Simulation of Stochastic Kinetic Models

Andrew Golightly & Colin S. Gillespie
School of Mathematics & Statistics,
Newcastle University, UK

A growing realisation of the importance of stochasticity in cell and molecular processes has stimulated the need for statistical models that incorporate intrinsic (and extrinsic) variability. In this chapter we consider stochastic kinetic models of reaction networks leading to a Markov jump process representation of a system of interest. Traditionally, the stochastic model is characterised by a chemical master equation. Whilst the intractability of such models can preclude a direct analysis, simulation can be straightforward and may present the only practical approach to gaining insight into a system's dynamics. We review exact simulation procedures before considering some efficient approximate alternatives.

Keywords: stochastic simulation; Markov jump process; time discretisation.

1 Introduction

Computational systems biology is typically concerned with developing dynamic simulation models of biological processes. Such models can be used to test understanding of systems of interest and perform *in silico* experimentation. Statistical methods based on the macroscopic rate equation (MRE), which describes the thermodynamic limit of a system via a set of coupled ordinary differential equations (ODEs) have been widely used [1, 2]. Such an approach may be appropriate when describing average concentrations within a population of cells or when modelling more global physiology, for example, at a tissue or organ level [3]. Single cell experiments and studies of noise in regulatory networks have revealed the importance of stochastic effects in intra-cellular processes and

The R code used in this chapter can be downloaded from the github repository: https://github.com/csgillespie/In-silico-Systems-Biology

in turn, this has motivated the need for models that incorporate intrinsic stochasticity [4]. A deterministic modelling approach fails to capture the stochastic (and discrete) nature of chemical kinetics at low concentrations. Reaction events are intrinsically stochastic, driven by Brownian motion. When such events take place, the effect is to change biochemical species numbers by an integer amount. Hence, as reactions take place, species numbers change abruptly and discretely. Such arising random fluctuations are often referred to as *intrinsic noise* [5]. Other sources of noise may be termed *extrinsic* (for example, due to variations in initial conditions or environmental conditions).

The aim of this chapter is to provide a concise introduction to the simulation of stochastic kinetic models through consideration of some commonly used simulation algorithms. Approximate strategies based on for example, time discretisation or the dispensation of the assumption of discrete states, are also explored. The remainder of this chapter is organised as follows. In Section 2, we briefly review stochastic chemical kinetics leading to a stochastic kinetic model of a system of interest, formulated as a Markov jump process. We consider exact simulation of the jump process via the Gillespie algorithm [6] in Section 3 before examining some recently proposed extensions which aim to increase the computational efficiency of the algorithm. Approximate simulation strategies such as the tau-leap [7], chemical Langevin equation [8, 9] and linear noise approximation [10] are considered in Section 4. The chapter concludes with a discussion in Section 6.

2 Stochastic chemical kinetics

In this section we represent a biological system of interest with a set of pseudo-biochemical reactions. There are a number of ways in which a system could be represented, from a qualitative diagram to a fully quantitative set of equations. A reaction network provides a flexible representation, allowing the modeller to specify the level of detail deemed appropriate. Once the assumptions about the underlying chemical kinetics have been made, simulation can take place.

2.1 Reaction networks

To fix notation, consider a biochemical reaction network involving u species $\mathcal{X}_1$, $\mathcal{X}_2, \ldots, \mathcal{X}_u$ and v reactions $R_1, R_2, \ldots, R_v$, written using standard chemical reaction notation as

$$\begin{aligned}
R_1: &\quad p_{11}\mathcal{X}_1 + p_{12}\mathcal{X}_2 + \cdots + p_{1u}\mathcal{X}_u \longrightarrow q_{11}\mathcal{X}_1 + q_{12}\mathcal{X}_2 + \cdots + q_{1u}\mathcal{X}_u \\
R_2: &\quad p_{21}\mathcal{X}_1 + p_{22}\mathcal{X}_2 + \cdots + p_{2u}\mathcal{X}_u \longrightarrow q_{21}\mathcal{X}_1 + q_{22}\mathcal{X}_2 + \cdots + q_{2u}\mathcal{X}_u \\
&\qquad\qquad \vdots \qquad\qquad\qquad\quad \vdots \qquad\qquad \vdots \\
R_v: &\quad p_{v1}\mathcal{X}_1 + p_{v2}\mathcal{X}_2 + \cdots + p_{vu}\mathcal{X}_u \longrightarrow q_{v1}\mathcal{X}_1 + q_{v2}\mathcal{X}_2 + \cdots + q_{vu}\mathcal{X}_u.
\end{aligned}$$

Let $X_{j,t}$ denote the number of molecules of species $\mathcal{X}_j$ at time t, and let X_t be the u-vector $X_t = (X_{1,t}, X_{2,t}, \ldots, X_{u,t})'$. Further, let $P = (p_{ij})$ be a $v \times u$ matrix of the coefficients

p_{ij} with $Q = (q_{ij})$ defined similarly. The $u \times v$ *stoichiometry matrix* S is defined by

$$S = (Q - P)'.$$

The matrices P, Q and S will typically be *sparse*. On the occurrence of a reaction of type i, the system *state* (X_t) is updated by adding the ith column of S. Consequently, if ΔR is a v-vector containing the number of reaction events of each type in a given time interval, then the system state should be updated by ΔX, where

$$\Delta X = S\,\Delta R.$$

The stoichiometry matrix therefore encodes important structural information about the reaction network. In particular, vectors in the left null-space of S correspond to *conservation laws* in the network, that is, any u-vector a satisfying $a'S = 0$ has the property (clear from the above equation) that $a'X_t$ remains constant for all t.

