Stochastic Simulation: Lecture 11

Christoph Reisinger

Oxford University Mathematical Institute

Modified from earlier slides by Prof. Mike Giles.

Continuous-time Markov Processes

In this two lecture we will consider the approximation of Continuous-time Markov Processes.

Probably the most important class of applications for these is in the stochastic modelling of chemical reactions in solution, so this is the context we will start with.

Chemical reaction:

$$A + B \longrightarrow C$$

Classical deterministic modelling in a "well-stirred" vessel gives a set of ODEs for the concentrations c_A, c_B, c_C:

$$\begin{aligned} \dot{c}_A &= -\kappa\, c_A\, c_B \\ \dot{c}_B &= -\kappa\, c_A\, c_B \\ \dot{c}_C &= +\kappa\, c_A\, c_B \end{aligned}$$

Continuous-time Markov Processes

This works well when there are lots of molecules of A and B in the solution, but there are applications (particularly in bio-chemistry) when there are very few, and then things become stochastic.

Let X_A, X_B, X_C be number of molecules of A, B, C in some well-mixed container. Reactions require a molecule of A to "bump into" a molecule of B and react, so

$$\mathbb{P}(\text{reaction in time interval } \mathrm{d}t) = \kappa \, X_A \, X_B \, \mathrm{d}t$$

and when a reaction happens

$$\begin{aligned} X_A &\to X_A - 1 \\ X_B &\to X_B - 1 \\ X_C &\to X_C + 1 \end{aligned}$$

Unit rate Poisson Process

A unit rate Poisson process $Y(\tau)$ is a continuous-time random counting process in which

- there is a set of increasing jump times $0 < \tau_1 < \tau_2 < \tau_3 < \ldots$
- setting $\tau_0 = 0$, then $Y(\tau) = j,$ for $\tau \in [\tau_j, \tau_{j+1}),\ j = 0, 1, \ldots$
- the jump intervals $\tau_{j+1} - \tau_j$ are i.i.d. exponential random variables, so for $t > 0$

$$\mathbb{P}(\tau_{j+1} - \tau_j > \tau) = \exp(-\tau)$$

and

$$\mathbb{P}(\tau_{j+1} < \tau + \mathrm{d}t \mid \tau_{j+1} > \tau > \tau_j) = \mathrm{d}t$$

Note: for any time $\tau > 0$, $Y(\tau)$ is a Poisson random variable with mean τ.

Continuous-time Markov Processes

Using a unit rate Poisson process to represent the number of reactions which have taken place we have

$$X(t) = X(0) + R(t) \begin{pmatrix} -1 \\ -1 \\ +1 \end{pmatrix}$$

where

$$R(t) = Y\left(\int_0^t \kappa X_A(s) X_B(s)\,\mathrm{d}s\right)$$

so the probability of a reaction in time interval $(t, t+\mathrm{d}t)$ is $\kappa X_A(t) X_B(t)\,\mathrm{d}t$.

Continuous-time Markov Processes

Generalising this, suppose we have d species, and multiple reactions, with the k-th reaction having an intensity function $\lambda_k(t)$ and with each such reaction changing the count of X_i by ζ_{ki}.

Then with independent unit rate processes for each reaction we have

$$X_i(t) = X_i(0) + \sum_k R_k(t)\ \zeta_{ki}$$

where we have the time-change representation

$$R_k(t) = Y_k\left(\int_0^t \lambda_k(s)\,\mathrm{d}s\right)$$

and for the most common law of *mass action kinetics*

$$\lambda_k(t) = \kappa_k \prod_{i=1}^{d} \frac{X_i!}{(X_i - \nu_{ki})!}\mathbf{1}_{\{X_i \geq \nu_{ki}\}}$$

when there are ν_{ki} inputs of species i in reaction k.

