

# Computational methods for continuous time Markov chains with applications to biological processes

David F. Anderson\*

\*anderson@math.wisc.edu

Department of Mathematics  
University of Wisconsin - Madison

Penn. State

January 13th, 2012

# Stochastic Models of Biochemical Reaction Systems

- ▶ Most common stochastic models of biochemical reaction systems are **continuous time Markov chains**.
- ▶ Often called **chemical master equation** type models in biosciences.

Common examples include:

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

# Stochastic Models of Biochemical Reaction Systems

- ▶ Most common stochastic models of biochemical reaction systems are **continuous time Markov chains**.
- ▶ Often called **chemical master equation** type models in biosciences.

Common examples include:

1. Gene regulatory networks.
2. Models of viral infection.
3. 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"

# Stochastic Models of Biochemical Reaction Systems

Path-wise methods can approximate values such as

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

For example,

1. Means:  $f(x) = x_j$ .
2. Moments/variances:  $f(x) = x_j^2$ .
3. Probabilities:  $f(x) = \mathbf{1}_{\{x \in A\}}$ .

or compute sensitivities

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

# Stochastic Models of Biochemical Reaction Systems

Path-wise methods can approximate values such as

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

For example,

1. Means:  $f(x) = x_j$ .
2. Moments/variances:  $f(x) = x_j^2$ .
3. Probabilities:  $f(x) = \mathbf{1}_{\{x \in A\}}$ .

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,

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

So, if we want

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

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,

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

So, if we want

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

we need

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

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

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

Can be bad if (i)  $\bar{N}$  is large, or (ii)  $\epsilon$  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  $O(\bar{N}\epsilon^{-2})$  could be prohibitively large.
2. For our models, we often have that  $\bar{N}$  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



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



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

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

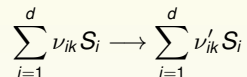
where  $Y$  is a **unit-rate** Poisson point process.

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**,  $S_1, \dots, S_d$ :

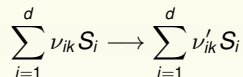


Denote reaction vector as

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

## Build up model: Random time change representation of Tom Kurtz

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



Denote reaction vector as

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

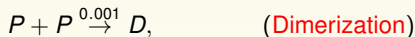
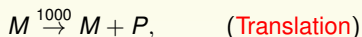
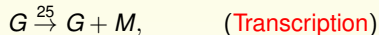
- The **intensity** (or **propensity**) of  $k$ th reaction is  $\lambda_k : \mathbb{Z}_{\geq 0}^d \rightarrow \mathbb{R}$ .
- By analogy with before

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

$Y_k$  are independent, unit-rate Poisson processes.

## Example

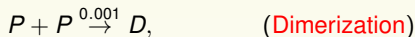
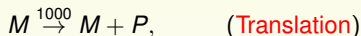
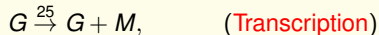
Consider a model of gene transcription and translation:



Then, if  $X = [X_M, X_P, X_D]^T$ ,

## Example

Consider a model of gene transcription and translation:



Then, if  $X = [X_M, X_P, X_D]^T$ ,

$$\begin{aligned} X(t) = X(0) &+ Y_1 (25t) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + Y_2 \left( 1000 \int_0^t X_M(s) ds \right) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ &+ Y_3 \left( 0.001 \int_0^t X_P(s)(X_P(s) - 1) ds \right) \begin{bmatrix} 0 \\ -2 \\ 1 \end{bmatrix} \\ &+ Y_4 \left( 0.1 \int_0^t X_M(s) ds \right) \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} + Y_5 \left( 1 \int_0^t X_P(s) ds \right) \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \end{aligned}$$

## 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  $O(\bar{N}\epsilon^{-2})$  could be prohibitively large.
2. For our models, we often have that  $\bar{N}$  is very large.

Let's try an **approximate scheme**.

# Tau-leaping: Euler's method

Explicit tau-leaping<sup>1</sup> or Euler's method, was first formulated by Dan Gillespie in this setting .

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

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

---

<sup>1</sup>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

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

## 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

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

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

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

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

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

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  $\epsilon^{-1} \ll \bar{N}$ .

