Stochastic Simulation of Models Arising in the Life Sciences

David F. Anderson*

*anderson@math.wisc.edu
Department of Mathematics
University of Wisconsin - Madison

SIAM Life Sciences 2012
August 9th, 2012
Broad Outline

1. Give two slides on facts for exponential random variables.

2. Discuss mathematical models of interest,
   - stochastic models of biochemical (population) processes.
   - continuous time Markov chains.

3. Gillespie’s algorithm

4. Derive Different stochastic representations
   - random time change.
   - next reaction method.

5. Derive “tau-leaping”

6. Derive ODEs and diffusion/Langevin approximations,

7. If time: one more advanced idea – parameter sensitivities.
The exponential random variable

Lemma
If for $i = 1, \ldots, n$, the random variables $X_i \sim \text{Exp}(\lambda_i)$ are independent, then

$$X_0 \equiv \min_i \{X_i\} \sim \text{Exp}(\lambda_0), \quad \text{where} \quad \lambda_0 = \sum_{i=1}^{n} \lambda_i.$$ 

Lemma
For $i = 1, \ldots, n$, let the random variables $X_i \sim \text{Exp}(\lambda_i)$ be independent. Let $j$ be the index of the minimum of $\{X_i\}$. Then $j$ is a discrete random variable with probability mass function

$$P\{j = i\} = \frac{\lambda_i}{\lambda_0}, \quad \text{where} \quad \lambda_0 = \sum_{i=1}^{n} \lambda_i.$$
How to generate an exponential

Suppose we want an exp(\(\lambda\)) random variable from a uniform.

Lemma
If \(U\) is uniform(0, 1) and \(\lambda > 0\), then

\[
X = \frac{\ln(1/U)}{\lambda},
\]

is exponentially distributed with a parameter of \(\lambda\).

Proof.
For \(t \geq 0\),

\[
P\{X \leq t\} = P\left\{\frac{\ln(1/U)}{\lambda} \leq t\right\}
= P\left\{1/U \leq e^{\lambda t}\right\}
= P\{e^{-\lambda t} \leq U\}
= 1 - e^{-\lambda t}.
\]

\(\square\)
What type of models:

Substrate-enzyme kinetics:

\[ S + E \xrightleftharpoons[\kappa^{-1}]{\kappa^1} [SE] \xrightarrow{\kappa^2} P + E. \]

Gene transcription & translation:

\[ G \xrightarrow{\kappa^1} G + M \quad \text{transcription} \]
\[ M \xrightarrow{\kappa^2} M + P \quad \text{translation} \]
\[ M \xrightarrow{\kappa^3} \emptyset \quad \text{degradation} \]
\[ P \xrightarrow{\kappa^4} \emptyset \quad \text{degradation} \]
\[ G + P \xrightleftharpoons[\kappa^{-5}]{\kappa^5} B \quad \text{Binding/unbinding of Gene} \]

Cartoon representation:

\[ \text{Gene (n1), mRNA (n2), Protein (n3)} \]

\[ \text{off, on, degradation, binding/unbinding of Gene} \]

---

\[ ^1 \text{J. Paulsson, Physics of Life Reviews, 2, 2005 157 – 175.} \]
Another example: Viral infection

Let

1. \( T = \) viral template.
2. \( G = \) viral genome.
3. \( S = \) viral structure.
4. \( V = \) virus.

Reactions:

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

Population Example: Lotka-Volterra predator-prey model

Think of $A$ as a prey and $B$ as a predator.

\[ A \xrightarrow{\kappa_1} 2A, \quad A + B \xrightarrow{\kappa_2} 2B, \quad B \xrightarrow{\kappa_3} \emptyset, \]

with $X_A(0) = X_B(0) = 1000$ and $\kappa_1 = 2, \kappa_2 = .002, \kappa_3 = 2$.

Deterministic model. Let $x(t) = [A(t), B(t)]^T$.

\[
x'(t) = \kappa_1 x_1(t) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \kappa_2 x_1(t)x_2(t) \begin{bmatrix} -1 \\ 1 \end{bmatrix} + \kappa_3 x_2(t) \begin{bmatrix} 0 \\ -1 \end{bmatrix}
\]

or

\[
x(t) = x(0) + \kappa_1 \int_0^t x_1(s)ds \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \kappa_2 \int_0^t x_1(s)x_2(s)ds \begin{bmatrix} -1 \\ 1 \end{bmatrix} + \kappa_3 \int_0^t x_2(s)ds \begin{bmatrix} 0 \\ -1 \end{bmatrix}
\]
What type of models:

Modeling choices:

1. **Discrete** vs. continuous state space.
2. Deterministic vs **stochastic** dynamics.

