EXPLICIT POISSON TAU-LEAPING

Yang Cao (VT), Dan Gillespie, Linda Petzold (UCSB)

***

DG Current Support: **U.C. Santa Barbara**
Caltech (NIGMS , NIH)

DG Past Support: **Caltech** (DARPA/AFOSR, Beckman)
Molecular Sciences Institute (Sandia / DOE)
ONR

ASSUMPTIONS & DEFINITIONS

- A well-stirred chemical system at constant volume and temperature.
- N species $\{S_1,\ldots,S_N\}$. System state is $\mathbf{X}(t)=\left(X_1(t),\ldots,X_N(t)\right)$,

  $X_i(t)\equiv$ number of S_i molecules at time t.
- M reactions $\{R_1,\ldots,R_M\}$. Each R_j is described by two quantities:
  - State change vector: $\boldsymbol{\nu}_j \triangleq \left(\nu_{1j},\ldots,\nu_{Nj}\right)$, where

    $\nu_{ij}\equiv$ change induced in X_i by one R_j event.

    So R_j induces the transition $\mathbf{x}\to\mathbf{x}+\boldsymbol{\nu}_j$.
  - Propensity function: a_j, where

    $a_j(\mathbf{x})dt \triangleq$ probability, given $\mathbf{X}(t)=\mathbf{x}$, that R_j will fire in $[t,t+dt)$.

TWO EXACT CONSEQUENCES

- The function $P(\mathbf{x},t\,|\,\mathbf{x}_0,t_0) \triangleq \text{Prob}\{\mathbf{X}(t)=\mathbf{x}, \text{ given } \mathbf{X}(t_0)=\mathbf{x}_0\}$ satisfies the **chemical master equation** (CME):

$$\frac{\partial P(\mathbf{x},t\,|\,\mathbf{x}_0,t_0)}{\partial t} = \sum_{j=1}^{M}\left[a_j(\mathbf{x}-\boldsymbol{\nu}_j)P(\mathbf{x}-\boldsymbol{\nu}_j,t\,|\,\mathbf{x}_0,t_0) - a_j(\mathbf{x})P(\mathbf{x},t\,|\,\mathbf{x}_0,t_0)\right]$$

  - ➢ **But it's nearly always impossible to solve.**

- The **stochastic simulation algorithm** (SSA): If the system is in state $\mathbf{x}$ at time t, then with $a_0(\mathbf{x}) \triangleq \sum_{j'=1}^{M} a_{j'}(\mathbf{x})$,
  - the time τ to the next reaction is an exponential random variable with mean $1/a_0(\mathbf{x})$;
  - the index j of the next reaction is an integer random variable with probability $a_j(\mathbf{x})/a_0(\mathbf{x})$.

  $\therefore$ By generating such samples for τ and j, we can advance the system to the next reaction by replacing $t \leftarrow t+\tau$ and $\mathbf{x} \leftarrow \mathbf{x}+\boldsymbol{\nu}_j$.

  - ➢ **But simulating *every* reaction usually takes too much time.**

An <u>Approximate</u> Acceleration Strategy: *Explicit Tau-leaping*

- **MathFact**: If some "event" occurs in each next dt with probability $a\,dt$ (a can be any positive *constant*), then the number of times the event will occur in any specified time $\tau > 0$ is $\mathcal{P}(a\tau)$, the Poisson random variable with mean $a\tau$:

$$\text{Prob}\{\mathcal{P}(a\tau)=n\} = \frac{e^{-a\tau}(a\tau)^n}{n!} \quad (n=0,1,\ldots)$$

$$\langle \mathcal{P}(a\tau)\rangle = a\tau, \quad \text{var}\{\mathcal{P}(a\tau)\} = a\tau\,.$$

- In state $\mathbf{x}$ at time t, suppose we can find a $\tau > 0$ that satisfies the **…**

  Leap Condition: During $[t,t+\tau)$ every a_j stays $\approx$ constant.

