Székely *et al. BMC Systems Biology* 2012, **6**:85
http://www.biomedcentral.com/1752-0509/6/85

BMC Systems Biology

METHODOLOGY ARTICLE Open Access

A higher-order numerical framework for stochastic simulation of chemical reaction systems

Tamás Székely Jr[1*], Kevin Burrage[1,2], Radek Erban[3] and Konstantinos C Zygalakis[4,5]

Abstract

Background: In this paper, we present a framework for improving the accuracy of fixed-step methods for Monte Carlo simulation of discrete stochastic chemical kinetics. Stochasticity is ubiquitous in many areas of cell biology, for example in gene regulation, biochemical cascades and cell-cell interaction. However most discrete stochastic simulation techniques are slow. We apply Richardson extrapolation to the moments of three fixed-step methods, the Euler, midpoint and θ-trapezoidal τ-leap methods, to demonstrate the power of stochastic extrapolation. The extrapolation framework can increase the order of convergence of any fixed-step discrete stochastic solver and is very easy to implement; the only condition for its use is knowledge of the appropriate terms of the global error expansion of the solver in terms of its stepsize. In practical terms, a higher-order method with a larger stepsize can achieve the same level of accuracy as a lower-order method with a smaller one, potentially reducing the computational time of the system.

Results: By obtaining a global error expansion for a general weak first-order method, we prove that extrapolation can increase the weak order of convergence for the moments of the Euler and the midpoint τ-leap methods, from one to two. This is supported by numerical simulations of several chemical systems of biological importance using the Euler, midpoint and θ-trapezoidal τ-leap methods. In almost all cases, extrapolation results in an improvement of accuracy. As in the case of ordinary and stochastic differential equations, extrapolation can be repeated to obtain even higher-order approximations.

Conclusions: Extrapolation is a general framework for increasing the order of accuracy of any fixed-step stochastic solver. This enables the simulation of complicated systems in less time, allowing for more realistic biochemical problems to be solved.

Keywords: Stochastic simulation algorithms, τ-leap, High-order methods, Monte Carlo error

Background

Biochemical systems with small numbers of interacting components have increasingly been studied in recent years, as they are some of the most basic systems in cell biology [1-3]. Stochastic effects can strongly influence the dynamics of such systems. Applying deterministic ordinary differential equation (ODE) models to them, which approximate particle numbers as continuous concentrations, can lead to confusing results [4,5]. In some cases, even systems with large populations cannot be accurately modelled by ODEs. For instance, when close to a bifurcation regime, ODE approximations cannot reproduce the behaviour of the system for some parameter values [6]. Stochastic systems can be modelled using discrete Markov processes. The density of states of a well-stirred stochastic chemical reaction system at each point in time is given by the chemical master equation (CME) [7,8]. The stochastic simulation algorithm (SSA) [9] is an exact method for simulating trajectories of the CME as the system evolves in time.

The SSA can be computationally intensive to run for realistic problems, and alternative methods such as the τ-leap have been developed to improve performance [10].

*Correspondence: tamas.szekely@cs.ox.ac.uk
[1]Department of Computer Science, University of Oxford, Oxford, OX1 3QD, UK
Full list of author information is available at the end of the article

Instead of simulating each reaction, the τ-leap performs several reactions at once, thus 'leaping' along the history axis of the system. This means that, unlike the SSA, the τ-leap is not exact; accuracy is maintained by not allowing too many reactions to occur per step. The size of each timestep, τ, determines the number of reactions occurring during that step, given by a Poisson random number.

This gain in speed must be balanced with loss of accuracy: larger steps mean fewer calculations but reduced accuracy. Many common τ-leap implementations employ a variable stepsize, as using the optimal stepsize τ at each point is crucial for the accuracy of the method [10-12]. However a fixed-step implementation can be useful in some cases. Although it may be less efficient, it is much easier to implement than variable-step equivalents. More importantly, the extrapolation framework that we describe in this paper requires a fixed-step method.

The original τ-leap as described by Gillespie [10] is known as the Euler τ-leap, as it can be compared to the Euler method for solving ODEs. It has been shown to have weak order of convergence one under both the scaling $\tau \to 0$ (traditional scaling) [13,14] and $V \to \infty$ (large volume scaling) [15], where V is the volume of the system. In the same paper, Gillespie also proposed the midpoint τ-leap method [10], which has higher-order convergence in some cases [15,16]. Tian and Burrage [17] proposed a variant known as the Binomial τ-leap method that avoids issues with chemical species becoming negative. Only recently has more work been done on constructing higher-order stochastic methods. One such method is the random-corrected τ-leap [18], where at each timestep a random correction is added to the Poisson random number that determines the number of reactions in that step. Given a suitable random correction, the lowest order errors on the moments can be cancelled. In this way methods with up to weak order two convergence for both mean and covariance have been constructed. More recently, Anderson and Koyama [19] and Hu *et al.* [20] proposed another weak second-order method, the θ-trapezoidal τ-leap, which is an adaptation of the stochastic differential equation (SDE) solver of Anderson and Mattingly [21] for the discrete stochastic case.

In this paper we introduce a framework for improving the order of accuracy of existing fixed-step stochastic methods by using them in conjunction with Richardson extrapolation, a well-known technique for improving the order of accuracy of a numerical solver by combining sets of simulations with different stepsizes [22]. Extrapolation was originally developed for ODE solvers but has also been successfully applied to SDE methods [23]. Our approach has three main advantages:

(1) It increases the order of accuracy of the methods supplied to it. This is desirable for the obvious reason that the resulting solutions are more accurate, as well as that larger timesteps can be used to reach a certain level of accuracy, reducing the computational cost. This is discussed further in our Conclusions.
(2) It can be applied to *any* fixed-step solver, for instance inherently higher-order methods such as the θ-trapezoidal τ-leap or methods with an extended stability region such as stochastic Runge-Kutta methods [24].
(3) The resulting higher-order solutions can be extrapolated again to give solutions with even *higher* order, as there is no (theoretical) limit on the number of times a method can be extrapolated (although statistical errors can obscure the results if the method is too accurate - see Section Monte Carlo error).

Our extrapolated methods may be useful for researchers in biology and biochemistry, as they are easy to implement and can accurately and quickly simulate discrete stochastic systems that could otherwise be too computationally intensive.

We show how the extrapolation framework can be applied to fixed-step stochastic algorithms using the examples of the fixed-step Euler τ-leap, midpoint τ-leap [10] and θ-trapezoidal τ-leap [20] methods. The extrapolation procedure depends heavily on the the existence of an appropriate global error expansion for the weak error of the numerical method. Once this is known, extrapolation consists of simple arithmetic. We calculate such an expansion for an arbitrary weak first-order method; this allows us to use extrapolation in order to obtain higher-order solutions. The weak order of all the moments of such methods can be improved by extrapolation. To reinforce this, we perform a simple error analysis by comparing the equations for the true and numerical mean of the Euler τ-leap method; we see that its global error is order one, and extrapolating it increases the order to two for the case of zeroth-order and first-order reactions. Using numerical simulations, we demonstrate that this is true for two first-order and three higher-order test systems with the Euler, midpoint and θ-trapezoidal τ-leap methods. Moreover, the extrapolated methods have consistently lower errors, and in many cases visibly higher-order convergence in the first two moments (the lack of convergence in some of the simulations is discussed in Section Monte Carlo error). Finally, we demonstrate that the extrapolation framework can be used to give even higher-order numerical solutions by applying a second extrapolation to the Euler τ-leap method.

The rest of this paper is organized as follows. We begin with an overview of the SSA and the τ-leap methods we will use later. We then discuss Richardson extrapolation for ODEs and SDEs and introduce the extrapolated discrete stochastic framework. We give numerical results to

support our claims that extrapolation reduces the error of fixed-step methods. Finally, we discuss the Monte Carlo error and give our conclusions. The derivations of the global error expansions for SDEs and discrete stochastic methods and related material are presented in the Appendix.

Overview of stochastic simulation methods

SSA

Gillespie's SSA [9] is a statistically exact method for simulating paths from a Markov jump process. The two basic assumptions of the SSA are (i) that individual molecules are not explicitly tracked, and (ii) there are frequent non-reactive collisions. Thus we assume that the system is well-mixed and homogeneous.

The SSA simulates a system of biochemical reactions with N species and M reactions, interacting inside a fixed volume V at constant temperature. The populations of chemical species (as molecular numbers, not concentrations) at time t are represented as a state vector $\mathbf{x} \equiv \mathbf{X}(t) \equiv (X_1, \ldots, X_N)^T$. Reactions are represented by a stoichiometric matrix $\boldsymbol{\nu}_j \equiv (\nu_{1j}, \ldots, \nu_{Nj})^T$, where $j = 1, \ldots, M$, composed of M individual stoichiometric vectors. Each stoichiometric vector represents a reaction j occurring and the system changing from state $\mathbf{x}$ to $\mathbf{x} + \boldsymbol{\nu}_j$. Each reaction occurs in an interval $[t, t + \tau)$ with relative probability $a_j(\mathbf{x})dt$, where a_j is the propensity function of the j-th reaction. Propensity functions are given by the mass-action kinetics of the reactant chemical species. For more detail, the reader is referred to Ref. [9]. The variables $\mathbf{X}$, $\boldsymbol{\nu}_j$ and $a_j(\mathbf{X})$ fully characterise the system at each point in time.

Algorithm 1. SSA Direct Method
With the system in state $\mathbf{X}_n$ *at time* t_n*:*

1. Generate two random numbers r_1 and r_2 from the unit-interval uniform distribution $\mathcal{U}(0, 1)$.
2. Find the time until the next reaction $\tau = \frac{1}{a_0} \ln\left(\frac{1}{r_1}\right)$, where $a_0(\mathbf{X}_n) = \sum_{j=1}^{M} a_j(\mathbf{X}_n)$.
3. Find next reaction j from $\sum_{m=1}^{j-1} a_m(\mathbf{X}_n) < a_0 r_2 \leq \sum_{m=j}^{M} a_m(\mathbf{X}_n)$.
4. Update $t_{n+1} = t_n + \tau$ and $\mathbf{X}_{n+1} = \mathbf{X}_n + \boldsymbol{\nu}_j$.

The Direct Method requires two newly-generated random numbers at each timestep. Although there are other SSA implementations, such as the Next Reaction Method [25] and the Optimised Direct Method [26], which can be more economical, in general the SSA is computationally costly.

τ-leap method

The τ-leap algorithm leaps along the history axis of the SSA by evaluating groups of reactions at once [10]. This means significantly fewer calculations, i.e. shorter computational time, per simulation, but simulation accuracy is compromised: we do not know exactly how many reactions occurred during each time step, nor can we tell more precisely when each reaction occurs than in which timestep. The *leap condition* defines an upper bound for the size of each timestep τ: it must be so small that the propensities do not change significantly for its duration, i.e. the change in state from time t to $t + \tau$ is very small [10]. Since τ is small, the probability $a(\mathbf{x})\tau$ that a reaction occurs during $[t, t + \tau)$ is also small, so the number of times K_j each reaction fires over one timestep can be approximated by $\mathcal{P}(a_j(\mathbf{x})\tau)$, a Poisson random variable with mean and variance $a_j(\mathbf{x})\tau$ [10]. The Euler τ-leap algorithm is the basic τ-leap method, and corresponds to the Euler method for solving ODEs or the Euler-Maruyama method for solving SDEs.

Algorithm 2. Euler τ-leap method
With the system in state $\mathbf{X}_n$ *at time* t_n*, and a timestep* τ*:*

1. Generate M Poisson random numbers $k_j = \mathcal{P}(a_j(\mathbf{X}_n)\tau), j = 1, \ldots, M$.
2. Update $t_{n+1} = t_n + \tau$ and $\mathbf{X}_{n+1} = \mathbf{X}_n + \sum_{j=1}^{M} \boldsymbol{\nu}_j k_j$.

The Euler τ-leap has weak order one [13-15]. Although considerable work has been done on improving the mechanism for selecting the timesteps τ [10-12] and eliminating steps that would result in negative populations [17,27-29], this does not affect the order of the method, limiting its accuracy. Methods with higher order are the only way to improve the accuracy beyond a certain point. Realising this, Gillespie also proposed a higher-order τ-leap method, the midpoint τ-leap [10]. This is similar to the midpoint method for ODEs, where at each step an estimate is made of the gradient of $\mathbf{X}$ at $t_n + \tau/2$. $\mathbf{X}_n$ is then incremented using this extra information to give a more accurate approximation.

