2\pi(u, x) = 0.$$

The PDE in (14) together with the boundary and initial conditions can be solved numerically by discretizing the involved variables (u and x) and by applying a finite volume scheme. In Figure 1 we depict the distribution of the unnormalized quantity of As for two time points. The figures show three probability mass functions (pmf). The first one was obtained by solving the original CTMC, the second was calculated by the finite volume scheme applied to the PDE given in (14) and the third was obtained by simulating the HSJD given in (13). We created 10^6 simulation traces, which took about two hours to complete, and still insufficient to produce a smooth pmf. The finite volume scheme for the PDE was implemented in scientific python and the calculations took about 10 seconds. There is a good agreement between the results based on simulation and those based on the PDE. The comparison against the pmf obtained based on the CTMC reveals that the HSJD is a good approximation of the original process. In Figure 2 we depicted the average amount of As obtained from the original CTMC, from the HSJD approximation (using the PDE) and from the ODE approximation. There is a large time interval where the ODE approximation gives largely imprecise idea of the amount of As in the system. At time $u = 0.002$ the ODE curve is at about 12 while the mean in the original model is at about 127.

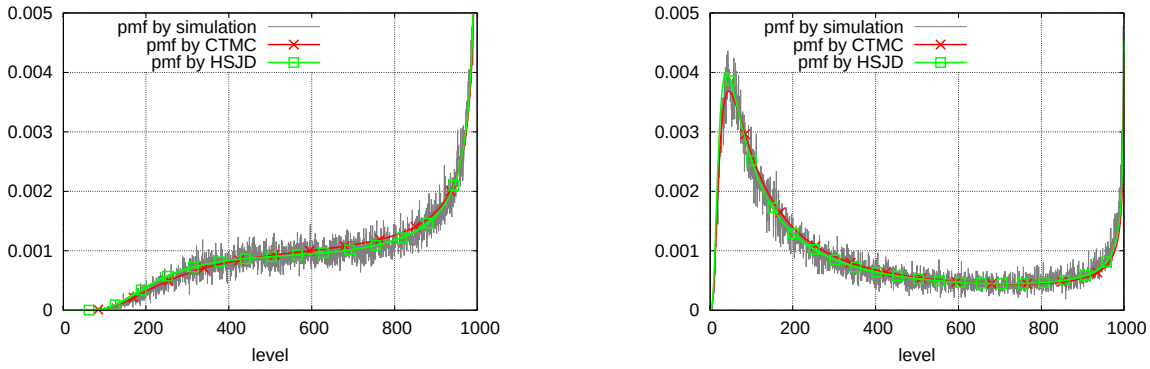

Figure 1. Probability mass function of the quantity of As in the model $A \to B, A + B \to 2B$, at time $u = 0.0012$ (left) and at time $u = 0.0016$ (right)

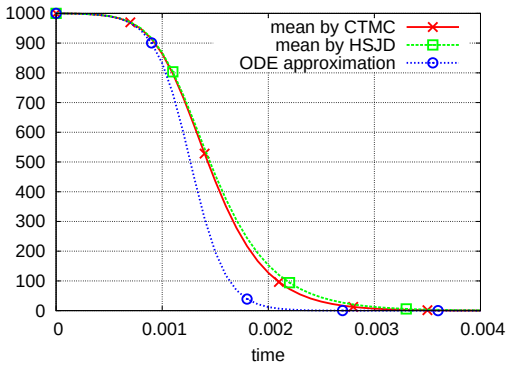

Figure 2. Average quantity of As in the model $A \to B, A + B \to 2B$, as function of time

The explanation for this behavior of the ODE approximation is that in the CTMC and in the HSJD the second reaction ($A + B \to 2B$) is inhibited up to the point when the other reaction ($A \to B$) takes place for the first time. On the contrary, in the ODE model the terms of the equations corresponding to the two reactions both start to contribute at the very beginning of the evolution of the model, resulting in an anticipated decay. This interpretation is confirmed by Fig. 1: at time $u = 0.0016$ the increase of the pmf toward 1000 indicates that there are realizations in which no reaction has taken place yet or the first took place shortly before.

We extend now the model with a one-way switch that modulates the speed of the reaction $A + B \to 2B$. The switch is represented by a third compound denoted by C. We have now three reactions:

$$A \to B, \quad A + B \to 2B, \quad C \to \emptyset$$

and the corresponding initial state is $(N, 0, 1)$. In state (i, j, k) the intensity of the first reaction is $\lambda_1 i$, the intensity of the second reaction is $\lambda_2 ij/N$ if $k = 1$ and it is $2\lambda_2 ij/N$ if $k = 0$ (i.e., the second reaction is two times faster if there is no C in the system), and the intensity of the third reaction is $c(i) = \max(0, \lambda_3(i - (N - S))/S)$. This means that the third reaction can occur only if $i > (N - S)$, its intensity is a linear function of i that reaches its maximum value λ_3 at $i = N$. As before we have $i + j = N$ and hence two variables, i and k, are sufficient to describe a state. In the associated CTMC the intensities of the possible transitions are $q_{(i,1),(i-1,1)} = \lambda_1 i + \lambda_2 i(N - i)/N$, $q_{(i,0),(i-1,0)} = \lambda_1 i + 2\lambda_2 i(N - i)/N$ and $q_{(i,1),(i,0)} = c(i)$. We will apply a HSJD approximation in which the number of As is described by a continuous quantity and the number of Cs is maintained discrete. This leads to a jump diffusion with a switch. We will refer as mode 1 (mode 2) the situation in which the number of Cs in the system is 1 (0). In both mode 1 and 2, the rate of the

reaction that changes the number of As is in density dependent form. The corresponding f function of Definition 1 is

$$f(x,l) = \begin{cases} \lambda_1 x_1 + \lambda_2 x_1(1-x_1) & \text{if } l = (-1,0) \text{ and } x_2 = 1 \\ \lambda_1 x_1 + 2\lambda_2 x_1(1-x_1) & \text{if } l = (-1,0) \text{ and } x_2 = 0 \\ 0 & \text{otherwise} \end{cases}$$

Note that after normalization of the fluid states the quantity $x_1 = \frac{i}{N}$ ranges from 0 to 1. The HSJD approximation is given now in (11). For the continuous part we have

$$\begin{aligned} d\tilde{Y}_1^{[N]}(u) = -I\left(0 < \tilde{Y}_1^{[N]}(u) < 1\right)\left(f\left(\tilde{Y}^{[N]}(u),(-1,0)\right)du + \frac{1}{\sqrt{N}}\sqrt{f\left(\tilde{Y}^{[N]}(u),(-1,0)\right)}dW_{-1}(u)\right) \\ -I\left(\tilde{Y}_1^{[N]}(u) = 1 \text{ or } \tilde{Y}_1^{[N]}(u) = 0\right)\frac{1}{N}dM^{[N]}(u) \end{aligned} \tag{15}$$

which is almost identical to (15) because in this model the transitions that change the discrete component do not change the continuous component. For the discrete component

$$d\tilde{Y}_2^{[N]}(u) = -I\left(\tilde{Y}_2^{[N]}(u) = 1\right)dJ^{[N]}(u) \tag{16}$$

where $J^{[N]}(u)$ is counting the occurrences of the reaction $C \to \emptyset$ with intensity $c\left(\tilde{Y}_1^{[N]}(u)\right)$. The initial condition is $\tilde{Y}^{[N]}(0) = (1,1)$.

