Computational methods for stochastic models of biochemical systems (and other population processes)

David F. Anderson*

*anderson@math.wisc.edu

Department of Mathematics

University of Wisconsin - Madison

Statistics seminar

December 7th, 2011

Stochastic Models of Biochemical Reaction Networks

- Most common stochastic models of biochemical reaction networks are continuous time Markov chains.
- Often called chemical master equation type models.

Common examples:

1. Gene regulatory networks.
2. Models of viral infection.
3. General cellular processes.
4. General population models (epidemic, predator-prey, etc.)

Stochastic Models of Biochemical Reaction Networks

- Most common stochastic models of biochemical reaction networks are continuous time Markov chains.
- Often called chemical master equation type models.

Common examples:

1. Gene regulatory networks.
2. Models of viral infection.
3. General cellular processes.
4. General population models (epidemic, predator-prey, etc.)

Path-wise simulation methods include:

Language in Biology	Language in Math
Gillespie's Algorithm	Sim. embedded DTMC
Next reaction method	Sim. random time change representation of Tom Kurtz
First reaction method	Sim. using exponential "alarm clocks"

Path-wise methods can approximate values such as

$$\mathbb{E}f(X(t))$$

For example,

1. Means.
2. Moments/variances.
3. Probabilities.

or compute sensitivities

$$\frac{d}{d\kappa}\mathbb{E}f(X^{\kappa}(t)).$$

Path-wise methods can approximate values such as

$$\mathbb{E}f(X(t))$$

For example,

1. Means.
2. Moments/variances.
3. Probabilities.

or compute sensitivities

$$\frac{d}{d\kappa}\mathbb{E}f(X^{\kappa}(t)).$$

Problem: solving using these algorithms can be computationally expensive.

First problem: joint with Des Higham

Our first problem: Approximate $\mathbb{E}f(X(T))$ to some desired tolerance, $\epsilon > 0$.

First problem: joint with Des Higham

Our first problem: Approximate $\mathbb{E}f(X(T))$ to some desired tolerance, $\epsilon > 0$.

Easy!

- Simulate the CTMC exactly,
- generate independent paths, $X_{[i]}(t)$, use the unbiased estimator

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} f(X_{[i]}(t)).$$

- stop when desired confidence interval is $\pm\,\epsilon$.

What is the computational cost?

Recall,

$$\mu_n = \frac{1}{n} \sum_{i=1}^{n} f(X_{[i]}(t)).$$

Thus,

$$\text{Var}(\mu_n) = O\left(\frac{1}{n}\right).$$

What is the computational cost?

Recall,

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} f(X_{[i]}(t)).$$

Thus,

$$\mathrm{Var}(\mu_n) = O\left(\frac{1}{n}\right).$$

So, if we want

$$\widehat{\sigma}_n = O(\epsilon),$$

we need

$$\frac{1}{\sqrt{n}} = O(\epsilon) \implies n = O(\epsilon^{-2}).$$

What is the computational cost?

Recall,

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} f(X_{[i]}(t)).$$

Thus,

$$\mathrm{Var}(\mu_n) = O\left(\frac{1}{n}\right).$$

So, if we want

$$\widehat{\sigma}_n = O(\epsilon),$$

we need

$$\frac{1}{\sqrt{n}} = O(\epsilon) \implies n = O(\epsilon^{-2}).$$

If $\overline{N}$ gives average cost (steps) of a path using exact algorithm:

$$\begin{aligned}\text{Total computational complexity} &= (\text{cost per path}) \times (\#\ \text{paths}) \\ &= O(\overline{N}\epsilon^{-2}).\end{aligned}$$

Can be bad if (i) $\overline{N}$ is large, or (ii) ϵ is small.

Benefits/drawbacks

Benefits:

1. Easy to implement.
2. Estimator

$$\mu_n = \frac{1}{n} \sum_{i=1}^{n} f(X_{[i]}(t))$$

   is unbiased.

Benefits/drawbacks

Benefits:

1. Easy to implement.
2. Estimator

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} f(X_{[i]}(t))$$

   is unbiased.

Drawbacks:

1. The cost of $O(\overline{N}\epsilon^{-2})$ could be prohibitively large.
2. For our models, we often have that $\overline{N}$ is very large.

We need to develop the model for better ideas....

Build up model: Random time change representation of Tom Kurtz

Consider the simple system

$$A + B \to C$$

where one molecule each of A and B is being converted to one of C.

Build up model: Random time change representation of Tom Kurtz

Consider the simple system

$$A + B \to C$$

where one molecule each of A and B is being converted to one of C.

Simple book-keeping: if $X(t) = (X_A(t), X_B(t), X_C(t))^T$ gives the state at time t,

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

where

- $R(t)$ is the # of times the reaction has occurred by time t, and
- $X(0)$ is the initial condition.

Build up model: Random time change representation of Tom Kurtz

Assuming intensity or propensity of reaction is

$$\kappa X_A(s) X_B(s),$$

Build up model: Random time change representation of Tom Kurtz

Assuming intensity or propensity of reaction is

$$\kappa X_A(s) X_B(s),$$

We can model

$$R(t) = Y\left(\int_0^t \kappa X_A(s) X_B(s) ds\right)$$

where Y is a unit-rate Poisson point process.

Build up model: Random time change representation of Tom Kurtz

Assuming intensity or propensity of reaction is

$$\kappa X_A(s) X_B(s),$$

We can model

$$R(t) = Y\left(\int_0^t \kappa X_A(s) X_B(s) ds\right)$$

where Y is a unit-rate Poisson point process.

Hence

$$\begin{pmatrix} X_A(t) \\ X_B(t) \\ X_C(t) \end{pmatrix} \equiv X(t) = X(0) + \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} Y\left(\int_0^t \kappa X_A(s) X_B(s) ds\right).$$

Build up model: Random time change representation of Tom Kurtz

- Now consider a network of reactions involving d chemical species, $S_1, \ldots, S_d$:

$$\sum_{i=1}^{d} \nu_{ik} S_i \longrightarrow \sum_{i=1}^{d} \nu'_{ik} S_i$$

Denote reaction vector as

$$\zeta_k = \nu'_k - \nu_k,$$

Build up model: Random time change representation of Tom Kurtz

- Now consider a network of reactions involving d chemical species, $S_1, \ldots, S_d$:

$$\sum_{i=1}^{d} \nu_{ik} S_i \longrightarrow \sum_{i=1}^{d} \nu'_{ik} S_i$$

  Denote reaction vector as

$$\zeta_k = \nu'_k - \nu_k,$$

- The intensity (or propensity) of kth reaction is $\lambda_k : \mathbb{Z}^d_{\geq 0} \to \mathbb{R}$.
- By analogy with before

$$X(t) = X(0) + \sum_k Y_k \left(\int_0^t \lambda_k(X(s)) ds \right) \zeta_k,$$

  Y_k are independent, unit-rate Poisson processes.

Back to our problem

Recall:

Benefits:

1. Easy to implement.
2. Estimator

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} f(X_{[i]}(t))$$

   is unbiased.

Drawbacks:

1. The cost of $O(\overline{N}\epsilon^{-2})$ could be prohibitively large.
2. For our models, we often have that $\overline{N}$ is very large.

Let's try an approximate scheme.

Tau-leaping: Euler's method

Explicit tau-leaping [1] or Euler's method, was first formulated by Dan Gillespie in this setting .

Tau-leaping is essentially an Euler approximation of $\int_0^t \lambda_k(X(s))ds$:

$$
\begin{aligned}
Z(h) &= Z(0) + \sum_k Y_k \left(\int_0^h \lambda_k(Z(s))\, ds \right) \zeta_k \\
&\approx Z(0) + \sum_k Y_k \Big(\lambda_k(Z(0))\, h \Big) \zeta_k \\
&\stackrel{d}{=} Z(0) + \sum_k \text{Poisson} \Big(\lambda_k(Z(0))\, h \Big) \zeta_k .
\end{aligned}
$$

[1] D. T. Gillespie, J. Chem. Phys., **115**, 1716 – 1733.

Euler's method

Path-wise representation for $Z(t)$ generated by Euler's method is

$$Z(t) = X(0) + \sum_k Y_k \left(\int_0^t \lambda_k(Z \circ \eta(s)) ds \right) \zeta_k,$$

where

$$\eta(s) = t_n \quad \text{if} \quad t_n \leq s < t_{n+1} = t_n + h$$

is a step function giving left endpoints of time discretization.