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

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

## Benefits/drawbacks

### Benefits:

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

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

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

### 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  $X \approx \hat{Z}_L$ , and  $\hat{Z}_L$  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  $X \approx \hat{Z}_L$ , and  $\hat{Z}_L$  is cheap.
- ▶ Suppose  $X, \hat{Z}_L$  can be generated simultaneously so that

$$\text{Var}(X - \hat{Z}_L)$$

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  $X \approx \hat{Z}_L$ , and  $\hat{Z}_L$  is cheap.
- ▶ Suppose  $X, \hat{Z}_L$  can be generated simultaneously so that

$$\text{Var}(X - \hat{Z}_L)$$

is small.

- ▶ Then use

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

## 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  $X \approx \hat{Z}_L$ , and  $\hat{Z}_L$  is cheap.
- ▶ Suppose  $X, \hat{Z}_L$  can be generated simultaneously so that

$$\text{Var}(X - \hat{Z}_L)$$

is small.

- ▶ Then use

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

## 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  $X \approx \hat{Z}_L$ , and  $\hat{Z}_L$  is cheap.
- ▶ Suppose  $X, \hat{Z}_L$  can be generated simultaneously so that

$$\text{Var}(X - \hat{Z}_L)$$

is small.

- ▶ Then use

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

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

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

## 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

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

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

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

# 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 appropriate choices of  $n_0$ ,  $n_\ell$ , and  $n_E$ , we define the estimators for the three terms above via

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

## 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 appropriate choices of  $n_0$ ,  $n_\ell$ , and  $n_E$ , we define the estimators for the three terms above via

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

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

## 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 appropriate choices of  $n_0$ ,  $n_\ell$ , and  $n_E$ , we define the estimators for the three terms above via

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

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

$$\widehat{Q}_0 \stackrel{\text{def}}{=} \frac{1}{n_0} \sum_{i=1}^{n_0} f(Z_{\ell_0,[i]}(T)),$$

and note that

$$\widehat{Q} \stackrel{\text{def}}{=} \widehat{Q}_E + \sum_{\ell=\ell_0+1}^L \widehat{Q}_\ell + \widehat{Q}_0$$

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 appropriate choices of  $n_0$ ,  $n_\ell$ , and  $n_E$ , we define the estimators for the three terms above via

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

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

$$\widehat{Q}_0 \stackrel{\text{def}}{=} \frac{1}{n_0} \sum_{i=1}^{n_0} f(Z_{\ell_0,[i]}(T)),$$

and note that

$$\widehat{Q} \stackrel{\text{def}}{=} \widehat{Q}_E + \sum_{\ell=\ell_0+1}^L \widehat{Q}_\ell + \widehat{Q}_0$$

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

$$\begin{aligned}Z_{13.1}(t) &= Y_1(13.1t), \\Z_{13}(t) &= Y_2(13t),\end{aligned}$$

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

The variance of difference is large:

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

## 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:

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

## 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 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 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

$$X(t) = X(0) + \sum_k Y_{k,1} \left( \int_0^t \lambda_k(X(s)) \wedge \lambda_k(Z_e \circ \eta_e(s)) ds \right) \zeta_k$$

$$+ \sum_k Y_{k,2} \left( \int_0^t \lambda_k(X(s)) - \lambda_k(X(s)) \wedge \lambda_k(Z_e \circ \eta_e(s)) ds \right) \zeta_k$$

$$Z_e(t) = Z_e(0) + \sum_k Y_{k,1} \left( \int_0^t \lambda_k(X(s)) \wedge \lambda_k(Z_e \circ \eta_e(s)) ds \right) \zeta_k$$

$$+ \sum_k Y_{k,3} \left( \int_0^t \lambda_k(Z_e \circ \eta_e(s)) - \lambda_k(X(s)) \wedge \lambda_k(Z_e \circ \eta_e(s)) ds \right) \zeta_k$$

## 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

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

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

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

## For approximate processes

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

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

Algorithm for simulation is equivalent in to  $\tau$ -leaping.