In order to describe the transient behavior of the model we need to refer to the pdf of the quantity of As in mode 1 and in mode 2. The continuous part of these densities will be denoted as

$$\pi_1(u,x) = \frac{\partial}{\partial x}\mathbb{P}\left\{\tilde{Y}_1^{[N]}(u) \le x, \tilde{Y}_2^{[N]}(u) = 1 | \tilde{Y}^{[N]}(0) = (1,1)\right\}, \quad \pi_2(u,x) = \frac{\partial}{\partial x}\mathbb{P}\left\{\tilde{Y}_1^{[N]}(u) \le x, \tilde{Y}_2^{[N]}(u) = 0 | \tilde{Y}^{[N]}(0) = (1,1)\right\}$$

with $0 < x < 1$, while the overall pdf of the quantity of As is

$$\pi(u,x) = \pi_1(u,x) + \pi_2(u,x).$$

The mode-specific masses at the boundaries will be referred to as

$$\pi_{1,0}(u) = \mathbb{P}\left\{\tilde{Y}^{[N]}(u) = (0,1) | \tilde{Y}^{[N]}(0) = (1,1)\right\}, \quad \pi_{1,1}(u) = \mathbb{P}\left\{\tilde{Y}^{[N]}(u) = (1,1) | \tilde{Y}^{[N]}(0) = (1,1)\right\}$$

and

$$\pi_{2,0}(u) = \mathbb{P}\left\{\tilde{Y}^{[N]}(u) = (0,0) | \tilde{Y}^{[N]}(0) = (1,1)\right\}, \quad \pi_{2,1}(u) = \mathbb{P}\left\{\tilde{Y}^{[N]}(u) = (1,0) | \tilde{Y}^{[N]}(0) = (1,1)\right\}$$

while their overall counterparts are

$$\pi_0(u) = \pi_{1,0}(u) + \pi_{2,0}(u) \qquad \text{and} \qquad \pi_1(u) = \pi_{1,1}(u) + \pi_{2,1}(u).$$

The Fokker-Planck equation that describes the evolution of the density in mode 1 is

$$\begin{aligned} \frac{\partial}{\partial u}\pi_1(u,x) = &\frac{\partial}{\partial x}\left(f\left((x,1),(-1,0)\right)\pi_1(u,x)\right) + \frac{\partial^2}{\partial x^2}\left(\frac{f((x,1),(-1,0))}{2N}\pi_1(u,x)\right) + \\ &\delta\left(1 - \frac{1}{N}\right)Nf((1,1),(-1,0))\pi_{1,1}(u) - c(x)\pi_1(u,x) \quad 0 < x < 1 \end{aligned} \tag{17}$$

where the last term is due to the switch. Similarly, in mode 2 we have

$$\begin{aligned} \frac{\partial}{\partial t}\pi_2(u,x) = &\frac{\partial}{\partial x}\left(f\left((x,0),(-1,0)\right)\pi_2(u,x)\right) + \frac{\partial^2}{\partial x^2}\left(\frac{f((x,0),(-1,0))}{2N}\pi_2(u,x)\right) + \\ &\delta\left(1 - \frac{1}{N}\right)Nf((1,0),(-1,0))\pi_{2,1}(u) + c(x)\pi_1(u,x) \quad 0 < x < 1 \end{aligned} \tag{18}$$

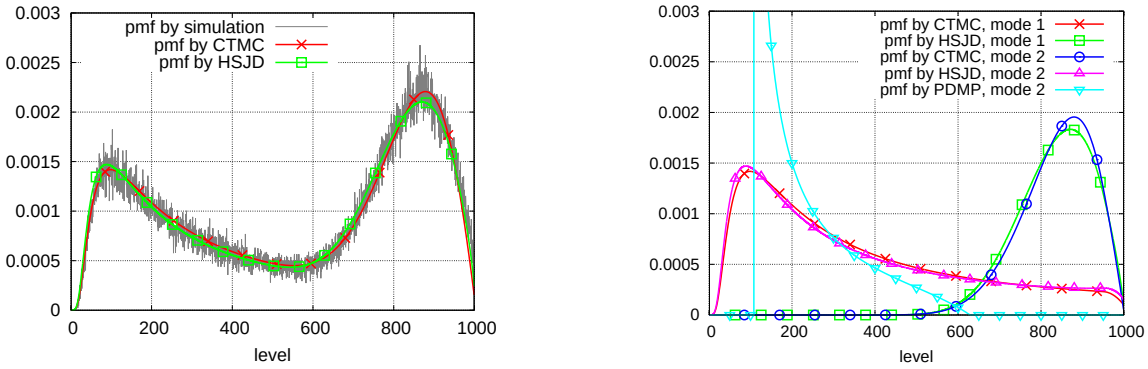

Figure 3. Distribution of the quantity of As at time $u = 0.003$; overall pmf (left) and pmf per mode (right).

Also the boundary conditions at 1 contain a term that takes into account the switch of the model. They are

$$\frac{\partial}{\partial t}\pi_{1,1}(u) = -f((1,1),(-1,0))\,\pi_1(u,1^-) - \frac{\partial}{\partial x}\left(\frac{f((x,1),(-1,0))}{2N}\pi_1(u,x)\right)\Big|_{x=1^-} - Nf((1,1),(-1,0))\pi_{1,1}(u) - c(1)\pi_{1,1}(u)$$

and

$$\frac{\partial}{\partial t}\pi_{2,1}(u) = -f((1,0),(-1,0))\,\pi_2(u,1^-) - \frac{\partial}{\partial x}\left(\frac{f((x,0),(-1,0))}{2N}\pi_2(u,x)\right)\Big|_{x=1^-} - Nf((1,1),(-1,0))\pi_{2,1}(u) - c(1)\pi_{1,1}(u)$$

For the boundary at 0, conditions similar to those reported for the model without switch can be written (note that the intensity of the switch is 0 at the lower boundary). This formulation does not allow one to distinguish the mass at 0 in mode 1 and the mass at 0 in mode 2. It is possible to distinguish these two masses, but this would require a slightly different treatment of the lower boundary.

We considered the model with the switch with the following parameters (in the unnormalized scale): $\lambda_1 = 3, \lambda_2 = 3000, \lambda_3 = 500, S = 50$ and $N = 1000$. We analyzed the model by the method of finite volumes and the execution times are similar to those reported for the model without switch. The distribution of the unnormalized quantity of As at $u = 0.003$ is depicted in Figure 3. Also in this case, there is a good agreement among the numerical solution of the PDEs, the simulation based results and the behavior of the original CTMC. In this example an approach based on pure ODEs is not feasible due to the presence of the switch, while a hybrid approach with a PDMP that switches between two ODEs is reasonable. On the right side of Figure 3 we depicted the distribution obtained by the PDMP approach as well. In mode 1 the PDMP approach leads to a single probability mass at about 848 because this approach provides a deterministic description inside each mode. This mass in not depicted in the figure. In mode 2 the PDMP approach gives a distribution because the time point at which the switch occurs is random. The support of this distribution is determined by the minimal and maximal times at which the switch changes from mode 1 to mode 2. The minimal time is 0 and in this case the level in mode 2 at $u = 0.003$ is about 109. The maximal time is about 0.0022 because in mode 1 the level is $N - S = 950$ at this time point. If the switch snaps at $u = 0.0022$ then the level in mode 2 at $u = 0.003$ is about 629. On the contrary, with the HSJD approach the level in mode 2 can be any value in the interval [0,1000]. The HSJD approach provides a much better approximation of the CTMC with respect to the PDMP one.

5. Simulation based results

In this section we analyze two biological models to provide a comparison of the quality and robustness of the approximation that we have proposed, with respect to more standard approaches. The pure jump-diffusion approximation (all quantities are fluid except at the boundaries) is referred to as SDE approximation both in the figures and

in the text, while the proposed hybrid switching jump diffusion as HSDE (Hybrid SDE). The deterministic approximation is referred to as ODE approximation and our comparisons involve also a PDMP approach which is referred to as HODE (Hybrid ODE).

The first model, which represents a viral intracellular kinetics, shows that

- the SDE approximation is more informative and accurate than that based on ODEs when there is a non-negligible probability to find the original discrete process on the barriers and the quantities under study have bi-modal behaviors;
- although the approximation provided by the SDE is less accurate than that generated by using a HODE approach, the jump-diffusion is more accurate on the barriers;
- the HSDE approach is the most accurate among all the methods discussed in this paper.

