Multilevel Monte Carlo and Tau-Leaping

Des Higham
Department of Mathematics and Statistics
University of Strathclyde
Outline

- SDEs: weak versus strong convergence
- Complexity of Monte Carlo/Euler–Maruyama
- Multilevel Monte Carlo
- Gillespie/Tau leaping

**Multilevel Monte Carlo**


**Multi-level for Gillespie/Tau leaping**

Anderson, Higham, *SIAM Multiscale Mod. & Sim.*, 2012
Stochastic Differential Equation (SDE)

\[ dS(t) = a(S(t)) \, dt + b(S(t)) \, dW(t) \]

\( S(0) \) given and \( 0 \leq t \leq T \)

Euler–Maruyama

\[ S_{n+1} = S_n + a(S_n)h + b(S_n)\Delta W_n \]

\( \Delta W_n := W(t_{n+1}) - W(t_n), \quad t_n = nh, \quad h = T/K \)

Assume that \( a \) and \( b \) are smooth and globally Lipschitz
Weak versus Strong

Weak Convergence
\[ |\mathbb{E} [S(t_n)] - \mathbb{E} [S_n]| \leq Ch \]

Strong Convergence
\[ \mathbb{E} \left[ \sup_{0 \leq n \leq K} |S(t_n) - S_n| \right] \leq Ch^{\frac{1}{2}} \]

Strong convergence + Markov inequality \(\Rightarrow\)
\[ P \left( |S(t_n) - S_n| \geq h^\alpha \right) \leq Ch^{\frac{1}{2}} - \alpha \]

Continuous Time/Higher Moments
\[ \mathbb{E} \left[ \sup_{0 \leq t \leq T} |S(t) - S(t)|^m \right] \leq C_{m,\delta} h^{m/2 - \delta} \]
Which is more relevant, weak or strong?

Conventional wisdom:

**Weak convergence** is usually enough. Most problems require **expected value** type information.

**Strong convergence** covers cases where we want to **visualize paths** or generate **time series** (e.g. to test a filtering algorithm or a parameter fitting algorithm).
Approximate $\mathbb{E}[S(T)]$ by applying E-M to get samples.

Let $\mu = \frac{1}{N} \sum_{i=1}^{N} S[i]$

Then

$$
\mathbb{E}[S(T)] - \mu = \mathbb{E}[S(T) - S_K + S_K] - \mu \\
= \mathbb{E}[S(T) - S_K] + \mathbb{E}[S_K] - \mu
$$

Confidence interval width is $O(h) + O(1/\sqrt{N})$

For confidence interval of $O(\epsilon)$, choose $h = 1/\sqrt{N} = \epsilon$

Computational cost is $N \times 1/h$

Hence, computational complexity is $O(\epsilon^{-3})$
The **Multilevel Monte Carlo** algorithm will achieve computational complexity of

\[ O(\epsilon^{-2} \log(\epsilon)^2) \]

using E-M, and giving good results in practice.

**A key ingredient:** Use a range of \( h \) values many paths at large \( h \), few paths at small \( h \).
Approximate $\mathbb{E}[f(S(T))]$, where $f$ is globally Lipschitz. $\epsilon$ is required accuracy (conf. int.)

Timesteps $h_l = M^{-l}T, \quad l = 0, 1, 2, \ldots, L$

$M$ is fixed and $L = \frac{\log \epsilon^{-1}}{\log M}$, so that $h_L = O(\epsilon)$

$\hat{P}_l$ denotes E-M approx. to $f(S(T))$ using $h_l$. Clearly

$$\mathbb{E}[\hat{P}_L] = \mathbb{E}[\hat{P}_0] + \sum_{l=1}^{L} \mathbb{E}[\hat{P}_l - \hat{P}_{l-1}]$$

