Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

SPEEDING UP STOCHASTIC AND DETERMINISTIC SIMULATION BY AGGREGATION: AN ADVANCED TUTORIAL

Mirco Tribastone

Systems Modeling and Analysis Unit
IMT School for Advanced Studies Lucca
Piazza San Francesco 19
Lucca, 55100, ITALY

Andrea Vandin

DTU Compute
Technical University of Denmark
Richard Petersens Plads, Building 324
DK-2800 Kgs. Lyngby, DENMARK

ABSTRACT

Dynamical models of systems across many branches of science and engineering can be mathematically represented in terms of stochastic processes such as Markov chains, or deterministically through a system of difference or differential equations. Unfortunately, in all but special cases these models do not enjoy analytical solutions, hence one is left with computer-based approaches by means of stochastic simulators and numerical solvers. As a consequence, the computational cost increases with the dimensionality of the model under consideration, hindering our capability of dealing with complex large-scale models arising from accurate mechanistic descriptions of real-world systems. This paper offers an advanced tutorial on an array of recently developed algorithms that seek to tame the complexity of these models by aggregating their constituting systems of equations, leading to lower-dimensional systems that preserve the original dynamics in some appropriate, formal sense.

1 INTRODUCTION

In many disciplines including biology, chemistry, computer science, ecology, economics, and physics, dynamical models are routinely used in order to understand and predict the behavior of various natural as well as engineered systems. Depending on the required level of abstraction and on the assumptions upon which the model is based, notable classes of models are *deterministic*, based on differential or difference equations, or *stochastic*, based, for example, on Markov processes. In biology and chemistry, ordinary differential equations (ODEs) are the underlying mathematical model of chemical reaction networks, describing the time-course evolution of the concentrations of the components (e.g., chemical species, proteins, or genes) in a (bio-)chemical system (Murray 2002). In computer science, Markov processes are of paramount relevance in the quantitative analysis of computing and networked systems including availability, performance, and reliability, see, e.g., (Stewart 2009). The Lotka-Volterra model is one celebrated instance in ecology, where the dynamics of the interactions in predator-prey systems is represented by a system of two coupled ordinary differential equations (Volterra 1931). In the simulation community, well-known *system dynamics* allows for the simulation of aggregate flows by means of difference/differential equations (Forrester 1961).

A major impediment to our capability of reasoning about the behavior of such systems is the well-known *curse of dimensionality*, also called the *state-space explosion* problem. Fundamentally, this is due to the fact that the size of the mathematical model tends to grow fast with the number of components of the system under consideration. It is an issue that may manifest itself in different forms depending on the specific types of considered systems and modeling formalisms. Here we give two instances:

- In models based on a discrete-state representation, such as Markov chains used for the analysis of *population processes* (Bortolussi et al. 2013), the number of states grows exponentially with the size of the population. For example, the number of states in a Markov chain underlying a closed

queueing network with K users/jobs and M stations is equal to $\binom{K+M-1}{M-1}$, corresponding to the number of ways in which it is possible to place K objects in M bins.

- In systems biology, protein phosphorylation has a fundamental role in the regulation of cellular life, as it can substantially affect the function of a protein by modifying its enzymatic activity or the possibility of binding with partners. A protein with n phosphorylation sites may exist in 2^n possible forms, each describing the (binary) phosphorylated/dephosphorylated state of a specific site (Whitmarsh and Davis 2016). Thus, kinetic models of multisite phosphorylation, usually based on ODEs, will need an exponential number of ODE variables for a detailed mechanistic description, e.g., (Salazar and Höfer 2009).

The most important consequence of state-space explosion is the high computational cost of the analysis for large models, due to the lack of analytical closed-form expressions in general. This cross-disciplinary problem has spurred a very intense line of investigation on *model reduction*, with the aim of deriving a lower-dimensional model whose (hopefully easier) solution can be formally related to the dynamics of the original one. An overview of the many methods that have been developed in the literature is beyond the scope of this paper. An exhaustive treatment of reduction techniques originating in control theory is given in (Antoulas 2005); Okino and Mavrovouniotis review simplification methods for chemical models (Okino and Mavrovouniotis 1998); a review concerning approaches in computational systems biology is provided in (Radulescu et al. 2012).

Here we present a tutorial on recent results on model reduction developed by this paper's authors and collaborators. Our techniques are centered on a notion of *aggregation* whereby each variable in the reduced model represents a sum of variables of the original one. Focusing on this type of relationship brings about a number of advantages. One concerns physical intelligibility. This is important especially when the model is to be used for prediction and validation purposes (Apri et al. 2012). More important, the aggregation is completely characterized in terms of the set of variables that contribute to each *macro-variable* in the reduced model. Here we consider aggregations that are induced by a *partitioning* of the variables of the original system: that is, each original variable must contribute to exactly one macro-variable. There are other possibilities, for instance based on a *covering* of the original state space, whereby a variable may appear in more than one macro-variable (Feret et al. 2009).

Casting the problem of model reduction to that of finding an appropriate partition of variables has a crucial algorithmic implication: we can leverage efficient *partition-refinement* algorithms that can compute the maximal (i.e., the most compact) aggregation (Cardelli et al. 2017a). Partition-refinement algorithms have been developed for Markov chain aggregation (Valmari and Franceschinis 2010), known in the literature as *lumpability* (Kemeny and Snell 1976). Our methods can be seen as a conservative generalization of lumpability methods to *polynomial dynamical systems* (PDS), see (Cardelli et al. 2017a), i.e., systems of nonlinear equations whose "next-state" update law is a polynomial function of the state variables; this covers Markov chains in the special case of an update law that is the linear map induced by the chain's transition matrix (Stewart 1994).

Our approach goes beyond Markov chain lumping in two fundamental ways:

i) It is based on the notion of *reaction network* (RN), a mathematical object that is a slight extension of a formal chemical reaction network (CRN). An RN can equivalently represent the PDS under consideration in terms of a finitary structure. The RN can be interpreted as the graphical counterpart of the transition matrix of a Markov chain. This allows extension of Markov-chain aggregation algorithms to deterministic nonlinear systems of equations.

ii) A CRN with stochastic mass-action kinetics (Voit 2013) —ubiquitous, e.g., in chemistry, epidemiology, and systems biology — is a special case of an RN. It can be seen as a domain-specific high-level language for chemical systems that defines the rules with which chemical species interact with each other, without having to explicitly enumerate the underlying state space of its underlying Markov chain (which may be exponentially larger than the size of the CRN description, or even

infinite) (Gillespie 1977). While traditional lumping techniques do require the availability of the state space, we can define a notion of aggregation *directly on the CRN* which implies lumpability at the underlying Markov chain level (Cardelli et al. 2017b).

In this paper we give a unified account of these results. We provide some background on Markov chain lumping in Section 2. We review the theory in Sections 3 and 4. In Section 5 we discuss the main features of *ERODE*, a software tool that implements the reduction techniques (Cardelli et al. 2017). Concluding remarks and a perspective on future work are discussed in Section 6.