  Then in $[t,t+\tau)$, R_j will occur $\approx \mathcal{P}(a_j(\mathbf{x})\tau)$ times, so

$$\mathbf{X}(t+\tau) \approx \mathbf{x} + \sum_{j=1}^{M} \mathcal{P}_j\left(a_j(\mathbf{x})\tau\right)\boldsymbol{\nu}_j\,.$$

The Explicit Tau-leaping Formula

$$\mathbf{X}(t+\tau) \approx \mathbf{x} + \sum_{j=1}^{M} \mathcal{P}_j\left(a_j(\mathbf{x})\tau\right)\boldsymbol{\nu}_j$$

- In principle feasible to implement because codes exist for generating random samples of $\mathcal{P}(\alpha)$ for any given $\alpha \geq 0$.
- Must take τ *small enough* that the Leap Condition is satisfied.
- But if τ is also *large enough* that at least some $a_j(\mathbf{x})\tau \gg 1$, then many firings of those R_j will occur in the leap, and the result may be faster than the single-reaction stepping procedure of the SSA.
- **Practical considerations for a viable simulation algorithm:**
  - How can we find the largest τ that satisfies the Leap Condition?
  - How can we avoid generating negative reactant populations?
  - How can we connect smoothly to the exact SSA for small τ ?

- Finding the largest τ that satisfies the Leap Condition -

With $\mathbf{X}(t) = \mathbf{x}$, let $\Delta_\tau a_j(\mathbf{x}) \equiv a_j\left(\mathbf{X}(t+\tau)\right) - a_j(\mathbf{x})$, a *random variable.*

A little history …

- **Version 1** of the Leap Condition required $\left|\Delta_\tau a_j(\mathbf{x})\right| \leq \varepsilon a_0(\mathbf{x}), \forall j$.
  - We shall take "$|Y| \leq B$" to mean: $\left|\langle Y \rangle\right| \leq B$ *and* $\mathrm{sdev}\{Y\} \leq B$.
  - Method needed $O(M^2)$ computations to find $\tau = \tau(\varepsilon, \mathbf{x})$.
- **Version 2**: Required $\left|\Delta_\tau a_j(\mathbf{x})\right| \leq \max\left(\varepsilon a_j(\mathbf{x}), c_j\right), \forall j$.
  - Fulfills the Leap Condition better ⇒ a more accurate simulation.
  - But still needed $O(M^2)$ computations to find $\tau = \tau(\varepsilon, \mathbf{x})$.
- ❖ **Version 3**: Requires $\left|\Delta_\tau x_i\right| \leq \max\left(\varepsilon_i x_i, 1\right), i \in I_{\mathrm{rs}}$, with $\varepsilon_i \equiv \varepsilon_i(\varepsilon, x_i)$ chosen so that *Version 2* of the Leap Condition is satisfied.
  - Gives the same (improved) simulation accuracy as Version 2,
  - But needs only $O(M)$ computations to find $\tau = \tau(\varepsilon, \mathbf{x})$.

Defining the functions $\varepsilon_i(\varepsilon, x_i)$ so that $\left|\Delta_\tau a_j(\mathbf{x})/a_j(\mathbf{x})\right| \le \varepsilon, \forall j$.

- Done *before* the simulation begins: For each *reactant* species S_i, determine $HOR(i) \equiv$ highest order of reaction in which S_i is a reactant.
  - If $HOR(i)=1$, take $\varepsilon_i = \varepsilon$.
  - If $HOR(i)=2$, take $\varepsilon_i = \varepsilon/2$, *except* if any reaction requires *two* S_i molecules take $\varepsilon_i = \varepsilon\Big/\left(2+(x_i-1)^{-1}\right)$.

Why? We want $\left|\Delta_\tau x_i/x_i\right| \le \varepsilon_i, \forall i$ to $\Rightarrow$ $\left|\Delta_\tau a_j/a_j\right| \le \varepsilon, \forall_j$. So …

- If $a_j = c_j x_i$, then $\Delta_\tau a_j = c_j \Delta_\tau x_i$; so $\Delta_\tau a_j/a_j = \Delta_\tau x_i/x_i$.
- If $a_j = c_j x_1 x_2$, then $\Delta_\tau a_j \doteq c_j x_2 \Delta_\tau x_1 + c_j x_1 \Delta_\tau x_2$; so $\Delta_\tau a_j/a_j \doteq \Delta_\tau x_1/x_1 + \Delta_\tau x_2/x_2$.
- If $a_j = c_j \frac{1}{2} x_i(x_i-1)$, then $\Delta_\tau a_j \doteq \frac{1}{2} c_j (x_i-1)\Delta_\tau x_i + \frac{1}{2} c_j x_i \Delta_\tau x_i$; so $\Delta_\tau a_j/a_j \doteq \left(\Delta_\tau x_i/x_i\right)\left(2+(x_i-1)^{-1}\right)$.