2.2 Markov jump process representation

Let us consider a bi-molecular reaction

$$\mathcal{X}_1 + \mathcal{X}_2 \longrightarrow \mathcal{X}_3\,.$$

This reaction will occur when a molecule of $\mathcal{X}_1$ collides with a molecule of $\mathcal{X}_2$ while moving around randomly, driven by Brownian motion. Consider a pair of such molecules in a container of fixed volume. Under fairly weak assumptions involving the container and its contents, it is possible to show that the collision hazard (or rate) is constant in a small time interval [8]. Therefore, for the reaction above, the probability of a given pair of molecules reacting in a time interval of length dt is $c\,dt$ for some constant c. Suppose now that there are x_1 molecules of $\mathcal{X}_1$ and x_2 molecules of $\mathcal{X}_2$. There are x_1x_2 possible pairs of molecules that could react so the probability of a reaction of this type occurring in a time interval of length dt is cx_1x_2dt. Note that this probability depends only on the current state of the system – this is the *Markov property*. Moreover, changes to the system state occur at discrete times (say $t_1, t_2 \ldots$) and there are finite periods of no change. Taking both of these properties together gives a *Markov jump process*, that is, a continuous time, discrete valued process that satisfies the Markov property.

We now consider the general case. Under the standard assumption of *mass-action stochastic kinetics*, each reaction R_i is assumed to have an associated rate constant, c_i, and a *propensity function*, $h_i(X_t, c_i)$, which define the overall *hazard* of a type i reaction occurring. That is, the system is a Markov jump process, and for an infinitesimal time increment dt, the probability of a type i reaction occurring in the time interval $(t, t + dt]$ is $h_i(X_t, c_i)dt$. Under mass-action stochastic kinetics, the hazard function is proportional to a product of binomial coefficients, with

$$h_i(X_t, c_i) = c_i \prod_{j=1}^{u} \binom{X_{j,t}}{p_{ij}}.$$

It should be noted that this hazard function differs slightly from the standard mass action rate laws used in continuous deterministic modelling, but is consistent (up to a constant of proportionality in the rate constant) asymptotically in the high concentration limit.

Order	Reactants	Products	Hazard	Description
0	$\emptyset$	$\mathcal{X}_1$	c_1	Influx
1	$\mathcal{X}_1$	$\emptyset$	$c_2 X_1$	Degradation
2	$\mathcal{X}_1 + \mathcal{X}_2$	$\mathcal{X}_3$	$c_3 X_1 X_2$	Catalysation
2	$2\mathcal{X}_1$	$\mathcal{X}_2$	$c_4 X_1(X_1 - 1)/2$	Dimerisation
3	$3\mathcal{X}_1$	$\mathcal{X}_3$	$c_5 X_1(X_1 - 1)(X_1 - 2)/6$	Trimerisation

Table 1: Example reactions and their associated hazards.

2.2.1 Reaction types

Some commonly encountered reaction types are given in Table 1 along with their associated hazards. For notational simplicity we remove the dependence of the current state on time when stating the hazards.

The zeroth order reaction may at first seem a little strange, since it appears that something is created from nothing. However, it can be useful for modelling a constant rate of production of a chemical species. First order reactions can be used to capture the spontaneous change of a molecule such as decay and dissociation. It is often desirable to write third or higher order reactions in terms of a series of reactions of order two or less. For example, the trimerisation reaction in Table 1 can be written as

$$2\mathcal{X}_1 \longrightarrow \mathcal{X}_2 \quad \text{and} \quad \mathcal{X}_2 + \mathcal{X}_1 \longrightarrow \mathcal{X}_3.$$

2.3 Chemical master equation

The chemical master equation (CME) refers to an ODE satisfied by the transition kernel of the Markov jump process. One such ODE can be derived as follows.

Let $p(x;t)$ denote the transition kernel of the jump process, that is, the probability that there will be at time t $x = (x_1, \ldots, x_u)'$ molecules of each respective species (assuming a well stirred spatially homogeneous volume Ω, and thermal equilibrium). Once this function is obtained, a fairly complete characterisation of the state of the system at time t is apparent. Now write $p(x; t + \Delta t)$ as the sum of the probabilities of the number of ways in which the network can arrive in state x at time $t + \Delta t$. We obtain

$$p(x; t + \Delta t) = \sum_{i=1}^{v} h_i(x - S^i, c_i) P(x - S^i; t)\Delta t + \left\{ 1 - \sum_{i=1}^{v} h_i(x, c_i)\Delta t \right\} p(x; t) \qquad (1)$$

where x is the state of the system at time t and S^i denotes the i^{th} column of the stoichiometry matrix S. Intuitively, the term $h_i(x - S^i, c_i)p(x - S^i; t)\Delta t$ is the probability that the system is one R_i reaction removed from state x at time t and then undergoes such a reaction in $(t, t + \Delta t)$. The second quantity in (1) is the probability that the system undergoes no reactions in $(t, t + \Delta t)$. We now observe that (1) leads to the ODE

$$\frac{d}{dt} p(x; t) = \sum_{i=1}^{v} \left\{ h_i(x - S^i, c_i) p(x - S^i; t) - h_i(x, c_i) p(x; t) \right\} . \qquad (2)$$

Equation (2) is most commonly referred to as the CME and is simply Kolmogorov's forward equation for the MJP. Unfortunately, the CME is only tractable for a handful of cases. The exactly solvable cases have been summarised by [11]. Hence, for most systems of interest, an analysis via the CME will not be possible and then stochastic simulation techniques such as those described in the next section will present the only practical approach to gaining insight into a system's dynamics. For further details of the master equation formalism in chemical kinetics, good reviews have been given by [10] and [4].

