Stochastic Simulation Algorithms for Chemical Reactions

Computer Science in Virginia Tech

Tae-Hyuk Ahn, Yang Cao, Layne T. Watson
Motivation

- Why do we use stochastic simulation methods in biochemical modeling?
  - In chemical systems, the small number of molecules of a few reactant species can result in dynamic behavior that is discrete and stochastic, rather than continuous and deterministic.
  - Stochastic simulation algorithm (SSA) using Monte Carlo methods is a stochastic method to simulate chemical systems, but the SSA is often slow because it simulates every reaction.
  - One remarkable attempt to improve the computational efficiency is the tau-leaping method.
  - QSSA and tQSSA are for stiff systems.
  - This paper compares computational efficiency and exactness between SSA, tau-leaping, implicit tau-leaping, QSSA, and tQSSA based on numerical experiments with simple chemical reactions and stiff systems.
Assumptions

- A well-stirred system at constant volume and temperature.
- $N$ species $\{S_1,\ldots,S_N\}$. System state $X(t) = (X_1(t),\ldots,X_N(t))$,
- $X_i(t)$ = number of $S_i$ molecules at time $t$.
- $M$ reactions $\{R_1,\ldots,R_M\}$. Propensity functions $a_1,\ldots,a_M$,
- $a_j(x)dt$ = probability, given $X(t) = x$, that $R_j$ will fire in next $dt$.
- When $R_j$ fires, the system’s state changes from $x$ to $x + \nu_j$,
- $\nu_j=(\nu_{1j},\ldots,\nu_{Nj})$, $\{\nu_{ij}\}$ is the “stoichiometric matrix”
Idea of Stochastic Simulation Algorithm

When will the next reaction occur?
What kind of reaction will it be?

\[ R_1: A + B \rightarrow C \]

<table>
<thead>
<tr>
<th></th>
<th>( t )</th>
<th>( t + \tau )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

System state X
Stochastic Simulation Algorithm

- \( p(\tau, j \mid x, t)d\tau = \) probability that, given the system state \( x \) at time \( t \), the next reaction will occur in the infinitesimal time interval \( [t + \tau, t + \tau + d\tau) \), and will be an \( R_j \) reaction.

- We call this \( p(\tau, j \mid x, t) \) as “reaction probability density function” because it is joint probability function of the two random variables.
  - \( \tau \) = “time to the next reaction”
  - \( j \) = “index of the next reaction”
Stochastic Simulation Algorithm

- Draw two independent samples \( r_1 \) and \( r_2 \) from \( U(0,1) \)

and take

\[
\tau = \frac{1}{a_0(X)} \ln \left( \frac{1}{r_1} \right)
\]

\( j = \) the smallest integer satisfying

\[
\sum_{j' = 1}^{j} a_{j'}(x) > r_2 a_0(x)
\]

- Update \( X \)

\[
X \leftarrow X + \nu_j
\]
Tau-leaping Method

- Idea – Many reactions can be simulated at each step with a preselected time $\tau$
- $\tau$ must be small enough to satisfy the “leap condition”: The expected state change induced by the leap must be sufficiently small that propensity functions remain nearly constant during the time step $\tau$
Tau-leaping Method: Some Details

- Agrees with SSA in the small step size limit

- Equivalent to forward Euler in the SDE and ODE regimes

\[ \Delta X = \nu P(a(x), \tau) \]

where

- \( \Delta X \) Change of state
- \( x \) Current state
- \( \tau \) Time step
- \( P(a, \tau) \) Poisson variable with parameters \( a \) and \( \tau \)
- \( a_j(x) \) Propensity functions
- \( \nu_{ij} \) Change in species \( i \) due to reaction \( j \)
Stiff Systems

- Exhibit slow and fast time scales. The fast scales are stable.

- Fast reactions almost cancel each other while slow reactions determine the trend.

- Explicit methods require unreasonably small time steps in order to maintain stability.