2 BACKGROUND

Polynomial dynamical systems

The reduction techniques presented in this paper concern PDS. These are systems of equations of the form:

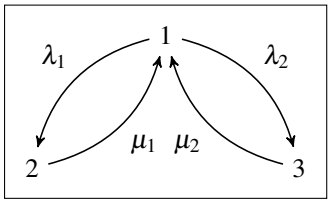
$$x_1'(t) = f_1(x_1(t),\ldots,x_N(t)) \qquad \ldots \qquad x_N'(t) = f_N(x_1(t),\ldots,x_N(t))$$

where x_1, …, x_N are the state variables, with $x_i(t)$ representing the state of the i-th variable at time t, f_1, …, f_N are the state update functions, assumed to be multivariate polynomials over the state variables, and the primed variables $x_1'(t)$, $\ldots x_N'(t)$ represent the "next state", i.e., the state reached upon application of the update function. This setting covers both discrete-time and continuous-time models. In the former case, the next state would read $x_i(t+1)$, for $1 \leq i \leq N$, i.e., the update function takes the system from the current step to the next one; in the latter case, the next state is the derivative with respect to time, hereafter denoted by the dot symbol, i.e., the time derivative of the i-th variable is denoted by $\dot{x}_i(t)$. The model description is completed by the initial condition $x_1(0)$, …, $x_N(0)$ that defines the initial state of the system.

The forthcoming results apply to either case, but in the following we will stick to examples in continuous time. In particular, we start with the special, but important case of a Markov chain.

Running example: a Markov chain reliability model

In order to fix ideas and relate to previous work on model aggregation, let us consider a toy example of a (continuous-time) Markov chain model for a simple repair system with $N = 3$ states. The model is pictorially depicted in the right inset. State 1 represents the operational state of some device. The transitions exponentially distributed with parameters λ_1 and λ_2 denote two different breakdown events (e.g., occurring in two different system components). Upon firing of either, the device is repaired, returning to state 1 with parameters μ_1 and μ_2. The stochastic behavior is completely characterized by the transition matrix $Q = (q_{ij})_{1 \leq i,j \leq N}$ of the Markov chain, which in this example reads thus:



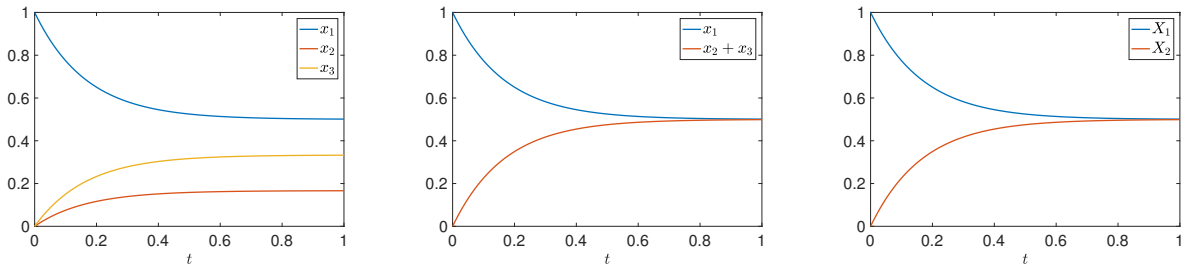
$$Q = \begin{bmatrix} -(\lambda_1+\lambda_2) & \lambda_1 & \lambda_2 \\ \mu_1 & -\mu_1 & 0 \\ \mu_2 & 0 & -\mu_2 \end{bmatrix} \tag{1}$$

where, as usual, the off-diagonal entries q_{ij} give the rate at which the process moves from state i to state j, and the diagonal elements are such that the each row sums to zero.

The forward equations of motions, giving the probabilities $x_i(t)$ that the Markov chain is found in each state i at time t, are provided as a PDS which is a system of linear ODEs. In our example, it reads:

$$\begin{aligned} \dot{x}_1(t) &= -(\lambda_1+\lambda_2)x_1(t) + \mu_1 x_2(t) + \mu_2 x_3(t) \\ \dot{x}_2(t) &= +\lambda_1 x_1(t) - \mu_1 x_2(t) \\ \dot{x}_3(t) &= +\lambda_2 x_1(t) - \mu_2 x_3(t) \end{aligned} \tag{2}$$

subject to a non-negative initial probability distribution, such that $\sum_i x_i(0) = 1$.



(a) Solutions of original model (2) (b) Original model with sums of solutions (c) Solution of the reduced model (3)

Figure 1: Correspondence between (2) and (3) for $\lambda_1 = 1$, $\lambda_2 = 2$, $\mu_1 = \mu_2 = 3$; initial conditions are $x_1(0) = 1$, $x_2(0) = x_3(0) = 0$, hence $X_1(0) = 1$, $X_2(0) = 0$.

Markov chain lumping

Let us now briefly review some results concerning the aggregation of Markov chains, which will set the stage for our generalization to PDS. Roughly speaking, lumpability identifies a partition of the states on whose blocks it is possible to define a Markov chain (the *lumped* Markov chain) whose behavior can be related to the original one (Kemeny and Snell 1976; Buchholz 1994).

Ordinary lumpability. With *ordinary lumpability*, each macro-state in the lumped Markov chain gives the (exact) sum of the probabilities of original states belonging to that partition block. To see this, let us assume that $\mu_1 = \mu_2 \equiv \mu$ in our running example. Then, we claim that the partition of states $\{\{1\},\{2,3\}\}$ form an ordinary lumpable partition. To see this, the equations (2) can be rewritten as follows:

$$\dot{x}_1(t) = -(\lambda_1 + \lambda_2)x_1(t) + \mu(x_2(t) + x_3(t))$$
$$\dot{x}_2(t) + \dot{x}_3(t) = +(\lambda_1 + \lambda_2)x_1(t) - \mu(x_2(t) + x_3(t))$$

We now apply the change of variables $X_1 = x_1$, $X_2 = x_2 + x_3$ (i.e., we define one variable for each block), getting the ODE system:

$$\dot{X}_1(t) = -(\lambda_1 + \lambda_2)X_1(t) + \mu X_2(t) \qquad \dot{X}_2(t) = +(\lambda_1 + \lambda_2)X_1(t) - \mu X_2(t) \tag{3}$$

This system satisfies the property that $X_1(t) = x_1(t)$ and $X_2(t) = x_2(t) + x_3(t)$ for all times whenever it is initialized such that $X_1(0) = x_1(0)$ and $X_2(0) = x_2(0) + x_3(0)$. Thus, we have indeed obtained a lower-dimensional system whose solution can be related to the original one. This correspondence is shown in Figure 1. The two plots in Figure 1b-1c are identical. This confirms that the solution of (3) can be related to that of the original model in (2) in terms of the sum of the reduced variables $x_2 + x_3$.