# Multi-level Monte Carlo: chemical kinetic setting

Can prove:

Theorem (Anderson, Higham 2011)

Suppose  $(X, Z_\ell)$  satisfy coupling. Then,

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

---

<sup>1</sup>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](http://arxiv.org:1107.2181). Also at [www.math.wisc.edu/~anderson](http://www.math.wisc.edu/~anderson).

# Multi-level Monte Carlo: chemical kinetic setting

Can prove:

Theorem (Anderson, Higham 2011)

Suppose  $(X, Z_\ell)$  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,

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

---

<sup>1</sup>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](http://arxiv.org:1107.2181). Also at [www.math.wisc.edu/~anderson](http://www.math.wisc.edu/~anderson).

## Multi-level Monte Carlo: an unbiased estimator

For well chosen  $n_0, n_\ell$ , and  $n_E$ . We have

$$\text{Var}(\widehat{Q}) = \text{Var} \left( \widehat{Q}_E + \sum_{\ell=\ell_0+1}^L \widehat{Q}_\ell + \widehat{Q}_0 \right) = O(\epsilon^2),$$

with

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

# 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.

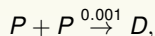
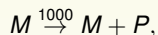
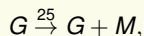
# Multi-level Monte Carlo: an unbiased estimator

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**.

## Example

Consider a model of gene transcription and translation:



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!)	$8.27 \times 10^{10}$

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	$6.2 \times 10^{10}$
$h = 3^{-6}$	$3,707.5 \pm 1.0$	4,750,000	6,207.9 S	$2.1 \times 10^{10}$
$h = 3^{-5}$	$3,693.4 \pm 1.0$	4,700,000	2,803.9 S	$6.9 \times 10^9$
$h = 3^{-4}$	$3,654.6 \pm 1.0$	4,650,000	1,219.0 S	$2.6 \times 10^9$

## 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!)	$8.27 \times 10^{10}$

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	$6.2 \times 10^{10}$
$h = 3^{-6}$	$3,707.5 \pm 1.0$	4,750,000	6,207.9 S	$2.1 \times 10^{10}$
$h = 3^{-5}$	$3,693.4 \pm 1.0$	4,700,000	2,803.9 S	$6.9 \times 10^9$
$h = 3^{-4}$	$3,654.6 \pm 1.0$	4,650,000	1,219.0 S	$2.6 \times 10^9$

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	$1.1 \times 10^9$
$M = 3, L = 5$	$3,714.7 \pm 1.0$	1,114.9 S	$9.4 \times 10^8$
$M = 3, L = 4$	$3,714.2 \pm 1.0$	1,656.6 S	$1.0 \times 10^9$
$M = 4, L = 4$	$3,714.2 \pm 1.0$	1,334.8 S	$1.1 \times 10^9$
$M = 4, L = 5$	$3,713.8 \pm 1.0$	1,014.9 S	$1.1 \times 10^9$

## 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!)	$8.27 \times 10^{10}$

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	$6.2 \times 10^{10}$
$h = 3^{-6}$	$3,707.5 \pm 1.0$	4,750,000	6,207.9 S	$2.1 \times 10^{10}$
$h = 3^{-5}$	$3,693.4 \pm 1.0$	4,700,000	2,803.9 S	$6.9 \times 10^9$
$h = 3^{-4}$	$3,654.6 \pm 1.0$	4,650,000	1,219.0 S	$2.6 \times 10^9$

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	$1.1 \times 10^9$
<b><math>M = 3, L = 5</math></b>	<b><math>3,714.7 \pm 1.0</math></b>	<b>1,114.9 S</b>	<b><math>9.4 \times 10^8</math></b>
$M = 3, L = 4$	$3,714.2 \pm 1.0$	1,656.6 S	$1.0 \times 10^9$
$M = 4, L = 4$	$3,714.2 \pm 1.0$	1,334.8 S	$1.1 \times 10^9$
$M = 4, L = 5$	$3,713.8 \pm 1.0$	1,014.9 S	$1.1 \times 10^9$

- ▶ 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**:

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

## 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. **Con:** Is substantially harder to implement; good software is needed.