3 Exact simulation methods

3.1 The Gillespie algorithm

Let $c = (c_1, c_2, \ldots, c_v)'$ and $h(X_t, c) = (h_1(X_t, c_1), h_2(X_t, c_2), \ldots, h_v(X_t, c_v))'$. Values for c and the initial system state x_0 completely specify the Markov process. Although the Markov jump process is rarely analytically tractable for interesting models, it is straightforward to forward-simulate exact realisations of this Markov process using a discrete event simulation method. This is due to the fact that if the current time and state of the system are t and X_t respectively, then the time to the next event can be shown to have an exponential distribution with rate parameter

$$h_0(X_t, c) = \sum_{i=1}^{v} h_i(X_t, c_i),$$

and the event will be a reaction of type R_i with probability $h_i(X_t, c_i)/h_0(X_t, c)$ independently of the waiting time. Forward simulation of process realisations in this way is typically referred to as *Gillespie's direct method* in the stochastic kinetics literature, after [6]. The procedure is summarised in algorithm 1. Note that the assumptions of mass-action kinetics, as well as the one-to-one correspondence between reactions and rate constants may both be relaxed. It is also worth mentioning that there is an equivalent alternative algorithm to Gillespie's direct method known as the *first reaction method* [12], although the direct method is typically to be preferred as it is more efficient. In particular, it requires just two random numbers to be simulated per event as opposed to the first reaction method, which requires v. That said, the first reaction method can be turned into a far more efficient method, known as the Gibson-Bruck algorithm [13]. We eschew the method here in favour of further examination of Gillespie's direct method, which we can speed up with a few clever "tricks".

3.2 Speeding up Gillespie's direct method

Not surprisingly, as the number of reactions and species increase, the length of time taken to perform a single iteration of the Gillespie algorithm also increases. We will examine some simple techniques for speeding up the method.

Algorithm 1 Gillespie's Direct Method

1. Set $t = 0$. Initialise the rate constants $c_1, \ldots, c_v$ and the initial molecule numbers $x_1, \ldots, x_u$.
2. Calculate $h_i(x, c_i)$, $i = 1, \ldots, v$ based on the current state, x.
3. Calculate the *combined hazard* $h_0(x, c) = \sum_{i=1}^{v} h_i(x, c_i)$.
4. Simulate the time to the next event, $t' \sim Exp(h_0(x, c))$ and put $t := t + t'$.
5. Simulate the reaction index, j, as a discrete random quantity with probabilities $h_i(x, c_i)/h_0(x, c)$, $i = 1, \ldots, v$.
6. Update x according to reaction j. That is, put $x := x + S^j$.
7. Output x and t. If $t < T_{max}$, return to step 2.

3.2.1 Hazards update

At each iteration, we update each of the v hazards, $h_i(x, c_i)$, $i = 1, \ldots, v$ - step 2 of algorithm 2. This requires v computations and is therefore $O(v)$. Naturally, after a single reaction has occurred, a better method is to only update the hazards that have changed. To this end, it is helpful to construct a dependency graph whose nodes represent reactions and a (directed) edge from one node to another indicates that one reaction affects the hazard of another.

3.2.2 Combined hazard update

At each iteration, we combine all v hazards to calculate the combined hazard

$$h_0(x, c) = \sum_{i=1}^{v} h_i(x, c_i)\,.$$

This is again $O(v)$. If we have used a dependency graph to determine which reaction hazards have changed after the last reaction occurrence then we can calculate the combined hazard by subtracting "old" hazard values (before the single reaction occurrence) and adding updated "new" hazard values. This is likely to be less demanding than recalculating h_0 from scratch.

3.2.3 Reaction selection

In this step we choose a reaction with probability proportional to its hazard, that is, we search for the j satisfying

$$\sum_{i=1}^{j-1} h_i(x, c_i) \;<\; U \times h_0(x, c) \;<\; \sum_{i=1}^{j} h_i(x, c_i)$$

where $U \sim U(0,1)$. To speed up this step, we can order each h_i in terms of size. One technique is to run a few pre-simulations for a short period of time $t << T_{max}$ [14]. The authors suggest reordering the hazard vector according to the relative occurrences of each reaction in the pre-simulations. Plainly, this method is not ideal as it is not clear how long to run the pre-simulations for, and the pre-simulations will be time consuming. Another method is to move h_i up one place in the hazard vector for each time reaction i is executed[15]. This swapping effectively reduces the search depth for a reaction at the next occurrence of that reaction. Note that the reordering only requires a swap of two memory addresses.

4 Approximate simulation methods

We have seen how to generate *exact* simulations from a stochastic kinetic model via the Gillespie algorithm and how to make the procedure efficient through the use of a few "tricks". However, if we are prepared to sacrifice the exactness of the simulation method, there is a potential for huge speed-ups.

One method is to divide up the time axis into small discrete chunks over which we approximate the underlying kinetics to allow advancement of the state from the start of one chunk to another in one step. We will work on the assumption that time intervals are small enough to assume constant reaction hazards over the interval.

4.1 Poisson and tau leap

Consider a Markov process with a constant hazard (say α) of events occurring throughout time, so that the first event follows an exponential $Exp(\alpha)$ distribution. It can then be shown that the number of events, say X, in the interval $(0,t]$ follows a Poisson $Po(\alpha t)$ distribution. A Markov process with constant hazard is known as a (homogeneous) Poisson process.

Given this basic property of the Poisson process, we assume that the number of reactions (of a given type) occurring in a short time interval has a Poisson distribution (independently of other reaction types). We can then simulate Poisson numbers of reaction events and update the system accordingly (algorithm 2).

The problem with the above method is that of choosing an appropriate time step Δt so that the method is fast but reasonably accurate. Clearly the smaller Δt, the more accurate, and the larger Δt, the faster. Another problem is that although one particular Δt may be good enough for one part of a simulation, it may not be appropriate for another. This motivates the idea of stepping ahead a variable amount of time τ, based on c and the current state of the system, x. This is the idea behind Gillespie's τ-leap algorithm.

The τ-leap method is an adaptation of the Poisson time step method to allow stepping ahead in time by a variable amount τ, where at each time step τ is chosen in an appropriate way in order to try and ensure a sensible trade-off between accuracy and speed. This is achieved by making τ as large (and hence fast) as possible whilst still

Algorithm 2 Poisson Leap method

1. Set $t = 0$. Initialise the rate constants and the initial molecule numbers x.
2. Calculate $h_i(x, c_i)$, for $i = 1, \ldots, v$, and simulate the v-dimensional reaction vector r, with ith entry a $Po(h_i(x, c_i)\Delta t)$ random quantity.
3. Update the state according to $x := x + Sr$.
4. Update $t := t + \Delta t$.
5. Output t and x. If $t < T_{max}$ return to step 2.

satisfying some constraint designed to ensure accuracy. In this context, the accuracy is determined by the extent to which the assumption of constant hazard over the interval is appropriate. Clearly whenever any reaction occurs some of the reaction hazards change, and so an assessment needs to be made of the magnitude of change of the hazards $h_i(x, c_i)$. Essentially, the idea is to choose τ so that the (proportional) change in all of the $h_i(x, c_i)$ is small.