Return to approximating $\mathbb{E}f(X(T))$

Let Z_L denote an approximate processes generated with time discretization step of h_L. Let

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} f(Z_{L,[i]}(t)).$$

Return to approximating $\mathbb{E}f(X(T))$

Let Z_L denote an approximate processes generated with time discretization step of h_L. Let

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} f(Z_{L,[i]}(t)).$$

We note

$$\mathbb{E}f(X(t)) - \mu_n = \left[\mathbb{E}f(X(t)) - \mathbb{E}f(Z_L(t))\right] + \mathbb{E}f(Z_L(t)) - \mu_n$$

Return to approximating $\mathbb{E}f(X(T))$

Let Z_L denote an approximate processes generated with time discretization step of h_L. Let

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} f(Z_{L,[i]}(t)).$$

We note

$$\mathbb{E}f(X(t)) - \mu_n = \left[\mathbb{E}f(X(t)) - \mathbb{E}f(Z_L(t))\right] + \mathbb{E}f(Z_L(t)) - \mu_n$$

Suppose have an order one method

$$\mathbb{E}f(X(t)) - \mathbb{E}f(Z_L(t)) = O(h_L).$$

Return to approximating $\mathbb{E}f(X(T))$

Let Z_L denote an approximate processes generated with time discretization step of h_L. Let

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} f(Z_{L,[i]}(t)).$$

We note

$$\mathbb{E}f(X(t)) - \mu_n = \left[\mathbb{E}f(X(t)) - \mathbb{E}f(Z_L(t))\right] + \mathbb{E}f(Z_L(t)) - \mu_n$$

Suppose have an order one method

$$\mathbb{E}f(X(t)) - \mathbb{E}f(Z_L(t)) = O(h_L).$$

We need:

1. $h_L = O(\epsilon)$.
2. $n = \epsilon^{-2}$.

Return to approximating $\mathbb{E}f(X(T))$

Let Z_L denote an approximate processes generated with time discretization step of h_L. Let

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} f(Z_{L,[i]}(t)).$$

We note

$$\mathbb{E}f(X(t)) - \mu_n = \left[\mathbb{E}f(X(t)) - \mathbb{E}f(Z_L(t))\right] + \mathbb{E}f(Z_L(t)) - \mu_n$$

Suppose have an order one method

$$\mathbb{E}f(X(t)) - \mathbb{E}f(Z_L(t)) = O(h_L).$$

We need:

1. $h_L = O(\epsilon)$.
2. $n = \epsilon^{-2}$.

Suppose a path costs $O(\epsilon^{-1})$ steps. Then

$$\begin{aligned}\text{Total computational complexity} &= (\#\text{ paths}) \times (\text{cost per path}) \\ &= O(\epsilon^{-3}).\end{aligned}$$

Benefits/drawbacks

Benefits:

1. Can drastically lower the computational complexity of a problem if $\epsilon^{-1} \ll \overline{N}$.

$$\text{CC of using exact} = \overline{N}\epsilon^{-2}$$

$$\text{CC of using approximate} = \epsilon^{-1}\epsilon^{-2}.$$

Benefits/drawbacks

Benefits:

1. Can drastically lower the computational complexity of a problem if $\epsilon^{-1} \ll \overline{N}$.

$$\text{CC of using exact} = \overline{N}\epsilon^{-2}$$

$$\text{CC of using approximate} = \epsilon^{-1}\epsilon^{-2}.$$

Drawbacks:

1. Convergence results usually give order of convergence. Can't give a precise h_L. Bias is a problem.

2. Tau-leaping has problems: what happens if you go negative?

3. Gone away from an unbiased estimator.

Multi-level Monte Carlo and control variates

- Suppose I want

$$\mathbb{E}X \approx \frac{1}{n}\sum_{i=1}^{n} X_{[i]},$$

but realizations of X are expensive.

Multi-level Monte Carlo and control variates

- Suppose I want

$$\mathbb{E}X \approx \frac{1}{n}\sum_{i=1}^{n} X_{[i]},$$

  but realizations of X are expensive.

- Suppose $X \approx \widehat{Z}_L$, and $\widehat{Z}_L$ is cheap.

Multi-level Monte Carlo and control variates

- Suppose I want

$$\mathbb{E}X \approx \frac{1}{n}\sum_{i=1}^{n} X_{[i]},$$

but realizations of X are expensive.

- Suppose $X \approx \widehat{Z}_L$, and $\widehat{Z}_L$ is cheap.

- Suppose $X, \widehat{Z}_L$ can be generated simultaneously so that

$$\mathrm{Var}(X - \widehat{Z}_L)$$

is small.

Multi-level Monte Carlo and control variates

- Suppose I want

$$\mathbb{E}X \approx \frac{1}{n}\sum_{i=1}^{n} X_{[i]},$$

but realizations of X are expensive.

- Suppose $X \approx \widehat{Z}_L$, and $\widehat{Z}_L$ is cheap.

- Suppose $X, \widehat{Z}_L$ can be generated simultaneously so that

$$\mathrm{Var}(X - \widehat{Z}_L)$$

is small.

- Then use

$$\mathbb{E}X = \mathbb{E}[X - \widehat{Z}_L] + \mathbb{E}\widehat{Z}_L$$

Multi-level Monte Carlo and control variates

- Suppose I want

$$\mathbb{E}X \approx \frac{1}{n}\sum_{i=1}^{n} X_{[i]},$$

but realizations of X are expensive.

- Suppose $X \approx \widehat{Z}_L$, and $\widehat{Z}_L$ is cheap.

- Suppose $X, \widehat{Z}_L$ can be generated simultaneously so that

$$\mathrm{Var}(X - \widehat{Z}_L)$$

is small.

- Then use

$$\mathbb{E}X = \mathbb{E}[X - \widehat{Z}_L] + \mathbb{E}\widehat{Z}_L \approx \frac{1}{n_1}\sum_{i=1}^{n_1}(X_{[i]} - \widehat{Z}_{L,[i]}) + \frac{1}{n_2}\sum_{i=1}^{n_2}\widehat{Z}_{L,[i]}.$$

Multi-level Monte Carlo and control variates

- Suppose I want

$$\mathbb{E}X \approx \frac{1}{n}\sum_{i=1}^{n} X_{[i]},$$

but realizations of X are expensive.

- Suppose $X \approx \widehat{Z}_L$, and $\widehat{Z}_L$ is cheap.

- Suppose $X, \widehat{Z}_L$ can be generated simultaneously so that

$$\mathrm{Var}(X - \widehat{Z}_L)$$

is small.

- Then use

$$\mathbb{E}X = \mathbb{E}[X - \widehat{Z}_L] + \mathbb{E}\widehat{Z}_L \approx \frac{1}{n_1}\sum_{i=1}^{n_1}(X_{[i]} - \widehat{Z}_{L,[i]}) + \frac{1}{n_2}\sum_{i=1}^{n_2}\widehat{Z}_{L,[i]}.$$

- Multi-level Monte Carlo (Mike Giles, Stefan Heinrich) = Keep going

$$\mathbb{E}X = \mathbb{E}(X - \widehat{Z}_L) + \mathbb{E}\widehat{Z}_L =$$

Multi-level Monte Carlo and control variates

- Suppose I want

$$\mathbb{E}X \approx \frac{1}{n}\sum_{i=1}^{n} X_{[i]},$$

but realizations of X are expensive.

- Suppose $X \approx \widehat{Z}_L$, and $\widehat{Z}_L$ is cheap.

- Suppose $X, \widehat{Z}_L$ can be generated simultaneously so that

$$\mathrm{Var}(X - \widehat{Z}_L)$$

is small.

- Then use

$$\mathbb{E}X = \mathbb{E}[X - \widehat{Z}_L] + \mathbb{E}\widehat{Z}_L \approx \frac{1}{n_1}\sum_{i=1}^{n_1}(X_{[i]} - \widehat{Z}_{L,[i]}) + \frac{1}{n_2}\sum_{i=1}^{n_2}\widehat{Z}_{L,[i]}.$$

- Multi-level Monte Carlo (Mike Giles, Stefan Heinrich) = Keep going

$$\mathbb{E}X = \mathbb{E}(X - \widehat{Z}_L) + \mathbb{E}\widehat{Z}_L = \mathbb{E}(Z - \widehat{Z}_L) + \mathbb{E}(\widehat{Z}_L - \widehat{Z}_{L-1}) + \mathbb{E}\widehat{Z}_{L-1} = \cdots$$

Multi-level Monte Carlo: an unbiased estimator

In our setting:

$\mathbb{E}f(X(t)) =$

Multi-level Monte Carlo: an unbiased estimator

In our setting:

$$\mathbb{E}f(X(t)) = \mathbb{E}[f(X(t)) - f(Z_L(t))] + \sum_{\ell=\ell_0+1}^{L} \mathbb{E}[f(Z_\ell(t)) - f(Z_{\ell-1}(t))] + \mathbb{E}f(Z_{\ell_0}(t)).$$

Multi-level Monte Carlo: an unbiased estimator

In our setting:

$$\mathbb{E}f(X(t)) = \mathbb{E}[f(X(t)) - f(Z_L(t))] + \sum_{\ell=\ell_0+1}^{L} \mathbb{E}[f(Z_\ell(t)) - f(Z_{\ell-1}(t))] + \mathbb{E}f(Z_{\ell_0}(t)).$$

For appropiate choices of n_0, n_ℓ, and n_E, we define the estimators for the three terms above via

$$\widehat{Q}_E \quad \stackrel{\text{def}}{=} \quad \frac{1}{n_E} \sum_{i=1}^{n_E} (f(X_{[i]}(T) - f(Z_{L,[i]}(T))),$$

Multi-level Monte Carlo: an unbiased estimator

In our setting:

$$\mathbb{E}f(X(t)) = \mathbb{E}[f(X(t)) - f(Z_L(t))] + \sum_{\ell=\ell_0+1}^{L} \mathbb{E}[f(Z_\ell(t)) - f(Z_{\ell-1}(t))] + \mathbb{E}f(Z_{\ell_0}(t)).$$

