Biochemical simulations: stochastic, approximate stochastic and hybrid approaches

Jürgen Pahle

Submitted: 1st July 2008; Received (in revised form): 13th October 2008

Abstract

Computer simulations have become an invaluable tool to study the sometimes counterintuitive temporal dynamics of (bio-)chemical systems. In particular, stochastic simulation methods have attracted increasing interest recently. In contrast to the well-known deterministic approach based on ordinary differential equations, they can capture effects that occur due to the underlying discreteness of the systems and random fluctuations in molecular numbers. Numerous stochastic, approximate stochastic and hybrid simulation methods have been proposed in the literature. In this article, they are systematically reviewed in order to guide the researcher and help her find the appropriate method for a specific problem.

Keywords: stochastic simulation; biochemical systems; approximate stochastic simulation; hybrid simulation methods; systems biology

INTRODUCTION

Biochemical networks, such as metabolism, signaling pathways or gene regulation, are characterized by a vast degree of interaction between many heterogeneous components. Often this results in complex temporal behavior that is difficult to comprehend. Therefore, computer simulations reproducing the dynamics of these systems have become an essential tool for researchers in the field [1, 2].

In most cases these simulations are based on deterministic modeling. Here, the systems are assumed to be continuous and to evolve deterministically. Their change over time can be described using ordinary differential equations (ODEs) [3]. The ODEs can then be solved by numerical integration to yield the dynamics of the system, and many efficient algorithms for that can be found in the literature, e.g. [4]. Since in the ODE approach, the biochemical systems are modeled as being continuous and deterministic, phenomena that occur due to the underlying discreteness of the systems and random fluctuations in molecular numbers, particularly in subsystems containing only few particles, are neglected.

Already in the 1970s stochastic algorithms have been devised that are able to account for fluctuations in discrete particle numbers [5] (the relationship between stochastic and deterministic models is studied in [6, 7]). However, these methods can be computationally demanding, and, therefore, numerous different (optimized/approximate/hybrid) stochastic algorithms have been developed recently in order to speed up the calculation.

The present article provides a systematic and comprehensive overview of the different stochastic simulation methods in order to help researchers...
seeking the appropriate method for their problem (see also [8–9]).

It should be mentioned that this review will be concerned with spatially homogeneous models only. When the spatial organization of biochemical systems should be taken into account, one has to resort to spatio-temporal simulation algorithms. For brief reviews on stochastic spatial approaches see e.g. [11–14]. Software capable of handling spatial models include, e.g. MCell [15, 16], MesoRD [17–19], Smoldyn [20, 21] or SmartCell [22, 23].

STOCHASTIC SIMULATIONS: ASSETS AND DRAWBACKS

Even though the deterministic approach has proven very successful, it comes with some issues:

- Small particle numbers in cellular subsystems (e.g. in signaling pathways) lead to random fluctuations which can change the dynamic behavior considerably [24].
- Bi- or multi-stable systems can not be described adequately [25].
- Stochasticity itself can be an important property of the system, e.g. in evolution, noise-induced amplification of signals or noise-driven divergence of cell fates (for a review see [26]).

For very small particle numbers (e.g. single genes) the concept of continuous concentrations is not appropriate [27].

Inherent stochastic fluctuations in molecule numbers can change the dynamic behavior of biochemical systems both quantitatively and qualitatively [28, 29]. One example is the production of proteins in random bursts [24]. Another example is the lysis/lysogeny switch in λ-phage-infected Escherichia coli cells [25]. This is not restricted to systems containing low-numbered species, but can also happen in systems particularly sensitive to noise or operating near a bifurcation point [30, 27].

Cells must act in a coordinated way despite ubiquitous molecular noise. Therefore, specific cellular systems to cope with or even exploit stochastic effects are likely to be very important and deserve further study. Stochastic modeling provides the appropriate framework to investigate, for example, the robustness of cells against random perturbations [31–38] or the constructive effects of noise [25, 39–41].

In all the above cases, random fluctuations in discrete particle numbers should be accounted for and stochastic modeling and simulation is required.

Noise can even help with elucidating the functioning of biochemical systems. Examples where noise is exploited to study the underlying cellular processes include [42–44].

However, stochastic simulations usually are much more computationally demanding than deterministic methods. Exact methods take time proportional to the number of single reaction events occurring during the simulation time.

One other drawback of stochastic modeling is that it still lacks behind in terms of appropriate analysis methods, such as stochastic bifurcation analysis [45], stochastic Metabolic Control Analysis (MCA) [46] or stochastic sensitivity analysis [47].