Ensuring that $\left|\Delta_\tau x_i\right| \le \max\{\varepsilon_i x_i, 1\}, \forall i \in I_{\text{rs}}$:

The basic tau-leaping formula $\Rightarrow$ $\Delta_\tau x_i = \sum_j \nu_{ij} \mathcal{P}_j\left(a_j\tau\right)$.

Since $\mathcal{P}_j$'s are statistically independent with means and variances $a_j\tau$,

$$\left\langle \Delta_\tau x_i \right\rangle = \sum_j \nu_{ij}(a_j\tau), \quad \operatorname{var}\{\Delta_\tau x_i\} = \sum_j \nu_{ij}^2 (a_j\tau).$$

The condition "$\left|\Delta_\tau x_i\right| \le \max\{\varepsilon_i x_i, 1\}$" will be considered satisfied iff

$$\left|\sum_j \nu_{ij} a_j \tau\right| \le \max\{\varepsilon_i x_i, 1\} \quad \& \quad \sqrt{\sum_j \nu_{ij}^2 a_j \tau} \le \max\{\varepsilon_i x_i, 1\}.$$

Solving these two equations for τ, we get the **tau-selection formula**:

$$\tau = \min_{i \in I_{\text{rs}}} \left\{ \frac{\max\{\varepsilon_i x_i, 1\}}{\left|\sum_j \nu_{ij} a_j(\mathbf{x})\right|}, \frac{\max\{\varepsilon_i x_i, 1\}^2}{\sum_j \nu_{ij}^2 a_j(\mathbf{x})} \right\}.$$

- Avoiding negative populations & Segueing to the SSA -

If the population of a *consumed* reactant species is small, it might get "overdrawn" during a tau-leap by too many reaction firings.

- Originally this was thought to be caused by $\mathcal{P}(m)$ being unbounded. But the more common cause was bounding $\left|\Delta_\tau a_j(\mathbf{x})\right|$ by "$\varepsilon a_0(\mathbf{x})$".
- With the bound "$\max\left(\varepsilon\, a_j(\mathbf{x}), c_j\right)$", negatives are rare. When they do occur, it's usually because two or more reaction channels (which in tau-leaping fire *independently*) deplete a *common reactant*.

The New Tau-Leaping Strategy …

- Uses a *second control parameter*, n_c; typically $5 \le n_\mathrm{c} \le 30$.
- Classifies any R_j with $a_j(\mathbf{x}) > 0$ that is within n_c firings of using up any reactant as ***critical***. All other reactions are called ***non-critical***.
- Is designed so that there will be **no more than one firing of a *critical* reaction in any leap**. This means that **no *critical* reaction will ever cause any population to go negative**.
- ***Reduces to the exact SSA*** **when *all* R_j are critical**.

THE EXPLICIT TAU-LEAPING PROCEDURE

0. Choose values for ε and n_c. For each reactant species S_i, set the appropriate function $\varepsilon_i(\varepsilon, x_i)$. Initialize $t \leftarrow 0$ and $\mathbf{x} \leftarrow \mathbf{x}_0$.

1. In state $\mathbf{x}$ at time t, evaluate all the $a_j(\mathbf{x})$. Then determine which reactions are *critical* and *non-critical* (per n_c).

2. If there are *no non-critical* reactions take $\tau' = \infty$; otherwise, compute the putative leap time τ' for the *non-critical* reactions as

