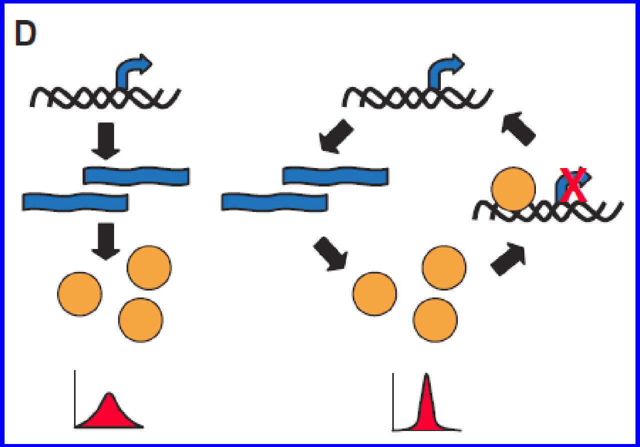
Review

- Origin, consequences, and control.

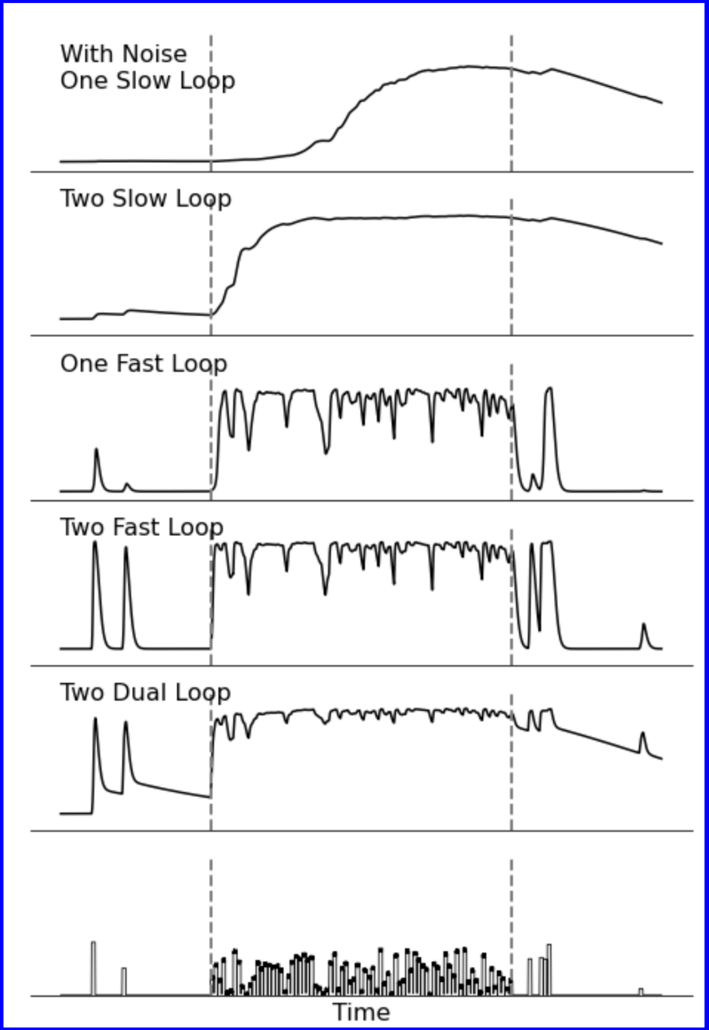
Intrinsic noise， **extrinsic noise**

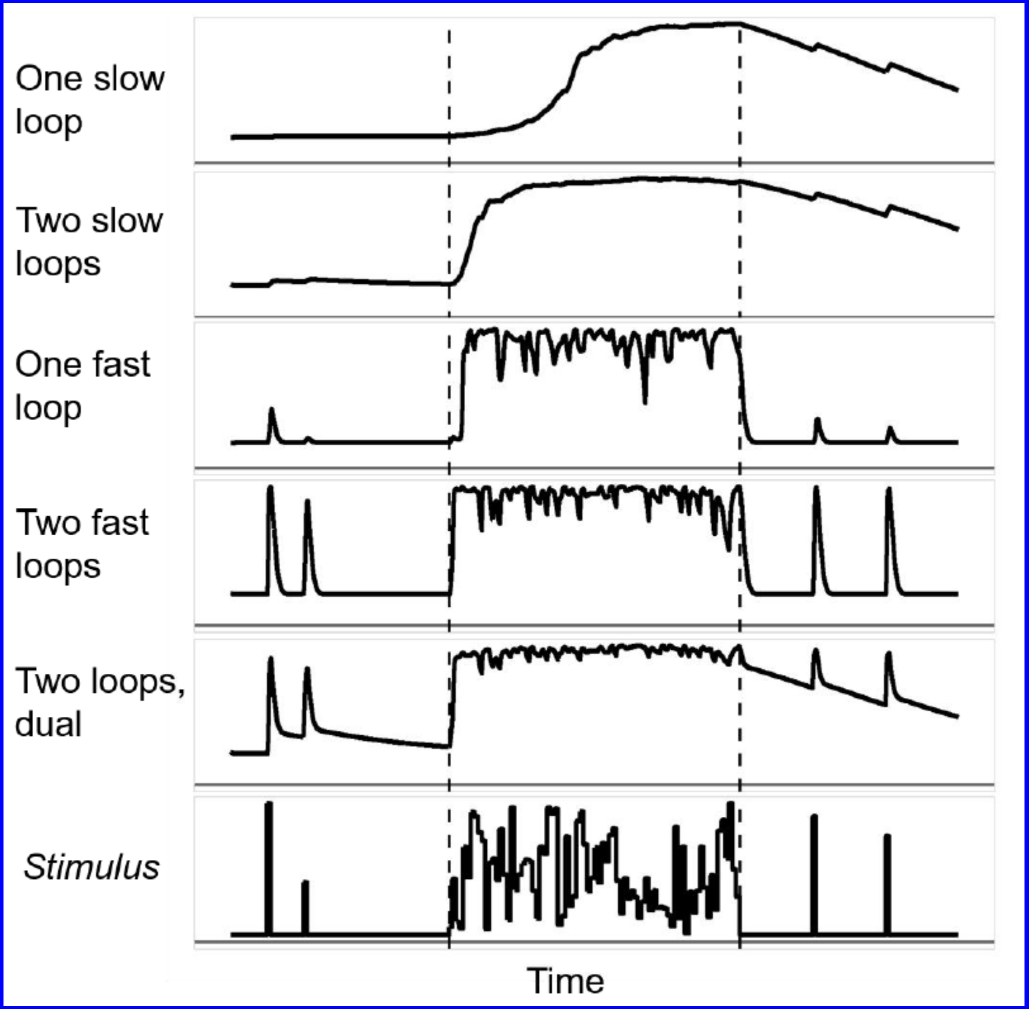
Extrinsic noise may affect cellular phenotypes more strongly than intrinsic noise.

Feedback loops, gene copy number can be exploited to control noise.



Comments on assignment 8





Chap4-2 Simulations methods of stochastic systems

1. Additive noise and Langevin equation

$$m\ddot{x}(t) = -m\gamma\dot{x}(t) + f(x(t)) + \xi(t)$$

$$\langle \xi(t)\xi(s) \rangle = 2D\delta(t-s) \qquad \langle \xi(t) \rangle = 0$$

$$\dot{x} = v$$

$$\dot{v} = -\gamma v + f(x(t)) + \xi(t)$$

Overdamped limit

$$\boxed{\dot{x} = f(x(t)) + \xi(t)}$$

$$x(t+\Delta t) - \mathrm{x(t)} = f(x(t))\Delta t + \int_t^{t+\Delta t} \xi(s)ds$$