- Implicit methods in general do not have step size limitations due to stability. Accuracy concerns alone determine the step size.
Implicit Tau-leaping

\[ \Delta X = \nu \tau a(x + \Delta X) + \nu P(a(x), \tau) - \nu \tau a(x) \]

- Based on the (explicit) tau method
- Only the \textbf{mean} part is implicit
- Tends to the backward Euler scheme as populations get larger
- Better suited for \textbf{stiff} problems
Quasi-Steady State Assumption

- System in stiff.
- In deterministic kinetics, the net rate of formation is zero when the fast reacting species are in a quasi-steady state.

Example for Michaelis-Menten kinetics

\[
E + S \iff E : S \rightarrow P + E
\]

\[
\frac{d[E : S]}{dt} = 0
\]

\[
S \rightarrow P,
\]

\[
a(s) = \frac{k_2 E_T[S]}{K_M + [S]}
\]
Total Quasi-Steady Sate Assumption

- QSSA eliminates fastest reacting variable under some assumptions.

- In Michalis-Menten kinetics, the necessary condition for the QSSA is $S_0 \gg E_T$.

- In a protein interaction network, however, the enzymes and substrates often swap their roles. Therefore QSSA assumption is not correct some models.

- The proper slow time scale is $\hat{[S]} = [S] + [E : S]$. 
SSA vs. Tau-leaping

- Irreversible Isomerization $S_1 \xrightarrow{c_1} 0$

Table 1. The number of runs and elapsed CPU time (sec) with the SSA and explicit tau-leaping method, where $t_f = 5$, $c_1 = 1$, $X_1 = 10^4$, and $\epsilon = 0.03$.

<table>
<thead>
<tr>
<th>Number of runs</th>
<th>1000</th>
<th>5000</th>
<th>10000</th>
<th>50000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSA</td>
<td>48.74</td>
<td>240.40</td>
<td>487.76</td>
<td>2433.28</td>
</tr>
<tr>
<td>Explicit Tau-leaping</td>
<td>2.73</td>
<td>14.21</td>
<td>28.37</td>
<td>143.29</td>
</tr>
</tbody>
</table>

Fig. 1. The SSA simulation (solid lines) and the explicit tau-leaping simulation (dotted lines) for the irreversible isomerization. The error control parameter $\epsilon$ is 0.03 (left) and 0.15 (right).
Goldbeter-Koshland switch

\[ D + S_p \xrightarrow{k_{1d}} D : S_p \xrightarrow{k_{2d}} S + D, \]
\[ E + S \xrightarrow{k_{1e}} E : S \xrightarrow{k_{2e}} S_p + E. \]

Table 2. The number of runs and elapsed CPU time (sec) for the SSA, explicit tau-leaping, implicit tau-leaping, QSSA, and tQSSA algorithms.

<table>
<thead>
<tr>
<th></th>
<th>1000</th>
<th>5000</th>
<th>10000</th>
<th>50000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSA</td>
<td>44.57</td>
<td>216.77</td>
<td>434.10</td>
<td>2175.45</td>
</tr>
<tr>
<td>Explicit Tau-leaping</td>
<td>4.75</td>
<td>18.23</td>
<td>46.57</td>
<td>181.31</td>
</tr>
<tr>
<td>Implicit Tau-leaping</td>
<td>18.43</td>
<td>89.13</td>
<td>181.29</td>
<td>887.44</td>
</tr>
<tr>
<td>QSSA</td>
<td>2.63</td>
<td>13.23</td>
<td>26.33</td>
<td>132.47</td>
</tr>
<tr>
<td>tQSSA</td>
<td>2.67</td>
<td>13.33</td>
<td>26.79</td>
<td>133.68</td>
</tr>
</tbody>
</table>
Results So Far

- SSA is slow, but exact.
- The explicit tau-leaping method improves computational efficiency for nonstiff systems, but can be unstable on stiff systems.
- The implicit tau-leaping is stable, but much slower than the explicit tau-leaping with appropriate correctness.
- In terms of CPU time, QSSA and tQSSA algorithms are the fastest approximate algorithms. But QSSA has conditional assumption. Therefore, tQSSA is better.
Under Development

- Comparing stochastic and deterministic results with budding yeast model in JigCell project.
- Hybrid method with automatic partition (slow and fast reactions) estimator.

Question?