For appropiate choices of n_0, n_ℓ, and n_E, we define the estimators for the three terms above via

$$\widehat{Q}_E \stackrel{\text{def}}{=} \frac{1}{n_E}\sum_{i=1}^{n_E}(f(X_{[i]}(T) - f(Z_{L,[i]}(T))),$$

$$\widehat{Q}_\ell \stackrel{\text{def}}{=} \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}(f(Z_{\ell,[i]}(T)) - f(Z_{\ell-1,[i]}(T))), \quad \text{for } \ell \in \{\ell_0+1,\ldots,L\}$$

Multi-level Monte Carlo: an unbiased estimator

In our setting:

$$\mathbb{E}f(X(t)) = \mathbb{E}[f(X(t)) - f(Z_L(t))] + \sum_{\ell=\ell_0+1}^{L} \mathbb{E}[f(Z_\ell(t)) - f(Z_{\ell-1}(t))] + \mathbb{E}f(Z_{\ell_0}(t)).$$

For appropiate choices of n_0, n_ℓ, and n_E, we define the estimators for the three terms above via

$$\begin{aligned}
\widehat{Q}_E &\stackrel{\text{def}}{=} \frac{1}{n_E}\sum_{i=1}^{n_E}(f(X_{[i]}(T) - f(Z_{L,[i]}(T))), \\
\widehat{Q}_\ell &\stackrel{\text{def}}{=} \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}(f(Z_{\ell,[i]}(T)) - f(Z_{\ell-1,[i]}(T))), \quad \text{for } \ell \in \{\ell_0+1,\ldots,L\} \\
\widehat{Q}_0 &\stackrel{\text{def}}{=} \frac{1}{n_0}\sum_{i=1}^{n_0} f(Z_{\ell_0,[i]}(T)),
\end{aligned}$$

and note that

$$\widehat{Q} \stackrel{\text{def}}{=} \widehat{Q}_E + \sum_{\ell=\ell_0}^{L} \widehat{Q}_\ell$$

is an unbiased estimator for $\mathbb{E}f(X(T))$.

Multi-level Monte Carlo: an unbiased estimator

In our setting:

$$\mathbb{E}f(X(t)) = \mathbb{E}[f(X(t)) - f(Z_L(t))] + \sum_{\ell=\ell_0+1}^{L} \mathbb{E}[f(Z_\ell(t)) - f(Z_{\ell-1}(t))] + \mathbb{E}f(Z_{\ell_0}(t)).$$

For appropiate choices of n_0, n_ℓ, and n_E, we define the estimators for the three terms above via

$$\begin{aligned}
\widehat{Q}_E &\stackrel{\text{def}}{=} \frac{1}{n_E}\sum_{i=1}^{n_E}(f(X_{[i]}(T) - f(Z_{L,[i]}(T))), \\
\widehat{Q}_\ell &\stackrel{\text{def}}{=} \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}(f(Z_{\ell,[i]}(T)) - f(Z_{\ell-1,[i]}(T))), \quad \text{for } \ell \in \{\ell_0+1,\ldots,L\} \\
\widehat{Q}_0 &\stackrel{\text{def}}{=} \frac{1}{n_0}\sum_{i=1}^{n_0} f(Z_{\ell_0,[i]}(T)),
\end{aligned}$$

and note that

$$\widehat{Q} \stackrel{\text{def}}{=} \widehat{Q}_E + \sum_{\ell=\ell_0}^{L} \widehat{Q}_\ell$$

is an unbiased estimator for $\mathbb{E}f(X(T))$.

So what is the coupling and the variance of the estimator?

How do we generate processes simultaneously

Suppose I want to generate:

- A Poisson process with intensity 13.1.
- A Poisson process with intensity 13.

How do we generate processes simultaneously

Suppose I want to generate:

- A Poisson process with intensity 13.1.
- A Poisson process with intensity 13.

- We could let Y_1 and Y_2 be independent, unit-rate Poisson processes, and set

$$Z_{13.1}(t) = Y_1(13.1t),$$
$$Z_{13}(t) = Y_2(13t),$$

Using this representation, these processes are independent and, hence, not coupled.

How do we generate processes simultaneously

Suppose I want to generate:

- A Poisson process with intensity 13.1.
- A Poisson process with intensity 13.

- We could let Y_1 and Y_2 be independent, unit-rate Poisson processes, and set

$$Z_{13.1}(t) = Y_1(13.1t),$$
$$Z_{13}(t) = Y_2(13t),$$

Using this representation, these processes are independent and, hence, not coupled.

The variance of difference is large:

$$\begin{aligned}\mathrm{Var}(Z_{13.1}(t) - Z_{13}(t)) &= \mathrm{Var}(Y_1(13.1t)) + \mathrm{Var}(Y_2(13t)) \\ &= 26.1t.\end{aligned}$$

How do we generate processes simultaneously

Suppose I want to generate:

- A Poisson process with intensity 13.1.
- A Poisson process with intensity 13.

How do we generate processes simultaneously

Suppose I want to generate:

- A Poisson process with intensity 13.1.
- A Poisson process with intensity 13.

- We could let Y_1 and Y_2 be independent unit-rate Poisson processes, and set

$$Z_{13.1}(t) = Y_1(13t) + Y_2(0.1t)$$
$$Z_{13}(t) = Y_1(13t),$$

The variance of difference is much smaller:

$$\text{Var}(Z_{13.1}(t) - Z_{13}(t)) = \text{Var}\left(Y_2(0.1t)\right) = 0.1t.$$

How do we generate processes simultaneously

More generally, suppose we want

1. non-homogeneous Poisson process with intensity $f(t)$ and
2. non-homogeneous Poisson process with intensity $g(t)$.

How do we generate processes simultaneously

More generally, suppose we want

1. non-homogeneous Poisson process with intensity $f(t)$ and
2. non-homogeneous Poisson process with intensity $g(t)$.

We can can let Y_1, Y_2, and Y_3 be independent, unit-rate Poisson processes and define

$$Z_f(t) = Y_1\left(\int_0^t f(s) \wedge g(s) ds\right) + Y_2\left(\int_0^t f(s) - (f(s) \wedge g(s))\, ds\right),$$

$$Z_g(t) = Y_1\left(\int_0^t f(s) \wedge g(s) ds\right) + Y_3\left(\int_0^t g(s) - (f(s) \wedge g(s))\, ds\right),$$

How do we generate processes simultaneously

More generally, suppose we want

1. non-homogeneous Poisson process with intensity $f(t)$ and
2. non-homogeneous Poisson process with intensity $g(t)$.

We can can let Y_1, Y_2, and Y_3 be independent, unit-rate Poisson processes and define

$$Z_f(t) = Y_1\left(\int_0^t f(s) \wedge g(s) ds\right) + Y_2\left(\int_0^t f(s) - (f(s) \wedge g(s))\, ds\right),$$
$$Z_g(t) = Y_1\left(\int_0^t f(s) \wedge g(s) ds\right) + Y_3\left(\int_0^t g(s) - (f(s) \wedge g(s))\, ds\right),$$

where we are using that, for example,

$$Y_1\left(\int_0^t f(s) \wedge g(s) ds\right) + Y_2\left(\int_0^t f(s) - (f(s) \wedge g(s))\, ds\right) = Y\left(\int_0^t f(s) ds\right),$$

where Y is a unit rate Poisson process.

Back to our processes

$$X(t) = X(0) + \sum_k Y_k \left(\int_0^t \lambda_k(X(s))ds \right) \zeta_k,$$

$$Z(t) = X(0) + \sum_k Y_k \left(\int_0^t \lambda_k(Z \circ \eta(s))ds \right) \zeta_k.$$

Back to our processes

$$X(t) = X(0) + \sum_k Y_k \left(\int_0^t \lambda_k(X(s))ds \right) \zeta_k,$$

$$Z(t) = X(0) + \sum_k Y_k \left(\int_0^t \lambda_k(Z \circ \eta(s))ds \right) \zeta_k.$$

Now couple

$$\begin{aligned} X(t) = X(0) &+ \sum_k Y_{k,1} \left(\int_0^t \lambda_k(X(s)) \wedge \lambda_k(Z_\ell \circ \eta_\ell(s))ds \right) \zeta_k \\ &+ \sum_k Y_{k,2} \left(\int_0^t \lambda_k(X(s)) - \lambda_k(X(s)) \wedge \lambda_k(Z_\ell \circ \eta_\ell(s))ds \right) \zeta_k \\ Z_\ell(t) = Z_\ell(0) &+ \sum_k Y_{k,1} \left(\int_0^t \lambda_k(X(s)) \wedge \lambda_k(Z_\ell \circ \eta_\ell(s))ds \right) \zeta_k \\ &+ \sum_k Y_{k,3} \left(\int_0^t \lambda_k(Z_\ell \circ \eta_\ell(s)) - \lambda_k(X(s)) \wedge \lambda_k(Z_\ell \circ \eta_\ell(s))ds \right) \zeta_k \end{aligned}$$

Back to our processes

$$X(t) = X(0) + \sum_k Y_k \left(\int_0^t \lambda_k(X(s))ds \right) \zeta_k,$$

$$Z(t) = X(0) + \sum_k Y_k \left(\int_0^t \lambda_k(Z \circ \eta(s))ds \right) \zeta_k.$$

Now couple

$$\begin{aligned} X(t) = X(0) &+ \sum_k Y_{k,1} \left(\int_0^t \lambda_k(X(s)) \wedge \lambda_k(Z_\ell \circ \eta_\ell(s))ds \right) \zeta_k \\ &+ \sum_k Y_{k,2} \left(\int_0^t \lambda_k(X(s)) - \lambda_k(X(s)) \wedge \lambda_k(Z_\ell \circ \eta_\ell(s))ds \right) \zeta_k \\ Z_\ell(t) = Z_\ell(0) &+ \sum_k Y_{k,1} \left(\int_0^t \lambda_k(X(s)) \wedge \lambda_k(Z_\ell \circ \eta_\ell(s))ds \right) \zeta_k \\ &+ \sum_k Y_{k,3} \left(\int_0^t \lambda_k(Z_\ell \circ \eta_\ell(s)) - \lambda_k(X(s)) \wedge \lambda_k(Z_\ell \circ \eta_\ell(s))ds \right) \zeta_k \end{aligned}$$

Algorithm for simulation is equivalent to next reaction method or Gillespie.