The second and more complex model, based on a transcription regulation phenomenon, is introduced to illustrate more clearly the last item of the previous list. This cannot be done with the first model because it is characterized by an exceedingly simple dynamic behavior.

The comparison of the different approaches has been carried out with a prototype implementation of the algorithm described in Section 3.3 and Appendix B, integrated in the GreatSPN framework [46]. A more detailed description of the tool is given in Section 5.3. The results computed by our prototype implementation have been processed through the R framework [48] to derive statistical information and graphics. All the results have been obtained on a 2.13 GHz Intel I7 processor with 8GB of RAM.

5.1. Viral intracellular kinetics

The first model, describes the intracellular kinetics of a generic virus and has been studied in [29]. It is described by the following six reactions:

$$\begin{aligned} gen &\xrightarrow{k_1} tem \\ tem &\xrightarrow{k_2} \emptyset \\ tem &\xrightarrow{k_3} tem + gen \\ gen + struct &\xrightarrow{k_4} \emptyset \\ tem &\xrightarrow{k_5} tem + struct \\ struct &\xrightarrow{k_6} \emptyset \end{aligned} \tag{19}$$

where *gen* represents the genomic viral nucleic acids, *tem* the template of viral nucleic acid transcribed to synthesize every viral component, and *struct* the viral structural protein. In details, reaction k_1 models the integration of the genomic viral nucleic acids into the host genome to form templates. Furthermore *gen* can be packaged (i.e. reaction k_4) within structural proteins to form progeny virus as described by the fourth reaction. After the initial virus infection, the amplification of the viral template is modeled by reaction k_3. Then, the synthesis of the viral structural protein is represented by reaction k_5. Finally, reactions k_2 and k_6 represent the degradation of *tem* and *struct*, respectively. The corresponding SPN model is shown in Figure A.9 of Appendix A.3.

As shown in [29] using linear stability analysis, the system exhibits two equilibriums: one in which all the components are null is unstable, while the other is stable. Hence, initializing the system close to the unstable equilibrium we can observe that the ODE-based methods always reaches the *stable* equilibrium, while stochastic simulation methods can reach also the *unstable* equilibrium with non-negligible probability.

We computed the transient behavior of the model along a time interval that extends from 0 to 200 days using Monte Carlo simulation, ODE, SDE, HODE, and HSDE under the assumptions that a single molecule of *tem* is present in the system at the beginning of the analysis and that the reaction kinetic constants are those reported in Table 1. The hybrid approaches consider *gen* and *tem* as discrete quantities, see [35].

Figure 4 reports a first comparison between the simulation of the original model and the approximations obtained with the four fluid or hybrid approximations. The comparison is performed by depicting the expected number of

μ_1	μ_2	μ_3	μ_4/N	μ_5	μ_6	*gen*	*tem*	*struct*	N
0.025	0.25	1.0	7.5×10^{-6}	1000	2.0	0	1	0	20000

Table 1. Rates and initial condition of the virus model in $days^{-1}$.

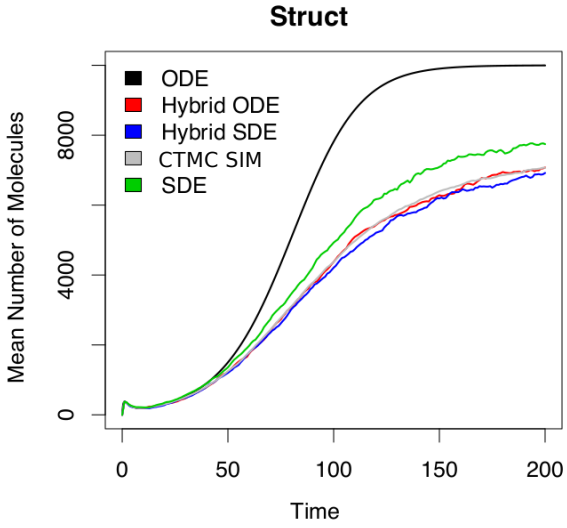

Figure 4. Mean of the number of *struct* moleculas as function of the time. Comparison between Monte Carlo simulation, ODE, HODE, SDE, HSDE.

molecules of *struct* as a function of time. Observe that confidence intervals for Monte Carlo simulation, SDE, and HODE are not explicitly reported on these diagrams to make them more readable; nevertheless these results have been computed using 5000 runs to insure a high level of confidence. The ODE approach deviates quite soon from the trajectory obtained from the simulation of the original process. In particular, it overestimates the expected number of molecules of *tem* by flattening around 10000, while the simulation suggests that this limit should be around 7000. Such inaccuracy is mainly due to the initial low number of reactants (i.e. *tem*), typical of this type of systems. Indeed, due to their deterministic nature, the ODEs cannot fall into the unstable equilibrium state unless they are initialized exactly there, while the CTMC can jump to the state for which the infection is blocked. Although the difference with the original trajectory is still substantial, SDE is much more accurate than ODE; the error between the curve generated using the SDE approach and that obtained by simulating the CTMC is confined within a 10% level. The HODE and HSDE approaches are instead capable of reproducing the mean behavior of the original process in a satisfactory manner.

A better picture of how the four approximations work is provided by Figure 5 where we focus on a single time instant to observe the distribution of the number of molecules of *struct*. In particular, we provide the comparison of the probability distribution of *struct* after 200 days computed by Monte Carlo simulation against those computed by using SDE, HODE and SDE. According to the figure, all three approaches provide a good representation of the original distribution by reproducing the bi-stability of the original process and the overall profile of the distribution. In particular, Figure 5 is structured in such a way that:

- On the background, we provide the distribution obtained by simulating the original CTMC in order to show that it is extremely sparse over the interval $[0, 20000]$ and characterized by a large probability mass, about 25 per cent, in zero.
- On the top of the left side of the figure, we focus on the probability to observe the quantity of *struct* smaller

than 0.1 and the probability to find *struct* extinct; in particular, we provide the comparison between these results obtained by simulating the original process and those generated by using SDE, HODE and HSDE.

- On the right side of the figure, we provide the comparison between the kernel estimates of the probability density function obtained by means of SDE, HODE and HSDE and the histogram generated by using the result of the simulation.

Figure 5 shows that, although all the three fluid approximations are able to reproduce the shape of the original distribution, they have very different behaviors around zero. Specifically, the HSDE approach is able to provide an accurate estimate of the probability mass that is present both *on* the barrier (e.g. in zero) and *around* the barrier (less than 0.1 molecules) whereas: i) the HODE approach, which is by definition not able to reach the barrier, fails to represent the first measure, but provides an accurate estimate of the probability mass present in the interval $(0, 0.1)$; ii) the SDE approximation underestimates both measures. This underestimation is in agreement with the overestimation of the expected number of molecules of *struct* depicted in Figure 4.

Even if all the approaches, but the one based on the ODEs, provide a good approximation of the bi-modal distribution, it is important to highlight that the four methods have very different computational costs. By using an Euler's step of 0.05, the solution of the ODE system required few milliseconds. Keeping the same integration step and computing 5000 trajectories, the SDE approach required $\approx$ 8 seconds, the integration of the HODE system has been obtained in $\approx$ 15 seconds and, finally, the computation of the HSDE trajectories has been carried out in $\approx$ 24 seconds. Thus, the SDE approximation is computationally the cheapest among those that provide a good estimate of the distribution of the process. Finally the Monte Carlo simulation required $\approx$ 210 minutes to compute the same number of trajectories.