Continuous-time Markov Processes

Example from a paper by Anderson and Higham (2012)

$$S_1 \underset{\kappa_2}{\overset{\kappa_1}{\rightleftarrows}} S_2, \quad 2S_2 \xrightarrow{\kappa_3} S_3,$$

then

$$\zeta_1 = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}, \quad \zeta_2 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}, \quad \zeta_3 = \begin{pmatrix} 0 \\ -2 \\ 1 \end{pmatrix},$$

and

$$\lambda_1 = \kappa_1 X_1, \quad \lambda_2 = \kappa_2 X_2, \quad \lambda_3 = \kappa_3 X_2 (X_2 - 1).$$

Stochastic Simulation Algorithm (SSA)

SSA is an exact simulation algorithm originally due to Gillespie (1976, 1977). There have been a number of variants published since – here I first describe his original "Direct Method".

Key idea: if we define $\lambda = \sum_k \lambda_k$, then

$$\begin{aligned} \mathbb{P}(\text{reaction } k \text{ occurs in next } \mathrm{d}t) &= \lambda_k \, \mathrm{d}t \\ \mathbb{P}(\text{some reaction occurs in next } \mathrm{d}t) &= \lambda \, \mathrm{d}t \\ \mathbb{P}(\text{next reaction is reaction } k) &= \lambda_k / \lambda \end{aligned}$$

Stochastic Simulation Algorithm (SSA)

Input: initial X, final time T
$t := 0$

while $t < T$ **do**
  compute λ_k and $\lambda := \sum_k \lambda_k$
  generate two uniform r.v.'s U_1, U_2
  next reaction time $t := t - \log(U_1)/\lambda$
  if $t < T$ **then**
    identify reaction k' s.t.

$$\sum_{k<k'} \frac{\lambda_k}{\lambda} < U_2 \leq \sum_{k \leq k'} \frac{\lambda_k}{\lambda}$$

    $X := X + \zeta_{k'}$
  end if
end while

Stochastic Simulation Algorithm (SSA)

Major issue: the cost is proportional to the total number of reactions that take place – could be millions.

This will be addressed by tau-leaping approximation, and MLMC.

Minor issues:

- for each reaction, the Direct Method requires 2 random numbers
- 2 key steps have costs proportional to the number of possible reactions.

The first of these is addressed by Gillespie's Next Reaction Method, and the second was addressed by Gibson & Bruck (2000).

Stochastic Simulation Algorithm (SSA)

Reaction k has the unit rate Poisson process

$$Y_k\left(\int_0^t \lambda_k(s)\,\mathrm{d}s\right)$$

with real jump times $t_1, t_2, \ldots$ and pseudo-times $\tau_1, \tau_2, \ldots$ where

$$\int_0^{t_n} \lambda_k(s)\,\mathrm{d}s = \tau_n \quad \Longrightarrow \quad \int_{t_n}^{t_{n+1}} \lambda_k(s)\,\mathrm{d}s = \tau_{n+1} - \tau_n$$

and $\tau_{n+1} - \tau_n = -\log U$ where U is an $(0, 1)$ uniform r.v.

Putting

$$T_k(t) = \tau_{n+1} - \tau_n - \int_{t_n}^{t} \lambda_k(s)\,\mathrm{d}s$$

means that reaction k occurs when T_k reaches 0.

Stochastic Simulation Algorithm (SSA)

Input: initial X, timers $T_k = -\log(U_k)$, final time T
$t := 0$

loop
  compute λ_k
  set $\Delta t = \min_k(T_k/\lambda_k)$, $k' = \text{argmin}_k\,(T_k/\lambda_k)$,
  $t := t + \Delta t$
  if $t > T$, **stop**
  $X := X + \zeta_{k'}$
  $T_{k'} = -\log U$
  for all $k \neq k'$ **do**
    $T_k := T_k - \lambda_k\,\Delta t$
  end for
end loop

Tau-leaping method

SSA is used extensively but it can be very costly – some simulations may involve millions of individual reactions, and may need to perform up to a million such calculations.

The tau-leaping method is an approximate simulation method.