For approximate processes

$$Z_\ell(t) = Z_\ell(0) + \sum_k Y_{k,1}\left(\int_0^t \lambda_k(Z_\ell \circ \eta_\ell(s)) \wedge \lambda_k(Z_{\ell-1} \circ \eta_{\ell-1}(s))ds\right)\zeta_k$$
$$+ \sum_k Y_{k,2}\left(\int_0^t \lambda_k(Z_\ell \circ \eta_\ell(s)) - \lambda_k(Z_\ell \circ \eta_\ell(s)) \wedge \lambda_k(Z_{\ell-1} \circ \eta_{\ell-1}(s))ds\right)\zeta_k$$

$$Z_{\ell-1}(t) = Z_{\ell-1}(0) + \sum_k Y_{k,1}\left(\int_0^t \lambda_k(Z_\ell \circ \eta_\ell(s)) \wedge \lambda_k(Z_{\ell-1} \circ \eta_{\ell-1}(s))ds\right)\zeta_k$$
$$+ \sum_k Y_{k,3}\left(\int_0^t \lambda_k(Z_{\ell-1} \circ \eta_{\ell-1}(s)) - \lambda_k(Z_\ell \circ \eta_\ell(s)) \wedge \lambda_k(Z_{\ell-1} \circ \eta_{\ell-1}(s))ds\right)\zeta_k,$$

Algorithm for simulation is equivalent in to τ-leaping.

Multi-level Monte Carlo: chemical kinetic setting

Can prove:

Theorem (Anderson, Higham 2011)

Suppose (X, Z_ℓ) *satisfy coupling. Then,*

$$\sup_{t \leq T} \mathbb{E}|X(t) - Z_\ell(t)|^2 \leq C_1(T) N^{-\rho} h_\ell + C_2(T) h_\ell^2.$$

[1] David F. Anderson and Desmond J. Higham, *Multi-level Monte Carlo for stochastically modeled chemical kinetic systems*. To appear in SIAM: Modeling and Simulation. Available at `arxiv.org:1107.2181`. Also at `www.math.wisc.edu/~anderson`.

Multi-level Monte Carlo: chemical kinetic setting

Can prove:

Theorem (Anderson, Higham 2011)

Suppose (X, Z_ℓ) *satisfy coupling. Then,*

$$\sup_{t \leq T} \mathbb{E}|X(t) - Z_\ell(t)|^2 \leq C_1(T) N^{-\rho} h_\ell + C_2(T) h_\ell^2.$$

Theorem (Anderson, Higham 2011)

Suppose $(Z_\ell, Z_{\ell-1})$ *satisfy coupling. Then,*

$$\sup_{t \leq T} \mathbb{E}|Z_\ell(t) - Z_{\ell-1}(t)|^2 \leq C_1(T) N^{-\rho} h_\ell + C_2(T) h_\ell^2.$$

---

[1] David F. Anderson and Desmond J. Higham, *Multi-level Monte Carlo for stochastically modeled chemical kinetic systems*. To appear in SIAM: Modeling and Simulation. Available at `arxiv.org:1107.2181`. Also at `www.math.wisc.edu/~anderson`.

Multi-level Monte Carlo: an unbiased estimator

For well chosen n_0, n_ℓ, and n_E. We have

$$\mathrm{Var}(\widehat{Q}) = \mathrm{Var}\left(\widehat{Q}_E + \sum_{\ell=\ell_0}^{L} \widehat{Q}_\ell\right) = O(\epsilon^2),$$

with

$$\text{Comp. cost} = \left[\epsilon^{-2}(N^{-\rho}h_L + h_L^2)\right]\overline{N} + \epsilon^{-2}\left(h_{\ell_0}^{-1} + \ln(\epsilon)^2 N^{-\rho} + \ln(\epsilon^{-1})\frac{1}{M-1}h_{\ell_0}\right)$$

Multi-level Monte Carlo: an unbiased estimator

Some observations:

1. Weak error plays no role in analysis: free to choose h_L.

Multi-level Monte Carlo: an unbiased estimator

Some observations:

1. Weak error plays no role in analysis: free to choose h_L.
2. Common problems associated with tau-leaping
   - Negativity of species numbers,

   does not matter. Just define process in a sensible way.

Multi-level Monte Carlo: an unbiased estimator

Some observations:

1. Weak error plays no role in analysis: free to choose h_L.
2. Common problems associated with tau-leaping
   - Negativity of species numbers,

   does not matter. Just define process in a sensible way.
3. The method is unbiased.

Multi-level Monte Carlo: an unbiased estimator

Some observations:

1. Weak error plays no role in analysis: free to choose h_L.

2. Common problems associated with tau-leaping

   - Negativity of species numbers,

   does not matter. Just define process in a sensible way.

3. The method is unbiased.

4. A pre-computation optimization problem can be used to efficiently allocate variance to different levels.

Multi-level Monte Carlo: an unbiased estimator

Some observations:

1. Weak error plays no role in analysis: free to choose h_L.
2. Common problems associated with tau-leaping
   - Negativity of species numbers,

   does not matter. Just define process in a sensible way.
3. The method is unbiased.
4. A pre-computation optimization problem can be used to efficiently allocate variance to different levels.
5. In this way: will never be appreciably slower than exact method + crude Monte Carlo.

Example

Consider a model of gene transcription and translation:

$$G \xrightarrow{25} G + M,$$

$$M \xrightarrow{1000} M + P,$$

$$P + P \xrightarrow{0.001} D,$$

$$M \xrightarrow{0.1} \emptyset,$$

$$P \xrightarrow{1} \emptyset.$$

Suppose:

1. initialize with: $G = 1$, $M = 0$, $P = 0$, $D = 0$,
2. want to estimate the expected number of dimers at time $T = 1$,
3. to an accuracy of $\pm$ 1.0 with 95% confidence.

Example

Method: Exact algorithm with crude Monte Carlo.

Approximation	# paths	CPU Time	# updates
3,714.2 $\pm$ 1.0	4,740,000	149,000 CPU S (41 hours!)	8.27 $\times 10^{10}$

Example

Method: Exact algorithm with crude Monte Carlo.

Approximation	# paths	CPU Time	# updates
$3{,}714.2 \pm 1.0$	4,740,000	149,000 CPU S (41 hours!)	8.27×10^{10}

Method: Euler tau-leaping with crude Monte Carlo.

Step-size	Approximation	# paths	CPU Time	# updates
$h = 3^{-7}$	$3{,}712.3 \pm 1.0$	4,750,000	13,374.6 S	6.2×10^{10}
$h = 3^{-6}$	$3{,}707.5 \pm 1.0$	4,750,000	6,207.9 S	2.1×10^{10}
$h = 3^{-5}$	$3{,}693.4 \pm 1.0$	4,700,000	2,803.9 S	6.9×10^{9}
$h = 3^{-4}$	$3{,}654.6 \pm 1.0$	4,650,000	1,219.0 S	2.6×10^{9}

Example

Method: Exact algorithm with crude Monte Carlo.

Approximation	# paths	CPU Time	# updates
$3{,}714.2 \pm 1.0$	4,740,000	149,000 CPU S (41 hours!)	8.27×10^{10}

Method: Euler tau-leaping with crude Monte Carlo.

Step-size	Approximation	# paths	CPU Time	# updates
$h = 3^{-7}$	$3{,}712.3 \pm 1.0$	4,750,000	13,374.6 S	6.2×10^{10}
$h = 3^{-6}$	$3{,}707.5 \pm 1.0$	4,750,000	6,207.9 S	2.1×10^{10}
$h = 3^{-5}$	$3{,}693.4 \pm 1.0$	4,700,000	2,803.9 S	6.9×10^{9}
$h = 3^{-4}$	$3{,}654.6 \pm 1.0$	4,650,000	1,219.0 S	2.6×10^{9}

Method: unbiased MLMC with $\ell_0 = 2$, and M and L detailed below.