Now, the practical question arises when stochastic simulation should be preferred over numerical integration of ODEs for a specific system one wants to analyze. Definite indicators necessitating stochastic modeling are (i) when the particle numbers are in a range where the concept of continuous concentrations is no longer appropriate, or (ii) when phenomena associated with stochasticity itself are the object of research. Beyond that most often rules of thumb based on the rough particle numbers in the system have been used to answer the question. This heuristic can be justified by the asymptotic equivalence of the Master Equation and the chemical Langevin equation (see Langevin method section) and the observation that often the relative magnitude of random fluctuations is inversely proportional to the square root of the system’s particle numbers [26]. However, giving a general threshold for the particle numbers, above which it is safe to employ deterministic methods, is surely impossible. The emergence of stochastic effects is very model-dependent and this has to be checked in each individual case [30, 27].

STOCHASTIC SIMULATION METHODS

The stochastic approaches are based on the notion of a stochastic reaction constant \( c_R \) for each reaction \( R \) in the system. \( c_R \) is the average probability that a specific combination of substrate particles in the system will react in the next infinitesimal time step \( dt \) according to reaction \( R \). Multiplied with the number of possible \( R \) substrate particle
combinations $h_{ij}$ (see [5] for details) we get a propensity $a_\mu$ for reaction $R_\mu$ as $a_\mu = h_{ij}c_{ij}$ with:

$$a_\mu \text{dt} = \text{the average probability that a reaction } R_\mu \text{ will occur in the system in the next infinitesimal time step } dt.$$  \hfill (1)

$a_\mu$ can be derived from the conventional deterministic reaction rate in the case of mass action kinetics.

It turns out that within this framework, the biochemical system can be identified with a (homogeneous) jump-type Markov process represented by a so-called Chemical Master Equation (CME) [5, 48], which describes the time evolution of the system state probability distribution. In general, this differential-difference equation is infinitely dimensional. Solving it, either analytically or numerically, still is a challenging task (the interested reader is referred to [49–51]). Therefore, one resorts to stochastic simulation methods that, using Monte Carlo calculation schemes, generate instances of the underlying stochastic process according to the Master Equation. However, since each run yields only one single trajectory, several runs have to be computed to be able to calculate statistics.

Stochastic simulation algorithms (SSA) can roughly be divided into exact, approximate or hybrid methods depending on whether or not they introduce approximations or combine different approaches into one calculation scheme.

**EXACT STOCHASTIC METHODS**

The so-called exact stochastic methods correctly account for inherent stochastic fluctuations and correlations. In addition, the discrete nature of the system is considered. Hence, they remain valid for very small particle numbers.

However, since they explicitly simulate each reaction event in the system, they have time complexity approximately proportional to the overall number of particles $\sum X_i$ present in the system. Therefore, they are prohibitively slow on large systems.

Gillespie [5] proposed two simple SSA, namely the Direct Method and the First Reaction Method. They are based on the so-called Reaction Probability Density Function:

$$P(\tau, \mu|x, t) = a_\mu(x) \exp \left(-\sum_{\mu=1}^{M} a_\mu(x) \tau \right).$$  \hfill (2)

This function determines the probability $P(\tau, \mu|x, t)\,d\tau$ that starting in state $x$ at time-point $t$ the next reaction in the system will occur in the time interval $[t+\tau, t+\tau+d\tau]$ and will be of type $R_\mu$. The reaction times are exponentially distributed (homogeneous Poisson process).

By iteratively drawing random numbers according to that density function and updating the system state according to the chosen reaction’s stoichiometry the system can be simulated over time, one reaction event after the other. The Direct Method and the First Reaction Method, as well as the other exact methods described below, are mathematically equivalent but differ in how they calculate samples of $P(\tau, \mu|x, t)$.

**Direct Method**

The Direct Method [5, 52] uses two random numbers per step to separately compute (i) the stochastic time step $\tau$ and (ii) the type of the next reaction $R_\mu$. A similar method was already proposed in [53]. The algorithm proceeds as follows:

1. First, the sum of all propensities for the $M$ possible individual reactions is calculated:

$$a_0 = \sum_{\mu=1}^{M} a_\mu$$  \hfill (3)

2. The stochastic time step is calculated:

$$\tau = -\frac{1}{a_0} \ln r_1$$  \hfill (4)

Here $r_1$ is denoting a uniformly distributed random number in the range $[0, 1]$.