Algorithm 3. Midpoint τ-leap method
With the system in state $\mathbf{X}_n$ *at time* t_n*, and a timestep* τ*:*

1. Calculate $\mathbf{X}' = \mathbf{X}_n + \frac{1}{2}\tau \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{X}_n)$.
2. Generate M Poisson random numbers $k_j = \mathcal{P}(a_j(\mathbf{X}')\tau), j = 1, \ldots, M$.
3. Update $t_{n+1} = t_n + \tau$ and $\mathbf{X}_{n+1} = \mathbf{X}_n + \sum_{j=1}^{M} \boldsymbol{\nu}_j k_j$.

Although under the scaling $\tau \to 0$ the midpoint τ-leap has the same order of accuracy in the mean as the Euler τ-leap method, under the large volume scaling it has weak order two [15,16]. Our numerical simulations also suggest that it gives higher-order approximations to the first two moments for both linear and non-linear systems (although this is not clear from the literature). However the local truncation error of its covariance is first-order [16].

θ-trapezoidal τ-leap method

Based on the SDE method of Anderson and Mattingly [21], the θ-trapezoidal τ-leap [20] is a weak second-order method. It consists of two steps, a predictor step with size $\theta\tau$ and a corrector step with size $(1-\theta)\tau$ that aims to cancel any errors made in the first step.

Algorithm 4. θ-trapezoidal τ-leap method

For a specified $\theta \in (0,1)$, $\alpha_1 = \frac{1}{2(1-\theta)\theta}, \alpha_2 = \frac{(1-\theta)^2+\theta^2}{2(1-\theta)\theta}$. *With the system in state* $\mathbf{X}_n$ *at time* t_n, *and a timestep* τ:

1. Generate M Poisson random numbers $k'_j = \mathcal{P}(a_j(\mathbf{X}_n)\theta\tau), j = 1,\ldots,M$.
2. Calculate predictor step $\mathbf{X}' = \mathbf{X}_n + \sum_{j=1}^{M} \boldsymbol{\nu}_j k'_j$.
3. Calculate $l_j = \max\left(\alpha_1 a_j(\mathbf{X}') - \alpha_2 a_j(\mathbf{X}_n), 0\right)$.
4. Generate M Poisson random numbers $k_j = \mathcal{P}(l_j(1-\theta)\tau), j = 1,\ldots,M$.
5. Update $t_{n+1} = t_n + \tau$ and $\mathbf{X}_{n+1} = \mathbf{X}' + \sum_{j=1}^{M} \boldsymbol{\nu}_j k_j$.

Specifically, the θ-trapezoidal τ-leap method was shown to have weak order of convergence two in the moments, and a local truncation error of $\mathcal{O}(\tau^3 V^{-1})$ for the covariance. $\tau = V^{-\beta}$, $0 < \beta < 1$ and in the analysis $V \to \infty$, but in simulations the system volume is kept constant; thus it seems that in practice this also results in weak second-order convergence in the covariance [20].

Methods

The extrapolation framework

We start with an ODE solver with stepsize h approximating the true solution $\mathbf{x}(T)$ by $\mathbf{x}_n^h$ (where $T = nh$), for which we assume the following global error expansion exists:

$$\mathbf{x}(T) - \mathbf{x}_n^h = \mathbf{e}_k(T)h^k + \mathbf{e}_{k+1}(T)h^{k+1} + \mathbf{e}_{k+2}(T)h^{k+2} + \ldots, \tag{1}$$

where k is the order of the numerical method and the $\mathbf{e}_k(T)$ are constants that only depend on the final integration time T. Extrapolating has the effect of cancelling the leading error term, resulting in a more accurate approximation. The existence of such an expansion is key to constructing a higher-order approximation, as the appropriate extrapolation coefficients must be used for the leading error terms to cancel. For example, in the case of a first-order method with stepsize h,

$$\mathbf{x}(T) - \mathbf{x}_n^h = \mathbf{e}_1(T)h + \mathbf{e}_2(T)h^2 + \mathcal{O}(h^3), \tag{2}$$

and similarly for stepsize $\frac{h}{2}$,

$$\mathbf{x}(T) - \mathbf{x}_{2n}^{h/2} = \mathbf{e}_1(T)\frac{h}{2} + \mathbf{e}_2(T)\frac{h^2}{4} + \mathcal{O}(h^3). \tag{3}$$

Setting $\hat{\mathbf{x}}_n^h = 2\mathbf{x}_{2n}^{h/2} - \mathbf{x}_n^h$ and using (3) and (2), we obtain

$$\mathbf{x}(T) - \hat{\mathbf{x}}_n^h = -\frac{h^2}{2}e_2(T) + \mathcal{O}(h^3), \tag{4}$$

which implies that $\hat{\mathbf{x}}_n^h$ is now a second-order approximation to $\mathbf{x}(T)$.

We define $z(k,q) = \mathbf{x}_n^{h_q}$ to be a series of approximations with order k and stepsize h_q to the true solution $\mathbf{x}(T)$, where $T = nh_1$, $h_q = T/p_q$ and $h_1 > h_2 > \ldots > h_q$. In general, one can use an order k method with stepsizes h_{q-1} and h_q (in the previous example, $h_{q-1} = h$ and $h_q = \frac{h}{2}$, i.e. $p = 2$), to arrive at an order $k+1$ estimate to $\mathbf{x}(T)$,

$$\mathbf{x}(T) - z(k, h_{q-1}) = \mathcal{O}(h^{k+1}),$$

where

$$z(k, h_{q-1}) = \frac{p^k\, z(k, q-1) - z(k,q)}{p^k - 1}.$$

This process can be repeated indefinitely. We can extrapolate from the initial approximations $z(1,1),\ldots,z(1,q)$ by combining the successive solutions in each column of the Romberg table:

z(1,1)			
	z(2,1)		
z(1,2)		z(3,1)	
	z(2,2)	⋮	z(q,q)
z(1,3)	⋮	z(3,q)	
⋮	z(2,q)		
z(1,q)			

For instance, in Eq. (4) we used (with $p = 2$) $\mathbf{x}_n^h = z(1,1)$ and $\mathbf{x}_{2n}^{h/2} = z(1,2)$ to find $\hat{\mathbf{x}}_n^h = z(2,1)$. Repeating with $\mathbf{x}_{2n}^{h/2}$ and $\mathbf{x}_{2n}^{h/4}$, we could extrapolate to find $\hat{\mathbf{x}}_{2n}^{h/2} = z(2,2)$. Then we could extrapolate $\hat{\mathbf{x}}_n^h$ and $\hat{\mathbf{x}}_{2n}^{h/2}$ to find a third-order approximation $\hat{\hat{\mathbf{x}}}_n^h = z(3,1)$, and so on.

Stochastic methods have two error measures: strong (comparing trajectories) and weak (comparing moments). In the context of extrapolation, we are interested in the global weak error, defined as

$$|\mathbb{E}(f(\mathbf{x}(T))) - \mathbb{E}(f(\mathbf{x}_n^h))|, \tag{5}$$

where $f : \mathbb{R}^N \mapsto \mathbb{R}$ is a suitable smooth functional, for example the first moment of one of the components of $\mathbf{x}$. $\mathbf{x}_n^h$ is a numerical approximation to the SDE

$$d\mathbf{X}_t = a(\mathbf{X}_t, t)dt + b(\mathbf{X}_t, t)dW_t, \tag{6}$$

where $a(\mathbf{x},t) : \mathbb{R}^{N+1} \mapsto \mathbb{R}^N$, $b(\mathbf{x},t) : \mathbb{R}^{N+1\times M} \mapsto \mathbb{R}^{N\times M}$ and W_t is a standard M-dimensional Wiener increment. Talay and Tubaro [23] derived a similar expansion to Eq. (1) for the global error when $\mathbf{x}_n^h$ was calculated using the Euler-Maruyama and Milstein schemes (outlined in

Appendix A). By using this expansion and the extrapolation framework, they were able to derive a second-order approximation to $\mathbb{E}(f(\mathbf{x}(T)))$. The crucial step in obtaining the global error expansion was to express it as a telescopic sum of the local errors. Liu and Li [30] also followed a similar procedure to derive a global error expansion for numerical methods for SDEs with Poisson jumps, thus allowing them to obtain higher-order weak approximations.

Extrapolation for discrete chemical kinetics

The extrapolation framework can be extended to the discrete stochastic regime. Since it requires two or more sets of approximations with given stepsizes (e.g. h and $h/2$), *it can only be used with a fixed-step method*: as more complex τ-leap methods vary τ at each step, it is not clear how to extrapolate them. However, this has the advantage of making our method very easy to program, as there is no complex programming overhead, for instance in choosing the timestep for τ-leap methods. We stress that we mostly use extrapolation to obtain higher-order approximations to the *moments* of the system (or their combinations, such as the covariance). In principle, given enough of the moments, the full probability distribution at some given time could be constructed. This is known as the Hamburger moment problem [31] and in general is a difficult problem to solve, as it might admit an infinite number of solutions. However, in some cases it is possible to reconstruct the full distribution from the extrapolated moments, as we have *a priori* knowledge about its shape. For instance, when the final distribution of states is known to be binomial, only the mean and variance are necessary for constructing the full extrapolated distribution (see Numerical Results, System 1).

In this section we focus on the Euler τ-leap method (ETL), since this choice simplifies the analysis, but we show in Appendix B that *any* fixed-stepsize method with known weak order can be extrapolated. In our numerical investigations we show results for the ETL, the midpoint τ-leap (MPTL) and the θ-trapezoidal τ-leap (TTTL) method. Extrapolating the ETL is very similar to extrapolating an ODE solver. The extrapolated ETL, which we call xETL from here on, involves running two sets of S ETL simulations for time $T = n\tau$.

Algorithm 5. Extrapolated Euler τ-leap method (xETL)

1. Run S ETL simulations with stepsize τ, to get ${}_s\mathbf{x}_n^\tau, s = 1, \ldots, S$.
2. Calculate desired moments $\mathbb{E}_S(f(\mathbf{x}_n^\tau)) = \frac{1}{S}\sum_{s=1}^{S} f({}_s\mathbf{x}_n^\tau)$.
3. Repeat steps 1 and 2 using stepsize $\tau/2$ to get $\mathbb{E}_S(f(\mathbf{x}_{2n}^{\tau/2}))$.
4. Take $\left(2\mathbb{E}_S(f(\mathbf{x}_{2n}^{\tau/2})) - \mathbb{E}_S(f(\mathbf{x}_n^\tau))\right)$ as the extrapolated approximation to the desired moment.

Algorithm 5 can be easily modified for use with any other fixed-step method, by replacing the ETL in Step 1 with the chosen method.

It is instructive to use a simple example to see analytically the effects of extrapolating the ETL. When the propensity functions are linear (i.e. the system only contains zeroth-order and first-order reactions), the equations for the moments are closed [32,33] and we can find explicitly the global error expansion for the first moment of our numerical solution (i.e. choose $f(\mathbf{x}) = \mathbf{x}$). The propensity functions can be written as

$$[\,a_1(\mathbf{x}), a_2(\mathbf{x}), \ldots, a_M(\mathbf{x})]^T = C\mathbf{x} + \mathbf{d}, \tag{7}$$

where $C \in \mathbb{R}^{M\times N}$, $\nu \in \mathbb{R}^{N\times M}$ and $\mathbf{d} \in \mathbb{R}^M$, and we define $W = \nu C$, i.e. $W \in \mathbb{R}^{N\times N}$. Thus at some timestep m, $m\tau < T$, the ETL gives (using matrix notation)

$$\mathbf{x}_{m+1}^\tau = \mathbf{x}_m^\tau + \nu\mathcal{P}(\tau(C\mathbf{x}_m^\tau + \mathbf{d})),$$

where $\mathbf{x}_m^\tau$ is an approximation to the true state vector $\mathbf{x}(t)$ at the m-th timestep (i.e. time t) with stepsize τ. Taking the expectation of both sides, the ETL evolves the mean as

$$\begin{aligned}\mathbb{E}(\mathbf{x}_{m+1}^\tau) &= \mathbb{E}(\mathbf{x}_m^\tau) + \nu\mathbb{E}(\mathcal{P}(\tau(C\mathbf{x}_m^\tau + \mathbf{d})))\\ &= \mathbb{E}(\mathbf{x}_m^\tau) + \nu\mathbb{E}(\mathbb{E}(\mathcal{P}(\tau(C\mathbf{x}_m^\tau + \mathbf{d}))|\mathbf{x}_m^\tau)) \\ &= (I + \tau W)\mathbb{E}(\mathbf{x}_m^\tau) + \tau\nu\mathbf{d},\end{aligned} \tag{8}$$

where I is the $N \times N$ identity matrix. Note that we cannot evaluate the expectation of $\mathcal{P}(\tau(C\mathbf{x}_m^\tau + \mathbf{d}))$ directly: because $\mathbf{x}_m^\tau$ is a random variable, we do not know the distribution of $\mathcal{P}(\mathbf{x}_m^\tau)$. Using the law of total expectation we can condition on $\mathbf{x}_m^\tau$ taking a specific value; $\mathcal{P}(\mathbf{x}_m^\tau)|\mathbf{x}_m^\tau$ does have a Poisson distribution. From (8), we see that at time $T = n\tau$,

$$\begin{aligned}\mathbb{E}(\mathbf{x}_n^\tau) = &\left(I + \tau nW + \binom{n}{2}\tau^2W^2 + \ldots\right)\boldsymbol{\mu}(0)\\ &+ W^{-1}(\tau nW + \frac{1}{2}\tau^2n^2W^2 + \ldots)\nu\mathbf{d}.\end{aligned} \tag{9}$$