Step-size parameters	Approx.	CPU Time	# updates
$M = 3$, $L = 6$	$3{,}713.9 \pm 1.0$	1,063.3 S	1.1×10^{9}
$M = 3$, $L = 5$	$3{,}714.7 \pm 1.0$	1,114.9 S	9.4×10^{8}
$M = 3$, $L = 4$	$3{,}714.2 \pm 1.0$	1,656.6 S	1.0×10^{9}
$M = 4$, $L = 4$	3714.2 ± 1.0	1,334.8 S	1.1×10^{9}
$M = 4$, $L = 5$	$3{,}713.8 \pm 1.0$	1,014.9 S	1.1×10^{9}

Example

Method: Exact algorithm with crude Monte Carlo.

Approximation	# paths	CPU Time	# updates
$3{,}714.2 \pm 1.0$	4,740,000	149,000 CPU S (41 hours!)	8.27×10^{10}

Method: Euler tau-leaping with crude Monte Carlo.

Step-size	Approximation	# paths	CPU Time	# updates
$h = 3^{-7}$	$3{,}712.3 \pm 1.0$	4,750,000	13,374.6 S	6.2×10^{10}
$h = 3^{-6}$	$3{,}707.5 \pm 1.0$	4,750,000	6,207.9 S	2.1×10^{10}
$h = 3^{-5}$	$3{,}693.4 \pm 1.0$	4,700,000	2,803.9 S	6.9×10^{9}
$h = 3^{-4}$	$3{,}654.6 \pm 1.0$	4,650,000	1,219.0 S	2.6×10^{9}

Method: unbiased MLMC with $\ell_0 = 2$, and M and L detailed below.

Step-size parameters	Approx.	CPU Time	# updates
$M = 3$, $L = 6$	$3{,}713.9 \pm 1.0$	1,063.3 S	1.1×10^{9}
$M = 3$, $L = 5$	$3{,}714.7 \pm 1.0$	1,114.9 S	9.4×10^{8}
$M = 3$, $L = 4$	$3{,}714.2 \pm 1.0$	1,656.6 S	1.0×10^{9}
$M = 4$, $L = 4$	3714.2 ± 1.0	1,334.8 S	1.1×10^{9}
$M = 4$, $L = 5$	$3{,}713.8 \pm 1.0$	1,014.9 S	1.1×10^{9}

- the exact algorithm with crude Monte Carlo demanded 140 times more CPU time than our unbiased MLMC estimator.

Example

Method: Exact algorithm with crude Monte Carlo.

Approximation	# paths	CPU Time	# updates
3,714.2 $\pm$ 1.0	4,740,000	149,000 CPU S (41 hours!)	8.27 $\times 10^{10}$

Unbiased Multi-level Monte Carlo with $M = 3$, $L = 5$, and $\ell_0 = 2$.

Level	# paths	CPU Time	Var. estimator	# updates
$(X, Z_{3^{-5}})$	3,900	279.6 S	0.0658	6.8 $\times 10^7$
$(Z_{3^{-5}}, Z_{3^{-4}})$	30,000	49.0 S	0.0217	8.8 $\times 10^7$
$(Z_{3^{-4}}, Z_{3^{-3}})$	150,000	71.7 S	0.0179	1.5 $\times 10^8$
$(Z_{3^{-3}}, Z_{3^{-2}})$	510,000	112.3 S	0.0319	1.7 $\times 10^8$
Tau-leap with $h = 3^{-2}$	8,630,000	518.4 S	0.1192	4.7 $\times 10^8$
Totals	N.A.	1031.0 S	0.2565	9.5 $\times 10^8$

Some conclusions about this method

1. Gillespie's algorithm is by far the most common way to compute expectations:
   - 1.1 Means.
   - 1.2 Variances.
   - 1.3 Probabilities.
2. The new method (MLMC) also performs this task with no bias (exact).

Some conclusions about this method

1. Gillespie's algorithm is by far the most common way to compute expectations:
   - 1.1 Means.
   - 1.2 Variances.
   - 1.3 Probabilities.
2. The new method (MLMC) also performs this task with no bias (exact).
3. Will be at worst the same speed as Gillespie (exact algorithm + crude Monte Carlo).
4. Will commonly be many orders of magnitude faster.

Some conclusions about this method

1. Gillespie's algorithm is by far the most common way to compute expectations:
   - 1.1 Means.
   - 1.2 Variances.
   - 1.3 Probabilities.
2. The new method (MLMC) also performs this task with no bias (exact).
3. Will be at worst the same speed as Gillespie (exact algorithm + crude Monte Carlo).
4. Will commonly be many orders of magnitude faster.
5. Applicable to essentially *all* continuous time Markov chains:

$$X(t) = X(0) + \sum_k Y_k \left(\int_0^t \lambda_k(X(s))ds \right) \zeta_k.$$

6. Makes no use of any specific structure or scaling in the problem.

Another example: Viral infection

Let

1. $T =$ viral template.
2. $G =$ viral genome.
3. $S =$ viral structure.
4. $V =$ virus.

Reactions:

$$
\begin{array}{ll}
\text{R1) } T + \text{stuff} \xrightarrow{\kappa_1} T + G & \kappa_1 = 1 \\
\text{R2) } G \xrightarrow{\kappa_2} T & \kappa_2 = 0.025 \\
\text{R3) } T + \text{stuff} \xrightarrow{\kappa_3} T + S & \kappa_3 = 1000 \\
\text{R4) } T \xrightarrow{\kappa_4} \emptyset & \kappa_4 = 0.25 \\
\text{R5) } S \xrightarrow{\kappa_5} \emptyset & \kappa_5 = 2 \\
\text{R6) } G + S \xrightarrow{\kappa_6} V & \kappa_6 = 7.5 \times 10^{-6}
\end{array}
$$

- R. Srivastava, L. You, J. Summers, and J. Yin, J. Theoret. Biol., 2002.
- E. Haseltine and J. Rawlings, J. Chem. Phys, 2002.
- K. Ball, T. Kurtz, L. Popovic, and G. Rempala, Annals of Applied Probability, 2006.
- W. E, D. Liu, and E. Vanden-Eijden, J. Comput. Phys, 2006.

Another example: Viral infection

Stochastic equations for $X = (X_G, X_S, X_T, X_V)$ are

$$X_1(t) = X_1(0) + Y_1\left(\int_0^t X_3(s)ds\right) - Y_2\left(0.025\int_0^t X_1(s)ds\right) - Y_6\left(7.5\times 10^{-6}\int_0^t X_1(s)X_2(s)ds\right)$$

$$X_2(t) = X_2(0) + Y_3\left(1000\int_0^t X_3(s)ds\right) - Y_5\left(2\int_0^t X_2(s)ds\right) - Y_6\left(7.5\times 10^{-6}\int_0^t X_1(s)X_2(s)ds\right)$$

$$X_3(t) = X_3(0) + Y_2\left(0.025\int_0^t X_1(s)ds\right) - Y_4\left(0.25\int_0^t X_3(s)ds\right)$$

$$X_4(t) = X_4(0) + Y_6\left(7.5\times 10^{-6}\int_0^t X_1(s)X_2(s)ds\right).$$

Another example: Viral infection

Reactions:

R1) $T+$ stuff $\stackrel{\kappa_1}{\to} T+G$	$\kappa_1 = 1$
R2) $G \stackrel{\kappa_2}{\to} T$	$\kappa_2 = 0.025$
R3) $T+$ stuff $\stackrel{\kappa_3}{\to} T+S$	$\kappa_3 = 1000$
R4) $T \stackrel{\kappa_4}{\to} \emptyset$	$\kappa_4 = 0.25$
R5) $S \stackrel{\kappa_5}{\to} \emptyset$	$\kappa_5 = 2$
R6) $G+S \stackrel{\kappa_6}{\to} V$	$\kappa_6 = 7.5 \times 10^{-6}$

If $T > 0$,

- reactions 3 and 5 are much faster than others.
- Looks like S is approximately Poisson$(500 \times T)$.

Another example: Viral infection

Reactions:

R1) $T+$ stuff $\stackrel{\kappa_1}{\to} T+G$	$\kappa_1 = 1$
R2) $G \stackrel{\kappa_2}{\to} T$	$\kappa_2 = 0.025$
R3) $T+$ stuff $\stackrel{\kappa_3}{\to} T+S$	$\kappa_3 = 1000$
R4) $T \stackrel{\kappa_4}{\to} \emptyset$	$\kappa_4 = 0.25$
R5) $S \stackrel{\kappa_5}{\to} \emptyset$	$\kappa_5 = 2$
R6) $G+S \stackrel{\kappa_6}{\to} V$	$\kappa_6 = 7.5 \times 10^{-6}$

If $T > 0$,

- reactions 3 and 5 are much faster than others.
- Looks like S is approximately Poisson$(500 \times T)$.

Can average out to get approximate process $Z(t)$.