## 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. **Con:** Is substantially harder to implement; good software is needed.
7. **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:



$$\kappa_1 = 1$$



$$\kappa_2 = 0.025$$



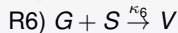
$$\kappa_3 = 1000$$



$$\kappa_4 = 0.25$$



$$\kappa_5 = 2$$



$$\kappa_6 = 7.5 \times 10^{-6}$$

- ▶ 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

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

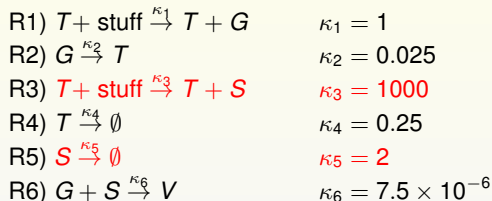
$$X_2(t) = X_2(0) + Y_3 \left( 1000 \int_0^t X_3(s) ds \right) - Y_5 \left( 2 \int_0^t X_2(s) ds \right) \\ - Y_6 \left( 7.5 \times 10^{-6} \int_0^t X_1(s) X_2(s) ds \right)$$

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

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

## Another example: Viral infection

Reactions:

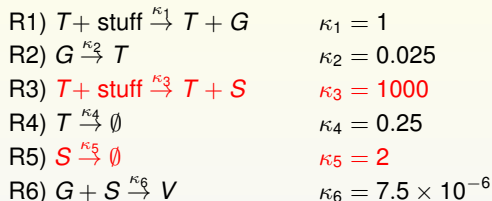


If  $T > 0$ ,

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

## Another example: Viral infection

Reactions:



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

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

---

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

## 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
$13.85 \pm 0.07$	75,000	24,800 CPU S	$1.45 \times 10^{10}$



## 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
$13.85 \pm 0.07$	75,000	24,800 CPU S	$1.45 \times 10^{10}$

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

Approximation	CPU Time	# updates
$13.91 \pm 0.07$	1,118.5 CPU S	$2.41 \times 10^8$

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

$$\sum_k \lambda'_k(X(\cdot)) \approx \bar{N} \gg 1.$$

There are therefore **two extreme parameters** floating around our models:

1. Some parameter  $N \gg 1$ , causing  $\bar{N} \gg 1$  (**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,  $\kappa'_k$ , may also vary over several orders of magnitude. We write

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

where the  $\beta_k$  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,  $\kappa'_k$ , may also vary over several orders of magnitude. We write

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

where the  $\beta_k$  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  $(X^N, Z_\ell^N)$  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  $(X^N, Z_\ell^N)$  satisfy coupling with  $X^N(0) = Z_\ell^N(0)$ . Then,

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

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

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



## Flavor of Proof

So,

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

## Flavor of Proof

So,

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

Hence,

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

Now work.

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**:

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

## 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

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



## 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

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

---

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

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

This lowers variance of estimator from

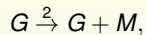
$$O(R^{-1} \epsilon^{-2}),$$

to

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

Lowered by order of magnitude (in  $\epsilon$ ).

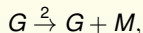
## Parameter Sensitivities



Want

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

# Parameter Sensitivities



Want

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

Method	# paths	Approximation	# updates	CPU Time
Likelihood ratio	689,600	$-312.1 \pm 6.0$	$2.9 \times 10^9$	3,506.6 S
Exact/Naive FD	246,200	$-318.8 \pm 6.0$	$2.1 \times 10^9$	3,282.1 S
CRP	26,320	$-320.7 \pm 6.0$	$2.2 \times 10^8$	410.0 S
<b>Coupled</b>	<b>4,780</b>	<b><math>-321.2 \pm 6.0</math></b>	<b><math>2.1 \times 10^7</math></b>	<b>35.3 S</b>

# 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](https://arxiv.org/abs/1107.2181). Also on my website:  
[www.math.wisc.edu/~anderson](http://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](https://arxiv.org/abs/1109.2890). Also on my website:  
[www.math.wisc.edu/~anderson](http://www.math.wisc.edu/~anderson).

Funding: NSF-DMS-1009275.