The Euler–Maruyama SDE approximation treats the drift and diffusion values as constant within a timestep, and only updates them at the end of the timestep.

Tau-leaping adopts the same idea, updating the λ_k only at the beginning/end of each timestep.

Within a timestep of size h, λ_k is fixed so the number of reactions of type k is $P(\lambda_k h)$ where $P(\mu)$ is a Poisson r.v. with mean μ.

Tau-leaping algorithm

Input: timestep h, initial state $\widehat{X}$, final time $T = N\,h$

for $n = 1, N$ **do**

  $\Delta\widehat{X} := 0$

  for each k **do**

    compute $\lambda_k(\widehat{X})$

    generate Poisson r.v.'s $R_k = Poiss(\lambda_k h)$

    $\Delta\widehat{X} := \Delta\widehat{X} + R_k\zeta_k$

  end for

  $\widehat{X} := \widehat{X} + \Delta\widehat{X}$

end for

Output: $f(\widehat{X})$

Tau-leaping method

The cost is $O(T/h)$, but there is now a discretisation error so that for $h \gg 1/\lambda$

$$\mathbb{E}[\, f(X_T) - f(\widehat{X}_T)\,] = O(h).$$

In next lecture will use MLMC to eliminate this error and also reduce the total cost.

Also, Poisson r.v.'s have an unbounded size, so there is a small but finite probability of ending up with negative population counts.

Set reaction rate to zero if negative count of one of the inputs.

Tau-leaping method

For large mean μ, the Poisson distribution is close to the Normal with mean μ and variance μ, rounded to the nearest integer.

This means that we approximately have

$$\widehat{X}_{n+1} = \widehat{X}_n + \sum_k \left(\lambda_k(\widehat{X}_n) h + \sqrt{\lambda_k(\widehat{X}_n)} \sqrt{h}\, Z_{kn} \right) \zeta_k$$

where Z_{kn} are i.i.d. unit Normals random variables.

This corresponds to the Euler–Maruyama discretisation of the *chemical Langevin* SDE approximation

$$\mathrm{d}X = \sum_k \left(\lambda_k(X)\, \mathrm{d}t + \sqrt{\lambda_k(X)}\, \mathrm{d}W_k \right) \zeta_k$$

The fact that MLMC would be very effective for this SDE suggests it might also be useful for tau-leaping.

Approximation hierarchy

Thus chemical kinetics can be modelled at 4 different levels:

- SSA – exact simulation of each and every reaction
- tau-leaping – regular updating of the propensity functions
- Langevin SDE – replacing Poisson distribution by Normal approximation
- ODEs – ignoring stochastic effects entirely

These involve a balance between cost and accuracy, but ideally we would like to achieve both low cost and high accuracy.

Towards multilevel simulation

Recall:

The SSA algorithm (and other equivalent methods) computes each reaction one by one – exact but very costly

"Tau-leaping" is equivalent to the Euler–Maruyama method for SDEs – the rates λ_k are frozen at the start of the timestep, so for each timestep of size h just need a sample from a Poisson distribution $Poiss(\lambda_k\, h)$ to obtain the number of reactions in that timestep.

i.e. for piecewise constant $\lambda(s)$,

$$Y\left(\int_0^{(n+1)\,h} \lambda(s)\,\mathrm{d}s\right) - Y\left(\int_0^{n\,h} \lambda(s)\,\mathrm{d}s\right) \sim Poiss(\lambda\, h)$$

Multilevel simulation

Anderson & Higham (2012) developed (and analysed) a very elegant and efficient multilevel version of this algorithm – big savings because finest level usually has 1000's of timesteps.

Key challenge: how to couple coarse and fine path simulations?