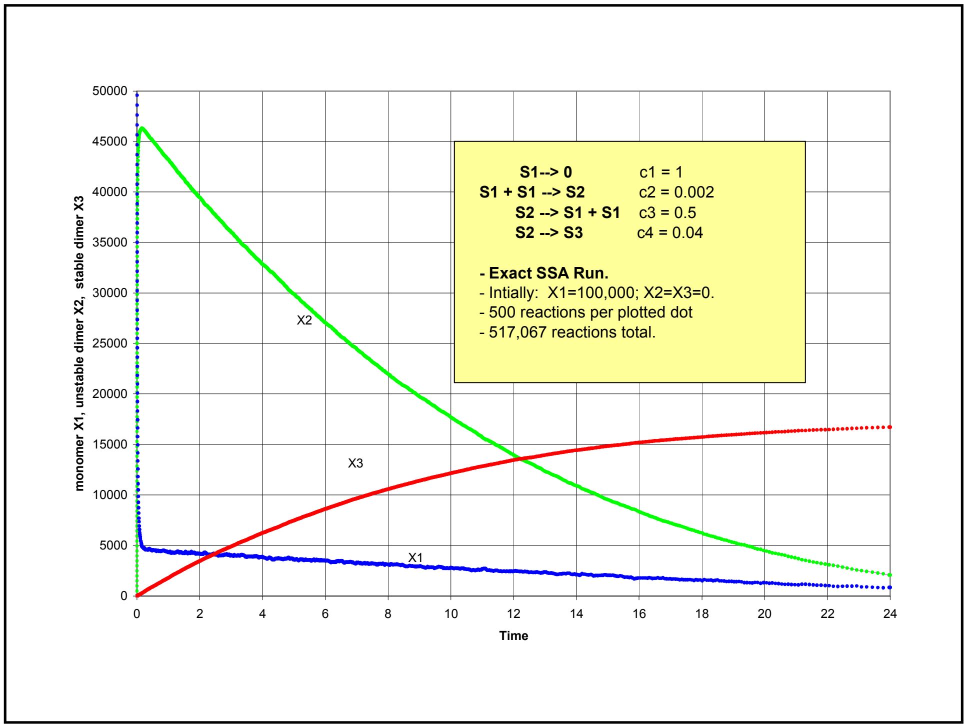
$$\tau' = \min_{i \in I_\mathrm{rs}} \left\{ \frac{\max\{\varepsilon_i x_i, 1\}}{\left|\sum_{j \in J_\mathrm{ncr}} \nu_{ij} a_j(\mathbf{x})\right|}, \frac{\max\{\varepsilon_i x_i, 1\}^2}{\sum_{j \in J_\mathrm{ncr}} \nu_{ij}^2 a_j(\mathbf{x})} \right\}.$$

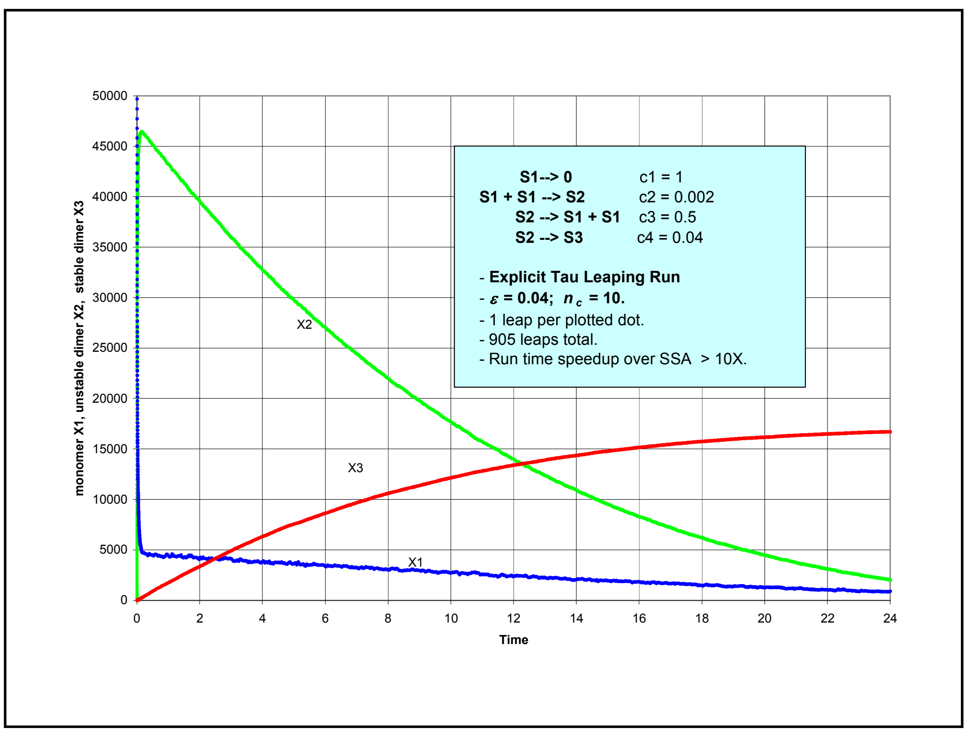
3. If there are *no critical* reactions take $\tau'' = \infty$; otherwise use the SSA to compute the time τ'' to, and the index j_c of, the *next critical* reaction.