A *preleap* check is typically implemented as follows. We can calculate the expected new state as $x' = x + S\mathrm{E}(r)$, where the ith element of $\mathrm{E}(r)$ is just $h_i(x, c_i)\tau$. We can then calculate the change in hazard at this "expected" new state and see if this is acceptably small. It is suggested that the magnitude of acceptable change should be a fraction of the cumulative hazard $h_0(x, c)$, ie.

$$|h_i(x', c_i) - h_i(x, c_i)| \leq \epsilon h_0(x, c), \quad \forall i.$$

Gillespie provides an approximate method for calculating the largest τ satisfying this property [7]. Note that if the resulting τ is as small (or almost as small) as the expected time leap associated with an exact single reaction update, then it is preferable to do just that. Since the time to the next event is $Exp(h_0(x, c))$, which has expectation $1/h_0(x, c)$, one should prefer an exact update if the suggested τ is less than (say) $2/h_0(x, c)$. A number of refinements have been made to this basic scheme and are summarised in [16].

4.2 Chemical Langevin equation

We have considered an approximation to the continuous time, discrete state space Markov jump process by discretising time. It therefore seems natural to consider a continuous state space approximation, leading to the *chemical Langevin equation* (CLE). The CLE can be constructed in a number of more or less formal ways. In particular, it can be derived as a high concentration limit of the Markov jump process, but we will present here an informal intuitive construction, and then provide brief references to more rigorous approaches.

Consider an infinitesimal time interval, $(t, t + dt]$. Over this time, the reaction hazards will remain constant almost surely. As in the previous section, we can therefore regard

the occurrence of reaction events as the occurrence of events of a Poisson process with independent realisations for each reaction type. Therefore, if we write dR_t for the v-vector of the number of reaction events of each type in the time increment, it is clear that the elements are independent of one another and that the ith element is a $Po(h_i(X_t, c_i)dt)$ random quantity. From this we have that $\mathrm{E}(dR_t) = h(X_t, c)dt$ and $\mathrm{Var}(dR_t) = \mathrm{diag}\{h(X_t, c)\}dt$ and so we can write

$$dR_t = h(X_t, c)dt + \mathrm{diag}\left\{\sqrt{h(X_t, c)}\right\} dW_t\,.$$

This is the Itô stochastic differential equation (SDE) which has the same infinitesimal mean and variance as the true Markov jump process (where dW_t is the increment of a v-dimensional Brownian motion). Now since $dX_t = S dR_t$, we can immediately deduce

$$dX_t = Sh(X_t, c)dt + S\,\mathrm{diag}\left\{\sqrt{h(X_t, c)}\right\} dW_t \tag{3}$$

as a SDE for the time evolution of X_t. As written, this SDE is a little unconventional, as the driving Brownian motion is of a different (typically higher) dimension than the state. This is easily remedied by noting that

$$\mathrm{Var}(dX_t) = S\,\mathrm{diag}\{h(X_t, c)\}S'dt,$$

which immediately suggests the alternative form

$$dX_t = S\,h(X_t, c)dt + \sqrt{S\,\mathrm{diag}\{h(X_t, c)\}S'}\,dW_t, \tag{4}$$

where now X_t and W_t are both u-vectors. Equation (4) is the SDE most commonly referred to as the *chemical Langevin equation* (CLE), and represents the diffusion process which most closely matches the dynamics of the associated Markov jump process. In particular, whilst it relaxes the assumption of discrete states, it keeps all of the stochasticity associated with the discreteness of state in its noise term. It also preserves many of the important structural properties of the Markov jump process. For example, (4) has the same conservation laws as the original stochastic kinetic model.

More formal approaches to the construction of the CLE usually revolve around the Kolmogorov forward equation for the Markov process, given by (2). A second-order Taylor approximation to this system of differential equations can be constructed, and compared to the corresponding forward equation for an SDE model (known in this context as the *Fokker-Planck equation*). Matching the second-order approximation to the Fokker-Planck equation leads to the CLE (4), as presented above; see [8] and [9] for further details and [17] for a recent discussion.

4.2.1 Numerical solution

As for ODE models, simulation typically proceeds using an approximate numerical solution, since the SDE in (4) can rarely be solved analytically. To understand the simplest such scheme, consider an arbitrary d-dimensional diffusion process satisfying

$$dX_t = \mu(X_t)dt + \sigma(X_t)dW_t$$

Algorithm 3 CLE method

1. Set $t = 0$. Initialise the rate constants and the initial molecule numbers x.
2. Calculate $h_i(x, c_i)$ and simulate the v-dimensional increment ΔW_t, with ith entry a $\mathrm{N}(0, \Delta t)$ random quantity.
3. Update the state according to
$$x := x + S\,h(x, c)\Delta t + S\,\mathrm{diag}\left\{\sqrt{h(X_t, c)}\right\}\,\Delta W_t$$
4. Update $t := t + \Delta t$
5. Output t and x. If $t < T_{max}$ return to step 2.

where $\mu(\cdot)$ is a d-vector known as the *drift* and $\sigma^2(\cdot) = \sigma(\cdot)\sigma(\cdot)'$ is a $d \times d$ matrix known as the *diffusion coefficient*. For small time steps Δt, the increments of the process can be well approximated by using the *Euler-Maruyama* discretisation

$$X_{t+\Delta t} - X_t \equiv \Delta X_t = \mu(X_t)\Delta t + \sigma(X_t)\Delta W_t\,,$$

where $\Delta W_t \sim \mathrm{N}(0, I\Delta t)$ and I is the $d \times d$ identity matrix. A system at time t can therefore be stepped to $t + \Delta t$ via

$$(X_{t+\Delta t}|X_t = x) \sim \mathrm{N}\left(x + \mu(x)\Delta t\,,\, \sigma(x)\sigma(x)'\Delta t\right).$$

Algorithm 3 describes the procedure for numerically integrating the CLE. Note that for simplicity, we use the form of the CLE given in (3).