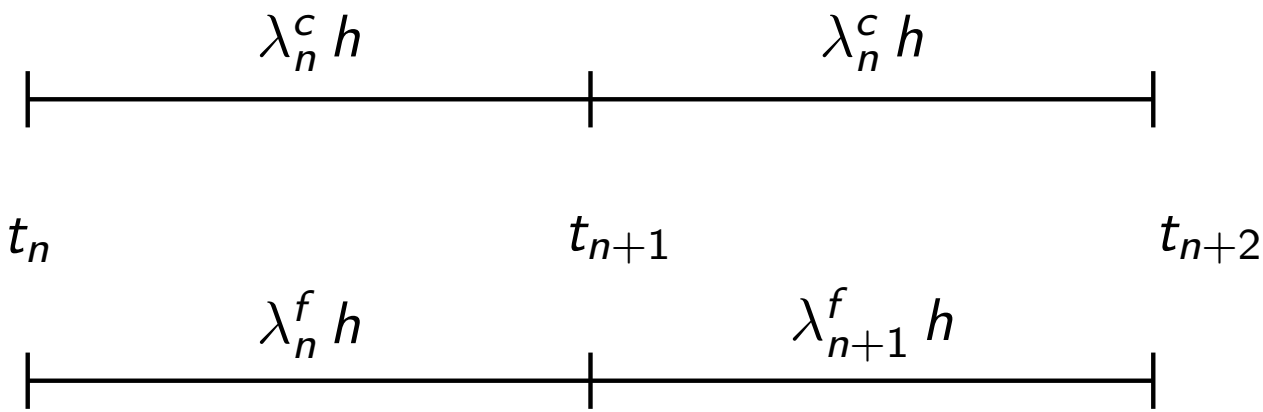
Crucial observation: for $t_1, t_2 \geq 0$

$$Poiss(t_1) + Poiss(t_2) \stackrel{d}{=} Poiss(t_1 + t_2)$$

Multilevel coupling

Solution (for uniform timesteps with refinement factor of 2)

- simulate the Poisson variable on the coarse timestep as the sum of two fine timestep Poisson variables
- couple the fine path and coarse path Poisson variables by using common variable based on smaller of two rates



If $\lambda_n^f < \lambda_n^c$, use $Poiss(\lambda_n^c\, h) \sim Poiss(\lambda_n^f\, h) + Poiss((\lambda_n^c - \lambda_n^f)\, h)$
If $\lambda_n^c < \lambda_n^f$, use $Poiss(\lambda_n^f\, h) \sim Poiss(\lambda_n^c\, h) + Poiss((\lambda_n^f - \lambda_n^c)\, h)$

Tau-leaping MLMC algorithm

Input: fine timestep h, final time $T = N\,h$, refinement factor M, initial states $\widehat{X}^f = \widehat{X}^c = X$

for $n = 1, N$ **do**
  for each k **do**
    compute λ_k^f, and also λ_k^c **if** $\mathrm{mod}(n-1, M) = 0$
    $R_{1,k} := Poiss(\min(\lambda_k^f, \lambda_k^c)h)$
    $R_{2,k} := Poiss(|\lambda_k^f - \lambda_k^c|h)$
    $\widehat{X}^f := \widehat{X}^f + (R_{1,k} + \mathbf{1}_{\lambda_k^f > \lambda_k^c} R_{2,k})\,\zeta_K$
    $\widehat{X}^c := \widehat{X}^c + (R_{1,k} + \mathbf{1}_{\lambda_k^c > \lambda_k^f} R_{2,k})\,\zeta_K$
  end for
end for

Numerical analysis

Anderson & Higham also analysed the variance and proved that

$$\mathbb{E}[\,\|\widehat{X}^f - \widehat{X}^c\|^2] = O(h).$$

Since the cost is $O(h^{-1})$ this is very similar to the Euler–Maruyama method applied to SDEs, and the overall complexity is $O(\varepsilon^{-2}|\log\varepsilon|^2)$ for ε RMS error (independent of the total number of reactions performed).

Extra bits

Once the timestep is reduced down to a size for which there are very few reactions per timestep, it makes sense to switch to SSA.