The probability density function of the SSA at time t is given by the CME [7,8]. The mean $\boldsymbol{\mu}(t) = \mathbb{E}(\mathbf{x}(t))$ can be found from the CME [34]; it evolves as

$$\frac{d\boldsymbol{\mu}(t)}{dt} = W\boldsymbol{\mu}(t) + \nu\mathbf{d}.$$

The solution of this is

$$\boldsymbol{\mu}(t) = e^{Wt}\boldsymbol{\mu}(0) + e^{Wt}\int_0^t e^{-Ws}\nu\mathbf{d}\,ds. \tag{10}$$

Using a Taylor expansion and basic manipulation, at $t = T$ this evaluates to

$$\boldsymbol{\mu}(T) = \left(I + TW + \frac{1}{2}T^2W^2 + \dots\right)\boldsymbol{\mu}(0) + W^{-1}\left(TW + \frac{1}{2}T^2W^2 + \dots\right)\nu\mathbf{d}. \quad (11)$$

Taking (11) minus (9) we see that the global error is

$$\boldsymbol{\mu}(T) - \mathbb{E}(\mathbf{x}_n^\tau) = \left(\frac{1}{2}\tau TW^2 + \mathcal{O}(\tau^2)\right)\boldsymbol{\mu}(0) + \left(\frac{1}{2}\tau TW + \mathcal{O}(\tau^2)\right)\nu\mathbf{d}. \quad (12)$$

Furthermore the extrapolated error is

$$\boldsymbol{\mu}(T) - \left(2\mathbb{E}(\mathbf{x}_{2n}^{\tau/2}) - \mathbb{E}(\mathbf{x}_n^\tau)\right) = \left(\frac{1}{6}\tau^2TW^3 + \mathcal{O}(\tau^3)\right)\boldsymbol{\mu}(0) + \left(\frac{1}{6}\tau^2TW^2 + \mathcal{O}^3\right)\nu\mathbf{d},$$

so the leading error term has been cancelled, leaving an order two approximation. Such a calculation would also apply for the MPTL. The difference is that for linear systems the MPTL is second-order convergent with respect to the mean [16], and similarly for the TTTL [20]. This should be taken into account in order to choose the correct extrapolation coefficients.

The above analysis only applies for the mean of a linear system, a very restricted case, but it is useful for demonstrating the basic principles of stochastic extrapolation. We employ a similar approach to Talay and Tubaro [23] and Liu and Li [30] to find a general expression for the global error expansion of the moments of a weak first-order discrete stochastic method; this is Appendix B. In Appendix C we explicitly evaluate this for the particle decay system and show that it is equivalent to Eq. [12] in this case. Appendix D contains the equations for the second moment for the case of linear systems.

Multimodal systems

As discussed before, one limitation of our approach is that only specific characteristics of the particle distribution can be extrapolated, rather than the full distribution. Typically we choose these to be the first and second moments, as for many systems these are the quantities of interest. However, in some cases the moments do not take values relevant to the actual dynamics of the system [35,36]. This occurs, for instance, in bimodal or multimodal systems, which have two or more stable states. Nevertheless, our method can be easily generalised to accommodate multimodal distributions as follows.

Algorithm 6. Extrapolated Euler τ-leap method (xETL) for multimodal systems

1. Run S ETL simulations with stepsize τ, to get ${}_s\mathbf{x}_n^\tau, s = 1, \dots, S$.
2. Plot histograms of the particle populations at time T and identify the stable states.
3. Choose point(s) at which to partition the S simulations into p subsets of $S_1, \dots, S_p$ simulations clustered around each stable state.
4. Calculate desired moments over the subsets of simulations, $\mathbb{E}_{S_i}(f(\mathbf{x}_n^\tau)) = \frac{1}{S_i}\sum_{s=1}^{S_i} f({}_s\mathbf{x}_n^\tau), i = 1, \dots, p$.
5. Repeat steps 1 and 4 using stepsize $\tau/2$ to get $\mathbb{E}_{S_i}(f(\mathbf{x}_{2n}^{\tau/2})), i = 1, \dots, p$.
6. Take $\left(2\mathbb{E}_{S_i}(f(\mathbf{x}_{2n}^{\tau/2})) - \mathbb{E}_{S_i}(f(\mathbf{x}_n^\tau))\right), i = 1, \dots, p$ as the extrapolated approximation to the desired moment for each of the p subsets of simulations.

Algorithm 6 is also simple to code and does not require significant extra computational time compared to Algorithm 5 because the dynamics of the system are found from the original simulations that are necessary for the extrapolation anyway. The point(s) at which the simulations are split into subsets can affect the accuracy of the results, so must be chosen with some care. In the Numerical Results (System 5), we apply Algorithm 6 to a bimodal system, and investigate the effects of the choice of splitting point.

Results and discussion

Numerical results

We simulate some example systems for various stepsizes τ over time $t = [0, T]$ using three fixed-step numerical methods: the Euler τ-leap (ETL), midpoint τ-leap (MPTL) and θ-trapezoidal τ-leap (TTTL) methods (with $\theta = 0.55$), and their extrapolated versions, the xETL, xMPTL and xTTTL. We plot the absolute weak errors in the mean and second moment, i.e.

$$|\mathbb{E}(\mathbf{x}(T)) - \mathbb{E}_S(\mathbf{x}_n^\tau)|, \quad |\mathbb{E}(\mathbf{x}(T)\mathbf{x}^T(T)) - \mathbb{E}_S(\mathbf{x}_n^\tau(\mathbf{x}_n^\tau)^T)| \quad (13a)$$

for the ETL, MPTL and TTTL methods and

$$|\mathbb{E}(\mathbf{x}(T)) - 2\mathbb{E}_S(\mathbf{x}_{2n}^{\tau/2}) + \mathbb{E}_S(\mathbf{x}_n^\tau)|, \quad |\mathbb{E}(\mathbf{x}(T)\mathbf{x}^T(T)) - 2\mathbb{E}_S(\mathbf{x}_{2n}^{\tau/2}(\mathbf{x}_{2n}^{\tau/2})^T) + \mathbb{E}_S(\mathbf{x}_n^\tau(\mathbf{x}_n^\tau)^T)| \quad (13b)$$

for the extrapolated methods. Here $\mathbf{x}(T)$ is the analytical solution at time T and $\mathbb{E}_S(f(\mathbf{x}_n^\tau))$ are the moments of its approximations given by S simulations of a fixed-step method with stepsize τ run for n steps. For the linear systems, the true solution is calculated analytically; for the

non-linear systems we use the value given by 10^6 or 10^7 repeats of the SSA (depending on the system). The error of a weak order α method with stepsize τ is approximately $C\tau^\alpha$, where C is an unknown constant. To easily see the order of accuracy of our results, we plot all the errors on log-log plots. Gradients are calculated using a least squares fit through the points. The highest level of Monte Carlo error, which can be calculated for the linear systems, is marked on the appropriate plots as a straight black line. Below this level, the absolute error results are, at least in part, essentially random (see Section Monte Carlo error). We note that in all test systems, the timesteps used were all in the useful τ-leaping regime: Poisson counts for each reaction channel varied between tens to hundreds.

System 1: Particle decay system

A simple test system is a particle decay,

$$X \xrightarrow{k} \emptyset, \quad k = 0.1.$$

The initial particle number was $X(0) = 10^4$ and the simulation time was $T = 10.4$. The units here, and in the systems below, are non-dimensional. This system is useful only as a test problem, but is first-order and easily tractable. The analytical mean and second moment are

$$\mathbb{E}(X(T)) = X(0)e^{-kT},$$

$$\mathbb{E}(X(T)^2) = X(0)e^{-kT} - X(0)e^{-2kT} + (X(0))^2 e^{-2kT}.$$

The average final particle numbers, calculated as above, were $\mathbb{E}(X(10.4)) = 3534.5$. We ran 10^8 simulations using timesteps $\tau = 0.05, \ldots, 0.8$. The errors on the mean and second moment are shown in Figure 1. In both cases, the ETL gives first-order errors and the xETL gives approximately second-order errors. The MPTL and TTTL also converge with second order. The errors of the xMPTL and xTTTL are very small, although they do not converge with any noticeable order. This is because the values of the absolute error for these methods are effectively given by their Monte Carlo error, rather than the bias (this is a recurring theme in stochastic simulations - see Section Monte Carlo error). The maximum level of Monte Carlo error was 0.0148 for the mean and 104.8 for the second moment.

In addition, because the final distribution of this system is known to be binomial [10], we can construct the distribution of the extrapolated solutions from just the mean and variance (Figure 2). The dashed lines are the distributions of ETL simulations with $\tau = 0.05$ (blue) and $\tau = 0.8$ (red), calculated from their histograms, and the circles are the full distributions of the xETL using $\tau = 0.05$ (blue) and $\tau = 0.8$ (red). The solution of the CME (black line) is the true distribution. Both the extrapolated solutions match the CME solution very well, in both mean and overall shape.

System 2: Chain decay system

This system consists of five chemical species, each undergoing a first-order reaction. It forms a closed chain of linear reactions, and is intended as a more complicated, but still linear, example system.

$$X_1 \xrightarrow{k_1} X_2 \xrightarrow{k_2} X_3 \xrightarrow{k_3} X_4 \xrightarrow{k_4} X_5 \xrightarrow{k_5} X_1, \quad k_j = 0.3,\ j = 1, \ldots, 5.$$

The initial populations were $\mathbf{X}(0) = (2500, 1875, 1875, 1875, 1875)^T$ and simulation time was $T = 16$. Since this system is linear, its expectation and covariance can be calculated analytically, as shown in Jahnke and Huisinga [32]; we used these to calculate the true second moment. The average final particle numbers, given by the analytical mean, were $\mathbb{E}(\mathbf{X}(16)) = (1971.3, 1996.7, 2025.1, 2020.9, 1986.1)^T$. We ran 10^8 simulations with $\tau = 0.1, \ldots, 1.6$. Figure 3 shows the absolute errors for the mean and second moment of X_1. The ETL

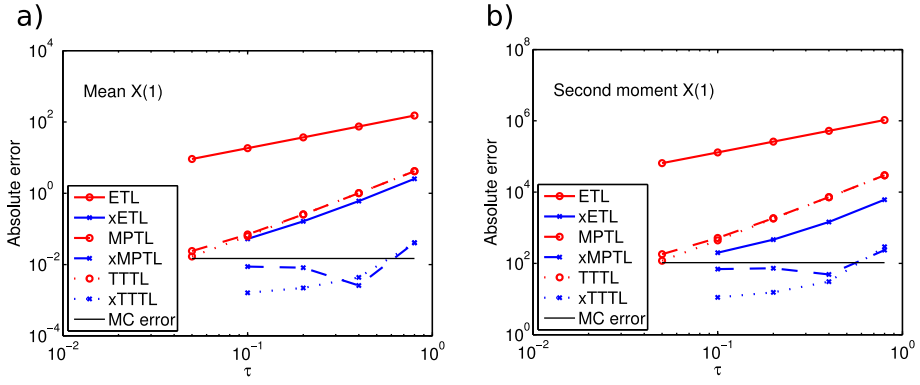

Figure 1 System 1 absolute errors. Absolute error (13) in **(a)** the mean for the ETL (gradient 1.0) and xETL (gradient 1.9), MPTL (gradient 1.9) and xMPTL (gradient 0.5), TTTL (gradient 2.0) and xTTTL (gradient 1.5); **(b)** the second moment for ETL (gradient 1.0) and xETL (gradient 1.6), MPTL (gradient 1.9) and xMPTL (gradient 0.5), TTTL (gradient 2.0) and xTTTL (gradient 1.5). The maximum Monte Carlo error levels (black straight lines) were 0.0148 (mean) and 104.8 (second moment). Results from 10^8 simulations, each run for $T = 10.4$.