As for ODEs, higher order numerical methods (such as the Milstein scheme) can be implemented for SDEs but are less widely used due to the complexity of the implementation [18].

4.3 Linear noise approximation

The linear noise approximation (LNA) generally possesses a greater degree of numerical and analytic tractability than the CLE. For example, the LNA solution involves (numerically) integrating a set of ODEs for which standard routines, such as the `lsoda` package [19], exist. Our brief derivation follows the approach of [20] to which we refer the reader for further details.

We begin by replacing the hazard function $h(X_t, c)$ in equation (4) with the rescaled form $\Omega f(X_t/\Omega, c)$ where Ω is the volume of the container in which the reactions are taking place. Note that the LNA approximates the CLE increasingly well as Ω and X_t become large, that is, as the system approaches its thermodynamic limit. The CLE then becomes

$$dX_t = \Omega S f(X_t/\Omega, c)dt + \sqrt{\Omega S \mathrm{diag}\{f(X_t/\Omega, c)\}S'}dW_t. \tag{5}$$

Algorithm 4 LNA method 1

1. Set $t = 0$. Initialise the rate constants and the initial molecule numbers x. Set $z_0 = x/\Omega$, $m_0 = (x - \Omega z_0)/\sqrt{\Omega}$ (i.e. a vector of zeros) and V_0 as the $u \times u$ matrix, with all entries equal to zero.
2. Numerically integrate the system of ODEs satisfied by z_t, m_t and V_t over $(t, t + \Delta t]$.
3. Update the state by drawing x from a $\mathrm{N}\left(\Omega z_{t+\Delta t} + \sqrt{\Omega} m_{t+\Delta t}\,,\, \Omega V_{t+\Delta t}\right)$ distribution.
4. Update $t := t + \Delta t$. Set $m_t = (x - \Omega z_t)/\sqrt{\Omega}$ and V_t as the $u \times u$ matrix, with all entries equal to zero.
5. Output t and x. If $t < T_{max}$ return to step 2.

We then obtain the LNA by writing the solution X_t of the CLE as a deterministic process plus a residual stochastic process [10],

$$X_t = \Omega z_t + \sqrt{\Omega} M_t. \tag{6}$$

Substituting into equation (5) gives

$$dz_t + \frac{1}{\sqrt{\Omega}} dM_t = S\, f(z_t + M_t/\sqrt{\Omega}, c)dt + \frac{1}{\sqrt{\Omega}} \sqrt{S \mathrm{diag}\{f(z_t + M_t/\sqrt{\Omega}, c)\} S'} dW_t. \tag{7}$$

We then Taylor expand the rate function around z_t to give

$$f(z_t + M_t/\sqrt{\Omega}, c) = f(z_t, c) + \frac{1}{\sqrt{\Omega}} F_t M_t + O(\Omega^{-1}) \tag{8}$$

where F_t is the $v \times u$ Jacobian matrix with (i, j)th element $\partial f_i(z_t, c)/\partial Z_{j,t}$ and we suppress the dependence of F_t on z_t and c for simplicity. Substituting equation (8) into (7) and collecting terms of $O(1)$ gives the MRE

$$\frac{dz_t}{dt} = S\, f(z_t, c). \tag{9}$$

Collecting terms of $O(1/\sqrt{\Omega})$ gives the SDE satisfied by the residual process as

$$dM_t = S\, F_t M_t dt + \sqrt{S \mathrm{diag}\{f(z_t, c)\} S'} dW_t. \tag{10}$$

Equations (6), (9) and (10) give the linear noise approximation of the CLE and therefore of the Markov jump process model.

Algorithm 5 LNA method 2

1. Set $t = 0$. Initialise the rate constants and the initial molecule numbers x. Set $z_0 = x/\Omega$, and V_0 as the $u \times u$ matrix, with all entries equal to zero.
2. Numerically integrate the system of ODEs satisfied by z_t and V_t over $(t, t + \Delta t]$.
3. Update the state by drawing x from a $\mathrm{N}\left(\Omega z_{t+\Delta t}\,,\, \Omega V_{t+\Delta t}\right)$ distribution.
4. Update $t := t + \Delta t$. Set $z_t = x/\Omega$ and V_t as the $u \times u$ matrix, with all entries equal to zero.
5. Output t and x. If $t < T_{max}$ return to step 2.

4.3.1 Solution of the linear noise approximation

For fixed or Gaussian initial conditions, that is $M_{t_1} \sim \mathrm{N}(m_{t_0}, V_{t_0})$, the SDE in (10) can be solved explicitly to give

$$(M_t|c) \sim \mathrm{N}\left(m_t\,,\, V_t\right)$$

where m_t is the solution to the deterministic ordinary differential equation (ODE)

$$\frac{dm_t}{dt} = S\,F_t m_t$$

and similarly

$$\frac{dV_t}{dt} = V_t F_t' S' + S\mathrm{diag}\{h(z_t)\}S' + S\,F_t V_t\,.$$

Note that we have dropped the dependence of both m_t and V_t on z_t and c to simplify the notation. Hence, the solution of the SDE in equation (10) requires the solution of a system of coupled ODEs; in the absence of an analytic solution to these equations, a numerical solver such as that described in [19] can be used. The approximating distribution of X_t can then be found as

$$X_t \sim \mathrm{N}\left(\Omega z_t + \sqrt{\Omega} m_t\,,\, \Omega V_t\right).$$

A realisation of X_t can then be obtained at discrete times via algorithm 4. With this approach, the ODE satisfied by z_t is essentially numerically integrated over the entire time horizon of interest. Hence, the accuracy of the LNA applied in this way (relative to the MJP) can become quite poor due to the difference between z_t and the true stochastic solution. An approach advocated by [21] to alleviate this problem is to restart z_t at each simulation time using the value of x_t. Hence, the system of ODEs satisfied by z_t and V_t are (numerically) solved over each interval $[t, t + \Delta t]$ with $z_t = x_t$ and V_t as a $u \times u$ matrix, with all entries equal to zero. Note that m_t is zero for all t and therefore the ODE satisfied by m_t need not be solved. Full details can be found in algorithm 5. Further discussion of the LNA including details of the LNA solution can be found in [10, 22, 23, 20].