3. Finally, the reaction taking place is determined. For this purpose, a second uniformly distributed random number $r_2$ is generated and the reaction $\mu$ chosen according to the following criteria:

$$\sum_{\mu=1}^{M-1} \frac{d_\mu}{a_0} \leq r_2 < \sum_{\mu=1}^{M} \frac{d_\mu}{a_0}$$  \hfill (5)

4. The corresponding reaction $R_\mu$ is realized, i.e. the number of the participating molecules is increased or decreased according to the stoichiometry, and the time is incremented by $\tau$. The whole process (1–4) is repeated as many times as necessary to reach the desired simulation time.
First Reaction Method

The First Reaction Method [5, 52] uses $M$ random numbers in each step to compute putative reaction times $\tau_\mu$ for each of the $M$ reactions $R_\mu$ in the system. The $\tau_\mu$ are exponentially distributed with parameter $a_\mu$. The reaction with the smallest reaction time $\tau_{\text{min}} = \min(\tau_1, \ldots, \tau_M)$, is executed and all putative reaction times are recalculated before the next step. Because of the wasteful use of random numbers and redundant recalculations the First Reaction Method is computationally inefficient. However, it can be numerically advantageous because, due to the machine-dependent resolution of floating-point numbers, the Direct Method cannot calculate rare reaction events amongst very frequent reactions in the system.

Next Reaction Method

Gibson and Bruck [54] improved the computational complexity of the First Reaction Method by the intelligent use of data structures:

- A so-called dependency graph stores dependencies between the reactions → redundant recalculations of $a_\mu$ are avoided.
- Absolute putative reaction times $\tau_\mu$ (since the beginning of simulation) are used instead of relative ones (since the last reaction event). Random numbers are ‘recycled’ during a reaction time update according to $\tau_\mu = (a_{\mu, \text{old}}/a_\mu(x)) (\tau_{\mu, \text{old}} - t) + t$.
- An indexed priority queue contains all reactions sorted according to their putative reaction times → the next reaction can always be found in the root of the tree in constant time.

In the Next Reaction Method recalculations of $a_\mu$ are minimized, and, asymptotically, only one random number per step is needed. Therefore it performs well, in particular, in the case of many reacting species and reactions.

Variants and extensions of the exact methods

Cao et al. (Optimized Direct Method [55]) and McCollum et al. (Sorting Direct Method [56]) reduce the computational cost of the Direct Method by sorting the reactions according to their propensities.

Li and Petzold (Logarithmic Direct Method [57]) use partial sums of propensities and a binary search for the next reaction in order to accelerate the computation.

Slepoy et al. [58] describe an algorithm that, in each iteration, chooses the next reaction event in constant-time independent of the number of different reactions.

The Direct Method has also been extended to accommodate delayed reactions often used to model, e.g. transcription or translocation processes [59–62].

Finally, there exist fast implementations that utilize special hardware, such as field-programmable gate arrays (FPGA) [63, 64] or computer clusters [65].

APPROMIMATE STOCHASTIC METHODS

The huge computational effort needed for exact stochastic simulation entailed a lively search for approximate simulation methods that sacrifice an acceptable amount of accuracy in order to speed up the simulation. The proposed methods often involve a grouping of reaction events (Probability-weighted Dynamic Monte Carlo (PW-DMC), $\tau$-Leap Method), i.e. they permit more than one reaction event per step.

One approximate stochastic method, StochSim, does not primarily aim at speeding up the simulation and shall be discussed first.

StochSim, mesoscopic approach

In the mesoscopic approach by Morton-Firth (now Firth) [66–68] single particles are distinguished and represented by separate software objects. Though, their positions and velocities in the reaction volume are disregarded. This characteristic makes StochSim mesoscopic, residing on a middle conceptual level between microscopic molecular dynamics and macroscopic population-based approaches.

In each step two particles (or one particle and one pseudoparticle for monomolecular reactions) are randomly chosen. Whether a reaction event takes place between the two, is then determined using a random number and a look-up table.

This procedure can be even more time-consuming than the Direct Method when many nonreactive selections are chosen. The advantages are that multistate particles are possible (with different reaction rates for each configuration) and that the life-cycle of single particles can be traced. However, this could also be accomplished by extending one of the exact algorithms to keep track of single molecules.
The approach has also been extended to cover simple 2D-spatial modeling with stationary particles [69].