Another example: Viral infection

Approximate process satisfies.

$$\begin{aligned} Z_1(t) &= X_1(0) + Y_1\left(\int_0^t Z_3(s)ds\right) - Y_2\left(0.025\int_0^t Z_1(s)ds\right) \\ &\quad - Y_6\left(3.75\times 10^{-3}\int_0^t Z_1(s)Z_3(s)ds\right) \\ Z_3(t) &= X_3(0) + Y_2\left(0.025\int_0^t Z_1(s)ds\right) - Y_4\left(0.25\int_0^t Z_3(s)ds\right) \\ Z_4(t) &= X_4(0) + Y_6\left(3.75\times 10^{-3}\int_0^t Z_1(s)Z_3(s)ds\right). \end{aligned} \tag{1}$$

Now use

$$\mathbb{E}f(X(t)) = \mathbb{E}[f(X(t)) - f(Z(t))] + \mathbb{E}f(Z(t)).$$

Another example: Viral infection

$$X(t) = X(0) + Y_{1,1}\left(\int_0^t \min\{X_3(s), Z_3(s)\}ds\right)\zeta_1 + Y_{1,2}\left(\int_0^t X_3(s) - \min\{X_3(s), Z_3(s)\}ds\right)\zeta_1$$
$$+ Y_{2,1}\left(0.025\int_0^t \min\{X_1(s), Z_1(s)\}ds\right)\zeta_2 + Y_{2,2}\left(0.025\int_0^t X_1(s) - \min\{X_1(s), Z_1(s)\}ds\right)\zeta_2$$
$$+ Y_3\left(1000\int_0^t X_3(s)ds\right)\zeta_3$$
$$+ Y_{4,1}\left(0.25\int_0^t \min\{X_3(s), Z_3(s)\}(s)ds\right)\zeta_4 + Y_{4,2}\left(0.25\int_0^t X_3(s) - \min\{X_3(s), Z_3(s)\}(s)ds\right)\zeta_4$$
$$+ Y_5\left(2\int_0^t X_2(s)ds\right)\zeta_5$$
$$+ Y_{6,1}\left(\int_0^t \min\{\lambda_6(X(s)), \Lambda_6(Z(s))\}ds\right)\zeta_6 - Y_{6,2}\left(\int_0^t \lambda_6(X(s)) - \min\{\lambda_6(X(s)), \Lambda_6(Z(s))\}ds\right)\zeta_6$$

$$Z(t) = Y_{1,1}\left(\int_0^t \min\{X_3(s), Z_3(s)\}ds\right)\zeta_1 + Y_{1,3}\left(\int_0^t Z_3(s) - \min\{X_3(s), Z_3(s)\}ds\right)\zeta_1$$
$$+ Y_{2,1}\left(0.025\int_0^t \min\{X_1(s), Z_1(s)\}ds\right)\zeta_2 + Y_{2,3}\left(0.025\int_0^t Z_1(s) - \min\{X_1(s), Z_1(s)\}ds\right)\zeta_2$$
$$+ Y_{4,1}\left(0.25\int_0^t \min\{X_3(s), Z_3(s)\}(s)ds\right)\zeta_4 + Y_{4,3}\left(0.25\int_0^t Z_3(s) - \min\{X_3(s), Z_3(s)\}(s)ds\right)\zeta_4$$
$$+ Y_{6,1}\left(\int_0^t \min\{\lambda_6(X(s)), \Lambda_6(Z(s))\}ds\right)\zeta_6 - Y_{6,3}\left(\int_0^t \Lambda_6(Z(s)) - \min\{\lambda_6(X(s)), \Lambda_6(Z(s))\}ds\right)\zeta_6,$$

Another example: Viral infection

Suppose want

$$\mathbb{E}X_{virus}(20)$$

Given $T(0) = 10$, all others zero.

Method: Exact algorithm with crude Monte Carlo.

Approximation	# paths	CPU Time	# updates
13.85 ± 0.07	75,000	24,800 CPU S	1.45×10^{10}

Another example: Viral infection

Suppose want

$$\mathbb{E}X_{virus}(20)$$

Given $T(0) = 10$, all others zero.

Method: Exact algorithm with crude Monte Carlo.

Approximation	# paths	CPU Time	# updates
13.85 ± 0.07	75,000	24,800 CPU S	1.45×10^{10}

Method: $\mathbb{E}f(X(t)) = \mathbb{E}[f(X(t)) - f(Z(t))] + \mathbb{E}f(Z(t))$.

Approximation	CPU Time	# updates
13.91 ± 0.07	1,118.5 CPU S	2.41×10^{8}

Exact + crude Monte Carlo used:

1. 60 times more total steps.
2. 22 times more CPU time.

Mathematical Analysis

We had

$$X(t) = X(0) + \sum_k Y_k \left(\int_0^t \lambda_k'(X(s))ds \right) \zeta_k.$$

Assumed

$$\sum_k \lambda_k'(X(\cdot)) \approx \overline{N} \gg 1.$$

There are therefore two extreme parameters floating around our models:

1. Some parameter $N \gg 1$, causing $\overline{N} \gg 1$ (inherent to model).
2. h, the stepsize (inherent to approximation).

To quantify errors, need to account for both.

Mathematical Analysis: Scaling in style of Thomas Kurtz

For each species i, define the normalized abundance

$$X_i^N(t) = N^{-\alpha_i} X_i(t),$$

where $\alpha_i \geq 0$ should be selected so that $X_i^N = O(1)$.

Mathematical Analysis: Scaling in style of Thomas Kurtz

For each species i, define the normalized abundance

$$X_i^N(t) = N^{-\alpha_i} X_i(t),$$

where $\alpha_i \geq 0$ should be selected so that $X_i^N = O(1)$.

Rate constants, κ_k', may also vary over several orders of magnitude. We write

$$\kappa_k' = \kappa_k N^{\beta_k}$$

where the β_k are selected so that $\kappa_k = O(1)$.

Mathematical Analysis: Scaling in style of Thomas Kurtz

For each species i, define the normalized abundance

$$X_i^N(t) = N^{-\alpha_i} X_i(t),$$

where $\alpha_i \geq 0$ should be selected so that $X_i^N = O(1)$.

Rate constants, κ_k', may also vary over several orders of magnitude. We write

$$\kappa_k' = \kappa_k N^{\beta_k}$$

where the β_k are selected so that $\kappa_k = O(1)$.

Eventually leads to scaled model

$$X^N(t) = X^N(0) + \sum_k Y_k \left(N^\gamma \int_0^t N^{\beta_k + \alpha \cdot \nu_k - \gamma} \lambda_k(X^N(s)) ds \right) \zeta_k^N.$$

Results

$$X^N(t) = X^N(0) + \sum_k Y_k \left(N^\gamma \int_0^t N^{c_k} \lambda_k(X^N(s))ds \right) \zeta_k^N.$$

Let $\rho_k \geq 0$ satisfy

$$|\zeta_k^N| \approx N^{-\rho_k},$$

and set

$$\rho = \min\{\rho_k\}.$$

Theorem (A., Higham 2011)

Suppose $(Z_\ell^N, Z_{\ell-1}^N)$ *satisfy coupling with* $Z_\ell^N(0) = Z_{\ell-1}^N(0)$. *Then,*

$$\sup_{t \leq T} \mathbb{E}|Z_\ell^N(t) - Z_{\ell-1}^N(t)|^2 \leq C_1(T, N, \gamma) N^{-\rho} h_\ell + C_2(T, N, \gamma) h_\ell^2.$$

Results

$$X^N(t) = X^N(0) + \sum_k Y_k \left(N^\gamma \int_0^t N^{c_k} \lambda_k(X^N(s))ds \right) \zeta_k^N.$$

Let $\rho_k \geq 0$ satisfy

$$|\zeta_k^N| \approx N^{-\rho_k},$$

and set

$$\rho = \min\{\rho_k\}.$$

Theorem (A., Higham 2011)

Suppose (X^N, Z_ℓ^N) *satisfy coupling with* $X^N(0) = Z_\ell^N(0)$. *Then,*

$$\sup_{t \leq T} \mathbb{E}|X^N(t) - Z_\ell^N(t)|^2 \leq C_1(T, N, \gamma) N^{-\rho} h_\ell + C_2(T, N, \gamma) h_\ell^2.$$

Flavor of Proof

Theorem (A., Higham 2011)

Suppose (X^N, Z_ℓ^N) *satisfy coupling with* $X^N(0) = Z_\ell^N(0)$. *Then,*

$$\sup_{t \le T} \mathbb{E}|X^N(t) - Z_\ell^N(t)|^2 \le C_1(T, N, \gamma) N^{-\rho} h_\ell + C_2(T, N, \gamma) h_\ell^2.$$

$$\begin{aligned} X^N(t) =& X^N(0) + \sum_k Y_{k,1}\left(N^\gamma N^{c_k} \int_0^t \lambda_k(X^N(s)) \wedge \lambda_k(Z_\ell^N \circ \eta_\ell(s)) ds\right) \zeta_k^N \\ &+ \sum_k Y_{k,2}\left(N^\gamma N^{c_k} \int_0^t \lambda_k(X^N(s)) - \lambda_k(X^N(s)) \wedge \lambda_k(Z_\ell^N \circ \eta_\ell(s)) ds\right) \zeta_k^N \end{aligned}$$

$$\begin{aligned} Z_\ell^N(t) =& Z_\ell^N(0) + \sum_k Y_{k,1}\left(N^\gamma N^{c_k} \int_0^t \lambda_k(X^N(s)) \wedge \lambda_k(Z_\ell \circ \eta_\ell(s)) ds\right) \zeta_k^N \\ &+ \sum_k Y_{k,3}\left(N^\gamma N^{c_k} \int_0^t \lambda_k(Z_\ell^N \circ \eta_\ell(s)) - \lambda_k(X^N(s)) \wedge \lambda_k(Z_\ell^N \circ \eta_\ell(s)) ds\right) \zeta_k^N \end{aligned}$$