Deterministic dynamics (continuous):

- Write down a system of ODEs or integral equations.

Discrete stochastic dynamics. Make the following assumptions:

- Well mixed **(Markov)**,
- state changes only occur due to reactions happening,
- the reaction choice is random,
- This gives DTMC,
- **holding times** are exponentially distributed.

It is known that we can:

- simulate paths using **Gillespie’s algorithm**,
- Write down **chemical master equation** (dynamics of probabilities),
- Write down stochastic equations.
What is Gillespie’s algorithm?

We can already state the basics of Gillespie’s algorithm.

Repeat the following steps:

1. Determine how long until the next reaction takes place: \( \Delta \).
   - Requires an appropriate exponential random variable.

2. Determine which reaction is the next to occur (discrete RV over reactions).

3. Update the state by addition/subtraction according to correct reaction and time by \( \Delta \).

4. Repeat for as long as you want.

Similar to someone simply telling you Euler’s method for an ODE, I doubt this gave you much insight into the processes.

My hope is that more mathematical equations/structure will.
Biochemical models

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

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

- $\nu_k \in \mathbb{Z}_d^d$: number of molecules of each chemical species consumed in the $k$th reaction.

- $\nu'_k \in \mathbb{Z}_d^d$: number of molecules of each chemical species created in the $k$th reaction.

Denote reaction vector as

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

- Example: each instance of the reaction $S_1 + S_2 \rightarrow S_3$ changes the state of the system by the reaction vector:

$$\zeta_k = \nu'_k - \nu_k = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}.$$

- The state of the system $X(t) \in \mathbb{Z}_d_{\geq 0}$ gives the number of molecules of each species in the system at time $t$.

- So if $k$th reaction happens at time $s \geq 0$, then

$$X(s) = X(s-) + \zeta_k.$$

- The intensity (or propensity) of $k$th reaction is $\lambda_k : \mathbb{Z}_d_{\geq 0} \rightarrow \mathbb{R}$.

NOTE:

- $\lambda_k$ gives transition rate from $x$ to $x + \zeta_k$.

- Total rate out of $x$ is $\lambda_0(x) = \sum_k \lambda_k(x)$. 
Mass-action kinetics

The standard intensity function chosen is mass-action kinetics:

\[ \lambda_k(x) = \kappa_k \left( \prod_i \nu_{ik}! \right) \left( \frac{x}{\nu_k} \right) = \kappa_k \prod_i \frac{x_i!}{(x_i - \nu_{ik})!}. \]

Example: If \( S_1 \rightarrow \text{anything} \), then \( \lambda_k(x) = \kappa_k x_1 \).

Example: If \( S_1 + S_2 \rightarrow \text{anything} \), then \( \lambda_k(x) = \kappa_k x_1 x_2 \).

Example: If \( 2S_2 \rightarrow \text{anything} \), then \( \lambda_k(x) = \kappa_k x_2(x_2 - 1) \).
Chemical master equation

Called the forward equation in mathematics.

1. For $x \in \mathbb{Z}^d_{\geq 0}$, let
   \[ P(x, t|\pi) \]
   be the probability that the system is in state $x$ at time $t$, given an initial distribution of $\pi$.

2. Then, the forward equations are
   \[
   \frac{d}{dt} P(x, t|\pi) = -\lambda_0(x)P(x, t|\pi) + \sum_k \lambda_k(x - \zeta_k)P(x - \zeta_k, t|\pi).
   \]

Some points:

1. Tough to get intuition from these equations for a given system (I think).
2. Can be solved for small, simple systems. Can then compute, for example, expectations.
3. System is often infinite dimensional and impossible to solve.

People often turn to simulation: Gillespie’s algorithm.
Gillespie’s Algorithm: simulate DTMC + exponential holding times

1. Initialize. Set the initial number of molecules of each species and set $t = 0$.

2. Calculate the intensity/propensity function, $\lambda_k$, for each reaction.

3. Set $\lambda_0 = \sum_{k=1}^{M} \lambda_k$.

4. Generate two independent uniform(0,1) random numbers $r_1$ and $r_2$.

5. Set

$$\Delta = \frac{1}{\lambda_0} \ln\left(\frac{1}{r_1}\right)$$

(equivalent to generating an exponential RV with parameter $\lambda_0$).