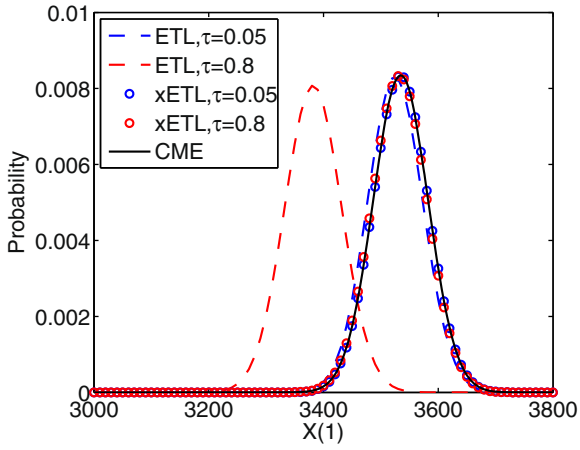

Figure 2 Constructing full extrapolated distributions of System 1. Distribution of states at time $T = 10.4$ for ETL with $\tau = 0.05$ (blue dashes), ETL with $\tau = 0.8$ (red dashes), xETL with $\tau = 0.05$ (red circles), xETL with $\tau = 0.8$ (blue circles). The analytical solution is given by the CME (black line). The extrapolated distributions match the CME solution very closely.

is again approximately weak order one and the xETL weak order two. Again, the errors for the MPTL, TTTL, xMPTL and xTTTL are very low, although again their order of convergence is not quite as high as expected. We believe this is due to the unusually high accuracy of these methods for closed systems compared to a maximum Monte Carlo error level of 0.0132 (mean) and 52.8 (second moment), which are high relative to the bias of these methods.

System 3: Michaelis-Menten system

The Michaelis-Menten is a common non-linear test system, and represents an enzyme (X_2) reacting with a substrate (X_1) to make a product (X_4). The enzyme and substrate form a complex (X_3), which can either dissociate or undergo a reaction to a product plus the original enzyme. It has four chemical species in three reactions:

$$\begin{aligned} X_1 + X_2 &\xrightarrow{k_1} X_3, & k_1 &= 10^{-5}, \\ X_3 &\xrightarrow{k_2} X_1 + X_2, & k_2 &= 0.2, \\ X_3 &\xrightarrow{k_3} X_2 + X_4, & k_3 &= 0.2. \end{aligned}$$

Simulation time was $T = 16$ and the initial populations were $\mathbf{X}(0) = (10^4, 2 \times 10^3, 2 \times 10^4, 0)^T$. We used 10^8 simulations with $\tau = 0.1, \ldots, 1.6$. There is no analytical solution, so in this case we approximated it with 10^7 SSA simulations. The average final state, given by the SSA, was $\mathbb{E}(\mathbf{X}(16)) = (5927.0, 18716.2, 3283.8, 20789.2)^T$.

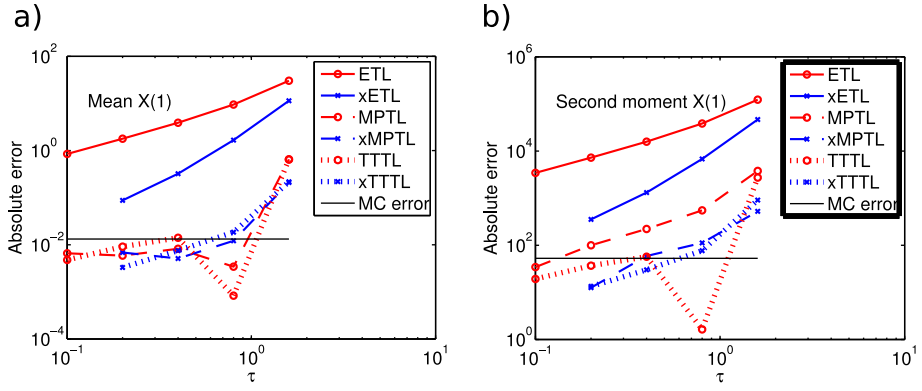

Figure 3 System 2 absolute errors. Absolute error (13) of X_1 in **(a)** the mean for ETL (gradient 1.3) and xETL (gradient 2.4), MPTL (gradient 1.2) and xMPTL (gradient 1.6), TTTL (gradient 1.1) and xTTTL (gradient 1.9); **(b)** the second moment for ETL (gradient 1.3) and xETL (gradient 2.4), MPTL (gradient 1.6) and xMPTL (gradient 1.7), TTTL (gradient 1.0) and xTTTL (gradient 2.0). The maximum Monte Carlo error levels (black straight lines) were 0.0132 (mean) and 52.8 (second moment). Results from 10^8 simulations, each run for $T = 16$.

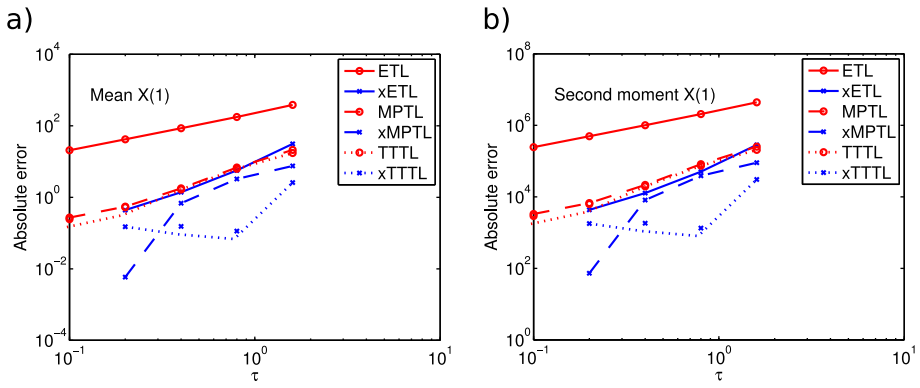

Figure 4 System 3 absolute errors. Absolute error (13) of X_1 in **(a)** the mean for ETL (gradient 1.0) and xETL (gradient 2.0), MPTL (gradient 1.6) and xMPTL (gradient 3.3), TTTL (gradient 1.6) and xTTTL (gradient 1.2); **(b)** the second moment for ETL (gradient 1.0) and xETL (gradient 2.0), MPTL (gradient 1.6) and xMPTL (gradient 3.3), TTTL (gradient 1.6) and xTTTL (gradient 1.2). Results from 10^8 simulations, each run for $T = 16$.

The errors in the mean and second moment for X_1 are shown in Figure 4. The ETL converges with order one for 10^8 simulations, and the xETL with order two. The MPTL and TTTL have a similar accuracy to the xETL, with approximate order two, with the xMPTL and xTTTL of approximate order three.

We investigated the effects of the coefficient of variation (CV, standard deviation divided by mean) on this system. The CVs of each species, averaged across all τ, in System 3 at $T = 16$ were $\mathrm{CV}(\mathbf{X}(16)) = (0.01, 0.003, 0.02, 0.004)^T$. In general, a higher CV indicates that the system is more noisy. We chose a new set of parameters for System 3 to give higher CVs: $\mathbf{X}_{\mathrm{new}}(0) = (100, 50, 200, 0)^T, \boldsymbol{k}_{\mathrm{new}} = (10^{-4}, 0.05, 0.07)^T$. The CVs using these parameters were $\mathrm{CV}(\mathbf{X}_{\mathrm{new}}(16)) = (0.05, 0.03, 0.1, 0.07)^T$, very different from the original CVs. However the relative errors (absolute error divided by average SSA state) at $\tau = 0.1$ were very similar: $\mathrm{error}_{\mathrm{rel}}(\mathbf{X}(16)) = (0.004, 0.0005, 0.003, 0.001)^T$ for the original system and $\mathrm{error}_{\mathrm{rel}}(\mathbf{X}_{\mathrm{new}}(16)) = (0.001, 0.0002, 0.0007, 0.002)^T$ with the new parameters (note that it is not useful to average the errors across all τ). This shows that higher CV does not necessarily mean higher errors, and the two are indicators of different characteristics of the system. We have focused on the errors as this is the characteristic that we want to improve.

System 4: Two-enzyme mutual inhibition system

This is a more realistic system involving 8 chemical species and 12 reactions [37]. It represents two enzymes, E_A and E_B, which catalyse the production of compounds A and B, respectively. In a classic example of double-negative feedback, each product inhibits the activity of the other enzyme. For this reason, the system is bistable in A and B:

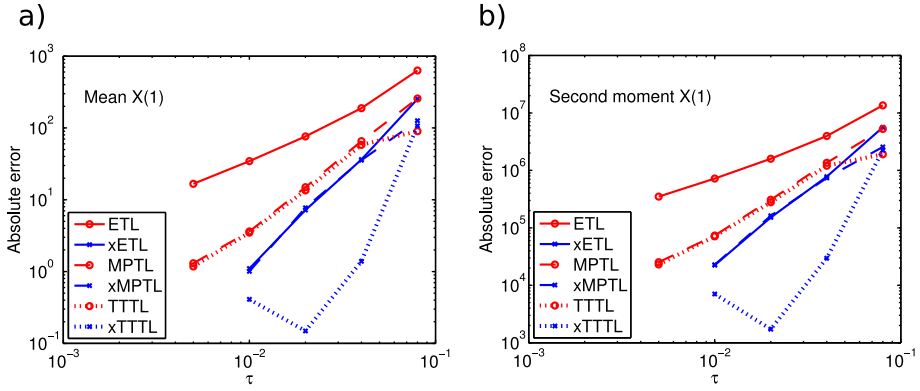

Figure 5 System 4 absolute errors. Absolute error (13) of X_1 in **(a)** the mean for ETL (gradient 1.3) and xETL (gradient 2.6), MPTL (gradient 1.9) and xMPTL (gradient 2.3), TTTL (gradient 1.7) and xTTTL (gradient 2.7); **(b)** the second moment for ETL (gradient 1.3) and xETL (gradient 2.6), MPTL (gradient 2.0) and xMPTL (gradient 2.3), TTTL (gradient 1.7) and xTTTL (gradient 2.9). Results from 10^7 simulations, each run for $T = 3.2$.

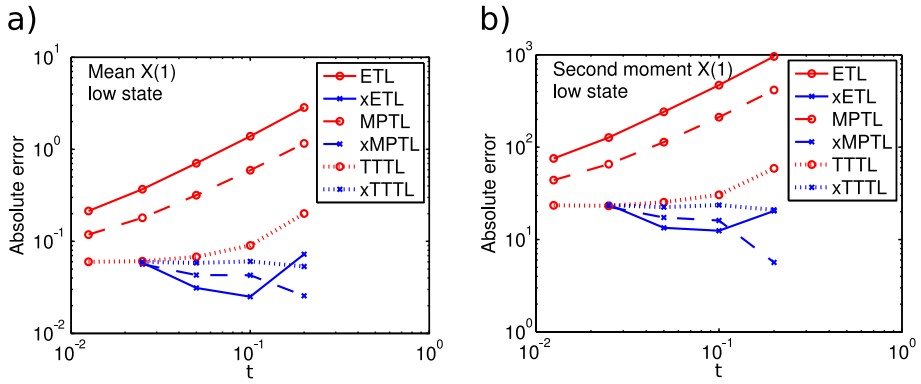

Figure 6 System 5 low peak absolute errors. Absolute error (13) of X_1 in **(a)** the mean for ETL (gradient 0.9) and xETL (gradient 0.06), MPTL (gradient 0.8) and xMPTL (gradient -0.3), TTTL (gradient 0.4) and xTTTL (gradient 0.0); **(b)** the second moment for ETL (gradient 0.9) and xETL (gradient -0.1), MPTL (gradient 0.8) and xMPTL (gradient -0.6), TTTL (gradient 0.3) and xTTTL (gradient 0.0). Results from 10^8 simulations, each run for $T = 5$.

when there are many particles of A, few particles of B can be produced, and vice versa. The reactions are

$$
\begin{aligned}
E_A &\xrightarrow{k_1} E_A + A, && k_1 = 15,\\
E_B &\xrightarrow{k_2} E_B + B, && k_2 = 15,\\
E_A + B &\underset{k_4}{\overset{k_3}{\rightleftharpoons}} E_AB, && k_3 = 5\times 10^{-4}, k_4 = 2,\\
E_AB + B &\underset{k_6}{\overset{k_5}{\rightleftharpoons}} E_AB_2, && k_5 = 10^{-3}, k_6 = 6,\\
A &\xrightarrow{k_7} \emptyset, && k_7 = 5,\\
E_B + A &\underset{k_9}{\overset{k_8}{\rightleftharpoons}} E_BA, && k_8 = 5\times 10^{-4}, k_9 = 2,\\
E_BA + A &\underset{k_{11}}{\overset{k_{10}}{\rightleftharpoons}} E_BA_2, && k_{10} = 10^{-3}, k_{11} = 6,\\
B &\xrightarrow{k_{12}} \emptyset, && k_{12} = 5.
\end{aligned}
$$

We simulated this system for $T = 3.2$ using initial populations of

$$\mathbf{X}(0) = (2\times 10^4, 1.5\times 10^4, 9500, 9500, 2000, 500, 2000, 500)^T,$$

where $\mathbf{X} = (A, B, E_A, E_B, E_AB, E_AB_2, E_BA, E_BA_2)^T$. We ran 10^7 simulations of the τ-leap using $\tau = 0.005, \ldots, 0.08$. 10^6 SSA simulations were used as an approximation to the analytical values. The final state of the system as given by the SSA mean was $\mathbb{E}(\mathbf{X}(3.2)) = (10420.5, 4884.4, 3594.7, 1528.0, 4592.7, 3812.6, 3853.8, 6618.2)^T$. The errors for X_1 in this system are shown in Figure 5. The ETL and xETL gave errors of approximately order one and two, respectively. The MPTL and TTTL were again approximate order two; the xMPTL had errors very similar to those of the xETL, and the xTTTL had very low errors of approximate order three.