Probability-Weighted Dynamic Monte Carlo
The PW-DMC [70] (PW-DMC method) is a rather ad hoc approach in which reactions with high probability are allowed to fire multiple times. Instead of considering single fast reaction events separately, several events are grouped together and simulated as if they were one event. This reduces the effective propensity \( a_\mu \) of fast reactions and can reduce computation times, but has the major drawback that fluctuations can be misdescribed.

\( \tau \)-Leap Methods
In 2001, Gillespie [71] developed an approximate stochastic simulation method named \( \tau \)-Leap Method to accelerate the simulation procedure. This method avoids the meticulous reconstruction of every individual reaction event. Instead, it leaps along the time axis in steps of length \( \tau \) containing many single reaction events. \( \tau \) has to fulfill the so-called Leap Condition: it must be small enough that no significant change in the propensities \( a_\mu \) during \([t, t + \tau]\) occurs. Then, the reaction channels effectively decouple and the number of firings \( K_\mu \) during time \( \tau \), starting from state \( x \) at time \( t \), can be approximated by Poisson distributed random variables:

\[
K_\mu(t; x, t) = P(a_\mu(x), \tau) = \frac{(a_\mu(x) \tau)^k}{k!} e^{-a_\mu(x) \tau}.
\]

In each step a Poisson random number \( k_\mu \) is drawn for each reaction \( R_\mu \) and the system state is updated according to:

\[
x(t + \tau) = x(t) + \sum_{\mu=1}^{M} k_\mu \Delta x_\mu.
\]

Due to the drawing of Poisson random numbers, each \( \tau \)-leap is more expensive than one step of the Direct or First Reaction Method. However, since many single reaction events can be leaped over when \( \tau \) is large enough, the simulation can be much faster after all. Figure 1 illustrates the procedure.

The most important question remaining is how to choose an appropriate \( \tau \)-value. Determining this value involves a trade-off between accuracy of the simulation and computation time. In the original article, Gillespie proposed a simple \( \tau \)-choosing strategy. This has been improved upon in [72] and [73].

![Figure 1: Schema of the \( \tau \)-Leap Method.](https://academic.oup.com/bib/article/10/1/53/229127)

A number of variants and extensions of \( \tau \)-Leaping have been developed recently that tackle some of the issues of the \( \tau \)-Leap Method from 2001:

- Rathinam et al. [74] draw on the analogy to numerical integration of ODEs and describe an ‘Implicit \( \tau \)-Leaping Method’ in order to deal with stiffness in the system. This method has been improved by Cao and Petzold in [75]. The algorithm in [76] automatically switches between explicit and implicit \( \tau \)-Leaping.
- Tian and Burrage [77] and Chatterjee et al. [78] suggested to use (bounded) binomial random numbers instead of Poisson distributed ones. This so-called ‘Binomial \( \tau \)-Leaping’ avoids one of the problems of the \( \tau \)-Leap Method, namely the generation of negative particle numbers in some cases. An extension is the ‘Multinomial \( \tau \)-Leaping Method’ in [79].
- An alternative approach to avoid negative particle numbers has been developed by Cao et al. [80]. In this ‘Modified \( \tau \)-Leaping Method’ exact stochastic simulation, allowing only one reaction event per step, is used for particle numbers that are critically low.
- Burrage and Tian [81] constructed ‘Poisson Runge-Kutta Methods (PRK)’ to increase the efficiency of \( \tau \)-Leaping.
- The ‘R-Leaping’ in [82] and the ‘K-Leap Method’ in [83] are variants of the \( k_\mu \)-Leaping method described in [71].
- The ‘Unbiased \( \tau \)-Leaping Method’ [84] attempts to correct for the bias in \( \tau \)-Leaping.
- ‘Binomial \( \tau \)-DSSA’ [85] extends \( \tau \)-Leaping by delayed reactions.
- Finally, there is also a version for spatial simulations called ‘Br-SSSA’ by Marquez-Lago et al. 2007 [86].
Langevin Method