6. Find $\mu \in [1, \ldots, M]$ such that

$$\frac{1}{\lambda_0} \sum_{k=1}^{\mu-1} \lambda_k < r_2 \leq \frac{1}{\lambda_0} \sum_{k=1}^{\mu} \lambda_k,$$

which is equivalent to choosing from reactions $[1, \ldots, M]$ with the $k$th reaction having probability $\lambda_k/\lambda_0$.

7. Set $t = t + \Delta$ and update the number of each molecular species according to reaction $\mu$: $X \leftarrow X + \zeta_\mu$.

8. Return to step 2 or quit.
Simulation and Monte Carlo

We can now generate paths of biochemical and/or population systems and can consider the following general question:

- How can I approximate $\mathbb{E}f(X(T))$ to some desired tolerance, $\epsilon > 0$?

Note that for us (those studying biochemical/population processes), this covers many topics:

1. Straight expectations and variances:

   $$\mathbb{E}X_i(t), \quad \mathbb{E}\left[X_i(t)^2\right],$$

   for some $i \in \{1, \ldots, d\}$ and $t \geq 0$.

2. Expected average values:

   $$\mathbb{E}\left[\frac{1}{t} \int_0^t X_i(s)ds\right],$$

3. Expected hitting times,

4. Any probability,

   $$P(X(t) \in A) = \mathbb{E}[1\{X(t) \in A\}].
A standard simulation problem

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, consistent estimator

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

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

What is the computational cost?

Recall,
\[
\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^{n} f(X_{[i]}(t)).
\]

Thus,
\[
\text{Var}(\hat{\mu}_n) = O \left( \frac{1}{n} \right).
\]

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:

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

Can be bad if (i) $\bar{N}$ is large, or (ii) $\epsilon$ is small.
Benefits/drawbacks

Benefits:
1. Easy to implement.
2. Estimator 
\[ \hat{\mu}_n = \frac{1}{n} \sum_{i=1}^{n} f(X_i(t)) \]
   is unbiased and consistent.

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.

For different ideas, a better mathematical understanding of the underlying processes is then useful.

Question: is there a mathematical equation that this CTMC satisfies?

Answer: Yes, there are a few. Each needs an understanding of Poisson processes.
Background information: The Poisson process

A Poisson process, \( Y(\cdot) \), is a model for a series of random observations occurring in time.

(a) Let \( \{e_i\} \) be i.i.d. exponential random variables with parameter one.

(b) Now, put points down on a line with spacing equal to the \( e_i \):

\[
\begin{array}{cccccccc}
X & X & X & X & X & X & X & X \\
\hline
\leftrightarrow & \leftrightarrow & \leftrightarrow & \cdots & \\
e_1 & e_2 & e_3 & \cdots & t
\end{array}
\]

- Let \( Y(t) \) denote the number of points hit by time \( t \).
- In the figure above, \( Y(t) = 6 \).
The Poisson process

Let

- $Y$ be a unit rate Poisson process.
- Define $Y_\lambda(t) \equiv Y(\lambda t)$,

Then $Y_\lambda$ is a Poisson process with parameter $\lambda$.

Intuition: The Poisson process with rate $\lambda$ is simply the number of points hit (of the unit-rate point process) when we run along the time frame at rate $\lambda$. 
The Poisson process

There is no reason \( \lambda \) needs to be constant in time, in which case

\[
Y_\lambda(t) \equiv Y \left( \int_0^t \lambda(s)ds \right)
\]

is an nonhomogeneous Poisson process with propensity/intensity \( \lambda(t) \geq 0 \).

Thus

\[
P\{ Y_\lambda(t + \Delta t) - Y_\lambda(t) > 0 | \mathcal{F}_t \} = 1 - \exp \left\{ - \int_t^{t+\Delta t} \lambda(s)ds \right\} \approx \lambda(t)\Delta t.
\]

Points:

1. We have “changed time” to convert a unit-rate Poisson process to one which has rate or intensity or propensity \( \lambda(t) \).