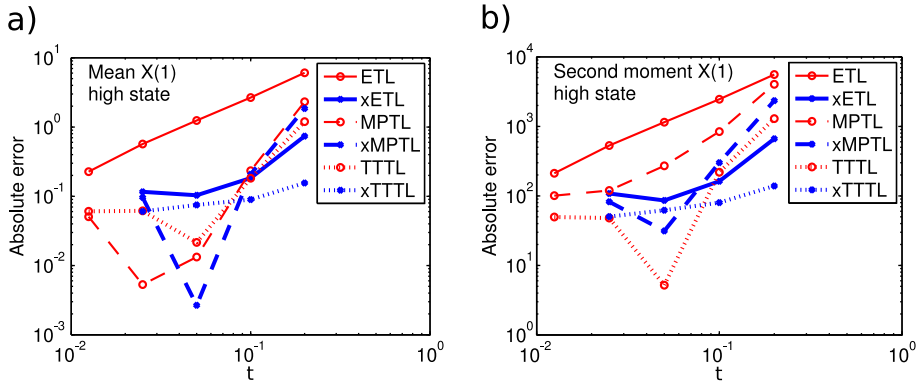

Figure 7 System 5 high peak absolute errors. Absolute error (13) of X_1 in **(a)** the mean for ETL (gradient 1.2) and xETL (gradient 0.9), MPTL (gradient 1.6) and xMPTL (gradient 1.9), TTTL (gradient 1.0) and xTTTL (gradient 0.4); **(b)** the second moment for ETL (gradient 1.2) and xETL (gradient 0.9), MPTL (gradient 1.3) and xMPTL (gradient 1.8), TTTL (gradient 1.2) and xTTTL (gradient 0.5). Results from 10^8 simulations, each run for $T = 5$.

System 5: Schlögl system

The last example we give illustrates how our method could work for systems that have multimodal distributions. The Schlögl system consists of four reactions [38] and is bimodal:

$$\begin{aligned} A + 2X &\xrightarrow{k_1} 3X, & k_1 &= 3 \times 10^{-7}, \\ 3X &\xrightarrow{k_2} A + 2X, & k_2 &= 10^{-4}, \\ B &\xrightarrow{k_3} X, & k_3 &= 10^{-3}, \\ X &\xrightarrow{k_4} B, & k_4 &= 3.5, \end{aligned}$$

where the populations of species A and B are held constant at 10^5 and 2×10^5 respectively, so only the numbers of X can change. This is a common benchmark system for computational simulation algorithms. Under certain parameter configurations, this system has two stable states for X, one high and one low. When $X(0)$ is low, the system usually settles in the lower equilibrium, and vice versa for high $X(0)$. We used $X(0) = 250$, an intermediate value that gives a bimodal distribution. We ran 10^8 simulations until $T = 5$ using $\tau = 0.0125, \ldots, 0.2$. Because of the bimodality, we separated the data into two sets. As the Schlögl system is simple enough to be solved using a CME solver (see e.g. [39]), we could calculate the density of states of X at $T = 5$. When simulating more complicated systems, this can be approximated by a histogram of the distribution of species at T (from the initial ETL simulations, for instance) to get a reasonable idea of its shape. Average final particle number was calculated from the CME to be $\mathbb{E}(X(5)) = 101.3$ for the low peak and $\mathbb{E}(X(5)) = 546.2$ for the high peak. Figures 6 and 7 show the absolute errors for this system (low and high peaks, respectively). The error levels seem to be similar in both cases. The general trend was for the ETL, MPTL and TTTL to converge with approximate order one, whereas the extrapolated solutions did not have a clear order of convergence. It is clear that in this case also, the Monte Carlo error is interfering with our ability to see a clear order of convergence.

The gradient of the MPTL mean error seems to change from approximately one (Figure 6) to around 1.5 (Figure 7). It is unlikely that this is due to Monte Carlo error, as the error of the MPTL is high enough that this should not be an issue. In fact, this is probably due to the large volume limit behaviour of the MPTL, discussed after Algorithm 3. Because the mean of the high peak is several times higher than the mean of the low peak, the system is closer to the large volume limit and the weak order of the MPTL increases accordingly. Once in the large volume limit, the gradient is expected to be two.

The point at which the data is separated must be chosen carefully, as it can influence the error results. Figure 8 shows the distribution of X at $T = 5$ calculated using a CME solver. The three choices of splitting points that we compare have also been marked. We chose $X = 300$; this is a fairly obvious splitting point in our case. To support this, we tested the effects of splitting the data given by the ETL at $X = 250$ and $X = 350$. We calculated the relative change in mean, second moment and variance between these two splitting values, averaged over all simulations of the ETL and all five timesteps: Table 1 shows that in this case the percentage difference is relatively small for the mean, becomes larger for the second moment, and is very high for the variance. This implies that the choice of splitting point is very important in this case, and should be carefully considered.

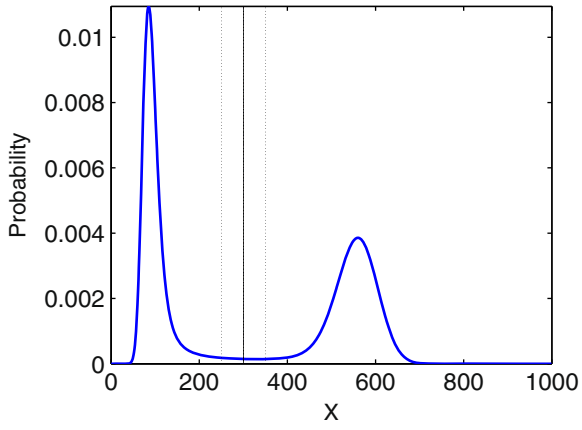

Figure 8 System 5 full distribution, $T = 5$. The distribution of System 5 calculated using a CME solver for $T = 5$. The three splitting points we have compared in this paper are illustrated.

Higher extrapolation

In principle it is possible to use extrapolation in conjunction with some fixed-step method such as the ETL to obtain increasingly higher-order approximations using the appropriate extrapolation coefficients from the Romberg table. In practice, the usefulness of repeated extrapolations is debatable, as each adds extra computational overhead and the higher accuracy can be obscured by Monte Carlo fluctuations. However it might be useful to create up to third or fourth-order methods in this way. We tried double-extrapolating the ETL on our

Table 1 Relative differences in moments for different splitting values of Schlögl system

Moment	Low peak	High peak
Mean	6.4%	1.6%
Second moment	21.8%	2.5%
Variance	81.6%	50.5%

Relative differences in moments $\mathbb{E}(f(X))$ for data split at $X = 250$ and $X = 350$ for the Schlögl system as percentages of their values when split at $X = 300$, i.e. $100 * |\mathbb{E}(f(X))_{\text{split}350} - \mathbb{E}(f(X))_{\text{split}250}| / \mathbb{E}(f(X))_{\text{split}300}$.

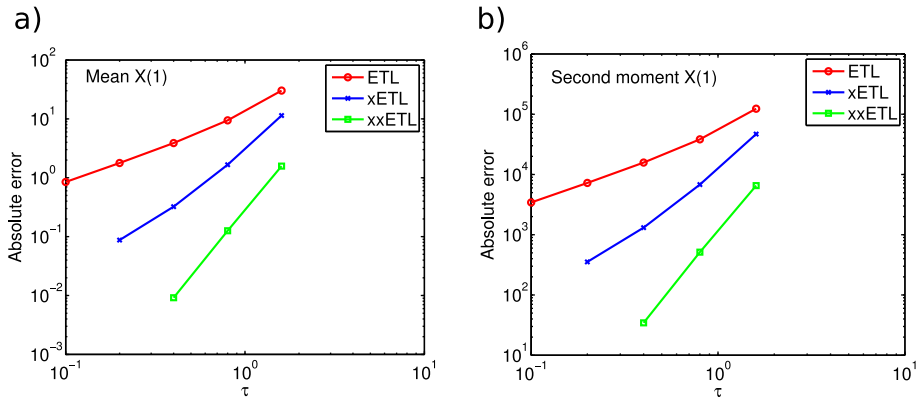

Figure 9 System 2, double-extrapolated absolute errors. Absolute error of X_1 for the Euler τ-leap (ETL), extrapolated Euler τ-leap (xETL) and double-extrapolated Euler τ-leap (xxETL) algorithm for **(a)** the mean, gradients 1.3 (ETL), 2.3 (xETL), 3.7 (xxETL), and **(b)** the second moment, gradients 1.3 (ETL), 2.4 (xETL), 3.8 (xxETL). Results from 10^8 simulations. Double-extrapolation produces consistently higher-order results.

test systems, with reasonable success. Figure 9 shows the ETL, xETL and double-extrapolated Euler τ-leap (xxETL) errors on species X_1 of System 2; in this case we can see that the xxETL results are approximately order three (or higher). Double-extrapolating gave substantially lower errors in almost every system, but it was not always easy to determine the order of convergence. A good example of this is System 3 in Figure 10. In such systems with relatively high Monte Carlo error, the approximate solutions from the xxETL were obscured by these fluctuations. The order of accuracy of the xxETL could be successfully seen for most molecular species of System 2, but it was not possible for the other systems as this would require a significant increase in the number of simulations, in order to further reduce the Monte Carlo error.

Monte Carlo error

The weak error we calculate numerically is

$$|\mathbb{E}(f(\mathbf{x}(T))) - \mathbb{E}_S(f(\mathbf{x}_n^\tau))|. \tag{14}$$

This error can be separated into two parts

$$\left[\mathbb{E}(f(\mathbf{x}(T))) - \mathbb{E}_\infty(f(\mathbf{x}_n^\tau))\right] + \left[\mathbb{E}_\infty(f(\mathbf{x}_n^\tau)) - \mathbb{E}_S(f(\mathbf{x}_n^\tau))\right], \tag{15}$$

where $\mathbb{E}_\infty(f(\mathbf{x}_n^\tau))$ are the theoretical values of the moments calculated by an infinite number of simulations with stepsize τ. The first term is the truncation error of the moments from their analytical solutions, i.e. the bias of the method, which depends only on the choice of timestep. The second term is the Monte Carlo error, which depends only on the number of simulations and is given by $\frac{C}{\sqrt{S}}$, where C is some constant and S the number of simulations. The Monte Carlo error can be so large that it overwhelms the bias of the underlying numerical method completely; in this case all of the numerical results are, in effect, incorrect, as they are random fluctuations.