If the value of $\tau$ can be chosen big enough such that every reaction channel on average produces a very large number of firings ($\langle P(a_\mu(x), \tau) \rangle = a_\mu(x) \tau \gg 1$) while still satisfying the Leap Condition, the Poisson random variables can be approximated by normal random variables $N_\mu$ [87]:

$$K_\mu(t; x, \tau) = P_\mu(a_\mu(x), \tau) 
\approx N_\mu(a_\mu(x)\tau, a_\mu(x)\tau)$$

$$= a_\mu(x)\tau + \sqrt{a_\mu(x)\tau} \cdot N_\mu(0, 1)$$

In this case, the $\tau$ value is called to be 'macroscopically infinitesimal'. The Poisson random variables for the system state update can be replaced by normal random variables which are easier to calculate. Conceptually, the procedure is now equivalent to the (chemical) Langevin equation, a stochastic differential equation (SDE):

$$x(t + \tau) = x(t) + \sum_{\mu=1}^{M} v_\mu a_\mu(x)\tau + \sum_{\mu=1}^{M} v_\mu \times \sqrt{a_\mu(x)\tau} \cdot N_\mu$$

with $N_\mu$ unit normal random variables.

In the limiting case $a_\mu(x)\tau \rightarrow \infty$, the last (noise) term becomes negligibly small compared with the second (see [87]) yielding the Euler update method for the numerical integration of ODEs. The Langevin Method therefore illustrates how the SSA are connected to the deterministic method through a series of approximations (SSA $\rightarrow$ $\tau$-Leap Method $\rightarrow$ Langevin Method $\rightarrow$ deterministic reaction rate equations).

Complex stochastic kinetics

The stochastic formalism is based on irreversible elementary reactions.

How to correctly handle derived kinetics, where a number of elementary reactions have been lumped into one complex reaction, is an important question because they are used in many models and often kinetic parameters for the underlying elementary reactions are not even known.

Bundschuh et al. 2003 [88] observed that a naive direct stochastic simulation of higher order kinetics can lead to a failure of correctly describing the fluctuations and even the mean of particle numbers. Nevertheless, in many cases the use of higher order terms in stochastic simulation is indeed justified as is discussed in Rao and Arkin [89] and Cao et al. [90] for the stochastic quasi-steady-state approximation (QSSA) and Michaelis–Menten kinetics (see also the Stochastic quasi-equilibrium approximations section below).

Hybrid methods

The basic idea of hybrid simulation methods is to combine the advantages of complementary simulation approaches: the whole system is subdivided into appropriate parts and different simulation methods operate on these parts at the same time. Figure 2 shows an exemplary schematic view of this procedure. Here, we have two subsystems containing the fast and slow reactions, respectively. Fast reactions often involve high-numbered species, e.g. in metabolism. Slow reactions or reactions involving low-numbered species can frequently be found in signal transduction or gene expression systems. The two subsystems are simulated iteratively by using adequate simulation methods, for instance, numerical integration of ODEs and stochastic simulation, respectively.

Mathematically, this corresponds to a partitioning of the CME (for a rigorous derivation of the partitioning process and mathematical details of the approximations employed, see [91–93]).

In between two reaction events in the slow/discrete subset of reactions, the fast subset evolves due to the action of the fast reactions only. This means that, during that time, its behavior can be approximated using ODEs, SDEs or approximate stochastic simulation methods independently of the slow subset.

However, the reaction propensities $a_\mu$ of slow reactions are, in general, also dependent on species whose concentrations are changed by fast reactions. Taking this into account, the slow subset no longer constitutes a homogeneous Poisson process. Instead,
it has to be described by a Master Equation with time-varying transition propensities $a_\mu(t)$. Gillespie [94] derives the correct Reaction Probability Density Function for this case:

$$P(\tau, \mu | x, t) = a_\mu(t + \tau) \exp\left(- \int_0^{t+\tau} \sum_{1 \leq \mu \leq M} a_\mu(t) \, dt\right).$$  \hspace{1cm} (11)

Fast reactions involving high-numbered species (e.g., metabolism) can be simulated efficiently with hybrid algorithms. These reactions, in particular, would slow down a pure stochastic simulation considerably. This potential speed-up and that random fluctuations are considered where necessary are the main advantages of hybrid approaches.

Nevertheless, there are still some open questions:

- Synchronization of the subnetworks? (Provision for time-varying $a_\mu$ in the slow subnet. Conversion between continuous concentrations and discrete particle numbers.)
- Reliable criteria for the partitioning?
- How to handle fast reactions with small particle numbers involved?
- Dynamic repartitioning $\rightarrow$ additional computational overhead.

For instance, dynamic/automated (re-)partitioning is not only important when the system state varies considerably over time, e.g., in oscillating systems. It also makes the methods much more user friendly by avoiding tedious modeler intervention. However, since dynamic partitioning causes computational overhead, the decision whether it should be used or not is always associated with a trade-off between user friendliness/accuracy and simulation speed. For very small systems it is probably not worthwhile to employ dynamic partitioning. A fixed partitioning can also be used when the subsystems are clearly different in terms of reaction velocities, for instance metabolic pathways coupled to signaling or reaction systems coupled by fast diffusion.