2. Will use similar time changes of unit-rate processes to build models of interest.
Models of interest

Consider the simple system

\[ A + B \rightarrow C \]

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

**Biological intuition for standard stochastic model:**

\[ P\{\text{reaction occurs in } (t, t + \Delta t)|\mathcal{F}_t\} \approx \kappa X_A(t)X_B(t)\Delta t \]

where

- \( \kappa \) is a positive constant, the reaction rate constant.
Models of interest

\[ A + B \rightarrow C \]

Simple book-keeping: if \( X(t) = (X_A(t), X_B(t), X_C(t)) \) 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.

Goal: represent \( R(t) \) in terms of Poisson process.
Models of interest

Recall that for $A + B \rightarrow C$ our intuition was

$$P\{\text{reaction occurs in } (t, t + \Delta t) | \mathcal{F}_t \} \approx \kappa X_A(t) X_B(t) \Delta t,$$

and that for an inhomogeneous Poisson process with rate $\lambda(t)$ we have

$$P\{Y_\lambda(t + \Delta t) - Y_\lambda(t) > 0 | \mathcal{F}_t \} = 1 - \exp \left\{ - \int_0^{\Delta t} \lambda(t) dt \right\} \approx \lambda(t) \Delta t.$$

This suggests 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 process.

Hence

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

This equation uniquely determines $X$ for all $t \geq 0$. 
General stochastic models of biochemical reactions

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

  \[ \sum_{i=1}^{d} \nu_{ik} S_i \rightarrow \sum_{i=1}^{d} \nu'_{ik} S_i \]

- $\nu_k \in \mathbb{Z}_{\geq 0}^d$: number of molecules of each chemical species consumed in the $k$th reaction.

- $\nu'_k \in \mathbb{Z}_{\geq 0}^d$: number of molecules of each chemical species created in the $k$th reaction.

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.
Think of \( A \) as a prey and \( B \) as a predator.

\[
\begin{align*}
A & \xrightarrow{\kappa_1} 2A, \\
A + B & \xrightarrow{\kappa_2} 2B, \\
B & \xrightarrow{\kappa_3} \emptyset,
\end{align*}
\]

with \( X_A(0) = X_B(0) = 1000 \) and \( \kappa_1 = 2, \kappa_2 = .002, \kappa_3 = 2. \)

**Deterministic model.** Let \( x(t) = [A(t), B(t)]^T \).

\[
x(t) = x(0) + \kappa_1 \int_0^t x_1(s)ds \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \kappa_2 \int_0^t x_1(s)x_2(s)ds \begin{bmatrix} -1 \\ 1 \end{bmatrix} + \kappa_3 \int_0^t x_2(s)ds \begin{bmatrix} 0 \\ -1 \end{bmatrix}
\]

**Stochastic model.** Let \( X(t) = [A(t), B(t)]^T \).

\[
X(t) = X(0) + Y_1 \left( \kappa_1 \int_0^t X_1(s)ds \right) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + Y_2 \left( \kappa_2 \int_0^t X_1(s)X_2(s)ds \right) \begin{bmatrix} -1 \\ 1 \end{bmatrix} + Y_3 \left( \kappa_3 \int_0^t X_2(s)ds \right) \begin{bmatrix} 0 \\ -1 \end{bmatrix}
\]
Lotka-Volterra

Think of \( A \) as a prey and \( B \) as a predator.

\[
A \xrightarrow{\kappa_1} 2A, \quad A + B \xrightarrow{\kappa_2} 2B, \quad B \xrightarrow{\kappa_3} \emptyset,
\]

with \( A(0) = B(0) = 1000 \) and \( \kappa_1 = 2, \kappa_2 = .002, \kappa_3 = 2. \)
Wait! I just pulled a fast one on you.

1. I told you how to simulate a CTMC (DTMC + holding times)

2. Then developed a CTMC model for biochemical processes.

3. Is simulation of the model I developed equivalent to the simulation strategy I proposed for CTMCs?

Answer: no, but this is more subtle than you might expect.

\[ X(t) = X(0) + \sum_k Y_k \left( \int_0^t \lambda_k(X(s)) \, ds \right) \zeta_k. \]
Statistically exact simulation methods

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

**Alternative to the Gillespie algorithm:** at time \( t \) let

\[ T_k = \int_0^t \lambda_k(X(s)) \, ds \]

\[ P_k = \min\{ s > T_k : Y_k(s) > Y_k(T_k) \}. \]

Assuming no other reactions fire first, amount of time until reaction \( k \) fires given by

\[ \lambda_k \Delta t_k = P_k - T_k \]

\[ \Rightarrow \Delta t_k = \frac{P_k - T_k}{\lambda_k}. \]
The (modified) Next Reaction Method: simulating the random time change representation

1. Initialize system and set $T_k = 0$ for all $k$.

2. For each $k$, set $P_k = \ln(1/r_k)$, where $r_k \sim \text{unif}(0, 1)$.

3. For each $k$ set: $\Delta t_k = (P_k - T_k)/\lambda_k$.

4. $\Delta = \min\{\Delta t_k\}$. Reaction that fires is where min. achieved.