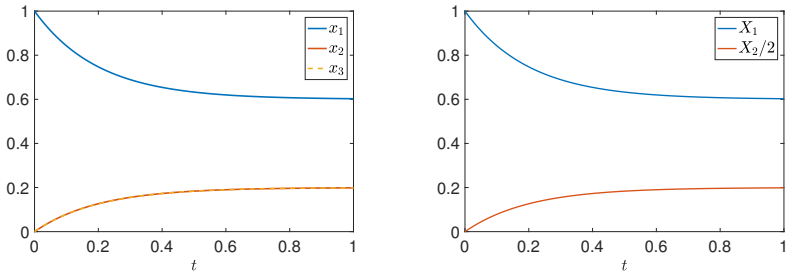
The reduced ODE system from (3) turns out to be induced by the two-state lumped Markov chain with the following transition matrix $\hat{Q}$:

$$\hat{Q} = \begin{bmatrix} -(\lambda_1 + \lambda_2) & \lambda_1 + \lambda_2 \\ \mu & -\mu \end{bmatrix} \tag{4}$$

An important feature of lumpability is that it is completely characterized by algebraic conditions on the original transition matrix. Indeed, a partition of blocks $X_1, \ldots, X_n$, is ordinary lumpable if and only if it holds that for any two blocks X_I, X_J and any states i, i' in X_I we have that

$$\sum_{j \in X_J} q_{i,j} = \sum_{j \in X_J} q_{i',j}.$$

In words, it must hold that the aggregate rate *toward any block* is the same for all states in a block. Such aggregate rate becomes the transition rate in the lumped Markov chain.



(a) Solution of original model (b) Solution of the reduced model (6)

Figure 2: Correspondence between (5) and (6) for $\lambda = 1$, $\mu = 3$, and $x_1(0) = 1$, $x_2(0) = x_3(0) = 0$, and $X_1(0) = 1$, $X_2(0) = 0$.

Exact lumpability. We observe that ordinary lumpability is exact but lossy, in the sense that from the lumped Markov chain one cannot recover the probabilities of the individual states in general. *Exact lumpability* is a notion which identifies a partition with equiprobable probabilities within each block. As with the ordinary case, we illustrate this behavior by considering the equations of motions of the Markov chain. Here we make the additional assumption that $\lambda_1 = \lambda_2 \equiv \lambda$. Then, the equations (2) become:

$$\dot{x}_1(t) = -2\lambda x_1(t) + \mu x_2(t) + \mu x_3(t) \qquad \dot{x}_2(t) = +\lambda x_1(t) - \mu x_2(t) \qquad \dot{x}_3(t) = +\lambda x_1(t) - \mu x_3(t) \tag{5}$$

We now claim that $\{\{1\},\{2,3\}\}$ form an exactly lumpable partition. Indeed, we can observe that if states 2 and 3 start with the same initial probabilities $x_2(0) = x_3(0)$, then their derivatives are equal at time $t = 0$; this implies that the probabilities $x_2(t)$ and $x_3(t)$ are equal at all time points. Thus, using the same steps as before, we get the lower-dimensional ODE system:

$$\dot{X}_1(t) = -2\lambda X_1(t) + \mu X_2(t) \qquad\qquad \dot{X}_2(t) = +2\lambda X_1(t) - \mu X_2(t) \tag{6}$$

Here, in addition to getting that $X_2(t) = x_2(t) + x_3(t)$, we also recover the individual probabilities by dividing the solution of each macro-variable by the size of its related partition block, i.e., $x_2(t) = x_3(t) = X_2(t)/2$. This is visualized in Figure 2. Instead, in all cases in which $x_2(0) \neq x_3(0)$, we cannot relate the solution of (6) to those of the original one (5) even if we set $X_2(0) = x_2(0) + x_3(0)$. This is exemplified in Figure 3, where Figure 3a plots the solution of (5) for initial conditions $x_1(0) = 0.5$, $x_2(0) = 0.3$ and $x_3(0) = 0.2$. As for Figure 2b, Figure 3b plots the solution of (6) using coherent initial conditions, i.e., $X_1(0) = x_1(0)$ and $X_2(0) = x_2(0) + x_3(0)$. However, differently from Figure 2, we now get two different trajectories, meaning that we cannot use (6) to relate to the solution of (5) for these initial conditions.

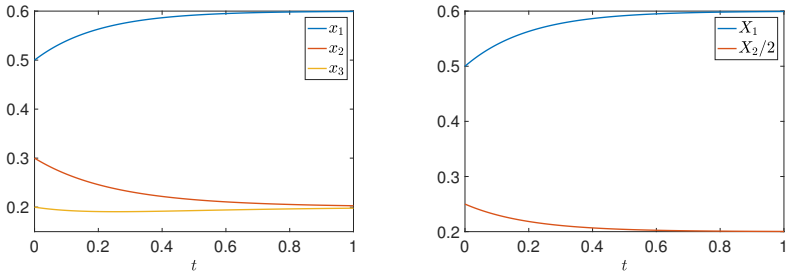
The algebraic criteria on the transition matrix which characterize exact lumpability are dual to those of ordinary lumpability. Here, a partition of blocks $X_1, \ldots, X_n$, is exactly lumpable if and only if it holds that for any two blocks X_I, X_J and any states i, i' in X_I we have that

$$\sum_{j \in X_J} q_{j,i} = \sum_{j \in X_J} q_{j,i'}.$$

That is, the aggregate rate *from any block* into all states in a block must be equal. Again, this aggregate rate becomes the transition rate in the lumped Markov chain.

We remark that ordinary lumpability imposes conditions for the *outgoing* transitions from states in the same block; exact lumpability has conditions on the *incoming* transitions. For this reason, these are also known as *forward* and *backward* criteria (Feret et al. 2012), respectively — terminology that will be used for our aggregation methods for PDS.

Lumping algorithms. The lumpability conditions allow us to check if a given partition induces an aggregated Markov chain. Efficient algorithms exist for computing the *maximal aggregation*, i.e., the



(a) Solution of (5). (b) Solution of the reduced model (6).

Figure 3: Non correspondence between (5) and (6) for $\lambda = 1$, $\mu = 3$, and $x_1(0) = 0.5$, $x_2(0) = 0.3$, $x_3(0) = 0.2$, and $X_1(0) = 0.5$, $X_2(0) = 0.5$.

coarsest lumpable partition (Valmari and Franceschinis 2010). They are based on *partition refinement*, a core approach for the minimization of foundational models in computer science (Paige and Tarjan 1987). Formally, they compute the coarsest partition that is a refinement of a given input partition, by iteratively splitting the blocks of the input partition until a fixed point. Importantly, these algorithms give freedom in choosing the input partition. This is important for two main reasons:

- In the case of ordinary lumpability, it allows isolation of states whose probability the modeler wishes to observe. This can be done by initializing the algorithm with a singleton block for each such observable state, which cannot be further split in subsequent iterations.
- In the case of exact lumpability, it allows for a pre-partitioning of the state space by placing equiprobable states within the same block of the initial partition; this is indeed a precondition for having equiprobable state probabilities at all time points.

An important feature of these algorithms is that they are not based on the analysis on the underlying ODEs of the Markov chain — the iterative splitting is of *structural* nature, based on the algebraic lumpability conditions. While it is in principle possible to restate such conditions on ODE properties, this will involve reasoning over uncountable state spaces, which is decidable using symbolic SAT-based methods, but computationally hard (Cardelli et al. 2016b). Instead using structural conditions on the transition matrix enables a very efficient algorithm which runs in $O(m \log n)$ time, where n is the number of states and m is the number of transitions of the original Markov chain (Valmari and Franceschinis 2010).

3 AGGREGATION OF POLYNOMIAL DYNAMICAL SYSTEMS

The forthcoming paragraphs of this section are devoted to the generalization of the principles of Markov chain aggregation to PDS. In order to do so, we need the following ingredients:

a) Definition of the PDS structural counterpart to the transition matrix of a Markov chain.
b) Generalization of ordinary and exact lumpability to PDS.
c) Definition of an aggregated PDS.
d) Development of an algorithm for computing the maximal aggregation.