$\Gamma(t) = \int_t^{t+\Delta t} \xi(s)ds = \sqrt{2D\Delta t}\,\omega$, ω is a Gaussian random number.

Gaussian distribution $\quad f（x）= \dfrac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad N(\mu, \sigma^2)$

2. Multiplicative noise

$$dX_i = f_i(\underline{X}(t))dt + \sum_{j=1}^{M} g_{i,j}(\underline{X}(t))dW_j(t) \quad i = 1, \ldots ,N,$$

$$I(j) = \int_t^{t+\Delta t} dW_j(t) \cong N_j(0,\Delta t)$$

$$X_i^{k+1} = X_i^k + f_i(\underline{X}^k(t))\Delta t + \sum_{j=1}^{M} g_{i,j}(\underline{X}^k(t))I(j)$$
$$+ \underbrace{\sum_{j_1=1}^{M}\sum_{j_2=1}^{M} L_{j_1} g_{i,j_2}(\underline{X}^k(t))I(j_1,j_2)}_{\text{extra term}} , \quad i = 1, \ldots ,N, \qquad (7)$$

$$I(j_1,j_2) = \int_t^{t+\Delta t} dW(t)_{j1} dW(t)_{j2},$$

$$L_{j_1} = \sum_{d=1}^{M} g_{d,j_1} \frac{\partial}{\partial X_d}.$$

Sotiropoulos V, Kaznessis YN (2008) The Journal of Chemical Physics 128: 014103.