5. Set $P_\mu = P_\mu + \ln(1/r)$, $r \sim \text{unif}(0, 1)$.

6. For each $k$, $T_k = T_k + \lambda_k \Delta$.

7. Update system and propensities.

8. Return to step 3 or quit.
Representation of Gillespie’s algorithm

**Question:** Can we derive a stochastic representation for Gillespie’s algorithm.

What do I need?

1. DTMC: uniform random variables!
   - Let \( \{U_0, U_1, U_2, \ldots \} \) be independent uniform(0,1) RVs.

2. Exponential holding times with rates \( \lambda_0(X(t)) \).
   - Let \( Y \) be a unit-rate Poisson process: will run at rate \( \lambda_0(X(t)) \).

Then, let

\[
R_0(t) = Y \left( \int_0^t \lambda_0(X(s))ds \right)
\]

\[
X(t) = X(0) + \sum_k \zeta_k \int_0^t 1_{(q_{k-1}(X(s-)), q_k(X(s-))]}(U_{R_0(s-)})dR_0(s),
\]

where

\[
q_k(x) = \sum_{\ell=1}^k \lambda_{\ell}(x)
\]
Recap

We now have two representations and, hence, simulation strategies for the generation of sample paths. One is more useful than the other:

1. Random time change representation of Tom Kurtz, which looks more like integrated ODE:

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

\[ Y_k \] are independent, unit-rate Poisson processes.

2. Gillespie’s algorithm (thinning of a Poisson process):

\[ R_0(t) = Y \left( \int_0^t \lambda_0(X(s)) ds \right) \]

\[ X(t) = X(0) + \sum_k \zeta_k \int_0^t \mathbf{1}_{(q_{k-1}(X(s-)), q_k(X(s-))]} \left( U_{R_0(s-)} \right) dR_0(s), \]

where

\[ q_k(x) = \sum_{\ell=1}^k \lambda_\ell(x) \]
At this point you could/should/may be asking....

Who cares?

I will try to convince you that the RTC

1. yields different computational/simulation methods,

2. gives more analytical understanding of the models
   – “Itô equations” for these models.
Deriving “Tau-leaping” is a snap with RTC.

Explicit “τ-leaping” \(^2\) was developed by Dan Gillespie in an effort to overcome the problem that each time-step may be prohibitively small.

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

\[
Z(\tau) = Z(0) + \sum_k Y_k \left( \int_0^\tau \lambda_k(Z(s)) \, ds \right) \zeta_k
\]

\[
\approx Z(0) + \sum_k Y_k \left( \lambda_k(Z(0)) \tau \right) \zeta_k
\]

\[
\overset{d}{=} Z(0) + \sum_k \text{Poisson}\left( \lambda_k(Z(0)) \tau \right) \zeta_k.
\]

One “intuitive” representation for $Z(t)$ generated by $\tau$-leaping is

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

where

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

is a step function giving left endpoints of time discretization.

**Question**: Is there a point for **Diffusion approximation** if only going to use to simulate?
What do we know about Poisson processes?

Recall that for a Poisson process, $Y$, the law of large numbers says

$$\frac{1}{N} Y(Nu) \approx u,$$

when $N \gg 1$.

What about fluctuations? The CLT gives

$$\frac{1}{N} Y(Nu) \approx u + \frac{1}{\sqrt{N}} W(u),$$

What do these imply for us?
Recall that process satisfies

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

Suppose that each \( X_i(t) = O(N) \) for some large \( N \gg 1 \). Then, scaling by \( N \) yields the equations:

\[ X^N(t) = X^N(0) + \sum_k \frac{1}{N} Y_k \left( N \int_0^t \lambda_k(X^N(s)) ds \right) \zeta_k. \]

So, LLN gives

\[ X^N(t) \approx X^N(0) + \sum_k \zeta_k \int_0^t \lambda_k(X^N(s)) ds \]