$\hat{Y}_0$ estimates $\mathbb{E}[\hat{P}_0]$ using $N_0$ paths, and $\hat{Y}_l$ estimates $\mathbb{E}[\hat{P}_l - \hat{P}_{l-1}]$ using $N_l$ paths:

$$\hat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} (\hat{P}_l^{[i]} - \hat{P}_{l-1}^{[i]})$$
Multilevel Monte Carlo ($M = 2$)
Strong convergence of E-M + glob. Lip. $f$ give

$$\text{var} \left[ \hat{P}_i - f(S(T)) \right] \leq \mathbb{E} \left[ \left( \hat{P}_i - f(S(T)) \right)^2 \right] = O(h_i)$$

and

$$\text{var} \left[ \hat{P}_i - \hat{P}_{i-1} \right]$$

$$\leq \left( \sqrt{\text{var} \left[ \hat{P}_i - f(S(T)) \right]} + \sqrt{\text{var} \left[ \hat{P}_{i-1} - f(S(T)) \right]} \right)^2 = O(h_i)$$

So $\hat{Y}_i = \frac{1}{N_i} \sum_{i=1}^{N_i} \left( \hat{P}_i^{[i]} - \hat{P}_{i-1}^{[i]} \right)$ has variance of $O(h_i/N_i)$
Recap: \( \mathbb{E}\left[\hat{P}_L\right] = \mathbb{E}\left[\hat{P}_0\right] + \sum_{l=1}^{L} \mathbb{E}\left[\hat{P}_l - \hat{P}_{l-1}\right] \)

Estimator for RHS is \( \hat{Y} := \hat{Y}_0 + \sum_{l=1}^{L} \hat{Y}_l \)

For \( l > 1 \), \( \hat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \left( \hat{P}_i^\text{[l]} - \hat{P}^\text{[l]}_{i-1} \right) \) and

\[
\text{var}\left[\hat{Y}_l\right] = O(h_l/N_l) \Rightarrow \text{var}\left[\hat{Y}\right] = \text{var}\left[\hat{Y}_0\right] + \sum_{l=1}^{L} O(h_l/N_l)
\]

Take \( N_l = O(\epsilon^{-2}Lh_l) \), to give \( \text{var}\left[\hat{Y}\right] = O(\epsilon^2) \)

Computational complexity is

\[
\sum_{l=0}^{L} N_l h_l^{-1} = \sum_{l=0}^{L} \epsilon^{-2} L h_l h_l^{-1} = L^2 \epsilon^{-2}
\]

Since \( L = \frac{\log \epsilon^{-1}}{\log M} \), this gives \( O(\epsilon^{-2}(\log \epsilon)^2) \)
Key Messages

Have to live with statistical error, so don’t be obsessed with accuracy of each sample

Can combine crude (cheap) samples with expensive (accurate) samples

Need to control the variance between pairs of paths at different refinements

Also, in the telescoping sum

\[ \mathbb{E} \left[ \hat{P}_L \right] = \mathbb{E} \left[ \hat{P}_0 \right] + \sum_{l=1}^{L} \mathbb{E} \left[ \hat{P}_l - \hat{P}_{l-1} \right] \]

if we have access to an exact sampler, we can construct a multilevel method with no bias
Consider the simple system

\[ A + B \rightarrow C \]

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

Simple book-keeping:

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

where \( R(t) \) counts reactions up to time \( t \).

Model

\[ R(t) = Y \left( \int_0^t \kappa X_A(t) X_B(t) \, ds \right) \]

where \( Y \) is a unit-rate Poisson process.
Now consider a network of reactions involving $d$ chemical species, $S_1, \ldots, S_d$:

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

Reaction vector $\zeta_k = \nu'_k - \nu_k$

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

By analogy with simple example

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

$Y_k$ are independent, unit-rate Poisson processes.
This representation is very closely linked with the **Next Reaction Method** of Gibson and Bruck (2000).