4. Take $\tau = \min(\tau', \tau'')$. Then set $\mathbf{X}(t+\tau) \doteq \mathbf{x} + \sum_{j \in J_\mathrm{ncr}} \mathcal{P}_j\left(a_j(\mathbf{x})\tau\right) \boldsymbol{\nu}_j$.

5. If $\tau'' \le \tau'$, replace $\mathbf{X}(t+\tau) \leftarrow \mathbf{X}(t+\tau) + \boldsymbol{\nu}_{j_\mathrm{c}}$.

6. Update $\mathbf{x} \leftarrow \mathbf{X}(t+\tau)$ and $t \leftarrow t+\tau$. Go to 1, or else stop.


S1--> 0 c1 = 1
S1 + S1 --> S2 c2 = 0.002
S2 --> S1 + S1 c3 = 0.5
S2 --> S3 c4 = 0.04
- Exact SSA Run.
- Intially: X1=100,000; X2=X3=0.
- 500 reactions per plotted dot
- 517,067 reactions total.
X2
X3
X1
monomer X1, unstable dimer X2, stable dimer X3
Time
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
0 2 4 6 8 10 12 14 16 18 20 22 24


S1--> 0 c1 = 1
S1 + S1 --> S2 c2 = 0.002
S2 --> S1 + S1 c3 = 0.5
S2 --> S3 c4 = 0.04
- Explicit Tau Leaping Run
- ε = 0.04; n c = 10.
- 1 leap per plotted dot.
- 905 leaps total.
- Run time speedup over SSA > 10X.
X2
X3
X1
monomer X1, unstable dimer X2, stable dimer X3
Time
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
0 2 4 6 8 10 12 14 16 18 20 22 24

Going from tau-leaping to *Langevin-leaping* …

- In $\mathbf{X}(t+\tau) \doteq \mathbf{x} + \sum_j \mathcal{P}_j\left(a_j(\mathbf{x})\tau\right)\boldsymbol{\nu}_j$: If for some j, $a_j(\mathbf{x})\tau \gg 1$, then

$$\mathcal{P}_j\left(a_j(\mathbf{x})\tau\right) \approx \mathcal{N}_j\left(a_j(\mathbf{x})\tau, a_j(\mathbf{x})\tau\right) = a_j(\mathbf{x})\tau + \sqrt{a_j(\mathbf{x})}\mathcal{N}_j(0,1)\sqrt{\tau}\ .$$

- ***A trick:*** Write the Poisson random number generator so that it returns $\mathcal{N}(\alpha,\alpha)$ for $\mathcal{P}(\alpha)$ when $\alpha \gg 1$. This is usually faster.
- Then if it happens that $a_j(\mathbf{x})\tau \gg 1$ for **all** j, tau-leaping will automatically become **Langevin leaping**:

$$\mathbf{X}(t+\tau) \doteq \mathbf{x} + \sum_j \boldsymbol{\nu}_j a_j\left(\mathbf{x}\right)\tau + \sum_j \boldsymbol{\nu}_j \sqrt{a_j\left(\mathbf{x}\right)}\,\mathcal{N}_j(0,1)\sqrt{\tau}$$

… and then going on to the *Reaction Rate Equation*

- Finally, if $a_j(\mathbf{x})\tau$ for all j is so much $\gg 1$ that $\sqrt{a_j(\mathbf{x})\tau} \ll a_j(\mathbf{x})\tau$, then the noise terms in the Langevin leaping formula can be dropped, and we get the Euler formula for the **reaction rate equation**:

$$\mathbf{X}(t+\tau) \doteq \mathbf{x} + \sum_j \boldsymbol{\nu}_j a_j\left(\mathbf{x}\right)\tau\ .$$

LIMITATIONS

- In explicit tau-leaping, the Leap Condition will always restrict τ to the time-scale of the *fastest* reactions.
- So for a system with a *large range of time scales* (e.g., ***stiff*** systems), explicit tau-leaping will seem slow.
- **Alternatives**
  - **Implicit Poisson tau-leaping**: A stochastic adaptation of the implicit Euler method for ODEs.
  - **The slow-scale SSA**: Skips over the fast reactions and simulates only the slow ones, but using specially modified propensity functions. An adaptation of the partial/rapid equilibrium method and the quasi steady-state method for ODEs.