Flavor of Proof

So,

$$X^N(t) - Z^N(t) = \sum_k Y_{k,2}\left(N^\gamma N^{c_k}\int_0^t \lambda_k(X^N(s)) - \lambda_k(X^N(s)) \wedge \lambda_k(Z^N_\ell \circ \eta_\ell(s))ds\right)\zeta^N_k$$
$$- Y_{k,3}\left(N^\gamma N^{c_k}\int_0^t \lambda_k(Z^N_\ell \circ \eta_\ell(s)) - \lambda_k(X^N(s)) \wedge \lambda_k(Z^N_\ell \circ \eta_\ell(s))ds\right)\zeta^N_k$$

Flavor of Proof

So,

$$X^N(t) - Z^N(t) = \sum_k Y_{k,2}\left(N^\gamma N^{c_k}\int_0^t \lambda_k(X^N(s)) - \lambda_k(X^N(s)) \wedge \lambda_k(Z^N_\ell \circ \eta_\ell(s))ds\right)\zeta^N_k$$
$$- Y_{k,3}\left(N^\gamma N^{c_k}\int_0^t \lambda_k(Z^N_\ell \circ \eta_\ell(s)) - \lambda_k(X^N(s)) \wedge \lambda_k(Z^N_\ell \circ \eta_\ell(s))ds\right)\zeta^N_k$$

Hence,

$$X^N(t) - Z^N(t) = M^N(t) + \sum_k N^\gamma \zeta^N_k N^{c_k}\int_0^t (\lambda_k(X^N(s)) - \lambda_k(Z^N_\ell \circ \eta_\ell(s)))ds.$$

Now work.

What about γ?

Example. Consider the family of models indexed by θ,

$$A \underset{\theta}{\overset{\theta}{\rightleftarrows}} B,$$

with,

$$X_A(0) = X_B(0) = \lfloor 1{,}000\,\theta^{-1} \rfloor,$$

where $\lfloor x \rfloor$ is the greatest integer less than or equal to x.

What about γ?

Example. Consider the family of models indexed by θ,

$$A \underset{\theta}{\overset{\theta}{\rightleftarrows}} B,$$

with,

$$X_A(0) = X_B(0) = \lfloor 1{,}000\,\theta^{-1} \rfloor,$$

where $\lfloor x \rfloor$ is the greatest integer less than or equal to x.

Note

$$\begin{aligned} \theta < 1 &\implies \gamma < 0 \\ \theta = 1 &\implies \gamma = 0 \\ \theta > 1 &\implies \gamma > 0. \end{aligned}$$

What about γ?

Example. Consider the family of models indexed by θ,

$$A \underset{\theta}{\overset{\theta}{\rightleftarrows}} B,$$

with,

$$X_A(0) = X_B(0) = \lfloor 1{,}000\,\theta^{-1} \rfloor,$$

where $\lfloor x \rfloor$ is the greatest integer less than or equal to x.

Note

$$\begin{aligned} \theta < 1 &\implies \gamma < 0 \\ \theta = 1 &\implies \gamma = 0 \\ \theta > 1 &\implies \gamma > 0. \end{aligned}$$

$$X_A(t) = X_A(0) + Y_1\left(\int_0^t \kappa(2,000\kappa^{-1} - X_A(s))ds\right) - Y_2\left(\int_0^t \kappa X_A(s)ds\right)$$

$$Z_A(t) = Z_A(0) + Y_1\left(\int_0^t \kappa(2,000\kappa^{-1} - Z_A \circ \eta(s))ds\right) - Y_2\left(\int_0^t \kappa Z_A \circ \eta(s)ds\right).$$

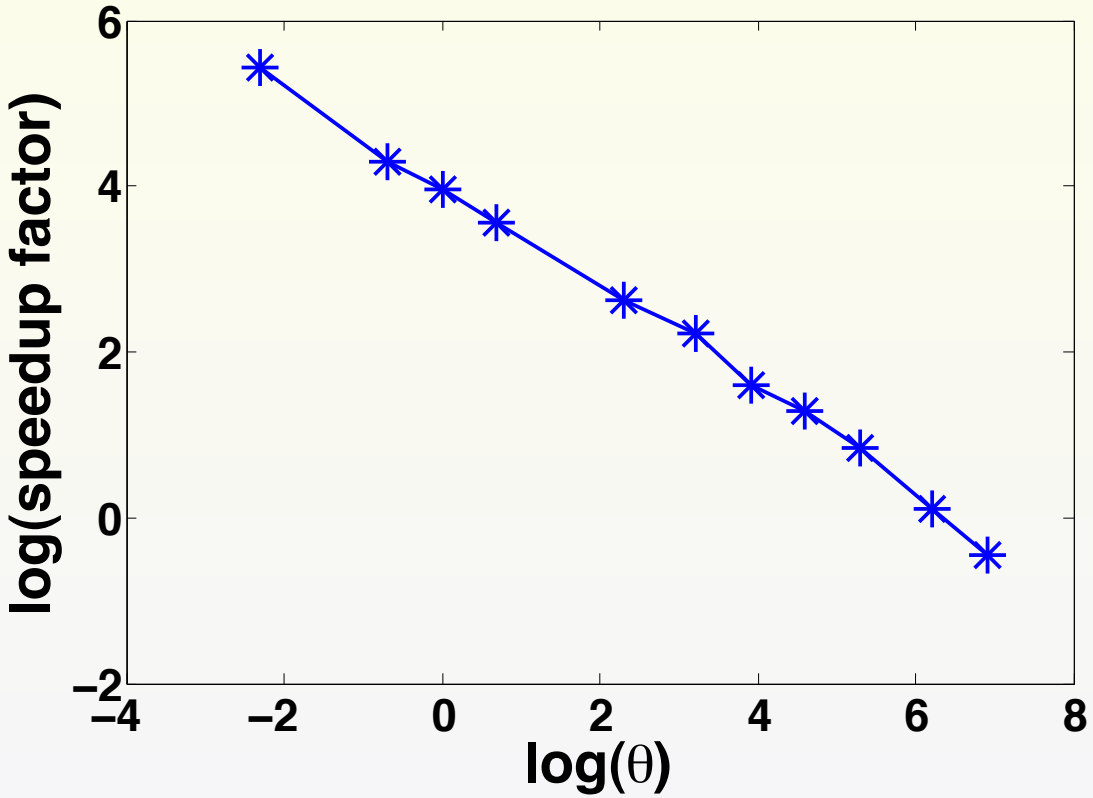
What about γ?

θ	Method	Estimate of $X_A(1)$	CPU time	Speedup
.1	Crude MC	9,999.97 $\pm$ 0.20	1,476.9 S	
.1	MLMC w/ $M = 3$, $L = 4$, $\ell_0 = 0$	10,000.07 $\pm$ 0.19	6.5 S	227.2
.5	Crude MC	1,999.90 $\pm$ 0.16	1,110.9 S	
.5	MLMC w/ $M = 3$, $L = 4$, $\ell_0 = 1$	2,000.10 $\pm$ 0.16	15 S	74.1
1	Crude MC	999.96 $\pm$ 0.11	1,464.4 S	
1	MLMC w/ $M = 3$, $L = 5$, $\ell_0 = 2$	999.99 $\pm$ 0.11	28 S	52.3
2	Crude MC	500.01 $\pm$ 0.11	739.9 S	
2	MLMC w/ $M = 6$, $L = 6$, $\ell_0 = 3$	499.96 $\pm$ 0.11	21	35.2
10	Crude MC	99.983 $\pm$ 0.044	900.2 S	
10	MLMC w/ $M = 3$, $L = 6$, $\ell_0 = 5$	99.965 $\pm$ 0.044	65 S	13.8
25	Crude MC	40.012 $\pm$ 0.028	898.0 S	
25	MLMC w/ $M = 3$, $L = 6$, $\ell_0 = 6$	39.996 $\pm$ 0.028	96.2 / 99.2 S	9.2
50	Crude MC	20.008 $\pm$ 0.0139	1,789.0 S	
50	MLMC w/ $M = 3$, $L = 7$, $\ell_0 = 7$	20.005 $\pm$ 0.0138	360 S	5.0
100	Crude MC	10.002 $\pm$ 0.0139	892.6 S	
100	MLMC w/ $M = 3$, $L = 7$, $\ell_0 = 7$	9.988 $\pm$ 0.0138	250 S	3.6
200	Crude MC	4.9996 $\pm$ 0.0088	1,120.3 S	
200	MLMC w/ $M = 3$, $L = 7$, $\ell_0 = 7$	4.9958 $\pm$ 0.0087	486 S	2.3
500	Crude MC	2.0029 $\pm$ 0.0044	1,781.6 S	
500	MLMC w/ $M = 3$, $L = 7$, $\ell_0 = 7$	1.9953 $\pm$ 0.0044	1,625.9 S	1.1
1,000	Crude MC	1.0038 $\pm$ 0.0043	897.2	
1,000	MLMC w/ $M = 3$, $L = 7$, $\ell_0 = 7$	1.0015 $\pm$ 0.0044	1,412.3 S	0.64

What about γ?

It appears that

$$\text{Speedup factor} = 54.6\,\theta^{-0.62}.$$



Log-log plot for the speedup factor of unbiased MLMC over an exact method. The best fit line, not shown, is

$$f(x) = 4 - 0.62x$$

.