Chemical Langevin equation

$$\frac{d\mathbf{X}(t)}{dt} \doteq \sum_{j=1}^{M} \nu_j a_j(\mathbf{X}(t)) + \sum_{j=1}^{M} \nu_j \sqrt{a_j(\mathbf{X}(t))}\,\Gamma_j(t) .$$

- Milstein method

$$X_i^{k+1} = X_i^k + \sum_{j=1}^{M} \upsilon_{ji}\alpha_j(\underline{X}^k)\Delta t + \sum_{j=1}^{M} \upsilon_{ji}\sqrt{\alpha_j(\underline{X}^k)}I(j)$$

$$+\frac{1}{2}\sum_{j_1=1}^{M}\sum_{j_2=1}^{M}\sum_{n=1}^{N} \upsilon_{j_1 n}\upsilon_{j_2 i}\sqrt{\frac{\alpha_{j_1}(\underline{X}^k)}{\alpha_{j_2}(\underline{X}^k)}}\frac{\partial\alpha_{j_1}}{\partial X_n}I(j_1,j_2)$$

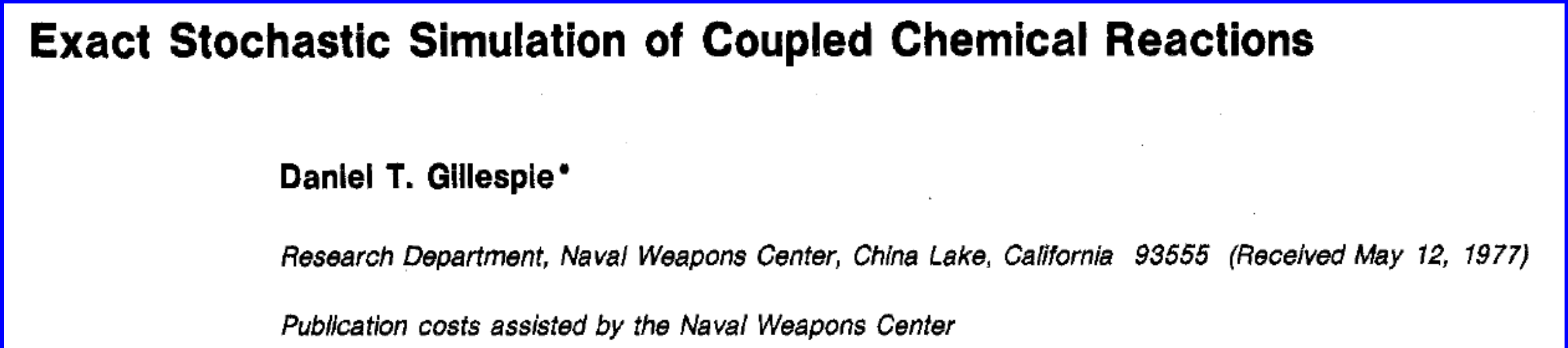
$$I(j) = \int_t^{t+\Delta t} dW_j(t) \cong N_j(0,\Delta t)$$

$$I(j_1,j_2) = \int_t^{t+\Delta t} dW(t)_{j1}dW(t)_{j2}$$

Sotiropoulos V, Kaznessis YN (2008) The Journal of Chemical Physics 128: 014103.

3.Methods for Simulating Discrete Stochastic Chemical Reactions

- Exact stochastic algorithm



Exact Stochastic Simulation of Coupled Chemical Reactions

Daniel T. Gillespie*

Research Department, Naval Weapons Center, China Lake, California 93555 (Received May 12, 1977)

Publication costs assisted by the Naval Weapons Center

- Tau-leap algorithm
- Chemical Langevin equation

Gillespie, D. T. (1977). "Exact stochastic simulation of coupled chemical reactions." The Journal of Physical Chemistry **81**(25): 2340-2361.

Discrete Stochastic Simulations (Chemical Reactions)

- We consider here the situation where different chemical can react with each other, governed by different reactions. In fact, the “chemicals” can be molecules, cells, or even organisms, and the “reactions” can be any interaction rules.
- We consider the situation where there are such a small number of chemicals that the stochastic element becomes significant.
- However, these methods are restricted to systems of uniformly mixed chemicals.