Reaction networks

Reaction networks (RNs) are the PDS structural analogue of a Markov chain transition matrix. An RN is represented by a finite set of *species* and a set of *reactions*. (The terminology adopted is aligned with chemistry because they represent a slight generalization of CRNs.) Denoting by $S_1, \ldots, S_N$ the species

(with the intuition that each species corresponds to an ODE variable), a reaction r is in the form

$$r: \qquad \rho_1 S_1 + \ldots + \rho_N S_N \xrightarrow{\alpha} \pi_1 S_1 + \ldots \pi_N S_N$$

where the coefficients ρ_i and π_i are nonnegative integers (the stoichiometry), for $i = 1, \ldots, N$, and α is a real number (the "rate"). Species appearing on the left-hand side of a reaction (before $\rightarrow$), are referred as *reagents*, while those appearing on the right-hand side are referred as *products*. Essentially, a formal chemical reaction requires that all rates be greater than zero, indicating the kinetic parameter of the reaction. Without such restriction, with an RN it is possible to represent any PDS (Cardelli et al. 2017a), essentially by encoding any monomial appearing in the derivatives of the PDS as a distinct reaction. We stress, however, that models of systems of various nature (e.g., biological, chemical, ecological, etc.) are usually provided directly in terms of a (chemical) RN (Goutsias and Jenkinson 2013).

For example, the Markov chain of (1) can be shown to correspond to the RN:

$$S_1 \xrightarrow{\lambda_1} S_2 \qquad S_1 \xrightarrow{\lambda_2} S_3 \qquad S_2 \xrightarrow{\mu_1} S_1 \qquad S_3 \xrightarrow{\mu_2} S_1 \tag{7}$$

The following, instead, is a simple variant that gives rise to a nonlinear ODE system:

$$S_4 + S_1 \xrightarrow{\lambda_1} S_2 \qquad S_4 + S_1 \xrightarrow{\lambda_2} S_3 \qquad S_2 \xrightarrow{\mu_1} S_1 \qquad S_3 \xrightarrow{\mu_2} S_1 \tag{8}$$

The ODE is derived by applying the rule based on the law of mass action (Voit 2013). We let x_i denote the variable associated with species S_i, obtaining:

$$\begin{aligned}
\dot{x}_1(t) &= -(\lambda_1 + \lambda_2) x_1(t) x_4(t) + \mu_1 x_2(t) + \mu_2 x_3(t) \\
\dot{x}_2(t) &= +\lambda_1 x_1(t) x_4(t) - \mu_1 x_2(t) \\
\dot{x}_3(t) &= +\lambda_2 x_1(t) x_4(t) - \mu_2 x_3(t) \\
\dot{x}_4(t) &= -(\lambda_1 + \lambda_2) x_1(t) x_4(t)
\end{aligned} \tag{9}$$

We remark that, differently from Markov chains, here the solutions are not restricted to represent probability distributions. For example, when using an RN to represent a biological system, the variables represent species concentrations.

Forward and backward equivalence

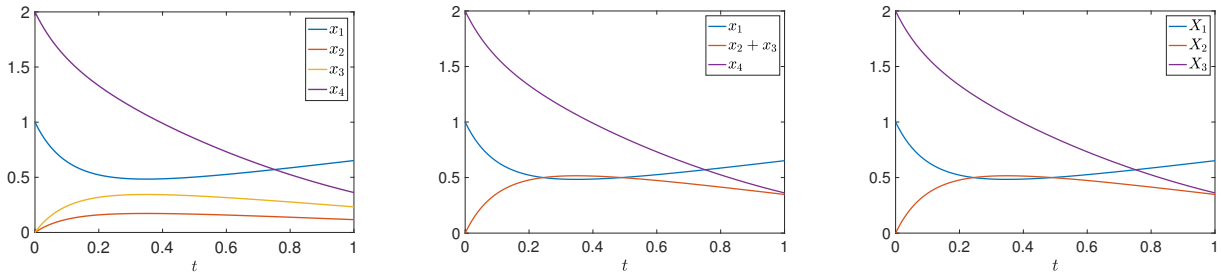
Forward equivalence (FE) is the PDS analogue of ordinary lumpability in Markov chains. Indeed, once again assuming $\mu_1 = \mu_2 \equiv \mu$, the reduced ODE system

$$\begin{aligned}
\dot{X}_1(t) &= -(\lambda_1 + \lambda_2) X_2(t) X_3(t) + \mu X_2(t) \\
\dot{X}_2(t) &= +\lambda_1 X_1(t) X_3(t) - \mu X_2(t) \\
\dot{X}_3(t) &= -(\lambda_1 + \lambda_2) X_2(t) X_3(t)
\end{aligned} \tag{10}$$

yields that $X_1(t) = x_1(t)$, $X_2(t) = x_2(t) + x_3(t)$, $X_3(t) = x_4(t)$ for all time points t; that is the partition of state variables $\{\{x_1\}, \{x_2, x_3\}, \{x_4\}\}$ is a forward equivalence.

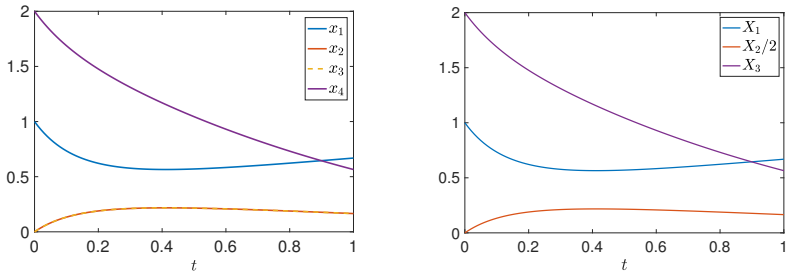
This is exemplified in Figure 4. As for the case of ordinary lumpability of Markov chains exemplified in Figure 1, the two plots in Figure 4b-4c are identical. This confirms that the solution of (10) can be related to that of the original PDS in (9) in terms of the sum of the reduced variables $x_2 + x_3$.

In a similar fashion, additionally assuming that $\lambda_1 = \lambda_2 \equiv \lambda$ yields that $x_2(t) = x_3(t)$ whenever these variables are initialized equally. That is the partition of state variables $\{\{x_1\}, \{x_2, x_3\}, \{x_4\}\}$ is a *backward equivalence* (BE), meaning that it identifies variables with equal trajectories whenever variables in the same



(a) Solution of original model (9) (b) Original model with sum of solutions (c) Solution of the reduced model (10)

Figure 4: Comparison of original PDS (9) and its FE reduction (10), for $\lambda_1 = 1$, $\lambda_2 = 2$, $\mu = \mu_1 = \mu_2 = 3$.



(a) Solution of original model (9) (b) Solution of the reduced model (11).

Figure 5: Solutions of PDS (9) and its BE reduction (11), for $\lambda = \lambda_1 = \lambda_2 = 1$, $\mu = \mu_1 = \mu_2 = 3$, and $x_1(0) = 1$, $x_2(0) = x_3(0) = 0$, $x_4(0) = 2$ and $X_1(0) = 1$, $X_2(0) = 0$, $X_3(0) = 2$.

block are initialized with the same initial conditions. Thus, using the same steps as before, we get the lower-dimensional PDS:

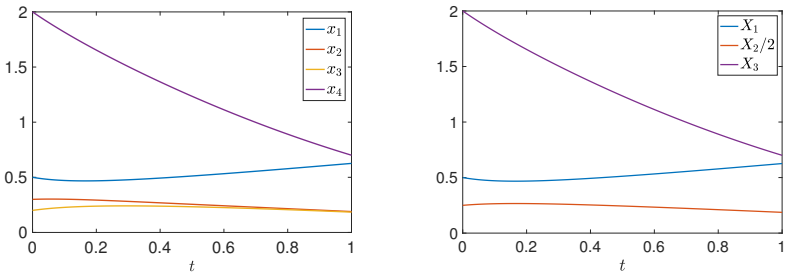
$$\dot{X}_1(t) = -2\lambda X_2(t)X_3(t) + \mu X_2(t) \qquad \dot{X}_2(t) = +\lambda X_1(t)X_3(t) - \mu X_2(t) \qquad \dot{X}_3(t) = -2\lambda X_2(t)X_3(t) \tag{11}$$

This is exemplified in Figure 5. Similarly to what discussed for exact lumpability of Markov chains (cf. Figure 3), the solution of a BE reduced model can be related to those of the original one if and only all variables in the same equivalence class have same initial condition. In Figure 5 we used $x_2(0) = x_3(0) = 0$, and hence $X_2(0) = x_2(0) + x_3(0) = 0$. Instead, Figure 6 shows the case for $x_2(0) = 0.3$, and $x_3(0) = 0.2$.

We remark that, in the cases so far, forward equivalence and backward equivalence are related to the same partition of variables; in general, these two notions are not comparable (Cardelli et al. 2015; Cardelli et al. 2016a).

The equivalences can be checked via structural RN-based conditions. The intuition behind how one can generalize the respective lumping notions can be provided by comparing (7) and (8). In the linear case (7), one needs two relate variables just in terms of the transitions toward a block, since each variable performs "reactions" on its own (i.e., there is only one species in the left-hand sides of the reaction). In the nonlinear case, instead, the presence of multiple species in the left-hand side is interpreted as an "environment" (formally, a multiset of species) which mutually regulates the possibility of an involved species in reacting. Thus, the structural conditions for forward and backward equivalence are stated in the following form:

> Given an RN over species $S_1, \ldots, S_N$, a partition of species is an equivalence for its underlying PDS if and only if, for any two blocks H_I, H_J and any species S_i, S_i' in block H_I
>
> $$\mathbf{r}(S_i, \eta, H_J) = \mathbf{r}(S_{i'}, \eta, H_J) \qquad \text{for all environments } \eta. \tag{12}$$



(a) Solution of (9). (b) Solution of the reduced model (11).

Figure 6: Solutions of model (9) and its BE reduction (11), for $\lambda_1 = \lambda_2 = 1$, $\mu_1 = \mu_2 = 3$, and $x_1(0) = 0.5$, $x_2(0) = 0.3$, $x_3(0) = 0.2$, $x_4(0) = 2$, and $X_1(0) = 0.5$, $X_2(0) = 0.5$, and $x_4(0) = 0.5$.

Depending on the considered equivalence, **r** is a specific function that maps an RN into a real number, which can be computed by inspection of the set of reactions (Cardelli et al. 2017a). We observe that the structural condition for forward and backward equivalence are conceptually analogous to the lumpability condition in that they compare two species/states with respect to blocks. Here, there is an additional dependence on the environment; this is analogous to the dependence on action labels for computational models based on probabilistic automata (Larsen and Skou 1991). When these notions are computed for a linear ODE system induced by a Markov chain, e.g. (7), then they collapse to the structural conditions of lumpability (in particular, the environment is always an empty multiset).

Aggregated PDS

For a given forward or backward equivalence, it is possible to compute the aggregated PDS by simple manipulations of the RN that encodes the original PDS. The operations essentially amount to *renaming* the species with a fixed representative species for each block of variables, scaling rate parameters by the multiplicities of the blocks containing the species in the left-hand sides, and *merging* reactions that, as a result of this, have equal left- and right-hand sides. For example, let us consider the case of forward equivalence in (8) when $\mu_1 = \mu_2 \equiv \mu$. Let us take S_2 as the representative of the block containing species S_2 and S_3. Then, the RN can be rewritten as follows:

$$S_4 + S_1 \xrightarrow{\lambda_1} S_2 \qquad S_4 + S_1 \xrightarrow{\lambda_2} S_2 \qquad S_2 \xrightarrow{\mu/2} S_1 \qquad S_2 \xrightarrow{\mu/2} S_1$$

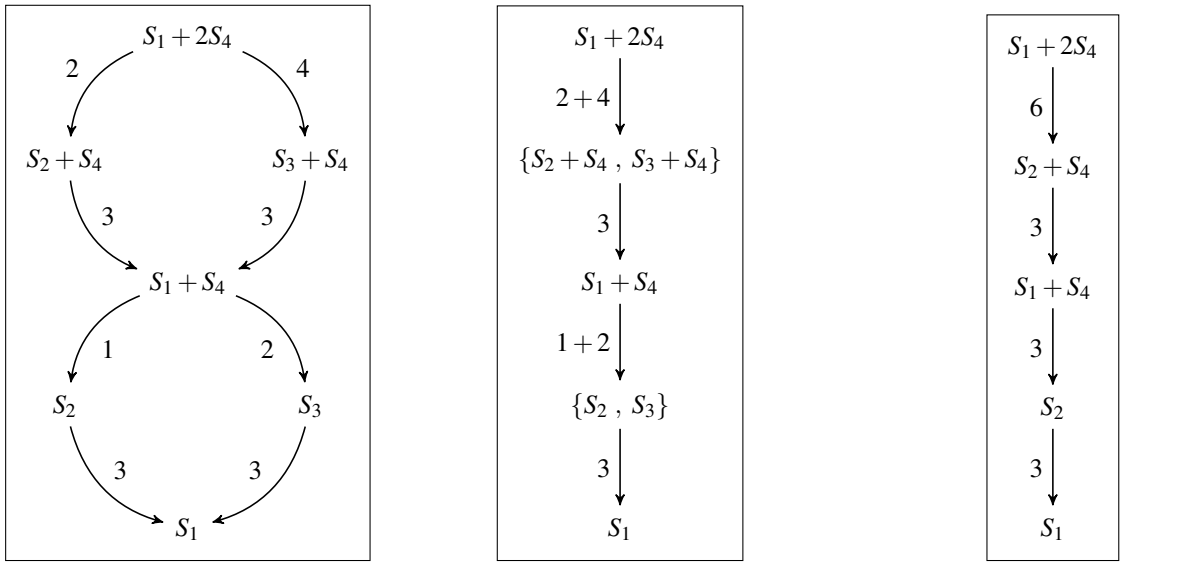
where the parameters of the last two reactions are scaled by 2. Then, we can merge the first two and the last two reactions by summing up the rates, leading to the reduced RN:

$$S_4 + S_1 \xrightarrow{\lambda_1+\lambda_2} S_2 \qquad S_2 \xrightarrow{\mu} S_1 \tag{13}$$

Then, it can be seen that this reduced RN corresponds to the reduced ODEs (10).

Maximal aggregation of PDS

Similarly to Markov chain lumpability, the structural conditions of forward and backward equivalence are based on a given candidate partition of variables. The first partition-refinement algorithm based on this notion was published in ref. (Cardelli et al. 2016a), covering PDS whose derivatives are polynomials of degree at most two; it has been subsequently extended to the general case of polynomials with arbitrary degree (Cardelli et al. 2017a). The algorithm computes the coarsest partition that refines a given input partition of species of an RN. It runs in polynomial time and space complexity, enabling the reduction of PDS with a few million variables in under 5 minutes on an ordinary laptop (Cardelli et al. 2017).



(a) Markov chain undelying the CRN (8) (b) Ordinary lumpable reduction of (a) (c) SMB reduction of (8)

Figure 7: The Markov chain (a) underlying the CRN (8) can be lumped into (b). This corresponds to the Markov chain underlying the CRN (13), which is an SMB reduction of (8).

4 SYNTACTIC MARKOVIAN BISIMULATION

RNs having all rates positive can be actually considered as CRNs with mass-action kinetics (Voit 2013). Other than the deterministic semantics based on ODEs considered so far in this text, CRNs also have a stochastic semantics based on population-based continuous time Markov chains (Gillespie 1977). The initial state of such Markov chain is given by a multi-set of species, where multiplicities denote the initial *population* (or count) of each species. The rest of the Markov chain can be generated by applying exhaustively all reactions to all states, starting from the initial one. The application of a reaction r on a state σ generates a new state σ' where all reagents of r are *consumed* (i.e., removed), and all products of r are *produced* (i.e., added). Following combinatorial arguments, the transition rate between the states σ and σ' is the kinetic constant of r multiplied by the number of possible ways in which the reagents of r can be matched with σ, corresponding to the number of possible interactions driven by r in σ.

Let us consider the RN from (8), with $\lambda_1 = 1$, $\lambda_2 = 2$, and $\mu_1 = \mu_2 = 3$. This is actually a CRN. The population-based continuous time Markov chain of this CRN for initial population $S_1 + 2S_4$ is given in Figure 7a. For example, we have a transition from the Markov chain state $(S_1 + 2S_4)$ to state $(S_2 + S_4)$ due to reaction $S_4 + S_1 \xrightarrow{\lambda_1} S_2$. The rate of the transition (2) is obtained by multiplying the rate $\lambda_1 = 1$ of the reaction by 2: the instance of S_1 in the starting state $(S_1 + 2S_4)$ can interact with either instances of S_4.

It can be shown that the following is an ordinary lumpable partition of such Markov chain:

$$\big\{\{(S_1+2S_4)\},\{(S_2+S_4),(S_3+S_4)\},\{(S_1+S_4)\},\{(S_2),(S_3)\},\{(S_1)\}\big\} \tag{14}$$

This means that we can rewrite Figure 7a in the lumped Markov chain depicted in Figure 7b having one state per equivalence class in (14). We note that, despite the partition (14) is defined for the states of the Markov chain in Figure 7a, it actually tells us that we can analyze the CRN in terms of cumulative information on the species S_2 and S_3. Indeed, the only two non-singleton blocks in (14) are invariant up to renaming of S_2 and S_3. For example, the state of Figure 7b corresponding to block $\{S_2+S_4\ ,\ S_3+S_4\}$ can be used to study the cumulative probability of being in a state with one instance of S_4, and one of either S_2 or S_3. We discover exactly this relation by means of a structural reduction technique, *syntactic Markovian bisimulation* (SMB) (Cardelli et al. 2017b), defined over the species and reactions of a CRN. SMB reduces

a CRN into a smaller one that preserves its stochastic semantics precisely in terms of ordinary lumpability, *but without having to analyze the full Markov chain*.

SMB is defined in terms of syntactic checks similar to those of forward equivalence in (12), and its relation with forward equivalence has been studied in (Cardelli et al. 2017b). Using an SMB-reduced CRN it is possible to derive directly a population-based Markov chain which corresponds to an ordinary lumpable reduction of the one of the original CRN. In particular, similarly to what done *by hand* in Figure 7b, all states of the Markov chain underlying the original CRN which are invariant up-to SMB-equivalent species are *automatically* collapsed into a single state in the Markov chain underlying the SMB-reduced CRN.

It can be shown that the partition of species $\{\{S_1\},\{S_2,S_3\},\{S_4\}\}$ is an SMB of the CRN from (8) (with $\lambda_1 = 1$, $\lambda_2 = 2$, and $\mu_1 = \mu_2 = 3$). By applying the same RN-to-RN transformation in Section 3, the CRN from (13) can also be interpreted as a reduced CRN which is equal up to ordinary lumpability, that is, the species S_2 in such reduced CRN represents the entire equivalence class $\{S_2,S_3\}$ of the source CRN. Figure 7c depicts the population-based Markov chain that would be directly generated starting from such reduced CRN, which stands in a one-to-one correspondence with that of Figure 7b.

The use of SMB has a number of advantages:

1. The population-based Markov chain underlying a CRN is typically very large or even infinite. Hence, it is often unfeasible to build the original Markov chain. Thanks to SMB, we avoid to build the original Markov chain, and we build instead directly a lumped one.
2. As for forward equivalence, the reduction induced by SMB is physically intelligible: the obtained reduced CRNs (and corresponding lumped Markov chains) are coarser versions of the original ones where we do not distinguish anymore among certain species.
3. Once we have computed an SMB-reduced CRN, we can use it for any initial condition.
4. When considering the stochastic semantics of a CRN, it is rarely the case that population Markov chains are built. Rather, the CRN is analyzed by means of stochastic simulations using, e.g., Gillespie's stochastic simulation algorithm (Gillespie 1977). Given that SMB preserves the stochastic semantics of CRNs, rather than the original CRN one can simulate its SMB reduction, which may have fewer species and reactions, hence leading to less expensive simulations.

5 TOOL SUPPORT

All analyses and reductions presented in this paper have been performed using ERODE, a mature tool for the evaluation and reduction of ordinary differential equations, and continuous time Markov chains. *ERODE* is a multi-platform application for Windows, Mac OS and Linux. It does not require any installation process, and it is available, together with a manual and sample models, at http://sysma.imtlucca.it/tools/erode.

Figure 8 contains a screenshot of *ERODE*, from which we appreciate:

- A fully-featured text editor based on the *XTEXT* framework which supports syntax highlighting, content assist, error detection and fix suggestions (top-middle of Fig. 8);
- A number of views, including a *project explorer* to navigate among different *ERODE* files (top-left of Fig. 8); an *outline* to navigate the parts of the currently open *ERODE* file (bottom-left of Fig. 8); a *plot view* to display ODE solutions and Markov chain simulations (top-right of Fig. 8); and a *console view* to display diagnostic information like warnings and model reduction statistics (bottom-right of Fig. 8).

The core layer of *ERODE* implements a number of minimization algorithms, including those for forward and backward equivalence. Finally, this layer provides support for numerical ODE solvers, using the Apache Commons Maths library (http://commons.apache.org/proper/commons-math/) or SUNDIALS (Hindmarsh, Brown, Grant, Lee, Serban, Shumaker, and Woodward 2005). When the input is a CRN, it can also be interpreted as a continuous time Markov chain, following an established approach (Gillespie 1977). Using the *FERN* library (Erhard et al. 2008), *ERODE* features continuous time Markov chain simulation.