Algorithm 6 Generic hybrid algorithm

1. Set $t = 0$. Initialise the rate constants and the initial molecule numbers x.
2. Classify reactions as fast or slow based on x.
3. Update fast reaction dynamics over $(t, t + \Delta t]$.
4. Based on the fast reaction events over $(t, t + \Delta t]$, determine if a slow reaction has occurred.
5. If no slow reactions have occurred, update x based on the fast reactions only. Set $t := t + \Delta t$ and goto step 7.
6. If (at least) one slow reaction has occurred, identify the time τ and type of the first slow reaction and update the state x to time τ. Set $t := \tau$.
7. Output t and x. If $t < T_{max}$ return to step 2.

4.4 Hybrid simulation strategies

Whilst the CLE and LNA approaches represent a computationally efficient alternative to exact simulation approaches such as the Gillespie algorithm, biochemical reactions describing processes such as gene regulation can involve very low concentrations of reactants [24] and ignoring the inherent discreteness in low copy number data traces is clearly unsatisfactory. The aim of a hybrid simulation strategy is to exploit the computational efficiency of methods such as the CLE and LNA whilst accurately describing the dynamics of low copy number species, thereby bridging the gap between exact and approximate algorithms. Hybrid simulation strategies for discrete-continuous stochastic kinetic models are reasonably well developed and involve partitioning reactions as fast or slow based on the likely number of occurrences of each reaction over a given time interval and the effect of each reaction on the number of reactants and products. Fast reaction events are then modelled as continuous (using for example the CLE) and the remaining slow reaction events are updated with an exact procedure. A generic hybrid procedure is given in algorithm 6.

The CLE is used by [25] to model fast reaction dynamics whilst modelling slow reaction events with a Markov jump process. Since the slow reaction hazards will necessarily be time-dependent, the time-dependent probability density of the "next reaction" algorithm is used compute the times of the slow reaction events. Discrete/CLE simulation strategies in the context of a simple gene regulatory system have been considered by [26] whilst [27] and [28] consider discrete/ODE approaches.

Label	Reaction	Hazard	Description
R_1	$\mathcal{X}_1 \xrightarrow{c_1} 2\mathcal{X}_1$	$c_1 X_1$	Prey reproduction
R_2	$\mathcal{X}_1 + \mathcal{X}_2 \xrightarrow{c_2} 2\mathcal{X}_2$	$c_2 X_1 X_2$	Prey death, predator reproduction
R_3	$\mathcal{X}_2 \xrightarrow{c_3} \emptyset$	$c_3 X_2$	Predator death

Table 2: Reaction list and hazards for the Lotka-Volterra system.

5 Example: Lotka-Volterra

As an example, we consider a Lotka-Volterra model of predator and prey interaction consisting of 3 reactions and 2 species, developed by [29] and [30]. The reaction list is given in Table 2. Although strictly speaking, $\mathcal{X}_1$ and $\mathcal{X}_2$ represent animal species, they could equally well be chemical species. In addition, the system is sufficiently complex to explore the auto-regulatory behaviour that is typical of many biochemical network models.

We aim to investigate the system dynamics through stochastic simulation. We therefore require key ingredients such as the stoichiometry matrix, which is

$$S = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}$$

and the vector of hazards, given by

$$h(X_t, c) = (c_1 X_{1,t}, c_2 X_{1,t} X_{2,t}, c_3 X_{2,t})' .$$

The chemical Langevin equation is characterised by the drift and diffusion functions of the SDE in equation (4). We obtain

$$S\, h(X_t, c) = \begin{pmatrix} c_1 X_{1,t} - c_2 X_{1,t} X_{2,t} \\ c_2 X_{1,t} X_{2,t} - c_3 X_{2,t} \end{pmatrix}$$

and

$$S \,\mathrm{diag}\{h(X_t, c)\} S' = \begin{pmatrix} c_1 X_{1,t} + c_2 X_{1,t} X_{2,t} & -c_2 X_{1,t} X_{2,t} \\ -c_2 X_{1,t} X_{2,t} & c_2 X_{1,t} X_{2,t} + c_3 X_{2,t} \end{pmatrix}.$$

To compute the linear noise approximation, we require $f(z_t, c)$ and the Jacobian matrix F_t. For simplicity, we take a fixed volume of $\Omega = 1$ (and note that for $\Omega \neq 1$, the hazard of R_2 should be $c_2 \Omega^{-1} X_1 X_2$ to scale appropriately with volume). We therefore obtain $f(z_t, c) = h(z_t, c)$ and

$$F_t = \begin{pmatrix} c_1 & 0 \\ c_2 z_{2,t} & c_2 z_{1,t} \\ 0 & c_3 \end{pmatrix}.$$

All simulations used initial conditions of $x_0 = (100, 100)'$ and rate constants $c = (0.5, 0.0025, 0.3)'$ as used in [31].