Chemical Reactions as a Poisson Process

- Under these assumptions, chemical reactions are a Poisson process, meaning each reaction occurs stochastically at a rate dependent only on the current state of the system, not dependent on previous states or events.
- This is also called a memoryless process
- We will look at several methods to simulate this.
- In what conditions does each method provide accurate results?
- How efficient is each method?

Four Views of a Poisson Process:

- If the rate of an event is r, then, $t = 1/r$ is mean time to next event.
- We consider four ways to step through time for a Poisson process
  - What is probability that an event will occur in time $T << t$? This is inefficient and we don’t use it.
  - At what time does the next event occur?
  - How many events occur in time T ~> t?
  - How many events occur in time $T >> t$?

Three Ways to Simulate a Poisson Process:

- Match time step to next event (1 event)
  - **Gillespie's "exact" algorithm**
  - Always accurate, but moderately expensive
- Take medium time steps (0 to ~20 events)
  - **"Tau-leap" algorithm**
  - less expensive and reasonably accurate if system doesn't change significantly in a time step (so if time step is well chosen);
- Take large time steps (> 10 events)
  - **Gillespie's "Chemical Langevin Equation" algorithm**
  - Accurate if system doesn't change significantly in a time step, but > 10 events occur. Least expensive

Reaction-Based Solving Methods:

- We are used to writing differential equations from chemical reactions.
- For example:

  $X+Y \rightarrow Z$ (rate a)
  $Z \rightarrow Y$ (rate b)

  Is converted to
  $dX/dt = -aXY;$
  $dY/dt = -aXY + bZ;$
  $dZ/dt = aXY - bZ;$

- But in stochastic systems the actual “events” or “reactions” is stochastic.
- And, when a reaction occurs, it affects many “chemicals” at once.
- So uncertainty in these chemicals is coupled.
- Thus, need to organize algorithms on the number and type of reactions that occur, not on the change in each chemical, as we are used to.

Define System:

- Define R reactions according to stoichiometry using two matrices and a column vector:
  - Matrix S is the substrates.
  - Matrix P is the products.
  - Vector K is the reaction rates.
- Example:

  $S_1 + S_2 \rightarrow S_3$ with rate constant c_1

  $2S_1 \rightarrow S_2$ with rate constant c_2,

  S = [1, 1, 0;   P = [0, 0, 1;   K = [c1
  2, 0, 0];   0, 1, 0];   c2];

- Initial conditions: X(0) for C chemicals: $[X_1, X_2, \ldots X_C]$

Gillespie Algorithm
Stochastic Simulation Algorithm

- Exact Algorithm to solve chemical reactions
- Essentially mimics real situation:
  - Given rates and molecule numbers,
  - How long until next change occurs?
  - Which is next event that occurs?
- Elegant and simple. Proposed in 1977
- Still used today.

Gillespie, D. T. (1977). "Exact stochastic simulation of coupled chemical reactions." The Journal of Physical Chemistry **81**(25): 2340-2361.

Step 1: Given the system state, determine the rate of each reaction, a_r.

- Reaction 1: $S_1 + S_2 \rightarrow S_3$, with rate constant c_1

  X_1, X_2 are the numbers of the reactant molecules

  Define the stoichiometry: $h_1 = X_1X_2$; this will give dependence on amounts of molecules.

  Then $a_1 = h_1c_1 = c_1 X_1X_2$ = rate for this reaction.