Likewise, on the one hand, taking into account time-varying $a_\mu(t)$ in the slow/discrete subsystem requires algorithms that are mathematically more involved. On the other hand, the increase in accuracy associated with it renders additional updates of the propensities $a_\mu(t)$ unnecessary so that larger steps can be taken.

Partitioning strategies based on reaction velocities usually aim at speeding up the simulation by taking the fast reactions out of the stochastically simulated subsystem. In contrast, particle number-based partitioning is more concerned with correctly treating reactions when particle numbers are so low that the system cannot be assumed continuous any longer. Usually a combination of different criteria are used for partitioning.

A number of hybrid algorithms [95–99] are based on a partitioning of the reaction system and the simulation of the slow subsystem using a stochastic algorithm which considers time-varying $a_\mu$, as described above. However, they differ in how exactly the nonhomogeneous Poisson process is sampled. Also, the partitioning strategies used are different, e.g., based on particle numbers, reaction velocities or (relative) level of fluctuations in particle numbers.

The hybrid methods in [91, 100, 101] also use a combination of stochastic simulation and numerical integration of ODEs/SDEs on a partitioned system, but do not consider time-varying $a_\mu$ in the slow subsystem. Instead, for instance, the method by Haseltine and Rawlings [91] introduces a ‘probability of no reaction’ in order to limit the approximation error.

Other methods, e.g., combine the Next Reaction Method with τ-Leaping [102], employ agent-based frameworks [103] or meta-algorithms [104], use an implicit partitioning scheme [105], or integrate three [106] or even four [107] different methods into a hybrid simulation scheme.

Table 1 gives an overview of the different hybrid simulation methods. They are characterized along different dimensions, namely which methods they integrate, whether the partitioning is dynamic/automated or user-defined, which partitioning policy is used and whether or not time-varying $a_\mu(t)$ in the slow/discrete subsystem(s) are considered or not (→ synchronization).

Hybrid modeling and simulation might become particularly important in the future because of the emergence of ever more complex and heterogeneous models that integrate signaling, metabolism and gene expression and, as such, require a simulation on multiple scales.

**Stochastic quasi-equilibrium approximation**

Several authors proposed the use of quasi-equilibrium approximations (QEA) in stochastic simulations, e.g. [92, 93, 108–112].
For instance, Cao et al. [93] devised the ‘Slow-Scale Simulation Method’ in which the system is subdivided reaction- and species-wise into a fast and a slow subsystem. A ‘virtual fast process’ is defined which consists of the fast species evolving under only the fast reactions, i.e. all slow reactions are turned off. This process needs to be stable in a way such that its relaxation time is much faster than the expected time to the next slow reaction. Then, on the slow time scale, it can be approximated using its asymptotic solution. Finally, the system is simulated considering only the slow reaction events explicitly. For this, the Direct Method can be used with modified \( a/C_2 \) (‘slow-scale propensity functions’). In each step, values for the fast species are determined by drawing random samples from the asymptotic probability distribution. Cao et al. give two simple examples for calculating the slow-scale propensities \( \pi_n \) in their article, but this can be difficult for more complex models.

Stochastic QEA can be regarded as residing in the middle between exact stochastic simulation methods and hybrid approaches. The system is partitioned into subsets of reactions or species as in hybrid simulation methods. However, only the slow/discrete subset is explicitly simulated.

**CONCLUSIONS**

Stochastic modeling and simulation is important whenever fluctuation phenomena play a role either as a destructive or constructive element.

Despite of the multitude of proposed stochastic methods, there seems to be none that fits all problems. For smaller models in terms of particle numbers and whenever a correct treatment of the fluctuations is required, one of the exact methods should be used. There exist a number of corresponding software tools (Table 2), e.g. Copasi or Dizzy [113, 114] (see Table 2), and these methods are also relatively straightforward to implement.