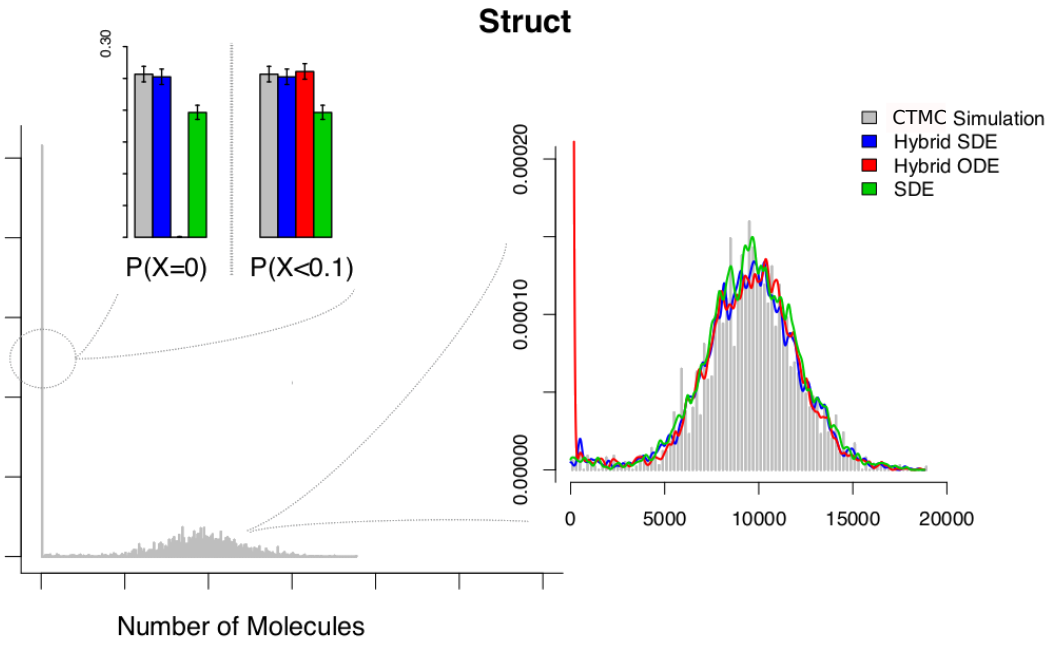

Figure 5. Distribution of *struct* moleculas at time 200.

5.2. Transcription regulation

The second model we consider, a transcriptional regulatory system described in [49], consists of the following 9 reactions

$$\begin{aligned} mRNA &\xrightarrow{k_1} mRNA + M \\ M &\xrightarrow{k_2} \emptyset \\ DNA \cdot D &\xrightarrow{k_3} DNA \cdot D + mRNA \\ mRNA &\xrightarrow{k_4} \emptyset \\ DNA + D &\xrightarrow{k_5} DNA \cdot D \\ DNA \cdot D &\xrightarrow{k_6} DNA + D \\ DNA \cdot D + D &\xrightarrow{k_7} DNA \cdot 2D \\ DNA \cdot 2D &\xrightarrow{k_8} DNA \cdot D + D \\ 2M &\xrightarrow{k_9} D \\ D &\xrightarrow{k_{10}} 2M \end{aligned} \tag{20}$$

and is graphically represented by the SPN reported in Figure A.10 without considering the sub-net in the dashed box.

In details, *mRNA* models the messenger RNA (mRNA) which is translated into a protein M by the reaction k_1. The mRNA transcription (i.e reaction k_3) can happen only when the transcription factor D occupies the DNA binding site R_1. Hence, the DNA binding in position R_1 of D is modeled by reaction k_5; while its unbinding is represented by reaction k_6. Moreover, we assume that a further binding in position R_2, disabling the basal transcription of mRNA, can happen only when the binding R_1 is already occupied by D. This is modeled by reaction k_7. Its corresponding unbinding is instead modeled by the reaction k_8 (i.e. reaction 8). The dimerization of M and D is represented by the reactions k_9 and k_{10}. Reactions k_2 and k_4 model the degradation of mRNA and M.

In order to highlight that the behavior at the boundary of the state space of the fluid components may have a strong impact on the overall dynamics of the system, we extend the model by adding the following three reactions:

$$\begin{aligned} E + M &\xrightarrow{k_{11}} EM \\ EM &\xrightarrow{k_{12}} E + M \\ EM &\xrightarrow{k_{13}} E + P \end{aligned} \tag{21}$$

where E is an enzyme catalyst of the production of a protein P, and EM is the complex generated by the binding between E and M. The corresponding sub-net is represented in the dashed box of Figure A.10 in Appendix A.3. These three additional reactions model the conversion of a protein M into a new protein P catalyzed by an enzyme E according to the well-known mass-action enzyme kinetics [50]. In particular, by means of reaction k_{11}, the enzyme E binds with the protein M to form the complex EM which in turn is converted into the product P and the enzyme E through reaction 13. Finally, reaction k_{12} represents the unbinding between E and M. We assumed as kinetic constants those reported in Table 1, and as an initial state, 40 molecules of D, 2 molecules of *DNA*, and 80 of E (Table3). From the structural analysis of this set of reactions we can observe that *DNA*, $DNA \cdot D$ and $DNA \cdot 2D$ are part of an invariant, so that the sum of their corresponding molecules is always constant. In our experiments this constant is set equal to 2. In this situation, also SDE fails because *DNA*,*DNA-D* and *DNA-2D* are not large enough to be approximated with a diffusion process. For this reason, we considered *DNA*,*DNA-D* and *DNA-2D* as discrete quantities and performed the transient analysis of the model up to 720 seconds by using only the hybrid approaches.

Figure 6(a) provides the comparison between the probability distributions of E after 720 seconds obtained with Monte Carlo simulation and those computed with the HSDE and the HODE approaches whereas Figure 6(b) reports the same measures obtained for M. Both the figures highlight how the HSDE approach provides a better approximation with respect to that of the HODE. Indeed, the HODE approach shows (in both the figures) a peak that is almost the

μ_1	μ_2	μ_3	μ_4	μ_5/N	μ_6	μ_7/N	μ_8	$\mu_9/(2 \cdot N)$	μ_{10}	μ_{11}/N	μ_{12}
0.043	0.0001	0.72	0.0039	0.014	0.48	0.00014	$8.8\ 10^{-12}$	0.029	0.5	0.001	0.0001

Table 2. Rates of the virus model in sec^{-1}

D	*DNA*	*DNA-D*	*DNA-2D*	*mRNA*	*M*	*E*	*EM*	*P*	*N*
40	2	0	0	0	0	80	0	0	650

Table 3. Initial condition of the virus model

double of that provided by the simulation of the original process. Furthermore, HSDE gives a good estimate of the probability mass present on the barriers whereas by construction the HODE approach is unable to reach the barriers. At last, the computation of 5000 trajectories by using 0.05 as integration step required ≈ 30 seconds with the HODE approach and ≈ 74 with HSDE. The Monte Carlo simulation of the same number of trajectories required ≈ 10 minutes.

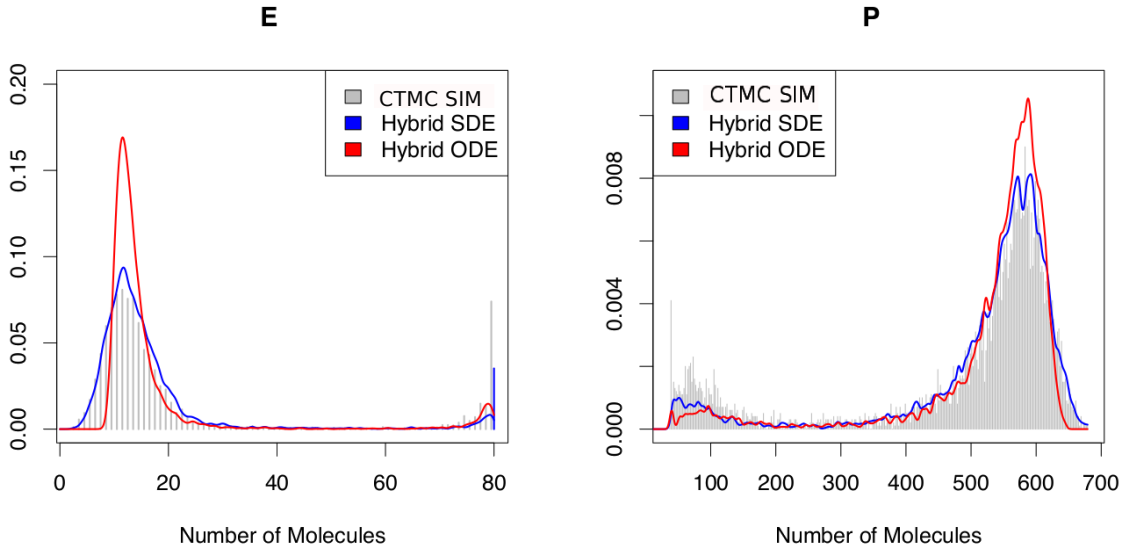

Figure 6. Comparison between distributions computed with Monte Carlo Simulation, HSDE, HODE.

5.3. The tool

The comparison of the different approaches has been carried out with a prototype implementation integrated in the GreatSPN framework [6] [46]. This allows us to use the new GreatSPN GUI (see Fig. 7) to easily design the SPN model and to generate the corresponding (H)ODE/(H)SDE solver.

This generation process requires the following three steps:

1. *PINV* computes the (lower and upper) bounds for all the quantities involved in the systems.
2. *PN2ODE* generates from an SPN model a C++ file implementing the corresponding (H)ODE/(H)SDE system. This system is directly encoded in C++ exploiting a set of specific classes (e.g. *SystemEquation*, *Equation*, ...) defined in an ad-hoc developed library;
3. *CREATE_SOLVE* compiles the previously generated C++ code with the ad-hoc developed library to generate the corresponding (H)ODE/(H)SDE solver. In particular this library implements the standard Euler method to solve ODE systems, the Euler-Maruyama method [26] to solve SDE systems[7], and the First-Reaction method [47]

[6] A VirtualBox image, in which this prototype implementation is installed, is available on demand sending an email to greatspn@di.unito.it.

[7] Currently, we are working to implement RungeKutta method in our framework, however it is important to highlight that the use of methods with better convergence order does not change the quality of the results presented in this section.

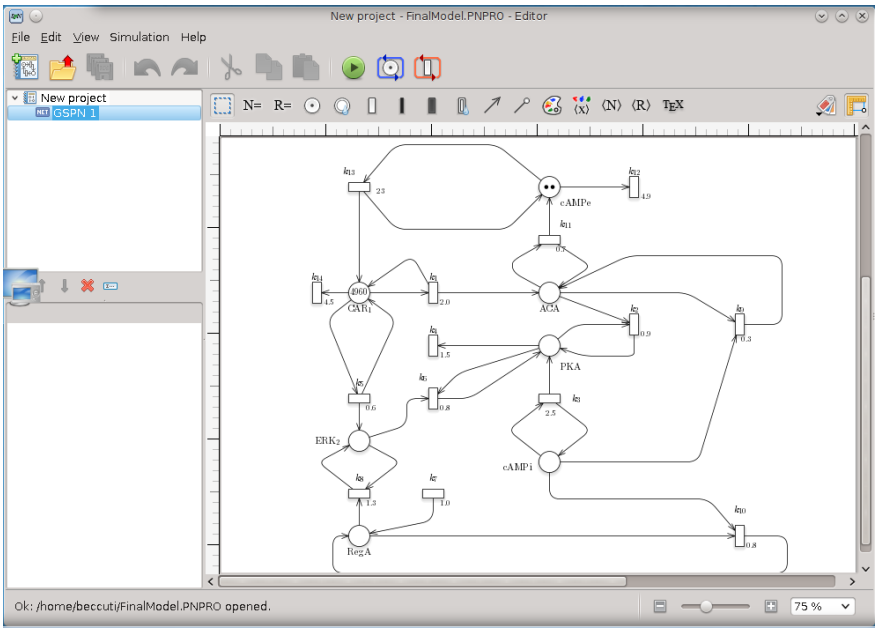

Figure 7. GreatSPN GUI.

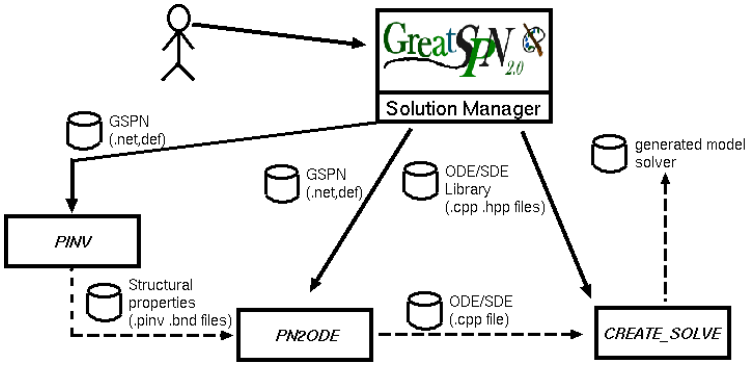

Figure 8. Framework architecture.

to simulate them. Moreover, it provides an extension of Euler-Maruyama method for computing the numerical solution of our (hybrid) jump diffusion approximation (see Appendix B where the pseudo-code which describes such extension is reported).

Then, the generated solver can be executed by command line as follows:

net_name.solver out_file_name type step num_runs max_time [-B place_bounds]

where *type* is used to specify which method will be used to solve the system (i.e. ODE, HODE, SDE, HSDE and SIM), *step* is the maximum Euler step, *num_runs* is the maximum number of runs (i.e., MaxRuns), and *max_time* is the final time for which the solution is computed. Optionally the user can specify the bounds (lower and upper) for each net place in a separate file, so that she/he will override those automatically computed from the SPN model by PINV (i.e. *-B place_bounds*).

6. Conclusions

In this paper we have provided numerical evidence that Kurtz's diffusion approximation can be extended to a jump diffusion approximation to address the case when the process reaches the boundary with non-negligible probability and that such jump diffusion approximation can be extended further in the framework of hybrid models. Our proposal allows to handle cases in which the number of certain objects represented in the original model does not grow unbounded, thus violating one of the conditions for density dependent Markov chains. In these cases we have shown that it is possible to apply the jump diffusion approximation only to those components of the model that are in density dependent form and are associated with high population levels. The remaining components are treated as discrete quantities. The resulting process is a hybrid switching jump diffusion, i.e., a Markov process with hybrid state space and jumps where the discrete state changes can be seen as switches that take the diffusion from one mode to another. We have shown that the stochastic differential equations that characterize this process can be derived automatically both from the description of the original Markov chains as well as from a model specified using a high level description language, like stochastic Petri nets. To support our proposal, we have applied the method for the analysis of four models of biological interest: a model of the crazy clock reaction and its variation with a switching behavior, a model describing viral infection kinetics and another representing a transcriptional regulatory mechanism. The results have been obtained in part by numerical integration of the Fokker Plank equation and in part by Monte Carlo simulations of the new approximating model. For all four examples, our method closely reproduces the behavior of the original CTMC with a substantial saving of execution time.

References

[1] T. Turner, S. Schnell, K. Burrage, Stochastic approaches for modelling in vivo reactions, Comp. Bio. Chem. 28 (2004) 165178.
[2] D. Wilkinson, Stochastic Modelling for Systems Biology, Chapman & Hall, 2006.
[3] D. T. Gillespie, Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem. 81 (25) (1977) 2340–2361.
[4] T. G. Kurtz, Solutions of ordinary differential equations as limits of pure jump Markov processes, Journal of Applied Probability 1 (7) (1970) 49–58.
[5] W. J. Stewart, Introduction to the Numerical Solution of Markov Chains., Princeton University Press, Princeton, New Jersey, USA, 1995.
[6] D. T. Gillespie, Approximate accelerated stochastic simulation of chemically reacting systems, J Chem Phys 115 (2001) 1716–1733.
[7] M. Rathinam, L. R. Petzold, Y. Cao, D. T. Gillespie, Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method, J Chem Phys 119 (24) (2003) 12784–12794.
[8] Y. Cao, D. T. Gillespie, L. R. Petzold, The slow-scale stochastic simulation algorithm, J Chem Phys 122 (1).
[9] T. Dayar, L. Mikeev, V. Wolf, On the numerical analysis of stochastic Lotka-Volterra models, in: Proc. of the Workshop on Computer Aspects of Numerical Algorithms (CANA10), 2010, pp. 289–296.
[10] M. Mateescu, V. Wolf, F. Didier, T. A. Henzinger, Fast adaptive uniformisation of the chemical master equation, IET systems biology 4 (6) (2010) 441–452.
[11] J. Zhang, L. T. Watson, Y. Cao, A modified uniformization method for the solution of the chemical master equation, Computers & Mathematics with Applications 59 (1) (2010) 573 – 584.
[12] J. Zhang, L. T. Watson, Y. Cao, Adaptive aggregation method for the chemical master equation, Int. J. of Computational Biology and Drug Design 2 (2) (2009) 134–148.
[13] F. Ciocchetta, A. Degasperi, J. Hillston, M. Calder, Some investigations concerning the CTMC and the ode model derived from bio-pepa., Electron. Notes Theor. Comput. Sci. 229 (2009) 145–163.

[14] F. Cordero, A. Horváth, D. Manini, L. Napione, M. D. 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, Theor. Comput. Sci. 412 (43) (2011) 6036–6057.
[15] A. Angius, A. Horváth, Product form approximation of transient probabilities in stochastic reaction networks, Electronic Notes on Theoretical Computer Science 277 (2011) 3–14.
[16] A. Angius, A. Horváth, V. Wolf, Quasi product form approximation for markov models of reaction networks, Transactions on Computational Systems Biology 7625 (XIV).
[17] T. G. Kurtz, Limit theorems and diffusion approximations for density dependent markov chains, in: Stochastic Systems: Modeling, Identification and Optimization, I, Springer, 1976, pp. 67–78.
[18] D. F. Anderson, T. G. Kurtz, Continuous time markov chain models for chemical reaction networks, in: Design and Analysis of Biomolecular Circuits, Springer, 2011, pp. 3–42.
[19] M. Beccuti, E. Bibbona, A. Horváth, R. Sirovich, A. Angius, G. Balbo, Analysis of Petri net models through stochastic differential equation, in: Proc. of International Conference on Application and Theory of Petri Nets and other models of concurrency (ICATPN'14), Tunis, Tunisia, 2014, arXiv:1404.0975.
[20] G. G. Yin, C. Zhu, Hybrid switching diffusions, Vol. 63 of Stochastic Modelling and Applied Probability, Springer, New York, 2010, properties and applications. doi:10.1007/978-1-4419-1105-6.
URL http://dx.doi.org/10.1007/978-1-4419-1105-6
[21] G. Horton, V. G. Kulkarni, D. M. Nicol, K. S. Trivedi, Fluid stochastic petri nets: Theory, application, and solution techniques, Eur. J. Op. Res. 105 (1) (1998) 184–201.
[22] M. Gribaudo, M. Sereno, A. Horváth, A. Bobbio, Fluid stochastic Petri nets augmented with flush-out arcs: Modelling and analysis, Discrete Event Dynamic Systems: Theory and Applications 11 (2001) 97–117.
[23] D. R. Cox, The analysis of non-markovian stochastic processes by the inclusion of supplementary variables, Mathematical Proceedings of the Cambridge Philosophical Society 51 (1955) 433–441.
[24] A. Pourranjbar, J. Hillston, L. Bortolussi, Don't just go with the flow: Cautionary tales of fluid flow approximation, in: M. Tribastone, S. Gilmore (Eds.), EPEW 2012, and UKPEW 2012, Vol. 7587 of Lecture Notes in Computer Science, Springer, Springer, 2012, p. 156–171.
[25] T. G. Kurtz, Strong approximation theorems for density dependent markov chains, Stochastic Processes and Their Applications 6 (3) (1978) 223–240.
[26] F. C. Klebaner, Introduction to stochastic calculus with applications, 3rd Edition, Imperial College Press, London, 2012.
[27] L. C. G. Rogers, D. Williams, Diffusions, Markov Processes, and Martingales: Volume 2, Itō calculus., Cambridge university press, 2000.
[28] D. T. Gillespie, The chemical langevin equation, J. Chem. Phys. 113 (2000) 297.
[29] R. Srivastava, L. You, J. Summers, J. Yin, Stochastic vs. deterministic modeling of intracellular viral kinetics, Journal of Theoretical Biology 218 (3) (2002) 309–321.
[30] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis, Modelling with Generalized Stochastic Petri Nets, J. Wiley, New York, NY, USA, 1995.
[31] M. H. A. Davis, Piecewise-deterministic markov processes: A general class of non-diffusion stochastic models, Journal of the Royal Statistical Society. Series B (Methodological) 46 (3) (1984) pp. 353–388.
[32] L. Bortolussi, Hybrid limits of continuous time markov chains, in: Quantitative Evaluation of Systems (QEST), 2011 Eighth International Conference on, 2011, pp. 3–12.
[33] G. Pola, M. Bujorianu, J. Lygeros, M. Benedetto, Stochastic hybrid models: An overview, in: Proc. IFAC Conf. Anal. Design Hybrid Syst, 2003, pp. 45–50.
[34] G. Caravagna, A. d'Onofrio, M. Antoniotti, G. Mauti, Stochastic Hybrid Automata with delayed transitions to model biochemical systems with delays, INFORMATION AND COMPUTATION 236 (SI) (2014) 19–34. doi:10.1016/j.ic.2014.01.010.
[35] S. Menz, J. C. Latorre, C. Schtte, W. Huisinga, Hybrid stochastic–deterministic solution of the chemical master equation, Multiscale Modeling & Simulation 10 (4) (2012) 1232–1262.
[36] T. A. Henzinger, L. Mikeev, M. Mateescu, V. Wolf, Hybrid numerical solution of the chemical master equation, in: Proceedings of the 8th International Conference on Computational Methods in Systems Biology, ACM, 2010, pp. 55–65.
[37] L. Bortolussi, Limit behavior of the hybrid approximation of stochastic process algebras, in: K. Al-Begain, D. Fiems, W. Knottenbelt (Eds.), 17th International Conference on Analytical and Stochastic Modeling Techniques and Applications (ASMTA 2010), Springer-Verlag, Springer-Verlag, 2010, p. 367–381.
[38] S. Intep, D. J. Higham, X. Mao, Switching and diffusion models for gene regulation networks, Multiscale Modeling & Simulation 8 (1) (2009) 30–45.
[39] E. L. Haseltine, J. B. Rawlings, Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics, The Journal of chemical physics 117 (15) (2002) 6959–6969.
[40] H. Salis, Y. Kaznessis, Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions, The Journal of chemical physics 122 (5) (2005) 054103.
[41] M. Bujorianu, J. Lygeros, Toward a general theory of stochastic hybrid systems, in: H. Blom, J. Lygeros (Eds.), Stochastic Hybrid Systems, Vol. 337 of Lecture Notes in Control and Information Science, Springer Berlin Heidelberg, 2006, pp. 3–30.
[42] J. Hu, J. Lygeros, S. Sastry, Towards a theory of stochastic hybrid systems, in: N. Lynch, B. Krogh (Eds.), Hybrid Systems: Computation and Control, Vol. 1790 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2000, pp. 160–173.
[43] W. Feller, Diffusion processes in one dimension, Trans. Amer. Math. Soc. 77 (1954) 1–31.
[44] W. Feller, The parabolic differential equations and the associated semi-groups of transformations, Ann. of Math. (2) 55 (1952) 468–519.
[45] P. Érdi, G. Lente, Stochastic Chemical Kinetics, Springer Series in Synergetics, Springer, 2014.
[46] J. Babar, M. Beccuti, S. Donatelli, A. S. Miner, Greatspn enhanced with decision diagram data structures, in: Proceedings of Applications and Theory of Petri Nets, 31st International Conference, PETRI NETS 2010, Braga, Portugal, June 21-25,, IEEE Computer Society, Los Alamitos, California, USA, 2010, pp. 308–317.
[47] J. Pahle, Biochemical simulations: stochastic, approximate stochastic and hybrid approaches, Briefings in Bioinformatics 10(1) (2009) 53–64.