Hence, a good ODE approximation is

\[ x'(t) = \sum_k \zeta_k \lambda_k(x(t)). \]
Please note the very common error: taking expectations of

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

tyields

\[ \mathbb{E}X(t) = \mathbb{E}X(0) + \sum_k \zeta_k \int_0^t \mathbb{E}\lambda_k(X(s))ds. \]

This is NOT the same as

\[ \mathbb{E}X(t) = \mathbb{E}X(0) + \sum_k \zeta_k \int_0^t \lambda_k(\mathbb{E}X(s))ds. \]

Which is equivalent to the usual ODE

\[ x'(t) = \sum_k \zeta_k \lambda_k(x). \]
Diffusion/Langevin approximation

Recall that Poisson processes satisfy (by the CLT)

\[ \frac{1}{N} Y(Nu) \approx u + \frac{1}{\sqrt{N}} W(u), \]

where \( W \) is a Brownian motion.

Hence,

\[ X_N^N(t) = X_N^N(0) + \sum_k \frac{1}{N} Y_k \left( N \int_0^t \lambda_k(X_N^N(s))ds \right) \zeta_k \]

\[ \approx X_N^N(0) + \sum_k \zeta_k \int_0^t \lambda_k(X_N^N(s))ds + \sum_k \zeta_k \frac{1}{\sqrt{N}} W_k \left( \int_0^t \lambda_k(X_N^N(s))ds \right) \]

or

\[ X_N^N(t) \approx X_N^N(0) + \sum_k \zeta_k \int_0^t \lambda_k(X_N^N(s))ds + \sum_k \zeta_k \frac{1}{\sqrt{N}} \int_0^t \sqrt{\lambda_k(X_N^N(s))}dW_k \]

Suggests approximate process of

\[ D_N^N(t) = X_N^N(0) + \sum_k \zeta_k \int_0^t \lambda_k(D_N^N(s))ds + \sum_k \zeta_k \frac{1}{\sqrt{N}} \int_0^t \sqrt{\lambda_k(D_N^N(s))}dW_k \]

which can be simulated. (Why not use tau-leaping, though?)
A specific computational problem: Gradient estimation

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 know how to estimate $J(\theta)$ using Monte Carlo.

However, what if we want

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

Thus, we want to know how sensitive our statistic is to perturbations in $\theta$.

Tells us:

1. How best to control a system for desired outcome.
2. Robustness of system to perturbations in parameters.
3. Which parameters we need to estimate well from data, etc.

There are multiple methods. We will consider:

- Finite differences.
Finite differencing
This method is pretty straightforward and is therefore used most.

Simply note that

\[ J'(\theta) \approx \frac{J(\theta + \epsilon) - J(\theta)}{\epsilon} = \frac{\mathbb{E} f(X^{\theta+\epsilon}(t)) - \mathbb{E} f(X^\theta(t))}{\epsilon} + O(\epsilon). \]

Centered differencing reduces bias to \( O(\epsilon^2) \).

The usual finite difference estimator is

\[ D_R(\epsilon) = \frac{1}{R} \sum_{i=1}^{R} \frac{f(X^{\theta+\epsilon}_{[i]}(t)) - f(X^\theta_{[i]}(t))}{\epsilon} \]

If generated independently, then

\[ \text{Var}(D_R(\epsilon)) = R^{-1} \epsilon^{-2} \text{Var}(f(X^{\theta+\epsilon}_{[i]}(t)) - f(X^\theta_{[i]}(t))) \]

\[ = O(R^{-1} \epsilon^{-2}). \]

Terrible. Worse than expectations.

How about common random numbers for variance reduction?
Common random numbers

It’s exactly what it sounds like. Reuse the random numbers used in the generation of

\[ X_{[i]}^{\theta+\epsilon}(t) \text{ and } X_{[i]}^\theta(t). \]

Why? Because:

\[
\text{Var}(f(X_{[i]}^{\theta+\epsilon}(t)) - f(X_{[i]}^{\theta+\epsilon}(t))) = \text{Var}(f(X_{[i]}^{\theta+\epsilon}(t))) + \text{Var}(f(X_{[i]}^\theta(t))) \\
- 2\text{Cov}(f(X_{[i]}^{\theta+\epsilon}(t)), f(X_{[i]}^\theta(t))).
\]

So, if we can “couple” the random variables, we can get a variance reduction! Sometimes substantial.
In the context of Gillespie’s algorithm, we simply reuse all the same random numbers (uniforms).