For an accelerated stochastic simulation of bigger systems, one of the approximate methods could be considered. However, some of them are ad hoc procedures tailored to specific problems rather than general stochastic solvers. The particle-based Stoch-Sim algorithm might suit, if one wants to model multistate molecules or to trace the life-cycle of single particles. The \( \tau \)-Leap Method or one of its variants seems to be a promising approach. However, this scheme is affected by stiffness which is often present in biochemical models. All approximate methods should be used with care. Their assumptions have to be thoroughly checked in each case. Otherwise they can misdescribe the fluctuations (e.g. the PW-DMC method and different forms of lumping) in some cases. With the exception of the \( \tau \)-Leap Method (implemented in [22, 114–116] general software tools have been mostly missing. However, this has already started to change (see e.g. [117]). Some of the approximate methods are rather cumbersome to implement and they often need intervention by the modeler.

### Table I: Overview and systematic classification of the different hybrid simulation methods

<table>
<thead>
<tr>
<th>Hybrid approaches</th>
<th>Methods combined</th>
<th>Dynamic partitioning</th>
<th>Partitioning criteria</th>
<th>Time-varying ( a )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alur et al. [103]</td>
<td>Direct Method/ODE</td>
<td>✓</td>
<td>Particle no.</td>
<td>✓/×</td>
</tr>
<tr>
<td>Haseltine and Rawlings [91]</td>
<td>Direct Method/ODE or SDE</td>
<td>✓</td>
<td>Heuristics</td>
<td>✓/×</td>
</tr>
<tr>
<td>Pahle [100, 101]</td>
<td>Next Reaction Method/ODE</td>
<td>✓</td>
<td>Particle no.</td>
<td>✓/×</td>
</tr>
<tr>
<td>Adalsteinsson et al. [118]</td>
<td>Direct Method/ODE</td>
<td>✓</td>
<td>User defined</td>
<td>✓/×</td>
</tr>
<tr>
<td>Bentele and Eils [95]</td>
<td>Next Reaction Method/ODE</td>
<td>✓</td>
<td>Relat. fluct., particle no.</td>
<td>✓/×</td>
</tr>
<tr>
<td>Burrage et al. [106]</td>
<td>Direct M./( \tau )-Leap/SDE</td>
<td>✓</td>
<td>Propensities, particle no.</td>
<td>✓/×</td>
</tr>
<tr>
<td>Kiehl et al. [100]</td>
<td>Direct Method/ODE</td>
<td>✓</td>
<td>User defined</td>
<td>✓/×</td>
</tr>
<tr>
<td>Neogi et al. [96]</td>
<td>Stoch. Sim./ODE</td>
<td>✓</td>
<td>Particle no.</td>
<td>✓/×</td>
</tr>
<tr>
<td>Puchalka and Kierzek [102]</td>
<td>Next Reaction M./( \tau )-Leap Method</td>
<td>✓</td>
<td>Substrate no., relat. prop.</td>
<td>✓/×</td>
</tr>
<tr>
<td>Takahashi et al. [104]</td>
<td>Next Reaction Method/ODE</td>
<td>✓</td>
<td>User defined</td>
<td>✓/×</td>
</tr>
<tr>
<td>Vasudeva and Bhalia [105]</td>
<td>Direct Method/ODE</td>
<td>✓</td>
<td>Propensities, particle no.</td>
<td>✓/×</td>
</tr>
<tr>
<td>Alfonsi et al. [97]</td>
<td>Next Reaction Method/SDE</td>
<td>✓</td>
<td>Propensities</td>
<td>✓/×</td>
</tr>
<tr>
<td>Salis and Kaznessis [121, 118]</td>
<td>Next Reaction Method/SDE</td>
<td>✓</td>
<td>Propensities, particle no.</td>
<td>✓/×</td>
</tr>
<tr>
<td>Griffith et al. [99]</td>
<td>Direct Method/ODE</td>
<td>✓</td>
<td>Propensities, particle no.</td>
<td>✓/×</td>
</tr>
<tr>
<td>Harris and Clancy [107]</td>
<td>Direct M./( \tau )-Leap/Langevin M./ODE</td>
<td>✓</td>
<td>Propensities</td>
<td>✓/×</td>
</tr>
<tr>
<td>Wagner et al. [122]</td>
<td>First Reaction M./discr. Gauss/ODE</td>
<td>✓</td>
<td>Error criterion</td>
<td>✓/×</td>
</tr>
</tbody>
</table>

✓/× denotes that the authors describe how this feature can be realized in principle but do not demonstrate it in their examples or implementation.
A very promising direction is the development of hybrid methods because they directly deal with the important problem of stiffness, which is often present in biochemical models. They appear to be flexible enough to allow for general stochastic solvers in the future even for very big and heterogeneous models. However, for the time being an established type of partitioning (reaction-wise and/or species-wise) and, above all, reliable criteria for an automatic and adaptive partitioning during the simulation are still missing. Hybrid algorithms are the most challenging methods to implement. Most of them also still need much user intervention. There exist already a few software tools, which allow for hybrid simulation, e.g. [65, 98, 101, 118], and this number is expected to grow in the future.