This formulation is useful when the propensity functions are linear. In this case, the moment equations are

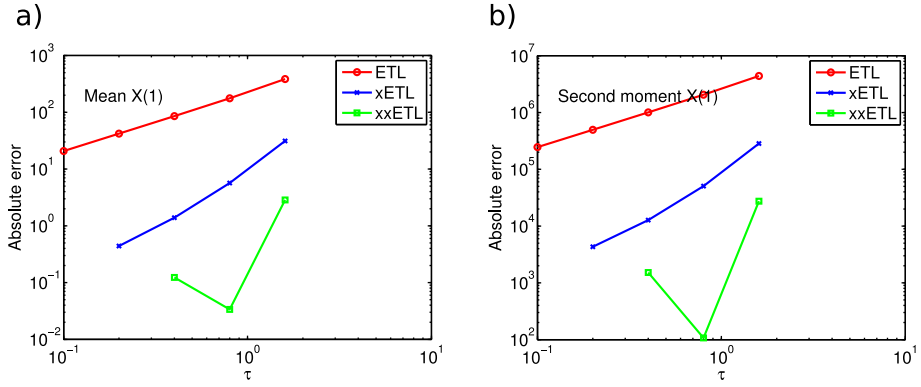

Figure 10 System 3, double-extrapolated absolute errors. Absolute error of X_1 for ETL, xETL and xxETL for **(a)** the mean, gradients 1.0 (ETL), 2.0 (xETL) and 2.3 (xxETL), and **(b)** the second moment, gradients 1.0 (ETL), 2.0 (xETL), 2.1 (xxETL). Results from 10^8 simulations. In this case, the xxETL errors do not show a clear order of accuracy, but are nonetheless smaller compared to the other methods.

closed, so $\mathbb{E}_\infty(f(\mathbf{x}_n^\tau))$ can be calculated for the appropriate numerical method. As an example, consider the mean of the ETL: its true value is given by Eq. (10) and the value of its numerical approximation can be found by iterating Eq. (8). In addition, a similar calculation can be found in Appendix D for the second moment. Unfortunately, this is not possible for non-linear systems, since in this case the equations describing the evolution of the moments are not closed any more [40].

Such a breakdown of the errors of System 1 with the ETL method is shown in Figure 11, using results from 10^8 simulations. We see that for the case of the ETL and xETL, the bias can easily be seen as the Monte Carlo errors are relatively low compared to the bias. However, when we extrapolate a second time, the bias of the resulting estimator is so low that Monte Carlo fluctuations completely obscure it, even with 10^8 simulations. This gives a poor approximation for the total error. The only way to reduce Monte Carlo error is to run more simulations or use variance reduction methods. This is a good illustration of why it is important to perform a suitable number of simulations to get accurate estimates of the moments.

Variance reduction methods, which aim to decrease the Monte Carlo error, are another useful way of reducing computational time: because the Monte Carlo error is lower, less simulations need to be run for a given accuracy, saving time. This is an important topic in its own right and we do not address it in this paper; we refer the interested reader to e.g. [41]. It is an active research area: recently Anderson and Higham [42] were able to significantly reduce the overal computational cost associated with the stochastic simulation of chemical kinetics, by extending the idea of multi level Monte Carlo for SDEs [43,44].

Conclusions

Throughout this paper, we have given reduced computational time as a motivation for using the extrapolation framework. As this is an important issue, we support here our claim that extrapolation speeds up simulations at a given error level. Although twice as many simulations must be run for a single extrapolation, the loss in computational time from more simulations is compensated for by the significant reduction in error. This is important, as slow runtime is often a limitation of stochastic methods. Total computational times and the corresponding errors in the mean (in brackets) of all three methods used in this paper and their extrapolated and double-extrapolated versions are shown in Tables 2 and 3 for Systems 1 and 4 (10^8 and 10^7 simulations, respectively). The estimated time to run the same number of SSA simulations is given for comparison. It should be noted that all the extrapolated and double-extrapolated τ-leap times are estimates: the time-consuming part of the extrapolation method is the two (or more) sets of simulations of the original method that must be run; the extrapolation itself is a fast arithmetic operation. The times for a single extrapolation are calculated as $runtime(\tau = h) + runtime(\tau = h/2)$, and for a double-extrapolation as $runtime(\tau = h) + runtime(\tau = h/2) + runtime(\tau = h/4)$. The extrapolated methods take several times longer to run, but the errors they give are several to hundreds of times lower. Most of the exceptions are where the error has clearly reached the Monte Carlo level (Tables 2 and 3, entries marked (-MC)). Besides extrapolation, the other obvious way to reduce error is to use a smaller timestep, so the real test for the effectiveness of extrapolation is to compare the runtimes of cases with similar error values. In Tables 2 and 3, we have marked in bold the extrapolated errors that can be directly compared to the base method (i.e. read up one row) and take less time to run, and similarly for double-extrapolated errors as compared to single-extrapolated ones. Although this varies for each system and simulation method, the general trend is that extrapolation takes less time to give a similar level of error, and this pattern was similar for our other example

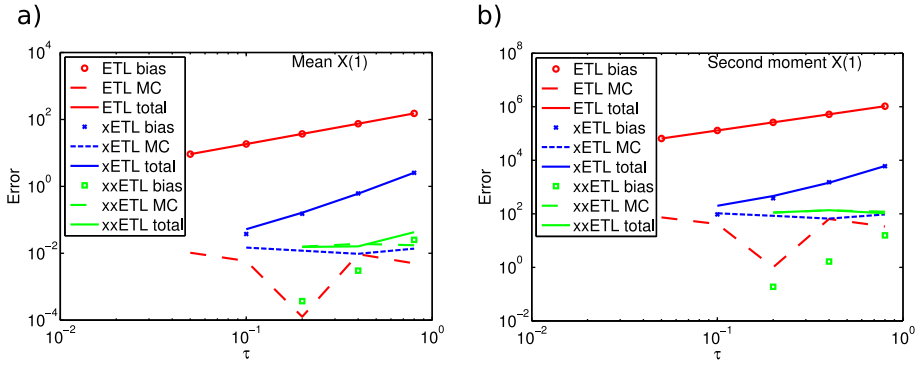

Figure 11 Split errors of System 1 using ETL. Split errors for System 1 for **(a)** the mean and **(b)** the second moment. The bias can be easily seen with the Euler τ-leap (ETL) or only a single extrapolation (xETL), but is obscured when we extrapolate a second time (xxETL). Results are from 10^8 simulations.

Table 2 Processing times of System 1

	τ				
	0.05	**0.1**	**0.2**	**0.4**	**0.8**
ETL	21.3 (9.2)	13.3 (18.4)	7.63 (37.1)	4.08 (74.7)	2.21 (152.0)
xETL	-	34.3 (0.05)	21 (0.16)	11.8 (0.61)	**6.34 (2.6)**
xxETL	-	-	(-MC) 42.1 (0.015)	(-MC) 25 (0.016)	**14 (0.043)**
MPTL	21.3 (0.024)	13.4 (0.070)	7.65 (0.25)	4.18 (1.0)	2.24 (4.2)
xMPTL	-	(-MC) 34.7 (0.0088)	(-MC) 21.1 (0.0082)	(-MC) 11.8 (0.0026)	**6.38 (0.041)**
xxMPTL	-	-	(-MC) 42.4 (0.0089)	(-MC) 25.2 (0.0090)	(-MC) 14 (0.0088)
TTTL	40.8 (0.017)	21.1 (0.063)	13.3 (0.26)	7.65 (1.1)	4.17 (4.2)
xTTTL	-	(-MC) 62.1 (0.0016)	(-MC) 34.6 (0.0022)	(-MC) 20.9 (0.0044)	**11.8 (0.041)**
xxTTTL	-	-	(-MC) 75.4 (0.0022)	(-MC) 42.2 (0.0031)	(-MC) 25.1 (0.011)

Processing times (in thousands of seconds) of 10^8 simulations of System 1 using Euler τ-leap, midpoint τ-leap and θ-trapezoidal τ-leap methods and their extrapolated and double-extrapolated versions. Absolute errors in the mean from the analytical values are included in brackets. For comparison, implementation of 10^8 simulations of an SSA using the Direct Method would take approximately 1.2×10^5 seconds. Entries marked (-MC) are thought to be below the Monte Carlo error level; bold entries are faster than the un/single-extrapolated versions with similar error level (i.e. compare with one row above).

systems. Thus we feel that in general, extrapolation is a worthwhile procedure.

Monte Carlo error is an unavoidable problem when using stochastic simulations. The statistical fluctuations inherent in stochastic systems can obscure the bias error (i.e. order of convergence) of the numerical method if their size relative to the bias is large, as the total error is made up of these two contributions. A large number S of simulations must be run, as the Monte Carlo error scales as $1/\sqrt{S}$. This error varies for each system. Figures 9 and 10 (and 11) show this clearly: both of the xxETLs have total errors of similar size for the same τ, but System 2 has relatively low Monte Carlo error, allowing us to see the bias of that system. However, we believe that System 3 has relatively high Monte Carlo error compared to its bias, implying that the xxETL errors we see in the figure are all due to statistical fluctuations. It should be noted that this seems to happen for all five test problems we use. The reason for this is that the extrapolated methods (and even the MPTL and TTTL, in some cases) have very high accuracy (i.e. low bias error). Since it is only possible to run a limited number of simulations, when the bias is very small, the total error will be given almost completely by the contribution from the Monte Carlo error.

A contrasting approach to reducing numerical errors is the multilevel Monte Carlo method. Originally developed for SDEs [43,44], it has recently been extended to discrete chemical kinetics [42]. By considering a formulation of the total error similar to Eq. (15), the multilevel Monte-Carlo method aims to reduce it by decreasing the Monte Carlo error. Here also many approximate solutions are generated with a variety of different timesteps. By intelligently combining many coarse-grained simulations with few fine-grained ones, it is possible to find a similar

Table 3 Processing times of System 4

	τ				
	0.005	**0.010**	**0.020**	**0.040**	**0.08**
ETL	191 (16.7)	82.3 (34.5)	46 (76.1)	32.6 (188.5)	19.7 (630.1)
xETL	-	293 (1.1)	**123 (7.2)**	**66.4 (36.3)**	35.1 (253.1)
xxETL	-	-	**279 (0.93)**	152 (2.5)	80.1 (36.0)
MPTL	165 (1.3)	84 (3.6)	44.9 (15.0)	24.1 (65.5)	12.3 (257.1)
xMPTL	-	244 (1.0)	129 (7.7)	69.1 (35.5)	36.3 (126.2)
xxMPTL	-	-	(-MC) 289 (1.2)	(-MC) 153 (1.5)	**81.3 (5.3)**
TTTL	277 (1.2)	155 (3.5)	82.4 (13.5)	44.6 (58.1)	23.3 (90.0)
xTTTL	-	(-MC) 430 (0.41)	(-MC) 239 (0.15)	**128 (1.4)**	68 (107.5)
xxTTTL	-	-	(-MC) 515 (0.45)	(-MC) 283 (0.37)	(-MC) 151 (16.9)

Processing times (in thousands of seconds) of 10^7 simulations of System 4 using Euler τ-leap, midpoint τ-leap and θ-trapezoidal τ-leap methods and their extrapolated and double-extrapolated versions. Absolute errors in the mean of X_1 from the SSA values are included in brackets. Running 10^7 simulations of an SSA using the Direct Method would take approximately 1.47×10^7 seconds. Entries marked (-MC) are thought to be below the Monte Carlo error level; bold entries are faster than the un/single-extrapolated versions with similar error level (i.e. compare with one row above).

level of accuracy to just using fine-grained simulations. In contrast, extrapolation uses the same number of coarse and fine-scale solutions and gives results which are more accurate than the fine-scale solution, by reducing the bias instead of the Monte Carlo error. In cases where the bias is obscured by statistical errors, using a combination of both extrapolation and the multilevel Monte Carlo method would be ideal, as it would reduce both sources of error. This is an interesting research question and we plan to address it in the future.

In this work, we have extended the extrapolation framework, which can increase the weak order of accuracy of existing numerical methods, to the discrete stochastic regime. To demonstrate the concept, we have applied it to three fixed-step methods, the Euler, midpoint and θ-trapezoidal τ-leap methods. Thus we have demonstrated numerically the effectiveness of extrapolation on a range of discrete stochastic numerical methods with different orders of accuracy for a variety of problems. The main requirement to use extrapolation with a numerical method is the existence of an expression for the global error that relates the error to the stepsize of the method. Analytically, this is all that must be found to show higher weak order convergence of the extrapolated method. To extrapolate once, only the leading error term need be known; further extrapolation requires knowledge of higher terms. We have found the form of the global weak error for a general weak order one method; the procedure is similar for higher-order methods. This is the real power of our approach: it can be applied to *any* fixed-step numerical method. Moreover, further extrapolations can raise the order of accuracy of the method indefinitely (although beyond a certain point the lower errors will be overtaken by Monte Carlo errors). We expect our method to be useful for more complex biochemical systems, for instance where frequent reactions must be simulated fast but accuracy is still important.