Next problem: parameter sensitivities.

Motivated by Jim Rawlings.

Next problem: parameter sensitivities.

Motivated by Jim Rawlings.

We have

$$X^\theta(t) = X^\theta(0) + \sum_k Y_k \left(\int_0^t \lambda_k^\theta(X^\theta(s))ds \right) \zeta_k.$$

and we define

$$J(\theta) = \mathbb{E}f(X^\theta(t)].$$

Next problem: parameter sensitivities.

Motivated by Jim Rawlings.

We have

$$X^\theta(t) = X^\theta(0) + \sum_k Y_k \left(\int_0^t \lambda_k^\theta(X^\theta(s))ds \right) \zeta_k.$$

and we define

$$J(\theta) = \mathbb{E}f(X^\theta(t)].$$

We want

$$J'(\theta) = \frac{d}{d\theta}\mathbb{E}f(X^\theta(t)).$$

Next problem: parameter sensitivities.

Motivated by Jim Rawlings.

We have

$$X^\theta(t) = X^\theta(0) + \sum_k Y_k \left(\int_0^t \lambda_k^\theta(X^\theta(s))ds \right) \zeta_k.$$

and we define

$$J(\theta) = \mathbb{E}f(X^\theta(t)].$$

We want

$$J'(\theta) = \frac{d}{d\theta}\mathbb{E}f(X^\theta(t)).$$

There are multiple methods. We consider finite differences:

$$J'(\theta) \approx \frac{J(\theta+\epsilon) - J(\theta)}{\epsilon} + O(\epsilon).$$

Next problem: parameter sensitivities.

Noting that

$$J'(\theta) = \frac{d}{d\theta}\mathbb{E}f(X^{\theta}(t)) = \frac{\mathbb{E}f(X^{\theta+\epsilon}(t)) - \mathbb{E}f(X^{\theta}(t))}{\epsilon} + o(\epsilon).$$

The usual finite difference estimator is

$$D_R(\epsilon) = \epsilon^{-1}\left[\frac{1}{R}\sum_{i=1}^{R} f(X_{[i]}^{\theta+\epsilon}(t)) - \frac{1}{R}\sum_{j=1}^{R} f(X_{[j]}^{\theta}(t))\right]$$

Next problem: parameter sensitivities.

Noting that

$$J'(\theta) = \frac{d}{d\theta}\mathbb{E}f(X^{\theta}(t)) = \frac{\mathbb{E}f(X^{\theta+\epsilon}(t)) - \mathbb{E}f(X^{\theta}(t))}{\epsilon} + o(\epsilon).$$

The usual finite difference estimator is

$$D_R(\epsilon) = \epsilon^{-1}\left[\frac{1}{R}\sum_{i=1}^{R} f(X_{[i]}^{\theta+\epsilon}(t)) - \frac{1}{R}\sum_{j=1}^{R} f(X_{[j]}^{\theta}(t))\right]$$

If generated independently, then

$$\mathrm{Var}(D_R(\epsilon)) = O(R^{-1}\epsilon^{-2}).$$

Next problem: parameter sensitivities.

Couple the processes.

$$X^{\theta+\epsilon}(t) = X^{\theta+\epsilon}(0) + \sum_k Y_{k,1}\left(\int_0^t \lambda_k^{\theta+\epsilon}(X^{\theta+\epsilon}(s)) \wedge \lambda_k^{\theta}(X^{\theta}(s))ds\right)\zeta_k$$
$$+ \sum_k Y_{k,2}\left(\int_0^t \lambda_k^{\theta+\epsilon}(X^{\theta+\epsilon}(s)) - \lambda_k^{\theta+\epsilon}(X^{\theta+\epsilon}(s)) \wedge \lambda_k^{\theta}(X^{\theta}(s))ds\right)\zeta_k$$
$$X^{\theta}(t) = X^{\theta}(0) + \sum_k Y_{k,1}\left(\int_0^t \lambda_k^{\theta+\epsilon}(X^{\theta+\epsilon}(s)) \wedge \lambda_k^{\theta}(X^{\theta}(s))ds\right)\zeta_k$$
$$+ \sum_k Y_{k,3}\left(\int_0^t \lambda_k^{\theta}(X^{\theta}(s)) - \lambda_k^{\theta+\epsilon}(X^{\theta+\epsilon}(s)) \wedge \lambda_k^{\theta}(X^{\theta}(s))ds\right)\zeta_k,$$

Use:

$$D_R(\epsilon) = \epsilon^{-1}\frac{1}{R}\sum_{i=1}^{R}\left[f(X_{[i]}^{\theta+\epsilon}(t)) - f(X_{[i]}^{\theta}(t))\right].$$

Next problem: parameter sensitivities.

Theorem (Anderson, 2011)

Suppose $(X^{\theta+\epsilon}, X^{\theta})$ *satisfy coupling. Then, for any* $T > 0$ *there is a* $C_{T,f} > 0$ *for which*

$$\mathbb{E} \sup_{t \leq T} \left(f(X^{\theta+\epsilon}(t)) - f(X^{\theta}(t)) \right)^2 \leq C_{T,f} \epsilon.$$

[1] David F. Anderson, *An efficient Finite Difference Method for Parameter Sensitivities of Continuous Time Markov Chains*. Submitted. Available at `arxiv.org:1109.2890`. Also at `www.math.wisc.edu/~anderson`.

Next problem: parameter sensitivities.

Theorem (Anderson, 2011)

Suppose $(X^{\theta+\epsilon}, X^{\theta})$ *satisfy coupling. Then, for any* $T > 0$ *there is a* $C_{T,f} > 0$ *for which*

$$\mathbb{E} \sup_{t \leq T} \left(f(X^{\theta+\epsilon}(t)) - f(X^{\theta}(t)) \right)^2 \leq C_{T,f} \epsilon.$$

This lowers variance of estimator from

$$O(R^{-1} \epsilon^{-2}),$$

to

$$O(R^{-1} \epsilon^{-1}).$$

Lowered by order of magnitude (in ϵ).

---

[1] David F. Anderson, *An efficient Finite Difference Method for Parameter Sensitivities of Continuous Time Markov Chains*. Submitted. Available at `arxiv.org:1109.2890`. Also at `www.math.wisc.edu/~anderson`.

Parameter Sensitivities

$$G \xrightarrow{2} G + M,$$

$$M \xrightarrow{10} M + P,$$

$$M \xrightarrow{k} \emptyset,$$

$$P \xrightarrow{1} \emptyset.$$

Want

$$\frac{\partial}{\partial k}\mathbb{E}\left[X^k_{\text{protein}}(30)\right], \quad k \approx 1/4.$$

Parameter Sensitivities

$$G \xrightarrow{2} G + M,$$

$$M \xrightarrow{10} M + P,$$

$$M \xrightarrow{k} \emptyset,$$

$$P \xrightarrow{1} \emptyset.$$

Want

$$\frac{\partial}{\partial k}\mathbb{E}\left[X^k_{\text{protein}}(30)\right], \quad k \approx 1/4.$$

Method	# paths	Approximation	# updates	CPU Time
Likelihood ratio	689,600	-312.1 $\pm$ 6.0	2.9×10^9	3,506.6 S
Exact	246,200	-318.8 $\pm$ 6.0	2.1×10^9	3,282.1 S
CRP (RTC 1)	26,320	-320.7 $\pm$ 6.0	2.2×10^8	410.0 S
CFD (RTC 2)	4,780	-321.2 $\pm$ 6.0	2.1×10^7	35.3 S

Analysis

Theorem

Suppose $(X^{\theta+\epsilon}, X^{\theta})$ *satisfy coupling. Then, for any* $T > 0$ *there is a* $C_{T,f} > 0$ *for which*

$$\mathbb{E} \sup_{t \leq T} \left(f(X^{\theta+\epsilon}(t)) - f(X^{\theta}(t)) \right)^2 \leq C_{T,f} \epsilon.$$

Proof:

Analysis

Theorem

Suppose $(X^{\theta+\epsilon}, X^{\theta})$ *satisfy coupling. Then, for any* $T > 0$ *there is a* $C_{T,f} > 0$ *for which*

$$\mathbb{E} \sup_{t \leq T} \left(f(X^{\theta+\epsilon}(t)) - f(X^{\theta}(t)) \right)^2 \leq C_{T,f} \epsilon.$$

Proof:

Key observation of proof:

$$X^{\theta+\epsilon}(t) - X^{\theta}(t) = M^{\theta,\epsilon}(t) + \int_0^t F^{\theta+\epsilon}(X^{\theta+\epsilon}(s)) - F^{\theta}(X^{\theta}(s)) ds.$$

Now work on Martingale and absolutely continuous part.

Thanks!

References:

1. David F. Anderson and Desmond J. Higham, *Multi-level Monte Carlo for continuous time Markov chains, with applications in biochemical kinetics*, to appear in SIAM: Multiscale Modeling and Simulation.

   Available at `arXiv.org:1107.2181`. Also on my website: `www.math.wisc.edu/~anderson`.

2. David F. Anderson, *Efficient Finite Difference Method for Parameter Sensitivities of Continuous time Markov Chains*, submitted.

   Available at `arXiv.org:1109.2890`. Also on my website: `www.math.wisc.edu/~anderson`.

Funding: NSF-DMS-1009275.