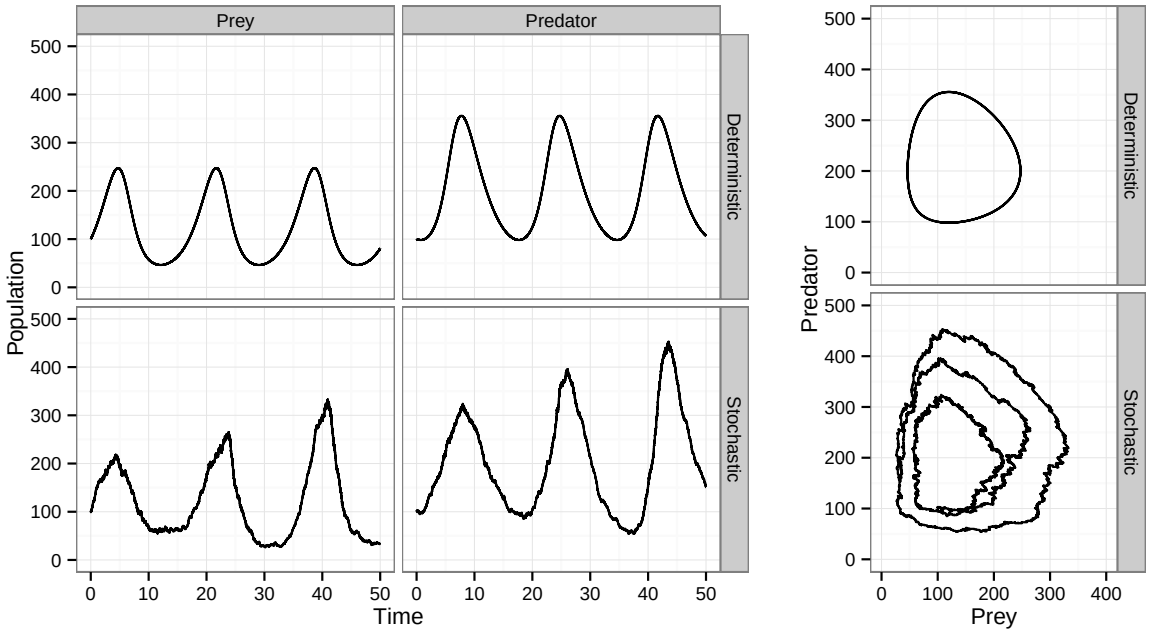

Figure 1: A single stochastic realisation of the Lotka-Volterra system using Gillespie's direct method and the deterministic solution.

Figure 1 shows a single stochastic realisation of the Lotka-Volterra system generated by Gillespie's direct method. For comparison, the deterministic MRE solution is shown. Note that with the stochastic solution, predator levels will eventually reach zero and the predator population will become extinct. The MRE solution on the other hand, is a perfectly repeating oscillation, carrying on indefinitely. It should be clear that for this system, the stochastic mean and the deterministic solution do not coincide.

Figure 2 shows the median, inter quartile range, upper and lower 2.5 percentiles for the prey population, using Gillespie's direct method, the CLE and both LNA approaches. The difference between the two LNA approaches is clear. Application of the LNA driven by a deterministic solution over the whole time-course of interest leads to a mismatch between the LNA solution and MJP solution. Restarting the deterministic solution at each simulation time, at the simulated value, alleviates this problem.

6 Discussion

Stochastic chemical kinetic theory provides a framework for model building that leads to a Markov jump process model from a simple list of biochemical reactions. Gillespie's direct method provides a straightforward way of simulating such processes on a computer. The algorithm can potentially be computationally intensive and therefore techniques that aim to reduce this cost (such as those considered in Section 3.2) can be of benefit. For systems involving many reaction channels and species, the computational cost of an

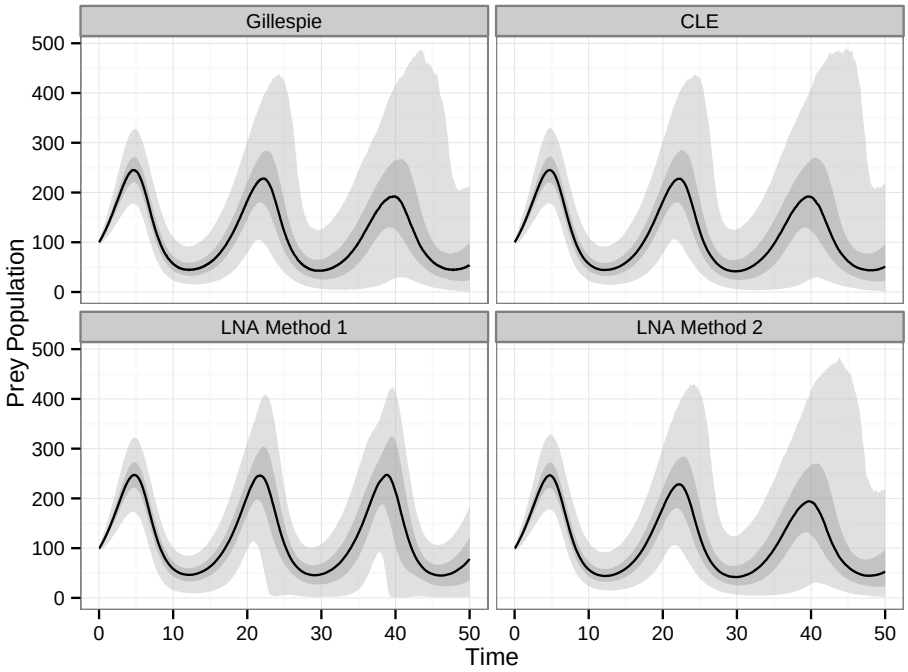

Figure 2: Median (solid), inter-quartile range (inner shaded region), upper and lower 2.5 percentiles (outer shaded region) based on 10^4 stochastic realisations of the Lotka-Volterra system using (a) Gillespie's direct method, (b) the CLE with $\Delta t = 0.01$, (c) the LNA (algorithm 4) and (d) the LNA (algorithm 5).

efficient implementation of the Gillespie algorithm may still preclude statistical analysis. The importance of approximate algorithms such as the CLE and LNA is then clear.