Appendices

SDE extrapolation

We summarise below the work of Talay and Tubaro [23] on extrapolating SDEs. The global weak error of a stochastic numerical scheme is

$$\text{error}(T,h) = \mathbb{E}(f(\mathbf{x}(T))) - \mathbb{E}(f(\mathbf{x}_n^h)). \tag{16}$$

To extrapolate this scheme, we must obtain an expression of the form (1). We start with the Kolmogorov backward equations,

$$\begin{aligned} u_t + \mathcal{L}u &= 0, \\ u(\mathbf{x},T) &= f(\mathbf{x}), \end{aligned}$$

where $u_t = \frac{\partial u}{\partial t}$, f is a homogeneous function on $[0,T]\times\mathbb{R}^N \to \mathbb{R}$ and $u(\mathbf{x},t) = \mathbb{E}(f(\mathbf{x}(T))|\mathbf{x}(t) = \mathbf{x})$. $\mathbf{x}$ are the solutions of the SDE (6) with initial conditions $\mathbf{x}(0) = \mathbf{x}_0$ and $\mathcal{L}$ is the generator of (6),

$$\mathcal{L} \equiv a(\mathbf{x},t)\cdot\nabla_{\mathbf{x}} + \frac{1}{2}b(\mathbf{x},t)b^T(\mathbf{x},t) : \nabla_{\mathbf{x}}\nabla_{\mathbf{x}},$$

with $A : B = \sum_{i,j} A_{ij}B_{ij}$. Crucially, (16) can then be written as

$$\text{error}(T,h) = \mathbb{E}u(\mathbf{x}_0,0) - \mathbb{E}u(\mathbf{x}_n^h, T).$$

The first term is known; the second can be found by recursively calculating the error between $\boldsymbol{u}$ when it is evaluated from adjacent timesteps:

$$\mathbb{E}\boldsymbol{u}(\mathbf{x}_n^h,T) = \mathbb{E}\boldsymbol{u}(\mathbf{x}_0,0) + h^2\sum_{k=0}^{n-1}\mathbb{E}\psi(\mathbf{x}_k^h,kh) + h^2K_n^h,$$

where $\psi(\mathbf{x},t)$ is an error function that is well-behaved and specific to each numerical method and K_n^h is a constant of unit order. For the case of the Euler-Maruyama and Milstein methods, the second term is $\mathcal{O}(h)$, so their global weak error is [23]

$$\text{error}(T,h) = -h\int_0^T \mathbb{E}\psi(\mathbf{x}(s),t)ds + \mathcal{O}(h^2). \tag{17}$$

Eq. (17) shows that the Euler-Maruyama and Milstein methods have global weak order one. It is easy to see that extrapolating them leads to solutions with global weak order two.

Discrete stochastic global error expansion

Here we derive a global error expansion similar to Eqs. (1) and (17) for the numerical approximation of moments by a weak first-order discrete stochastic method, such as the Euler or midpoint τ-leap methods. Once the form of the error expansion is known, it is clear what extrapolation coefficients to use and extrapolation is simple. Similarly to the case of SDEs, we start with the Kolmogorov backward equations

$$\partial_t u + \mathcal{L}u = 0 \qquad \text{in } \mathbb{Z}_+^N\times[0,T), \tag{18a}$$

$$u(\mathbf{x},t) = f(\mathbf{x}) \qquad \text{on } \mathbb{Z}_+^N\times\{T\}. \tag{18b}$$

Here

$$\mathcal{L}u \equiv \sum_{j=1}^{M} a_j(\mathbf{x})\left(u(\mathbf{x}+\nu_j) - u(\mathbf{x})\right)$$

and

$$u(\mathbf{x},t) = \mathbb{E}\left(f(\mathbf{X}_T)|\mathbf{X}_t = \mathbf{x}\right),$$

with $\mathbf{X}_t = \mathbf{X}(t)$. Using semigroup notation it is possible to formally denote the solution of (18) as

$$u(\mathbf{x},t) = e^{(T-t)\mathcal{L}}f(\mathbf{x}),$$

which can be useful for quantifying the local error of a numerical approximation. By applying a stochastic

Taylor expansion [45] to jump processes [16], the one-step expansion for the expectation of f calculated with a first-order numerical method should satisfy

$$\begin{aligned}\mathbb{E}(f(\mathbf{X}_h^h)|\mathbf{X}_0^h = \mathbf{x}) &= \mathcal{A}_h f(\mathbf{x}) \\ &= f(\mathbf{x}) + h\mathcal{L}f(\mathbf{x}) + \sum_{j=2}^{\infty} \frac{h^j}{j!}\mathcal{A}_j f(\mathbf{x}),\end{aligned} \tag{19}$$

where $\mathcal{A}_j$ are difference operators associated with the numerical method in hand. Furthermore we have for the true one-step expansion

$$\begin{aligned}\mathbb{E}(f(\mathbf{X}_h)|\mathbf{X}_0^h = \mathbf{x}) &= e^{\mathcal{L}h} f(\mathbf{x}) \\ &= f(\mathbf{x}) + h\mathcal{L}f(\mathbf{x}) + \sum_{j=2}^{\infty} \frac{h^j}{j!}\mathcal{L}^j f(\mathbf{x}).\end{aligned} \tag{20}$$

Remark 1. *An important element in our derivation of a global error expansion relates to the boundedness of $u(\mathbf{x}, t)$, and its discrete derivative. This boundedess is guaranteed [14] when the number of molecules in the chemical system is conserved or decreases with time. Proving this in the general case where zeroth-order reactions can add molecules to the system is a non-trivial task. One way around this problem is to set the propensity functions $a_j(\mathbf{x})$ to zero outside a large but finite domain [14]; this is the approach we follow here.*

Now if we let $e(f, T, h)$ define the global error in the numerical approximation of $\mathbb{E}(f(\mathbf{X}_T))$ at time $T = nh$ by a first-order numerical method with timestep h we see that

$$\begin{aligned}e(f, T, h) &= \mathbb{E}(f(\mathbf{X}_T)|\mathbf{X}_0 = \mathbf{x}) - \mathbb{E}(f(\mathbf{X}_T^h)|\mathbf{X}_0^h = \mathbf{x}) \\ &= u(\mathbf{x}, 0) - \mathbb{E}(u(\mathbf{X}_T^h, T)|\mathbf{X}_0^h = \mathbf{x}) \\ &= \sum_{i=1}^{N} \mathbb{E}(u(\mathbf{X}_{(i-1)h}^h, (i-1)h)) - \mathbb{E}(u(\mathbf{X}_{ih}^h, ih)) \\ &= \sum_{i=1}^{N} \mathbb{E}(u(\mathbf{X}_{ih}^*, ih)) - \mathbb{E}(u(\mathbf{X}_{ih}^h, ih)),\end{aligned}$$

where we have used the result in Figure 12, and for notational simplicity omitted the dependence of the expectations on the initial conditions. If we take $g_i(\mathbf{y}) = u(\mathbf{y}, ih) = e^{\mathcal{L}(T-ih)} f(\mathbf{y})$, then

$$\mathbb{E}(u(\mathbf{X}_{ih}^*, ih)) - \mathbb{E}(u(\mathbf{X}_{ih}^h, ih)) = \mathbb{E}(g_i(\mathbf{X}_{ih}^*)) - \mathbb{E}(g_i(\mathbf{X}_{ih}^h)).$$

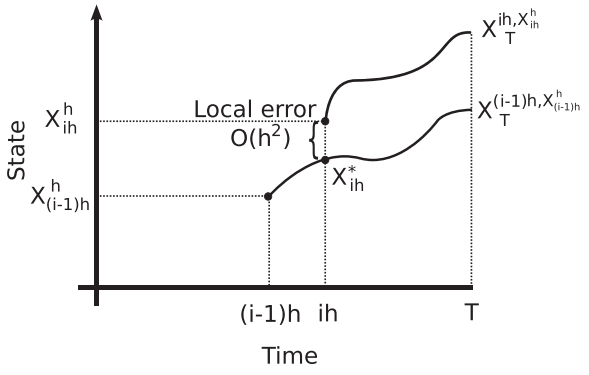

Figure 12 Local error. Representation of the local error in terms of the function $u(\mathbf{x}, t) = \mathbb{E}(f(\mathbf{X}_T^{t,\mathbf{x}}))$.

Applying Eqs. (19) and (20) to g_i,

$$\begin{aligned}e(f, T, h) &= \sum_{i=1}^{N} \mathbb{E}\left[\frac{h^2}{2}(\mathcal{L}^2 - A_2)g_i(\mathbf{X}_{(i-1)h}^h) + h^3 R_i^h\right] \\ &= \sum_{i=1}^{N} \mathbb{E}\left[\frac{h^2}{2}(\mathcal{L}^2 - A_2)e^{-h\mathcal{L}}g_{i-1}(\mathbf{X}_{(i-1)h}^h)\right] \\ &\quad + h^3\mathbb{E}(R_i^h) \\ &= \sum_{i=1}^{N} \frac{h^2}{2}\mathbb{E}\left[(\mathcal{L}^2 - A_2)e^{-h\mathcal{L}}u(\mathbf{X}_{(i-1)h}^h, (i-1)h)\right] \\ &\quad + h^3\mathbb{E}(R_i^h),\end{aligned}$$

where we have used the fact that $g_i = e^{-h\mathcal{L}}g_{i-1}$. We thus obtain

$$e(f, T, h) = \sum_{i=1}^{N} h^2\mathbb{E}(\psi_e((\mathbf{X}_{(i-1)h}^h, (i-1)h))) + h^3\mathbb{E}(\tilde{R}_i^h), \tag{21}$$

where

$$\psi_e(\mathbf{x}, t) = \frac{1}{2}\left(\mathcal{L}^2 - \mathcal{A}_2\right)u(\mathbf{x}, t), \tag{22}$$

and

$$\mathbb{E}(\tilde{R}_i^h) \leq C(T),$$

from our assumptions on the boundedness of $u(\mathbf{x}, t)$. Furthemore it is easy to see that

$$\sum_{i=1}^{N} h\mathbb{E}|\psi_e((\mathbf{X}_{(i-1)h}^h, (i-1)h)| \leq C(T).$$

Using this and results from Talay and Tubaro [23] we find that

$$e(f,T,h) = -h\int_0^T \mathbb{E}(\psi_e(\mathbf{X}_s,s))ds + \mathcal{O}(h^2),$$

where $\psi_e(\mathbf{x},t)$ is dependent on the numerical method and given by (22).

An example explicit calculation of the global error expansion for a linear system

In Eq. (12) we calculated the global error on the mean for a linear system using the equations for its true and numerical solutions. We show how this approach connects to the general formulation of the errors given by Eq. (21) by using System 1 as an example. We choose this example as it is one-dimensional and linear and is simple enough that the global error Eq. (12) can be calculated explicitly from the general formulation. We define the following notation for the discrete difference:

$$f^{\nu} = f(x+\nu) - f(x),$$

where we have assumed that we have only one chemical species subject to only one chemical reaction. For this particular case,

$$\mathcal{L}^2 f = a^2 f^{\nu\nu} + a^2 a^{\nu} f^{\nu\nu} + a a^{\nu} f^{\nu},$$

while for the ETL

$$\mathcal{A}_2 f = a^2 f^{\nu\nu},$$

and thus

$$\begin{aligned}\psi_e(x,t) = &\frac{1}{2}(a^2(x)(a(x+\nu)-a(x))(u(x+2\nu,t)\\ &-2u(x+\nu,t)+u(x,t))\\ &+\frac{1}{2}a(x)(a(x+\nu)-a(x))(u(x+\nu,t)-u(x,t)).\end{aligned} \tag{23}$$

For particle decay, $\nu = -1$ and $a(x) = \kappa x$. For $g(x) = x$, the solution to (18) is

$$u(x,t) = xe^{-\kappa(T-t)},$$

so (23) becomes

$$\psi_e(x,t) = \frac{1}{2}\kappa^2 x e^{-\kappa(T-t)}.$$

Thus the global error given by the general formulation is

$$\begin{aligned}&\sum_{i=1}^{N} h^2 \mathbb{E}(\psi_e((X^h_{(i-1)h},(i-1)h)))\\ &= \frac{\kappa^2 h^2}{2}\sum_{i=1}^{N}\mathbb{E}(X^h_{(i-1)h})e^{-\kappa(T-(i-1)h)}\\ &= \frac{\kappa^2 h^2}{2}\sum_{i=1}^{N}\left[(1-\kappa h)e^{\kappa h}\right]^{i-1} e^{-\kappa T}\mathbb{E}(X^h_0)\\ &= \frac{\kappa^2 Th}{2}\left(\frac{\left[(1-\kappa h)e^{\kappa h}\right]^N - 1}{(1-\kappa h)e^{\kappa h}-1}\right)\frac{e^{-\kappa T}}{N}\mathbb{E}(X^h_0)\\ &= \frac{\kappa^2 Th}{2}\mathbb{E}(X^h_0) + \mathcal{O}(h^2),\end{aligned}$$

which agrees exactly with what one obtains by calculating Eq. (12) for System 1 (i.e. setting $d = 0$, $W = -\kappa$).