See, for example,

*A modified next reaction method for simulating chemical systems with time dependent propensities and delays*,

Essentially an Euler approximation of \( \int_0^t \lambda_k(X(s)) \, ds \):

\[
Z(h) = Z(0) + \sum_k Y_k \left( \int_0^h \lambda_k(Z(s)) \, ds \right) \zeta_k \\
\approx Z(0) + \sum_k Y_k \left( \lambda_k(Z(0)) \, h \right) \zeta_k \\
\overset{d}{=} Z(0) + \sum_k \text{Poisson} \left( \lambda_k(Z(0)) \, h \right) \zeta_k.
\]
Path-wise representation for $Z(t)$ generated by Euler

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

where

\[ \eta(s) = \left\lfloor \frac{s}{h} \right\rfloor h, \quad \implies \eta(s) = t_n \quad \text{if} \quad t_n \leq s < t_{n+1} = t_n + h \]

is a step function giving left endpoints of time discretization.
Dilemma

Need $h$ to be small for tau-leaping to be accurate. But as $h \to 0$ tau-leaping becomes arbitrarily more expensive than exact Gillespie.

To get a useful conclusion, don’t consider a fixed system. Anderson, Ganguly and Kurtz (2011) considered the limit $V \to \infty$ and $h \to 0$ with, for some $\gamma > 0$,

$$V = h^{-\gamma}$$

This relation brings together the idea of large system size (where exact simulation is expensive and tau-leaping has an advantage) and small $h$ (where accuracy of tau-leaping can be analyzed).

They showed that tau-leaping is then $O(h)$ accurate.
Couple two tau-leaps at different resolutions

Example, generate:
- a Poisson process with intensity 13.1
- a Poisson process with intensity 13

Let $Y_1$ and $Y_2$ be independent, unit-rate Poisson processes, and set $Z_{13.1}(t) = Y_1(13.1t)$ and $Z_{13}(t) = Y_2(13t)$. Then

$$\text{Var}(Z_{13.1}(t) - Z_{13}(t)) = \text{Var}(Y_1(13.1t)) + \text{Var}(Y_2(13t)) = 26.1t$$

Alternatively,

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

Variance of difference is much smaller:
$$\text{Var}(Z_{13.1}(t) - Z_{13}(t)) = \text{Var}(Y_2(0.1t)) = 0.1t$$
More generally

Suppose we want

- non-homogeneous Poisson process with intensity $f(t)$
- 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

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

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

We are using that, for example,

$$Y_1 \left( \int_0^t f(s) \wedge g(s) \, ds \right) + Y_2 \left( \int_0^t f(s) - (f(s) \wedge g(s)) \, ds \right) = Y \left( \int_0^t f(s) \, ds \right)$$
This idea can be used to derive a multilevel tau-leap algorithm.

Straightforward to implement (in a way that guarantees not to be more expensive than Gillespie).

Using Gillespie at the finest level $\rightarrow$ unbiased estimator
### Example of Gillespie/Tau-leaping/multilevel

\[
\begin{align*}
G & \xrightarrow{25} G + M \\
M & \xrightarrow{1000} M + P \\
P + P & \xrightarrow{0.001} D \\
M & \xrightarrow{0.1} \emptyset \\
P & \xrightarrow{1} \emptyset
\end{align*}
\]

Start with 1 gene

Estimate expected number of dimers at \( t = 1 \)

<table>
<thead>
<tr>
<th>Method</th>
<th>Solution</th>
<th>Updates</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gillespie/MC</td>
<td>3714.6 ± 1</td>
<td>8.3 \times 10^{10}</td>
<td>1.5 \times 10^{5} sec</td>
</tr>
<tr>
<td>Tau-leap/MC</td>
<td>3708.4 ± 1</td>
<td>1.7 \times 10^{10}</td>
<td>2.0 \times 10^{4} sec</td>
</tr>
<tr>
<td>Tau/Gill/MLMC</td>
<td>3713.9 ± 1</td>
<td>5.8 \times 10^{8}</td>
<td>1.7 \times 10^{3} sec</td>
</tr>
</tbody>
</table>
Multi-level approach dramatically improves Monte Carlo simulation when samples contain discretization errors

Compute many (cheap) samples at low resolution and few (expensive) samples at high resolution

Now available for Gillespie/tau-leaping

MLMC is currently being pursued in many directions

See home page of Mike Giles (Oxford) for a comprehensive, up-to-date multilevel bibliography