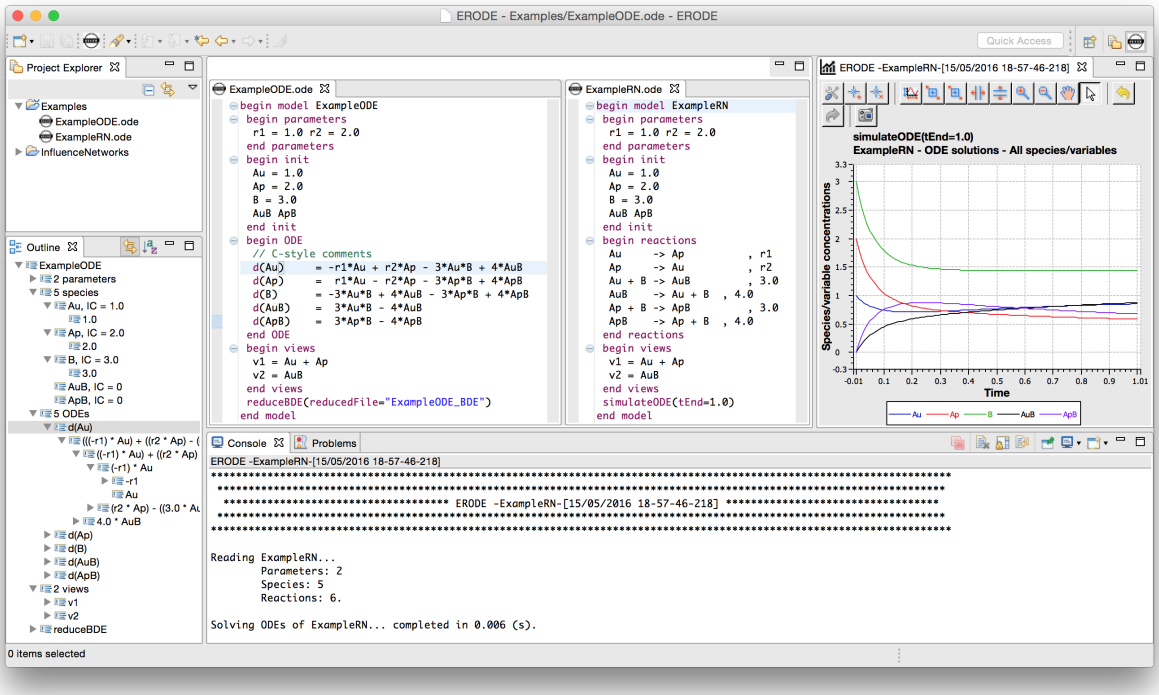


Figure 8: A screenshot of *ERODE*.

6 CONCLUSION

In this paper we have reviewed our recent work on techniques for the aggregation of polynomial dynamical systems (PDS) using efficient partition-refinement algorithms. An important feature of these techniques is the possibility of preserving variables of interest to the modeler, thus restricting the aggregation to variables that, whilst contributing to the overall dynamics, are not necessarily of direct concern to stakeholders.

Our methods are domain agnostic. Thus, they are in principle applicable in several contexts. For example, aggregations can be used as a general-purpose preprocessing step in simulation studies based on the *system dynamics* approach, which is inherently based on systems of difference/differential equations. The preservation of modeler-defined observables makes the aggregation transparent to *hybrid* simulation techniques (Mustafee et al. 2017), mixing for instance discrete-event simulation and system dynamics (Caudill and Lawson 2013), by keeping untouched the input/output variables of the dynamical system. We note that our focus on polynomial derivatives is not particularly restrictive. Indeed, it is possible to algorithmically transform systems with other nonlinearities (i.e., rational expressions, exponentials, trigonometric functions) into PDS by appropriately introducing auxiliary variables (Liu et al. 2015).

Our lumping-based aggregation techniques are orthogonal to other reduction methods for dynamical systems, for instance those based on quasi-equilibrium or quasi-steady state assumptions (Segel and Slemrod 1989). Thus, a combined application may achieve further reductions. Clearly, they are also independent from the underlying algorithms for the numerical solution, thus they can be used to further speed up the parallel simulations, e.g., (Lakshmiranganatha and Muknahallipatna 2017).

Our syntactic Markovian bisimulation is similar in spirit to forward and backward equivalence for PDS, but it applies to chemical reaction networks (CRN) that are interpreted as Markov chains. Here, the main benefit is that the aggregation analysis is made at the CRN level, which is typically (exponentially) more compact than the underlying Markov chain, enabling a significantly more efficient aggregation than traditional lumping approaches.

CRNs with stochastic dynamics are at the basis of several population-based models (Goutsias and Jenkinson 2013); their relationship with agent-based models has been recently investigated (Warnke et al. 2016). These models typically make the implicit assumption of relatively few classes (e.g., susceptible, infected, and recovered) of *homogenous*, statistically indistinguishable individual agents, each class being represented with a single variable that counts their population levels. Even under this assumption, which may not be always realistic (Reinhardt and Uhrmacher 2017), our methods can prove advantageous. For instance, in such population models evolving over a network (Goutsias and Jenkinson 2013), symmetries in the network topology can induce aggregations (Vandin and Tribastone 2016).

The techniques reviewed in this paper fundamentally hinge on the assumption of *exact* aggregation. It is natural to investigate aggregation techniques that can deal with *heterogenous* populations, aiming to collapse the behavior of nearly-similar states/variables into a single macro-variable (Iacobelli and Tribastone 2013). Some recent work has considered this issue by framing the problem into a mathematical framework for dealing with uncertain dynamical systems (Bortolussi and Gast 2016), or by computing lower and upper bounds on the homogeneization error by means of differential inequalities (Tschaikowski and Tribastone 2016). We refer the interested reader to the paper (Cardelli et al. 2018) for details on our own recent approach to this problem, which is developed as conservative extension of the equivalence relations presented here.

REFERENCES

Antoulas, A. 2005. *Approximation of Large-Scale Dynamical Systems*. Advances in Design and Control. SIAM.

Apri, M., M. de Gee, and J. Molenaar. 2012. "Complexity reduction preserving dynamical behavior of biochemical networks". *Journal of Theoretical Biology* 304(0):16 – 26.

Bortolussi, L., and N. Gast. 2016. "Mean Field Approximation of Uncertain Stochastic Models". In *46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*.

Bortolussi, L., J. Hillston, D. Latella, and M. Massink. 2013. "Continuous approximation of collective system behaviour: A tutorial". *Performance Evaluation* 70(5):317–349.

Buchholz, P. 1994. "Exact and Ordinary Lumpability in Finite Markov Chains". *Journal of Applied Probability* 31(1):59–75.

Cardelli, L., M. Tribastone, M. Tschaikowski, and A. Vandin. 2015. "Forward and Backward Bisimulations for Chemical Reaction Networks". In *26th International Conference on Concurrency Theory, CONCUR*, 226–239.

Cardelli, L., M. Tribastone, M. Tschaikowski, and A. Vandin. 2016a. "Efficient Syntax-driven Lumping of Differential Equations". In *Tools and Algorithms for the Construction and Analysis of Systems — 21st International Conference, TACAS*.