The stochastic simulation methods examined here are by no means exhaustive and indeed, there is a vast literature in this area. For example, we can derive moment equations from the chemical master equation to obtain fast approximations to the stochastic mean and variance of the system [32, 33, 34]. Alternatively, we can take advantage of multi-core processors; models can be partitioned into smaller sub-systems and simulated independently [35].

Computing details

All simulations were performed on a machine with 4 GB of RAM and with an Intel quad-core CPU. The simulation code for the Lotka-Volterra model was written in R [36]. The graphics were created using the ggplot2 R package [37]. The R code used in this chapter can be downloaded from the github repository:

https://github.com/csgillespie/In-silico-Systems-Biology

It was worth noting that using R is useful when developing algorithms that have relatively simple behaviour, for larger models, this method does not scale well. Instead, simulators

written in compiled languages, such as C/C++ or Java are preferred; particularly if they can input/export SBML.

References

[1] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[2] B. Finkenstadt, E. Heron, M. Komorowski, K. Edwards, S. Tang, C. Harper, J. Davis, M. White, A. Millar, and D. Rand. Reconstruction of transcriptional dynamics from gene reporter data using differential equations. *Bioinformatics*, 24(24):2901, 2008.

[3] B. Calderhead and M. Girolami. Statistical analysis of nonlinear dynamical systems using geometric sampling methods. *Interface Focus*, 1(6):821–835, 2011.

[4] D. J. Wilkinson. Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature Reviews Genetics*, 10:122–133, 2009.

[5] P. S. Swain, M. B. Elowitz, and E.D. Siggia. Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proceedings of the National Academy of Sciences*, 99(20):12795–12800, 2002.

[6] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81:2340–2361, 1977.

[7] D. T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, 115(4):1716–1732, 2001.

[8] D. T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188:404–425, 1992.

[9] D. T. Gillespie. The chemical Langevin equation. *Journal of Chemical Physics*, 113(1):297–306, 2000.

[10] N. G. van Kampen. *Stochastic Processes in Physics and Chemistry*. North-Holland, 2001.

[11] D. A. McQuarrie. Stochastic approach to chemical kinetics. *Journal of Applied Probability*, 4:413–478, 1967.

[12] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22:403–434, 1976.

[13] M. A. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *Journal of Physical Chemistry A*, 104(9):1876–1889, 2000.

[14] Y. Cao, H. Li, and L. Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting system. *Journal of Chemical Physics*, 121(9):4059–4067, 2004.

[15] J. M. McCollum, G. D. Peterson, M. L. Simpson, and N. F. Samatova. The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Journal of Computational Biology and Chemistry*, 30(1):39–49, 2006.

[16] W Sandmann. Streamlined formulation of adaptive explicit-implicit tau-leaping with automatic tau selection. In *Winter Simulation Conference (WSC), Proceedings of the 2009*, pages 1104–1112. IEEE, 2009.

[17] A. Golightly and D. J. Wilkinson. Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface Focus*, 1(6):807–820, 2011.

[18] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer Verlag, New York, 1992.

[19] L. Petzold. Automatic selection of methods for solving stiff and non-stiff systems of ordinary differential equations. *SIAM Journal on Scientific and Statistical Computing*, 4(1):136–148, 1983.

[20] D. J. Wilkinson. *Stochastic Modelling for Systems Biology*. Chapman & Hall/CRC Press, Boca Raton, Florida, 2nd edition, 2012.

[21] P. Fearnhead, V. Giagos, and C. Sherlock. Inference for reaction networks using the Linear Noise Approximation. Available from http://arxiv.org/pdf/1205.6920, 2012.

[22] J. Elf and M. Ehrenberg. Fast evaluation of fluctuations in biochemical networks with a linear noise approximation. *Genome Research*, 13(11):2475–2484, 2003.

[23] M. Komorowski, B. Finkenstadt, C. Harper, and D. Rand. Bayesian inference of biochemical kinetic parameters using the linear noise approximation. *BMC Bioinformatics*, 10(1):343, 2009.

[24] P. Guptasarma. Does replication-induced transcription regulate synthesis of the myriad low copy number proteins of *Escherichia coli*? *BioEssays*, 17:987–997, 1995.

[25] H. Salis and Y. Kaznessis. Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. *Journal of Chemical Physics*, 122:054103, 2005.

[26] D. Higham, S. Intep, X. Mao, and L. Szpruch. Hybrid simulation of autoregulation within transcription and translation. *BIT Numerical Mathematics*, (51):177–196, 2011.

[27] T. R. Kiehl, R. M. Matteyses, and M. K. Simmons. Hybrid simulation of cellular behavior. *Bioinformatics*, 20(3):316–322, 2004.

[28] A. Alfonsi, E. Cances, G. Turinici, B. Ventura, and W. Huisinga. Adaptive simulation of hybrid stochastic and deterministic models for biochemical systems. *ESAIM: Proceedings*, 14:1–13, 2005.

[29] A. J. Lotka. *Elements of Physical Biology*. Williams and Wilkens, Baltimore, 1925.

[30] V. Volterra. Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558–60, 1926.

[31] R. J. Boys, D. J. Wilkinson, and T. B. L. Kirkwood. Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing*, 18:125–135, 2008.

[32] C S Gillespie. Moment-closure approximations for mass-action models. *IET Systems Biology*, 3(1):52–8, 2009.

[33] C A Gómez-Uribe and G C Verghese. Mass fluctuation kinetics: capturing stochastic effects in systems of chemical reactions through coupled mean-variance computations. *The Journal of Chemical Physics*, 126(2):024109, 2007.

[34] I Krishnarajah, A R Cook, G Marion, and G Gibson. Novel moment closure approximations in stochastic epidemics. *Bulletin of Mathematical Biology*, 67(4):855–873, July 2005.

[35] C S Gillespie. Stochastic simulation of chemically reacting systems using multi-core processors. *The Journal of Chemical Physics*, 136(1):014101, 2012.

[36] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[37] Hadley Wickham. *ggplot2: elegant graphics for data analysis*. Springer New York, 2009.