See Table 2 for a list of selected software systems which support stochastic simulation of biochemical systems. In addition, there is also the SBML stochastic test suite [119], which facilitates testing the accuracy of stochastic simulation software.

Table 2: List of selected software systems for biochemical systems capable of stochastic simulation

<table>
<thead>
<tr>
<th>Software</th>
<th>Methods</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioNetS</td>
<td>Direct Method, Langevin Eq., Hybrid Solver</td>
<td>[123, 118]</td>
</tr>
<tr>
<td>Cellware</td>
<td>Direct M., Next Reaction M., r-Leap (explicit), Langevin-type Eq. with Poisson noise-term</td>
<td>[115, 124, 125]</td>
</tr>
<tr>
<td>Copasi</td>
<td>Next Reaction Method, Hybrid Solver</td>
<td>[113, 101]</td>
</tr>
<tr>
<td>Dizzy</td>
<td>Direct M., Next Reaction M., r-Leap</td>
<td>[114, 126]</td>
</tr>
<tr>
<td>Dynetica</td>
<td>Direct Method,</td>
<td>[127, 128]</td>
</tr>
<tr>
<td>E-Cell</td>
<td>Direct M., Next Reaction M., r-Leap (implicit and explicit), Langevin Eq., Hybrid Solver</td>
<td>[116, 129]</td>
</tr>
<tr>
<td>Hy3S/SynBioSS</td>
<td>Direct M., Next Reaction M., SDE Solver, Hybrid Solver</td>
<td>[98, 110]</td>
</tr>
<tr>
<td>Jarnac</td>
<td>Direct Method</td>
<td>[130]</td>
</tr>
<tr>
<td>Kinetikit</td>
<td>Direct Method, Hybrid Solver</td>
<td>[31, 105]</td>
</tr>
<tr>
<td>MCell</td>
<td>Spatial stochastic simulation</td>
<td>[15, 16]</td>
</tr>
<tr>
<td>MesoRD</td>
<td>Spatial stochastic simulation</td>
<td>[17, 18, 19]</td>
</tr>
<tr>
<td>Moleculizer</td>
<td>First Reaction Method</td>
<td>[132, 133]</td>
</tr>
<tr>
<td>Smart Cell</td>
<td>Direct M., r-Leap, Spatial stochastic simulation</td>
<td>[22, 23]</td>
</tr>
<tr>
<td>Smoldyn</td>
<td>Spatial stochastic simulation</td>
<td>[20, 21]</td>
</tr>
<tr>
<td>StochKit</td>
<td>Direct M., Optimized Direct M.,</td>
<td>[117, 65]</td>
</tr>
<tr>
<td></td>
<td>Different r-Leaping Methods,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ssStochastic SSA, multiscale SSA</td>
<td></td>
</tr>
<tr>
<td>StochSim</td>
<td>Stochastic mesoscopic approach</td>
<td>[135, 68, 67]</td>
</tr>
<tr>
<td>Stocks</td>
<td>Direct Method (lin. growing</td>
<td>[136, 137]</td>
</tr>
<tr>
<td></td>
<td>vol. &amp; and cell division possible)</td>
<td></td>
</tr>
<tr>
<td>Stode</td>
<td>Direct Method</td>
<td>[138, 139]</td>
</tr>
</tbody>
</table>

The type of the simulation engines implemented and corresponding references are given.

Key Points

- The dynamics of biochemical systems can be considerably affected by random fluctuations in discrete particle numbers.
- Stochastic modeling and simulation are required to study these phenomena.
- A multitude of different stochastic simulation methods have been proposed that can be classified as exact, approximate or hybrid.
- Approximate methods have been developed to speed up the computationally expensive exact stochastic simulation.
- Hybrid methods are particularly promising because they combine different approaches in order to deal with stiffness in biochemical models.

Acknowledgements

I would like to thank Ursula Kummer and Sven Sahle for many interesting discussions. Since the field has been growing considerably, in particular during the last few years, it is almost impossible to cover every method proposed. Therefore, my apologies to all authors whose work I did not cite in this article either due to space limitations or plain ignorance.

References