Cardelli, L., M. Tribastone, M. Tschaikowski, and A. Vandin. 2016b. "Symbolic computation of differential equivalences". In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL*, 137–150.

Cardelli, L., M. Tribastone, M. Tschaikowski, and A. Vandin. 2017a. "Maximal aggregation of polynomial dynamical systems". *Proceedings of the National Academy of Sciences* 114(38):10029–10034.

Cardelli, L., M. Tribastone, M. Tschaikowski, and A. Vandin. 2017b. "Syntactic Markovian Bisimulation for Chemical Reaction Networks". In *Models, Algorithms, Logics and Tools: Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday*, edited by L. Aceto, G. Bacci, G. Bacci, A. Ingólfsdóttir, A. Legay, and R. Mardare, 466–483. Cham: Springer International Publishing.

Cardelli, L., M. Tribastone, M. Tschaikowski, and A. Vandin. 2018. "Guaranteed Error Bounds on Approximate Model Abstractions through Reachability Analysis". In *15th International Conference on Quantitative Evaluation of Systems (QEST)*. To appear.

Cardelli, L., M. Tribastone, A. Vandin, and M. Tschaikowski. 2017. "ERODE: A Tool for the Evaluation and Reduction of Ordinary Differential Equations". In *Tools and Algorithms for the Construction and Analysis of Systems — 23rd International Conference, TACAS*.

Caudill, L., and B. Lawson. 2013. "A hybrid agent-based and differential equations model for simulating antibiotic resistance in a hospital ward". In *2013 Winter Simulations Conference (WSC)*, 1419–1430.

Erhard, F., C. C. Friedel, and R. Zimmer. 2008. "FERN - a Java framework for stochastic simulation and evaluation of reaction networks". *BMC Bioinformatics* 9(1):356.

Feret, J., V. Danos, J. Krivine, R. Harmer, and W. Fontana. 2009. "Internal coarse-graining of molecular systems". *Proceedings of the National Academy of Sciences* 106(16):6453–6458.

Feret, J., T. Henzinger, H. Koeppl, and T. Petrov. 2012. "Lumpability abstractions of rule-based systems". *Theoretical Computer Science* 431:137–164.

Forrester, J. W. 1961. *Industrial dynamics*. MIT Press.

Gillespie, D. 1977, December. "Exact Stochastic Simulation of Coupled Chemical Reactions". *Journal of Physical Chemistry* 81(25):2340–2361.

Goutsias, J., and G. Jenkinson. 2013. "Markovian dynamics on complex reaction networks". *Physics Reports* 529(2):199–264.

Hindmarsh, A. C., P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. 2005. "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers". *ACM Transactions on Mathematical Software (TOMS)* 31(3):363–396.

Iacobelli, G., and M. Tribastone. 2013. "Lumpability of fluid models with heterogeneous agent types". In *The 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*.

Kemeny, J., and J. Snell. 1976. *Finite Markov Chains*. Berlin: Springer New York, Heidelberg.

Lakshmiranganatha, S., and S. S. Muknahallipatna. 2017. "Parallel in time solution of ordinary differential equation for near real-time transient stability analysis". In *2017 Winter Simulation Conference (WSC)*, 4592–4593.

Larsen, K. G., and A. Skou. 1991. "Bisimulation through probabilistic testing". *Information and Computation* 94(1):1–28.

Liu, J., N. Zhan, H. Zhao, and L. Zou. 2015. "Abstraction of Elementary Hybrid Systems by Variable Transformation". In *Formal Methods*, edited by N. Bjørner and F. de Boer, 360–377.

Murray, J. D. 2002. *Mathematical Biology I: An Introduction*. 3rd ed. Springer.

Mustafee, N., S. Brailsford, A. Djanatliev, T. Eldabi, M. Kunc, and A. Tolk. 2017. "Purpose and benefits of hybrid simulation: Contributing to the convergence of its definition". In *2017 Winter Simulation Conference (WSC)*, 1631–1645.

Okino, M. S., and M. L. Mavrovouniotis. 1998. "Simplification of Mathematical Models of Chemical Reaction Systems". *Chemical Reviews* 2(98):391–408.

Paige, R., and R. Tarjan. 1987. "Three Partition Refinement Algorithms". *SIAM Journal on Computing* 16(6):973–989.

Radulescu, O., A. N. Gorban, A. Zinovyev, and V. Noel. 2012. "Reduction of dynamical biochemical reactions networks in computational biology". *Frontiers in Genetics* 3(131).

Reinhardt, O., and A. M. Uhrmacher. 2017. "An Efficient Simulation Algorithm for Continuous-time Agent-based Linked Lives Models". In *Proceedings of the 50th Annual Simulation Symposium*, 9:1–9:12.

Salazar, C., and T. Höfer. 2009. "Multisite protein phosphorylation — from molecular mechanisms to kinetic models". *FEBS Journal* 276(12):3177–3198.

Segel, L., and M. Slemrod. 1989. "The quasi-steady-state assumption: A case study in perturbation". *SIAM Review* 31(3):446–477.

Stewart, W. J. 1994. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press.

Stewart, W. J. 2009. *Probability, Markov Chains, Queues, and Simulation*. Princeton University Press.

Tschaikowski, M., and M. Tribastone. 2016. "Approximate reduction of heterogeneous nonlinear models with differential hulls". *IEEE Transactions on Automatic Control* 61(4):1099–1104.

Valmari, A., and G. Franceschinis. 2010. "Simple $O(m \log n)$ Time Markov Chain Lumping". In *Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS*, 38–52.

Vandin, A., and M. Tribastone. 2016. "Quantitative Abstractions for Collective Adaptive Systems". In *Formal Methods for the Quantitative Evaluation of Collective Adaptive Systems - 16th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM*, 202–232.

Voit, E. O. 2013. "Biochemical Systems Theory: A Review". *ISRN Biomathematics* 2013:53.

Volterra, V. 1931. "Variations and fluctuations of the number of individuals in animal species living together". *Animal Ecology*.

Warnke, T., O. Reinhardt, and A. M. Uhrmacher. 2016. "Population-based CTMCs and agent-based models". In *2016 Winter Simulation Conference (WSC)*, 1253–1264.

Whitmarsh, A. J., and R. J. Davis. 2016. "Multisite phosphorylation by MAPK". *Science* 354(6309):179–180.

AUTHOR BIOGRAPHIES

MIRCO TRIBASTONE is Associate Professor of Computer Science at IMT School for Advanced Studies Lucca, Italy, having held positions at the University of Southampton, UK, and at the Ludwig-Maximilians University of Munich Germany. He is Visiting Professor under a DFG Mercator Fellowship at the Technical University of Braunschweig, Germany. His main interests are in the analysis of dynamical systems, with applications to concurrency theory, computational biology, and software performance engineering. His e-mail address is mirco.tribastone@imtlucca.it.

ANDREA VANDIN is Assistant Professor at DTU the Technical University of Denmark. Before that he was an Assistant Professor at IMT School for Advanced Studies Lucca, Italy until 2017, and Senior Research Assistant at University of Southampton, UK, until 2015. His main research interests are in the development of scalable techniques for the formal quantitative system analysis. He is interested in applying his research in practice, thus he provided tool support for most of his contributions. His e-mail address is anvan@dtu.dk.