Formulae for the second moment of the Euler τ-leap in the case of linear systems

We can write the formulae for the numerical and analytical evaluation of the second moment of the ETL. This is useful for finding the relative contributions of the bias and Monte Carlo errors, as well as for explicitly calculating the local errors. The ETL evolves the second moment in time as

$$\begin{aligned}\mathbb{E}(\mathbf{x}^{\tau}_{m+1}(\mathbf{x}^{\tau}_{m+1})^T) = \mathbb{E}\Big[&\left(\mathbf{x}^{\tau}_m + \nu\mathcal{P}(\tau(C\mathbf{x}^{\tau}_m+\mathbf{d}))\right)\\ &\left(\mathbf{x}^{\tau}_m + \nu\mathcal{P}(\tau(C\mathbf{x}^{\tau}_m+\mathbf{d}))\right)^T\Big].\end{aligned} \tag{24}$$

Using the law of total expectation and writing $S^{\tau}_m = \mathbb{E}(\mathbf{x}^{\tau}_m(\mathbf{x}^{\tau}_m)^T)$, $B^{\tau}_m = \mathrm{diag}(C\mathbb{E}(\mathbf{x}^{\tau}_m)))$ and $\boldsymbol{D} = \mathrm{diag}(\mathbf{d})$, we obtain

$$\begin{aligned}S^{\tau}_{m+1} = &S^{\tau}_m + \tau W S^{\tau}_m + \tau S^{\tau}_m W^T + \tau\boldsymbol{\mu}^{\tau}_m\mathbf{d}^T\nu^T + \tau\nu\mathbf{d}(\boldsymbol{\mu}^{\tau}_m)^T\\ &+\tau^2 W S^{\tau}_m W^T + \tau^2 W\boldsymbol{\mu}^{\tau}_m\mathbf{d}^T\nu^T + \tau^2\nu\mathbf{d}(\boldsymbol{\mu}^{\tau}_m)^T W^T\\ &+\tau^2\nu\mathbf{d}\mathbf{d}^T\nu^T + \tau\nu B^{\tau}_m\nu^T + \tau\nu D\nu^T.\end{aligned} \tag{25}$$

We can now iterate this formula in order to obtain the numerical approximation for the second moment of the ETL at any timestep.

The behaviour of the second moment in time as given by the CME, $S(t) = \mathbb{E}(\mathbf{x}(t)\mathbf{x}^T(t))$, is [33,40]

$$\begin{aligned}\frac{dS(t)}{dt} = &WS(t) + S(t)W^T + \boldsymbol{\mu}(t)\mathbf{d}^T\nu^T + \nu\mathbf{d}\boldsymbol{\mu}^T(t)\\ &+\nu B(t)\nu^T + \nu D\nu^T,\end{aligned} \tag{26}$$

where $B(t) = \mathrm{diag}(C\boldsymbol{\mu}(t))$. By solving Eq. (26) and iterating Eq. (25), we can use the formulation of Eq. (15) to quantify the bias and Monte Carlo errors of the ETL.

A similar approach can also be used for the MPTL and TTTL.

List of abbreviations

ODE;Ordinary differential equation, SDE;Stochastic differential equation, CME;Chemical Master Equation, SSA;Stochastic simulation algorithm, ETL;Euler τ-leap, xETL;Extrapolated Euler τ-leap, xxETL;Double-extrapolated Euler τ-leap, MPTL, xMPTL, xxMPTL; Midpoint τ-leap and extrapolated and double-extrapolated versions,TTTL, xTTTL, xxTTTL;Theta-trapezoidal τ-leap and extrapolated and double-extrapolated versions, CV;Coefficient of variation

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

The main ideas were jointly discussed and developed by all authors, and all authors wrote the manuscript. The global error expansion was mainly calculated by KCZ. TSz performed the simulations and analysed them. All authors have read and approved the final version of the manuscript.

Acknowledgements

TSz is supported by the Engineering and Physical Sciences Research Council through the Systems Biology Doctoral Training Centre, University of Oxford. This publication was based on work supported in part by Award No. KUK-C1-013-04, made by King Abdullah University of Science and Technology (KAUST). The research leading to these results has received funding from the European Research Council under the *European Community's* Seventh Framework Programme (FP7/2007-2013) /ERC *grant agreement* No. 239870. RE would also like to thank Somerville College, University of Oxford, for a Fulford Junior Research Fellowship; Brasenose College, University of Oxford, for a Nicholas Kurti Junior Fellowship; the Royal Society for a University Research Fellowship; and the Leverhulme Trust for a Philip Leverhulme Prize. TSz would like to thank Manuel Barrio for his discussions and help with the simulations. KCZ would like to thank James Lottes for his invaluable comments regarding the global error expansion.

Author details

[1]Department of Computer Science, University of Oxford, Oxford, OX1 3QD, UK. [2]Department of Mathematics, Queensland University of Technology, Brisbane, Qld 4001, Australia. [3]Mathematical Institute, University of Oxford, Oxford, OX1 3LB, UK. [4]School of Mathematics, University of Southampton, Southampton, SO17 1BJ, UK. [5]Mathematics Section, École Polytechnique Fédérale de Lausanne, Station 8, CH-1015 Lausanne, Switzerland.

Received: 28 February 2012 Accepted: 22 June 2012
Published: 15 July 2012

References

1. McAdams HH, Arkin A: **It's a noisy business! Genetic regulation at the nanomolar scale.** *Trends Genet* 1999, **15:**65–69.
2. Elowitz M, Levine A, Siggia E, Swain P: **Stochastic gene expression in a single cell.** *Science* 2002, **297:**1183–1186.
3. Fedoroff N, Fontana W: **Small numbers of big molecules.** *Science* 2002, **297:**1129–1131.
4. Kaern M, Elston T, Blake W, Collins J: **Stochasticity in gene expression: from theories to phenotypes.** *Nature Rev Genet* 2005, **6:**451–464.
5. Wilkinson DJ: **Stochastic modelling for quantitative description of heterogeneous biological systems.** *Nature Rev Genet* 2009, **10:**122–133.
6. Erban R, Chapman SJ, Kevrekidis I, Vejchodsky T: **Analysis of a stochastic chemical system close to a SNIPER bifurcation of its mean-field model.** *SIAM J Appl Mathematics* 2009, **70:**984–1016.
7. McQuarrie DA: **Stochastic approach to chemical kinetics.** *J Appl Probability* 1967, **4:**413–478.
8. Gillespie DT: **A rigorous derivation of the Chemical Master Equation.** *Physica A* 1992, **188:**404–425.
9. Gillespie DT: **Exact stochastic simulation of coupled chemical reactions.** *J Phys Chem* 1977, **81:**2340–2361.
10. Gillespie DT: **Approximate accelerated stochastic simulation of chemically reacting systems.** *J Chem Phys* 2001, **115:**1716–1733.
11. Gillespie DT, Petzold LR: **Improved leap-size selection for accelerated stochastic simulation.** *J Chem Phys* 2003, **119:**8229–8234.
12. Cao Y, Gillespie DT, Petzold LR: **Efficient step size selection for the tau-leaping simulation method.** *J Chem Phys* 2006, **124:**044109.
13. Rathinam M, Petzold LR, Cao Y, Gillespie DT: **Consistency and stability of tau-leaping schemes for chemical reaction systems.** *Multiscale Model Simul* 2005, **4:**867–895.
14. Li T: **Analysis of explicit tau-Leaping schemes for simulating chemically reacting systems.** *Multiscale Model Simul* 2007, **6:**417–436.
15. Anderson DF, Ganguly A, Kurtz TG: **Error analysis of tau-leap simulation methods.** *Ann Appl Probability* 2011, **21:**2226–2262.
16. Hu Y, Li T, Min B: **The weak convergence analysis of tau-leaping methods: revisited.** *Commun Math Sci*. in press.
17. Tian TH, Burrage K: **Binomial leap methods for simulating stochastic chemical kinetics.** *J Chem Phys* 2004, **121:**10356–10364.
18. Hu Y, Li T: **Highly accurate tau-leaping methods with random corrections.** *J Chem Phys* 2009, **130:**124109.
19. Anderson DF, Koyama M: **Weak error analysis of numerical methods for stochastic models of population processes** 2011. Arxiv.org:1102.2922.
20. Hu Y, Li T, Min B: **A weak second order tau-leaping method for chemical kinetic systems.** *J Chem Phys* 2011, **135:**024113.
21. Anderson DF, Mattingly JC: **A weak trapezoidal method for a class of stochastic differential equations.** *Commun Math Sci* 2011, **9:**301–318.
22. Hairer E, Nørsett SP, Wanner G: *Solving ordinary differential equations: Nonstiff problems*. 2nd edition. Berlin: Springer; 1993.
23. Talay D, Tubaro L: **Expansion of the global error for numerical schemes solving stochastic differential equations.** *Stochastic Anal Appl* 1990, **8:**483–509.
24. Rué P, Villa-Freixà J, Burrage K: **Simulation methods with extended stability for stiff biochemical kinetics.** *BMC Sys Biol* 2010, **4:**110–123.
25. Gibson M, Bruck J: **Efficient exact stochastic simulation of chemical systems with many species and many channels.** *J Phys Chem A* 2000, **104:**1876–1889.
26. Cao Y, Li H, Petzold LR: **Efficient formulation of the stochastic simulation algorithm for chemically reacting systems.** *J Chem Phys* 2004, **121:**4059–4067.
27. Chatterjee A, Vlachos DG, Katsoulakis MA: **Binomial distribution based tau-leap accelerated stochastic simulation.** *J Chem Phys* 2005, **122:**024112.
28. Cao Y, Gillespie DT, Petzold LR: **Avoiding negative populations in explicit Poisson tau-leaping.** *J Chem Phys* 2005, **123:**054104.
29. Yates CA, Burrage K: **Look before you leap: A confidence-based method for selecting species criticality while avoiding negative populations in tau-leaping.** *J Chem Phys* 2011, **134:**084109.
30. Liu XQ, Li CW: **Weak approximation and extrapolations of stochastic differential equations with jumps.** *SIAM J Numer Anal* 2000, **37:**1747–1767.
31. Shohat J, Tamarkin JD: *The Problem of Moments*. New York: American Mathematical Society; 1943.
32. Jahnke T, Huisinga W: **Solving the Chemical Master Equation for monomolecular reaction systems analytically.** *J Math Biol* 2007, **54:**1–26.
33. Gadgil C, Lee C, Othmer H: **A stochastic analysis of first-order reaction networks.** *Bull Math Biol* 2005, **67:**901–946.
34. van Kampen NG: *Stochastic Processes in Physics and Chemistry*. 3rd edition. Amsterdam: Elsevier; 2007.
35. Shahrezaei V, Swain PS: **Analytical distributions for stochastic gene expression.** *Proc Nat Acad Sci USA* 2008, **105:**17256–17261.
36. Marquez-Lago T, Stelling J: **Counter-intuitive stochastic behavior of simple gene circuits with negative feedbacks.** *Biophys J* 2010, **98:**1742–1750.
37. Marquez-Lago T, Burrage K: **Binomial tau-leap spatial stochastic simulation algorithm for applications in chemical kinetics.** *J Chem Phys* 2007, **127:**104101.
38. Schlögl F: **Chemical reaction models for nonequilibrium phase-transitions.** *Z Phys* 1972, **253:**147–161.
39. MacNamara S, Burrage K, Sidje RB: **Multiscale modeling of chemical kinetics via the master equation.** *Multiscale Model Simul* 2007, **6:**1146–1168.

40. Mélykúti B, Burrage K, Zygalakis KC: **Fast stochastic simulation of biochemical reaction systems by alternative formulations of the chemical Langevin equation.** *J Chem Phys* 2010, **132:**164109.
41. Liu JS: *Monte Carlo strategies in scientific computing*. 3rd edition. New York: Springer; 2001.
42. Anderson DF, Higham DJ: **Multi-level Monte Carlo for continuous time Markov chains, with applications in biochemical kinetics.** *SIAM Multiscale Model Simul* 2012, **10:**146–179.
43. Kebaier A: **Statistical Romberg extrapolation: A new variance reduction method and applications to option pricing.** *Ann Appl Probability* 2005, **15:**2681–2705.
44. Giles MB: **Multilevel Monte Carlo path simulation.** *Operations Res* 2008, **56:**607–617.
45. Rößler A: **Stochastic Taylor expansions for the expectation of functionals of diffusion processes.** *Stochastic Anal App* 2004, **22:**1553–1576.

doi:10.1186/1752-0509-6-85
Cite this article as: Székely *et al.*: A higher-order numerical framework for stochastic simulation of chemical reaction systems. *BMC Systems Biology* 2012 **6**:85.