Anderson & Higham (2012) came up with a very nice way to couple the finest tau-leaping level to an SSA treatment, so the final algorithm is unbiased.

The key idea is the "coarse" path uses tau-leaping, and the "fine" path uses the exact updating of the rates λ, and each reaction k can be split into two reactions:

- one with rate $\min(\lambda^f, \lambda^c)$
- one with rate $|\lambda^f - \lambda^c|$

then use either Direct Method or Next Reaction Method for coupled simulation.

This leads to an $O(\varepsilon^{-2})$ complexity overall, with only a $(\log N)^2$ dependence on the number of reactions per path.

Extra bits

Model reduction: some biochemical reaction networks are very complex – can use a simpler approximate model (e.g. based on some forward-backward reactions being in equilibrium) as an additional "level"

Anderson & Higham (2012) also give an example of this.

Extra bits – adaptation

Adaptive time-stepping:

- Can be helpful to improve accuracy, especially when there is a fast initial transient.
- MLMC treatment essentially the same as for SDEs.

Adaptive treatment of reactions:

- some handled by SSA, some by tau-leaping, perhaps even some as Langevin SDEs
- This has been explored by Moraes *et al* (2016)

Extra bits – level 0 c.v.

Moraes *et al* (2016) also introduced an interesting control variate for the very coarsest tau-leaping level.

Start from

$$X(t) = X(0) + \sum_k Y_k \left(\int_0^t \lambda_k(X(s))\,\mathrm{d}s \right) \zeta_k,$$

replace Y_k by identity, since $\mathbb{E}[Y_k(s)] = s$, to get

$$Z(t) = X(0) + \sum_k \left(\int_0^t \lambda_k(Z(s))\,\mathrm{d}s \right) \zeta_k, \quad \Longrightarrow \dot{Z} = \sum_k \lambda_k \zeta_k$$

and then we have the approximation

$$\widetilde{X}(t) = X(0) + \sum_k Y_k \left(\int_0^t \lambda_k(Z(s))\,\mathrm{d}s \right) \zeta_k.$$

Extra bits – level 0 c.v.

Defining

$$K = \int_0^T \lambda_k(Z(s))\,\mathrm{d}s$$

then

$$\mathbb{E}[\widetilde{X}(T)] = X(0) + \sum_k K\,\zeta_k$$

and for any polynomial $f(X)$ can compute $\mathbb{E}[f(\widetilde{X}(T))]$.

$\widetilde{X}(T)$ can then be simulated using the same Y_k as the coarsest level tau-leaping $\widehat{X}$ simulation.

Key References

D.T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions", J. Comp. Phys., 22(4):403-434, 1976.

D.T. Gillespie, "Exact stochastic simulation of coupled chemical reactions", J. Phys. Chem. 81(25):2340-2361, 1977.

M.A. Gibson, J. Bruck "Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels", J. Phys. Chem., 104(9):1876-1889, 2000.

D.T. Gillespie, A. Ganguly, T.G. Kurtz, "Error analysis of tau-leaping simulation methods", Annals of Applied Probability, 21(6):2226-2262, 2011.

D.T. Gillespie, A. Hellander, L.R. Petzold, "Perspective: Stochastic algorithms for chemical kinetics", J. Chem. Phys., 138(17):170901, 2013.

Key references – multilevel

D.F. Anderson, D.J. Higham. "Multi-level Monte Carlo for continuous time Markov chains, with applications in biochemical kinetics". SIAM Multiscale Modelling and Simulation, 10(1):146-179, 2012.

D.F. Anderson, D.J. Higham, Y. Sun. "Complexity of multilevel Monte Carlo tau-leaping". SIAM Journal on Numerical Analysis, 52(6):3106-3127, 2014.

A. Moraes, R. Tempone, P. Vilanova. "A multilevel adaptive reaction-splitting simulation method for stochastic reaction networks". SIAM Journal on Scientific Computing, 38(4):A2091-A2117, 2016.