[48] R webpage, http://www.r-project.org/.
[49] J. Goutsias, Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems, The Journal of Chemical Physics 122 (184102). doi:10.1063/1.1889434.
[50] I. Segel, Enzyme kinetics: behavior and analysis of rapid equilibrium and steady state enzyme systems, Wiley classics library, Wiley, 1975. URL http://books.google.it/books?id=6uFqAAAAMAAJ
[51] P. Brémaud, Markov chains, Vol. 31 of Texts in Applied Mathematics, Springer-Verlag, New York, 1999, gibbs fields, Monte Carlo simulation, and queues.
[52] M. K. Molloy, Performance analysis using stochastic petri nets, IEEE Transactions on Computers 31 (9) (1982) 913–917. doi:http://dx.doi.org/10.1109/TC.1982.1676110.

Appendix A. Examples and properties of density dependent processes

Appendix A.1. Example for a density dependent family

We describe here a family of density dependent CTMCs which corresponds to a simple epidemic model. The groups involved in the process are that of the susceptible individuals and that of the infected ones. The state of the process is a pair (i, j) giving the number of susceptible individuals, i, and the number infected ones, j. We consider an area of size N and assume that there are three possible events. The transition that makes the number of susceptible individuals grow is proportional to the size of the area, i.e.,

$$q_{(i,j),(i+1,j)} = N\lambda_1$$

Infection occurs by the interaction of a susceptible individual and an already infected one. The intensity of the corresponding reaction in state (i, j) is proportional to the product $i \times j$ and inversely proportional to the size of the considered area, i.e., we have

$$q_{(i,j),(i-1,j+1)} = \lambda_2 \frac{ij}{N}$$

Infected individuals are assumed to become immune independently of each other and independently of the size of the area, i.e.,

$$q_{(i,j),(i,j-1)} = \lambda_3 j$$

All the above three intensities can be written in such a form that they are proportional to the size of the area and they depend on the density of the number of the individuals, i/N and j/N, (or do not depend on one or both state variables)

$$q_{(i,j),(i+1,j)} = N\lambda_1, \quad q_{(i,j),(i-1,j+1)} = N\lambda_2 \left(\frac{i}{N}\right)\left(\frac{j}{N}\right), \quad q_{(i,j),(i,j-1)} = N\lambda_3 \left(\frac{j}{N}\right)$$

Density dependent CTMCs are those whose transition intensities are in the above form. The formal definition is provided in Definition 1.

For the above model the set of possible state changes is $C = \{(1, 0), (-1, +1), (0, -1)\}$ and function f required by Definition 1 is defined as

$$f(y, l) = \begin{cases} \lambda_1 & \text{if } l = (+1, 0) \\ \lambda_2 y_1 y_2 & \text{if } l = (-1, +1) \\ \lambda_3 y_2 & \text{if } l = (0, -1) \\ 0 & \text{otherwise} \end{cases}$$

Appendix A.2. Properties of density dependent processes

In order to gain a better understanding of the property of density dependence, let us introduce some general concept from the theory of Markov chains. Among the many books devoted to this topic, we refer the reader to [51].

For a general Markov chain $M(u)$ with state space $S \in \mathbb{Z}^k$ and instantaneous transition rates $q_{i,j}$, let us introduce the following key object

$$F_M(i) = \sum_{j \in S} (j - i) q_{i,j} \quad i \in S. \tag{A.1}$$

The function F_M will be referred to as the *generator* of the chain. Under suitable hypothesis the expectation of $M(u)$ solves the following *Dynkin equation* cf. [51, Chapter 9, Theorem 2.2]

$$\frac{d\mathbb{E}(M(u))}{du} = \mathbb{E}[F_M(M(u))]. \tag{A.2}$$

Two invariance properties of density dependent CTMCs can be then stated using the normalized chains.

Property 1. *The density dependence property of the family $X^{[N]}(u)$ is equivalent to require that for the family of the normalized CTMCs, $Z^{[N]}(u)$, the generator is constant. This constant generator, denoted by $F(y)$, is equal to $F_{Z^{[N]}}(y)$ for all N and hence*

$$F(y) = \sum_{l \in C} \frac{l}{N} \cdot p^{[N]}_{y,y+\frac{l}{N}} = \sum_{l \in C} \frac{l}{N} \cdot q^{[N]}_{Ny,Ny+l} = \sum_{l \in C} l f\,(y, l) \tag{A.3}$$

where $p^{[N]}_{i,j}$ and $q^{[N]}_{i,j}$ are the instantaneous transition rates of the processes $Z^{[N]}$ and $X^{[N]}$, respectively. Note that a necessary condition for a generator to be constant is that the entries of the vectors that describe the effect of the transitions, i.e., the entries of the vectors in C, cannot depend on N.

Property 2. *Each element of the family $Z^{[N]}(u)$ solves the same Dynkin equation*

$$\frac{d\mathbb{E}(Z^{[N]}(u))}{du} = \mathbb{E}[F(Z^{[N]}(u))]. \tag{A.4}$$

Note that (3) is analogous to the above Dynkin equation.

In case of a nearly density dependent family Property 1 does not hold but, as stated by the following property, the generator is still dominated by the function $F(y)$ given in (A.3) as the indexing parameter increases.

Property 3. *If the family $X^{[N]}(u)$ is nearly density dependent then*

$$F_{Z^{[N]}}(y) = F(y) + O\left(\frac{1}{N}\right),$$

from which it follows that

$$\lim_{N \to \infty} F_{Z^{[N]}}(y) = F(y) \tag{A.5}$$

Appendix A.3. Petri Nets

Petri Nets (PNs) are bipartite directed graphs with two types of nodes: places and transitions. The places, graphically represented as circles, correspond to the state variables of the system (e.g., chemical compounds), while the transitions, graphically represented as rectangles, correspond to the events (e.g., reaction occurrences) that can induce state changes. The arcs connecting places to transitions (and vice versa) express the relations between states and event occurrences. Places can contain tokens (e.g., molecules) drawn as black dots within the places. The state of a PN, called marking, is defined by the number of tokens in each place. The evolution of the system is given by the occurrence of enabled transitions, where a transition is enabled iff each of its input places contains a number of tokens greater than or equal to a given threshold defined by the cardinality of the corresponding input arc. A transition occurrence, called firing, removes a fixed number of tokens from its input places and adds a fixed number of tokens to its output places (according to the cardinality of its input/output arcs).

Stochastic Petri Nets (SPNs) are PNs where the firing of each transition is assumed to occur after a random delay (firing time) from the time it is enabled. In this paper we consider exponentially distributed random delays [52]. Accordingly, each transition is associated with a rate that represents the parameter of its firing delay distribution. Firing rates may be marking dependent. When a marking is entered an exponentially distributed random delay is sampled for each enabled transition. The transition with the lowest delay fires and the system changes marking accordingly. Consequently, the underlying stochastic process is a CTMC.

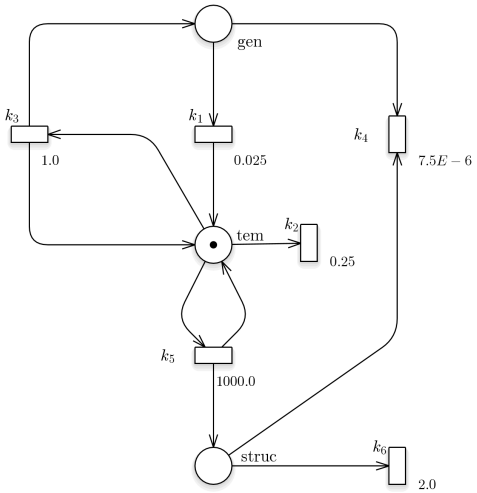

Figure A.9. PN model describing the intracellular kinetics of a generic virus

Formally, an SPN is defined by the following ingredients. P and T are the sets of places and transitions, respectively. The initial marking is given by a vector of non-negative integers of length $|P|$ and it is denoted by $\mathbf{m_0}$. The multiplicity of the input arcs of transition $t \in T$ is given by a vector of non-negative integers of length $|P|$ and it is denoted by I_t. Similarly, the multiplicity of the output arcs are given by the vector denoted by O_t.