This can be achieved simply by setting the “seed” of the random number generator before generating \( X^{\theta+\varepsilon} \) and \( X^{\theta} \).

Mathematically, this is equivalent to using the same Poisson process and uniforms in:

\[
R_{0}^{\theta+\varepsilon}(t) = Y \left( \int_{0}^{t} \lambda_{0}^{\theta+\varepsilon}(X^{\theta+\varepsilon}(s)) ds \right)
\]

\[
X^{\theta+\varepsilon}(t) = X^{\theta+\varepsilon}(0) + \sum_{k} \zeta_{k} \int_{0}^{t} 1_{(q_{k-1}^{\theta+\varepsilon}(X^{\theta+\varepsilon}(s-)), q_{k}^{\theta+\varepsilon}(X^{\theta+\varepsilon}(s-))]}(U_{R_{0}(s-)})(UR_{0}(s-)) \, dR_{0}^{\theta+\varepsilon}(s),
\]

\[
R_{0}^{\theta}(t) = Y \left( \int_{0}^{t} \lambda_{0}^{\theta}(X^{\theta}(s)) ds \right)
\]

\[
X^{\theta}(t) = X^{\theta}(0) + \sum_{k} \zeta_{k} \int_{0}^{t} 1_{(q_{k-1}^{\theta}(X^{\theta}(s-)), q_{k}^{\theta}(X^{\theta}(s-))]}(U_{R_{0}(s-)})(UR_{0}(s-)) \, dR_{0}^{\theta}(s),
\]
CRN + Gillespie is good idea.

1. Costs little in terms of implementation.
2. Variance reduction and gains in efficiency can be huge.

Thus, it is probably the most common method used today.

However:
- Over time, the processes decouple, often completely.

**Question**: Can we do better? Is there a natural way to couple processes using random time change?
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:

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

Using a fact: sum of homogeneous Poisson process is again a Poisson process.
Next problem: parameter sensitivities.

Couple the processes.

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

\[
X^\theta(t) = X^\theta(0) + \sum_k Y_{k,1} \left( \int_0^t \lambda_k^\theta(X^\theta(s)) \wedge \lambda_k^\theta(X^\theta(s)) ds \right) \zeta_k \\
+ \sum_k Y_{k,3} \left( \int_0^t \lambda_k^\theta(X^\theta(s)) - \lambda_k^{\theta+\epsilon}(X^{\theta+\epsilon}(s)) \wedge \lambda_k^\theta(X^\theta(s)) ds \right) \zeta_k,
\]
Next problem: parameter sensitivities.

**Theorem**

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

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

This lowers variance of estimator from

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

to

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

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

---

**Point:** a deeper mathematical understanding led to better computational method.

---

Example: birth and death

\[ \emptyset \xleftrightarrow{\theta} M. \]  

\[(1)\]
Example: genetic toggle switch

\[
\emptyset \overset{\lambda_1(X)}{\rightleftharpoons} X_1, \quad \emptyset \overset{\lambda_3(X)}{\rightleftharpoons} X_2,
\]

with intensity functions

\[
\lambda_1(X(t)) = \frac{\alpha_1}{1 + X_2(t)^\beta}, \quad \lambda_2(X(t)) = X_1(t)
\]

\[
\lambda_3(X(t)) = \frac{\alpha_2}{1 + X_1(t)^\gamma}. \quad \lambda_4(X(t)) = X_2(t),
\]

and parameter choice

\[
\alpha_1 = 50, \quad \alpha_2 = 16, \quad \beta = 2.5, \quad \gamma = 1.
\]

- Begin the process with initial condition \([0, 0]\) and
- consider the sensitivity of \(X_1\) as a function of \(\alpha_1\).
Example: genetic toggle switch

Figure: Time plot of the variance of the Coupled Finite Difference estimator versus the Common Reaction Path estimator for the model (2). Each plot was generated using 10,000 sample paths. A perturbation of \( \epsilon = 1/10 \) was used.
Simulation of CTMCs found in biology is usually relatively straightforward if you just want a few paths:

- Gillespie’s algorithm.
- Next reaction method.

These algorithms are precise simulations of different representations for the same models.

Once the naive method fails, the random time change understanding is usually the most helpful in deriving advanced computational methods:

1. tau-leaping,
2. variance reduction methods...
Thanks!

Some references:


