6th St.Petersburg Workshop on Simulation (2009) 715-719

Multistep Methods for Markovian Event Systems

Werner Sandmann[1]

Abstract

We consider multistep methods for accelerated trajectory generation in the simulation of Markovian event systems, which is particularly useful in cases where the length of trajectories is large, e.g. when regenerative cycles tend to be long, when we are interested in transient measures over a finite but large time horizon, or when multiple time scales render the system stiff.

1. Markovian Event Systems

Markovian models are widespread for modeling stochastic phenomena in a variety of domains. Typically, the models are given in a high-level description such as queueing networks, Petri nets, stochastic automata networks, or sets of coupled chemical reactions, amongst many others. In principle, they can be mapped to the stochastic process level in that they are uniquely defined by an initial probability distribution and a generator matrix. But in practice models tend to be very large. The size of the state space typically increases exponentially with the number of system components or, in other words, the model dimensionality. This effect is known as state space explosion and often causes models to be numerically intractable. One major advantage of simulation is that the state space need not be explicitly enumerated. Thus, a model description that reflects the event system character of the model is well suited, in particular for simulation purposes. In almost all relevant cases the structure of the underlying Markov chain is not arbitrary but state transitions correspond to certain events where similar events essentially have the same effect. Hence, they can be taken as specific discrete event systems [4], which provides a structured model description on an intermediate level of abstraction. For Markovian models the events need not be scheduled and the setting of Markovian event systems is also useful for numerical solution [6].

In order to describe a Markovian event system we have to define its state space and to specify all relevant events that may trigger state transitions. It is necessary to define under which conditions a certain event may occur, how it affects the system state and at which rate it occurs. Diverse formal specifications of Markovian event systems can be found in the literature. Here, we adopt the transition class formalism of [10]. Without loss of generality we assume that the state space is $\mathcal{S} \subseteq \mathbb{N}^d$. All events that trigger state transitions are classified according to their effects which yields transition classes. Formally, a transition

[1]University of Bamberg, Germany, E-mail: werner.sandmann@uni-bamberg.de

class is a triplet $\mathcal{C} = (\mathcal{U}, u, \alpha)$ where $\mathcal{U} \subseteq \mathbb{N}^d$ is the source state space containing all states in which the event or the corresponding state transition, respectively, is possible, $u : \mathcal{U} \to \mathbb{N}^d$ is the destination state function giving the new state $u(x) \in \mathbb{N}^d$ according to the state transition when the event occurs in state $x \in \mathcal{U}$, and $\alpha : \mathcal{U} \to \mathbb{R}$ is the transition rate function giving the rate $\alpha(x) \in \mathbb{R}$ at which the event or transition occurs in state $x \in \mathcal{U}$. Any Markovian model can be uniquely described by a set of such transition classes together with an initial distribution.

As a queueing network example consider a d-node tandem network with exponentially distributed service times where arrivals occur only at the first node according to a Poisson process with arrival rate λ. The service rates are denoted by $\mu_1, \ldots, \mu_d$ and the buffer capacities by $\nu_1, \ldots, \nu_d$. Hence, the different types of transitions are arrivals at node 1, moves from node i to node $i+1$, $0 < i < d$ and departures from node d. Therefore, $d+1$ transition classes are sufficient:

$\mathcal{C}_1 = (\mathcal{U}_1, u_1, \alpha_1)$, where

- $\mathcal{U}_1 = \{(x_1, \ldots, x_d) \in \mathbb{N}^d : x_1 < \nu_1\}$,
- $u_1 : \mathbb{N}^d \to \mathbb{N}^d, x \mapsto u_1(x) = (x_1 + 1, x_2, x_3, \ldots, x_d)$,
- $\alpha_1 : \mathbb{N}^d \to \mathbb{R}, x \mapsto \alpha_1(x) = \lambda$;

$\mathcal{C}_i = (\mathcal{U}_i, u_i, \alpha_i), i = 2, \ldots, d$, where

- $\mathcal{U}_i = \{(x_1, \ldots, x_d) \in \mathbb{N}^d : x_{i-1} > 0, x_i < \nu_i\}$,
- $u_i : \mathbb{N}^d \to \mathbb{N}^d, x \mapsto u_i(x) = (x_1, \ldots, x_{i-2}, x_{i-1}-1, x_i+1, x_{i+1}, \ldots, x_d)$,
- $\alpha_i : \mathbb{N}^d \to \mathbb{R}, x \mapsto \alpha_i(x) = \mu_{i-1}$;

$\mathcal{C}_{d+1} = (\mathcal{U}_{d+1}, u_{d+1}, \alpha_{d+1})$, where

- $\mathcal{U}_{d+1} = \{(x_1, \ldots, x_d) \in \mathbb{N}^d : x_d > 0\}$,
- $u_{d+1} : \mathbb{N}^d \to \mathbb{N}^d, x \mapsto u_d(x) = (x_1, \ldots, x_{d-1}, x_d - 1)$,
- $\alpha_{d+1} : \mathbb{N}^d \to \mathbb{R}, x \mapsto \alpha_d(x) = \mu_d$;

It becomes clear that state-dependent rates can be easily incorporated just by corresponding transition rate functions. Also the state space may be infinite, which is then implicitly given by dropping the restrictions on the source state spaces. Phase-type distributed interarrival and service times can be modeled by properly defined transition classes for any change from one to the next phase.

As a chemical reaction set consider the enzyme-catalyzed substrate conversion

$$E + S \underset{c_2}{\overset{c_1}{\rightleftharpoons}} ES \overset{c_3}{\rightharpoonup} E + P \tag{1}$$

where c_1, c_2, c_3 denote associated reaction rate constants such that the corresponding state-dependent reaction rate computes as c_i times the number of possible combinations of the required reactants. States of corresponding Markovian models are similarly defined as states of a queueing network, namely by the number of molecules of each species. If we successively number the species E, S, ES, P, a state $x = (x_1, x_2, x_3, x_4)$ expresses that there are x_1 E-molecules, x_2 S-molecules, x_3 ES-molecules, and x_4 P-molecules. Then the transition classes corresponding to the stoichiometric equation (1) are the following.

$\mathcal{C}_1 = (\mathcal{U}_1, u_1, \alpha_1)$, where

- $\mathcal{U}_1 = \{(x_1, \ldots, x_4) \in \mathbb{N}^4 : x_1, x_2 > 0\}$,
- $u_1 : \mathbb{N}^4 \to \mathbb{N}^4, x \mapsto u_1(x) = (x_1 - 1, x_2 - 1, x_3 + 1, x_4)$,
- $\alpha_1 : \mathbb{N}^4 \to \mathbb{R}, x \mapsto \alpha_1(x) = c_1 x_1 x_2$;

$\mathcal{C}_2 = (\mathcal{U}_2, u_2, \alpha_2)$, where

- $\mathcal{U}_2 = \{(x_1, \ldots, x_4) \in \mathbb{N}^4 : x_3 > 0\}$,
- $u_2 : \mathbb{N}^4 \to \mathbb{N}^4, x \mapsto u_2(x) = (x_1 + 1, x_2 + 1, x_3 - 1, x_4)$,
- $\alpha_2 : \mathbb{N}^4 \to \mathbb{R}, x \mapsto \alpha_2(x) = c_2 x_3$;

$\mathcal{C}_3 = (\mathcal{U}_3, u_3, \alpha_3)$, where

- $\mathcal{U}_3 = \{(x_1, \ldots, x_4) \in \mathbb{N}^4 : x_3 > 0\}$,
- $u_3 : \mathbb{N}^4 \to \mathbb{N}^4, x \mapsto u_3(x) = (x_1 + 1, x_2, x_3 - 1, x_4 + 1)$,
- $\alpha_3 : \mathbb{N}^4 \to \mathbb{R}, x \mapsto \alpha_3(x) = c_3 x_3$.

Obviously, compared to a desription via generator matrices the transition class formalism for Markovian event systems provides a huge gain in storage requirements and is also well suited for immediate implementation. An important point regarding computer implementations is that the state space and the generator matrix of the underlying Markov chain is implicitly coded by logical predicates and simple functions that are both easy to implement.

2. Multistep Methods

Simulation of Markovian models is straightforward. It essentially consists of generating trajectories according to the Markov chain dynamics. In discrete-time the next state is chosen according to the transition probabilities. In continuous-time the same is done according to the jump probabilities after generating the exponentially distributed state holding time. If the interest is in steady-state distributions, generation of holding times can be skipped even in the continuous-time case by simulating the uniformized discrete-time Markov chain instead but for transient measures trajectories of the CTMC are generated. However, if the time horizon is large or the system is stiff corresponding to multiple time scales this becomes exceedingly slow such that accelerated trajectory generation is desirable.

The basic idea of multistep methods is to accelerate the trajectory generation via advancing the simulation by appropriately chosen time steps rather than simulating each single event explicitly. Multistep simulation methods for stochastic models have been proposed in several contexts, including computer and communication networks [1, 7, 8] that need not be Markovian. Here, we cast multistep simulation approaches for Markovian networks in the setting of Markovian event systems, which is inspired by approaches in chemical physics [2, 5, 9]. where state spaces are potentially infinite, the reaction system typically evolves on multiple time scales.

Let C be the number of transition classes. For $i = 1, \ldots, C$ define $v_i = u_i(x) - x$ and denote by K_i the random variable describing the number of times that an event/transition according to $\mathcal{C}_i$ occurs in the time interval $[t, t + \tau)$. Then

$$X(t + \tau) = X(t) + \sum_{i=1}^{C} v_i K_i. \tag{2}$$

Accordingly, a general algorithmic framework for approximate trajectory generation where the simulation is advanced by pre-defined time steps instead of simulating every single event is as follows:

Init $t := t_0$, $x := x_0$ and t_{end};

while $t < t_{end}$

1. Compute all $\alpha_i(x)$ and $\alpha(x) := \alpha_1(x) + \cdots + \alpha_C(x)$;
2. Choose a step size τ according to some appropriate rule;
3. Compute suitable estimates $\hat{k}_1, \ldots, \hat{k}_C$ for $K_1, \ldots, K_C$;
4. Set $t := t + \tau$ and update the system state x according to (2).

If $X(t) = x$ and all transition rate functions are constant in $[t, t + \tau)$, the random variable K_i is Poisson distributed with mean $\tau\alpha_i(x)$, that is for $k \in \mathbb{N}_0$:

$$P(K_i = k) = \frac{(\tau\alpha_i(x))^k \exp(-\tau\alpha_i(x))}{k!}, \qquad i = 1, \ldots, C. \tag{3}$$

Note that even state-independent transition rate functions are not necessarily constant over time but some can vanish if the corresponding transition is not any longer possible in a state that has been reached in the meantime. Handling all transition rate functions as if they were indeed constant gives an appropriate rule for Step 2 of the above algorithm, which then yields an approximate scheme for trajectory generation. The quality of the approximation relies on the validity of the assumption of approximately constant transition rate functions, which must be formally specified and can then be used to control the approximation error.

Direct multistepping computes $\hat{k}_1, \ldots, \hat{k}_M$ as realizations of the corresponding Poisson random variables. Obviously, (2) then becomes an explicit deterministic expression for $X(t + \tau)$ as a function of x and obeys similarities to the explicit (forward) Euler method for solving systems of ordinary differential equations (ODEs). If the state components x_i are large and the Poisson random variates are approximated by their means, (2) becomes the explicit Euler formula for the deterministic event rate equations. Therefore, direct multistepping is also referred to as explicit tau-leaping in the context of chemically reacting systems [5, 12].

However, explicit ODEs solvers are known to become instable for stiff ODE systems and this effect turns over to direct multistepping for stiff Markovian systems. For stiff ODEs, implicit methods such as the implicit (backward) Euler method are often better suited, which motivates similar approaches to multistepping for Markovian models. Unfortunately, a completely implicit multistep simulation method would require to generate random variates according to the Poisson distribution with means $\tau\alpha_i(X(t+\tau)), i = 1, \ldots, C$, which depend on the unknown

random state $X(t+\tau)$. Instead, a partially implicit version is considered [9, 12]. Rewriting the random variables K_i as $K_i - \tau\alpha_i(X(t)) + \tau\alpha_i(X(t))$ and evaluating all transition rate functions in the last term at $X(t+\tau)$ instead of $X(t)$ yields

$$X(t+\tau) = X(t) + \sum_{i=1}^{C} v_i \Big(K_i - \tau\alpha_i(X(t)) + \tau\alpha_i(X(t+\tau)) \Big). \tag{4}$$

Then, in a first step, all K_i are approximated by computing realizations of Poisson random variables as with direct multistepping. Once these realizations, now denoted by $k_1, \ldots, k_C$, have been generated and given $X(t) = x$, (4) becomes an implicit deterministic equation that can be solved by, e.g., Newton iteration. Typically, the resulting estimate $\hat{x}(t+\tau)$ for $X(t+\tau)$ is not integer-valued. Therefore, in practice, the estimates to be used for the updating in Step 4 of the above algorithm are obtained by rounding the corresponding term in (4) to the nearest integer. That is

$$\hat{k}_i = \text{round}(k_i - \tau\alpha_i(x) + \tau\alpha_i(\hat{x}(t+\tau))). \tag{5}$$

As an alternative to (4), motivated by the properties of the trapezoidal rule for solving systems of deterministic ODEs, [3] proposed to substitute (4) by

$$X(t+\tau) = X(t) + \sum_{i=1}^{C} v_i \Big(K_i - \frac{\tau}{2}\alpha_i(X(t)) + \frac{\tau}{2}\alpha_i(X(t+\tau)) \Big), \tag{6}$$

which sometimes yields higher accuracy. However, it depends on the specific problem at hand whether (4) or (6) should be preferred.

3. Summary of Simulation Experiences

The concepts as presented in the previous section have been mostly applied to chemically reacting systems in recent years. In this context it has been often empirically demonstrated that the simulation of *some* stiff systems can be significantly accelerated. The integrated framework of Markovian event systems and the transition class formalism render possible to apply similar multistep methods to the simulation of Markovian models in a broader class of application domains. We have studied the simulation of Markovian queueing networks where phase-type distributions are allowed for both interarrival and service time distributions. Multiple time scales and associated stiff systems in the queueing network context results from, e.g., servers that may fail and can be repaired, where failure rates are by orders of magnitude smaller than repair rates. It turns out that multistep methods are particularly suitable for accelerated simulation of queueing networks. Even if the systems are not stiff, they are usually enormously complex and direct multistepping significantly accelerates simulation at the expense of only a small loss in accuracy. In particular, for state-independent interarrival and service rates the loss of accuracy is most often negligible. Moreover, combining direct/explicit and implicit multistepping is likely to yield further improvements [2, 11, 12]. Hence, multistep methods for Markovian event systems are very promising to provide efficient simulation methods within a broad range of application domains.

References

[1] S. Bohacek, J. P. Hespanha, J. Lee, and K. Obraczka. A hybrid systems modeling framework for fast and accurate simulation of data communication networks. In *Proceedings of ACM SIGMETRICS'03*, (2003). 58–69.

[2] Y. Cao, D. T. Gillespie, and L. R. Petzold. The adaptive explicit-implicit tau-leaping method with automatic tau selection. *J. Chemical Physics*, 126(22):224101–224101–9, (2007).

[3] Y. Cao and L. R. Petzold. Trapezoidal tau-leaping formula for the stochastic simulation of biochemical systems. In *Proceedings of the 1st Conference on Foundations of Systems Biology in Engineering*, (2005), 149–152.

[4] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, 2nd edition, (2008).

[5] D. T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chemical Physics*,(2001), 115:1716–1732.

[6] W. K. Grassmann. Finding transient solutions in Markovian event systems through randomization. In W. J. Stewart, editor, *Numerical Solution of Markov Chains*, (1991), ch. 18, 357–372. Marcel Dekker, Inc.

[7] Y. Guo, W. Gong, and D. Towsley. Time-stepped hybrid simulation (TSHS) for large scale networks. In *Proceedings of IEEE INFOCOM 2000*, (2000), vol. 2, 441–450.

[8] A. Kochut and A. U. Shankar. Timestep stochastic simulation of computer networks using diffusion approximation. In *Proceedings of MASCOTS 2006*, (2006), 247–254.

[9] M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *J. Chemical Physics*, (2003), 119:12784–12794.

[10] W. Sandmann. Structured description of Markovian network models and its potentials for efficient rare event simulation. In *Proceedings of HetNets'04*, (2004), P39/1–10.

[11] W. Sandmann. *Exposition and streamlined formulation of adaptive explicit-implicit tau-leaping.* Technical report, University of Bamberg (2009).

[12] W. Sandmann. Rare event simulation methodologies in systems biology. In G. Rubino and B. Tuffin, editors, *Rare Event Simulation using Monte Carlo Methods*, (2009), ch. 11. John Wiley & Sons.