In Figs.A.9 andA.10 are shown the SPN models for the two case studies presented in Sec.5.

The parameter of the exponential distribution of the firing delay of transition t in marking $\mathbf{m}$ is denoted by $\lambda_{t,\mathbf{m}}$. With the above notation the set of possible state changes is given by $C = \{l \mid l = O_t - I_t, t \in T\}$ because the overall effect of transition t is given by $O_t - I_t$. Transition t is enabled in marking $\mathbf{m}$ iff $\mathbf{m} \geq I_t$. And the transition intensities of the CTMC corresponding to the SPN can be written as

$$q_{\mathbf{m},\mathbf{m}+l} = \sum_{\forall t: l = O_t - I_t \wedge \mathbf{m} \geq I_t} \lambda_{t,\mathbf{m}}$$

A special form of transition intensity, which is particularly interesting in our context, arises when a transition models a chemical reaction behaving according to the stochastic law of mass action. In this case the intensity of transition t in marking $\mathbf{m}$ is given as

$$\lambda_{t,\mathbf{m}} = N^{1-\sum_{i=1}^{|P|} I_t(i)} \mu_t \prod_{i=1}^{|P|} \binom{\mathbf{m}(i)}{I_t(i)} \tag{A.6}$$

where V is the volume in which the reaction occurs, μ_t is the rate constant of the reaction, and $v(i)$ is the ith entry of the vector v. The above can be written separating the contributions with different orders in N as

$$\lambda_{t,\mathbf{m}} = N\left(\mu_t \prod_{i=1}^{|P|} \frac{1}{I_t(i)!} \left(\frac{\mathbf{m}(i)}{N}\right)^{I_t(i)} + O\left(\frac{1}{N}\right)\right)$$

which shows that models in which all transitions act according to the stochastic law of mass actions are nearly density dependent and that the function f required by the definitions of density dependence is given by

$$f(y, l) = \sum_{\forall t: l = O_t - I_t \wedge \mathbf{m} \geq I_t} \mu_t \prod_{i=1}^{|P|} \frac{y(i)^{I_t(i)}}{I_t(i)!}$$

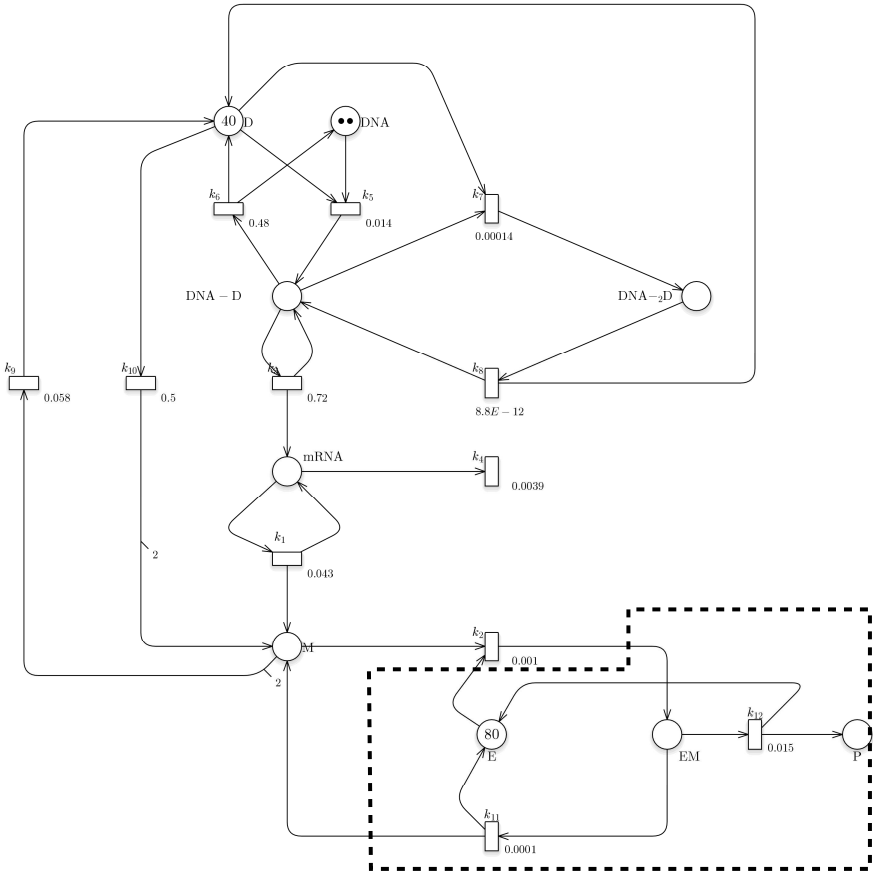

Figure A.10. SPN model describing the transcription regulation.

Algorithm 1 Algorithm for simulating HSJD systems

```
1:  function SolveSSDE(SSDE, step, MaxRuns, FinalTime)
    SSDE = HSJD system.
    step = step used in the Euler solution.
    MaxRuns = maximum number of runs.
    FinalTime = maximum time for each run.
    Value= vector encoding the state of the HSJD at the current step.
    PrValue= vector encoding the state at the previous step.
    ListESim= list of transitions on which discrete simulation step is performed.
    ListE= list of transitions on which diffusion step is performed.
2:      run=0;
3:      while (run ≤ MaxRuns) do
4:          time = 0.0;
5:          SSDE.Init(Value);
6:          while (time ≤ FinalTime) do
7:              print(time,Value);
8:              Value.Copy(PrValue);
9:              h = step;
10:             ListESim = SSDE.CheckBound(PrValue);
11:             e = ListESim.CheckFire(h);
12:             for (SDE ∈ SSDE) do
13:                 ListE=SDE.getEvent(ListESim);
14:                 SDE.computeEuler(Value,PrValue,h,ListE);
15:                 If (SDE.Check(e)) then SDE.computeSim(Value,PrValue,e);
16:             SSDE.Norm(Value);
17:             time += h;
18:         run++;
19: end function
```

Appendix B. Pseudo-code implementing (H)SDE solver

Algorithm 1 shows the pseudo-code implementing a simple simulation approach to analyze HSJD models in which its quantities can visit their barrier. The algorithm takes in input a HSJD system (called *SSDE*), a step size used for the Euler scheme (i.e *step*), the number of runs (i.e. *MaxRuns*), and a final time (i.e. *FinalTime*); and it prints the generates traces.

In details, the method *init()* at line 5 initializes the vector *Value* encoding the state of the HSJD; the method *Copy()*, called at line 8 before computing the state at the next step, copies the current values of vector *Value* in the vector *PrValue* storing the previous state. Method *CheckBound()* taking in input the current state returns the list of events currently in $\overset{\star}{C}{}_x^F$. Among these events the method *CheckFire()* at line 11 returns the first that is scheduled to fire (i.e., e) with firing delay smaller than the current solution step (i.e., h). Moreover, such method re-sets h value according to the selected firing delay time.

Then, for each SDE equation the method *getEvents()* returns the list of events (i.e. *ListE*) in $\overset{\circ}{C}{}_x^F$. This list is hence used by the method *computeEuler()* to update the current state using the standard Euler method. Moreover, the state may be updated by transition e iff e decrements/increments the considered state component. At the end of each step, the method *Norm()* is called to normalize the state vector taking into account the invariants of the system.