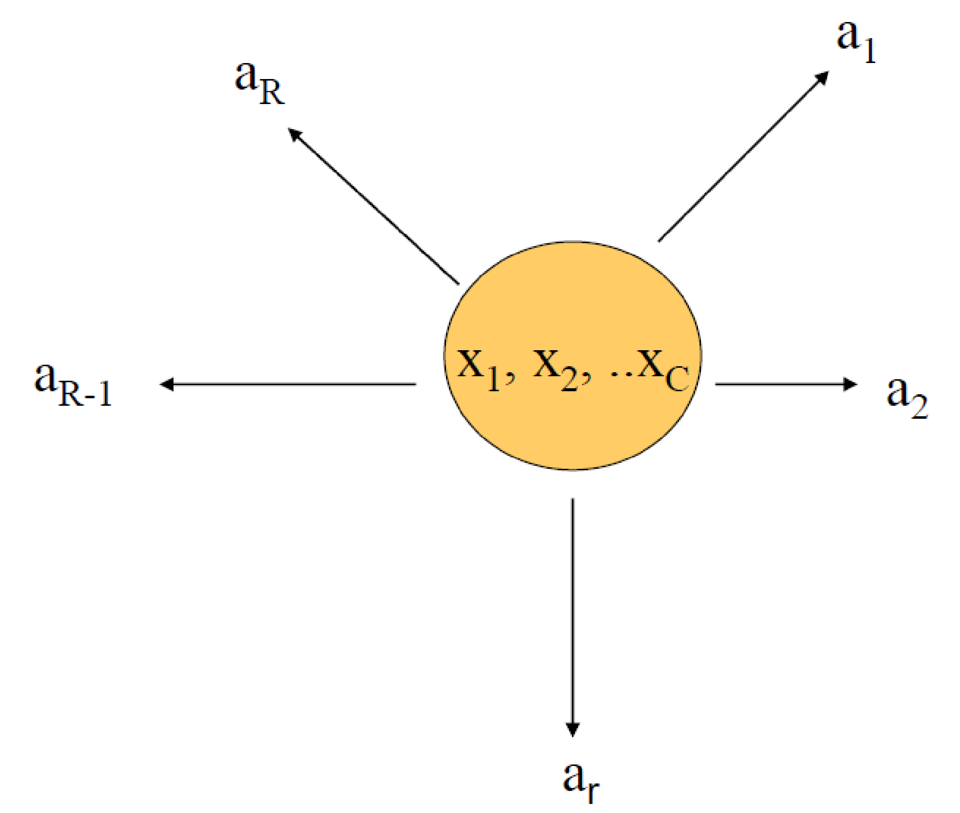
- Reaction 2: $2S_1 \rightarrow S_2$,

  $h_2 = X_1(X_1-1)/2$, $a_2 = h_2c_2 = c_2 X_1(X_1-1)/2$

- Finally, define: $a_0 = \Sigma a_r$ ($r = 1$ to R)

  This is the combined rate of all possible reactions

Gillespie Rates:



What is the residence time of this state?

How fast do we leave this state?

$$a_0 = \Sigma a_r$$

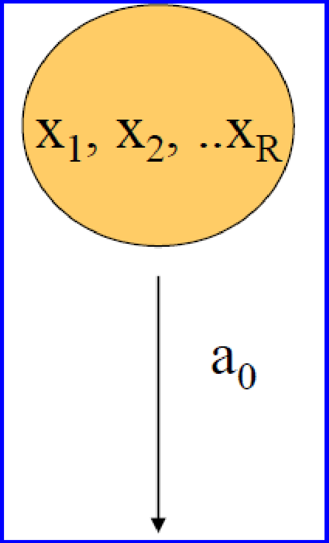
$(r = 1 \text{ to } R)$

is the total rate at which some reaction occurs.

What is the residence time of this state?

Current state

p is the probability that state still survives at time τ:
Simple Poisson Process



All other states

$$p(\tau) = e^{-a_0\tau}$$

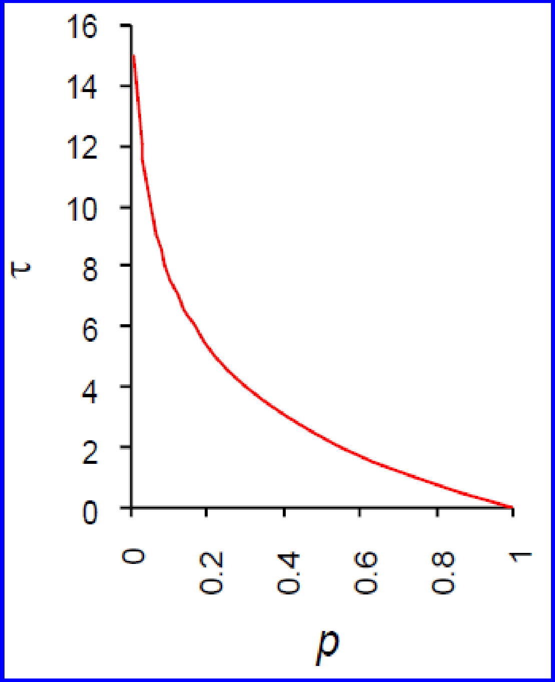
Thus:

$$\ln(p) = -a_0\tau$$

$$\ln\left(\frac{1}{p}\right) = a_0\tau$$

$$\frac{1}{a_0}\ln\left(\frac{1}{p}\right) = \tau$$

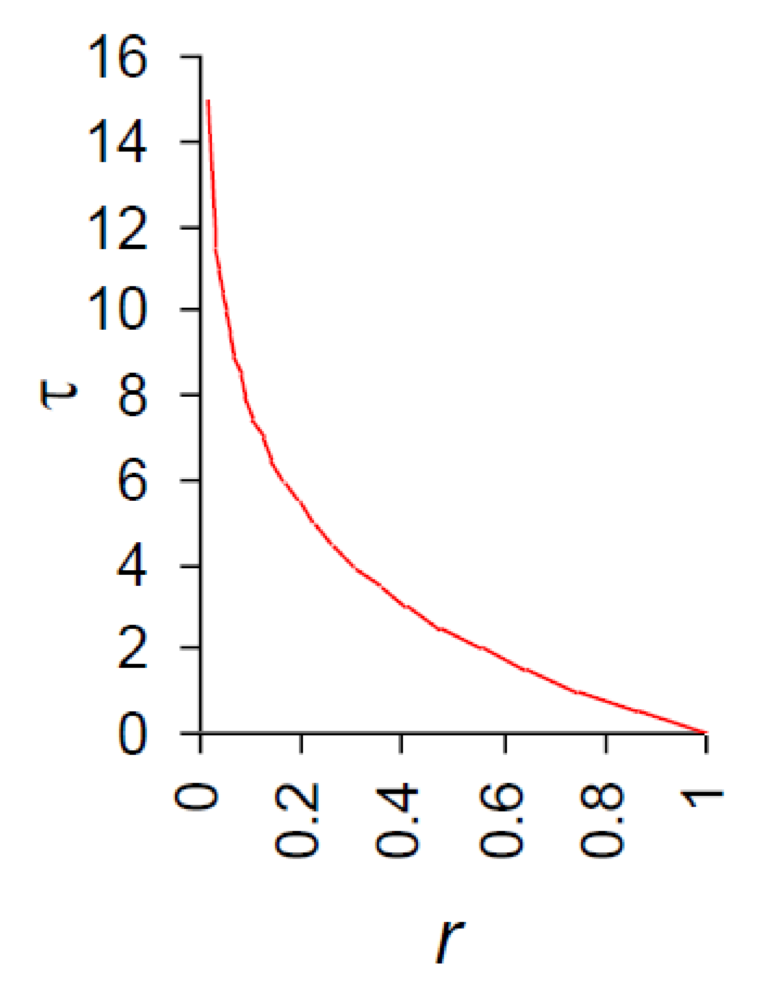
Exponential decay



As *p* decays from 1 to 0, τ increases from 0 to infinity

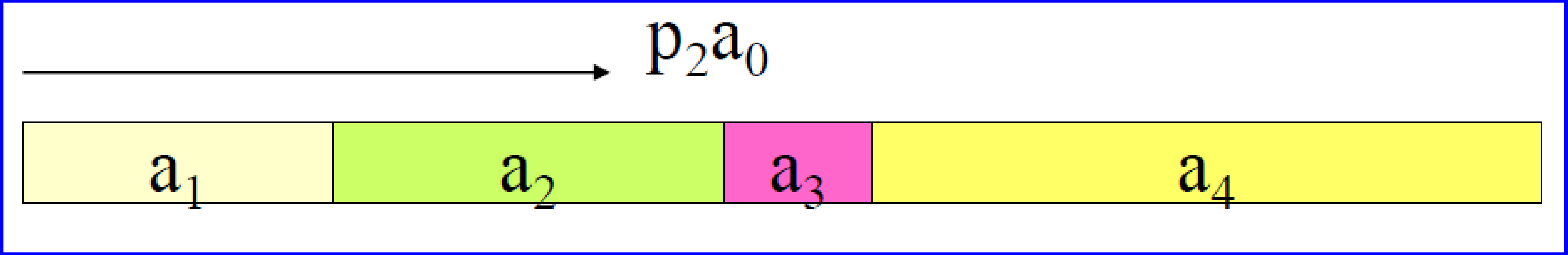
Step 2 When does the next reaction occur …

- Pick p_1, a uniform random number from 0 to 1
- Let $\tau = \frac{1}{a_0}\ln\left(\frac{1}{p_1}\right)$
- This is time of the next event.
- (Note that the time step doesn't have to be predetermined, and is exact.)



Step 2 …and which reaction is it?

- Determine which reaction occurs at time t:
- Pick p_2, another uniform random number from 0 to 1
- Find r, such that: $$\sum_{i=1}^{r-1} a_i < p_2 a_0 < \sum_{i=1}^{r} a_i$$
- Think about dividing a_0 into R pieces of length a_r
- p_2 determines r based on weighting:

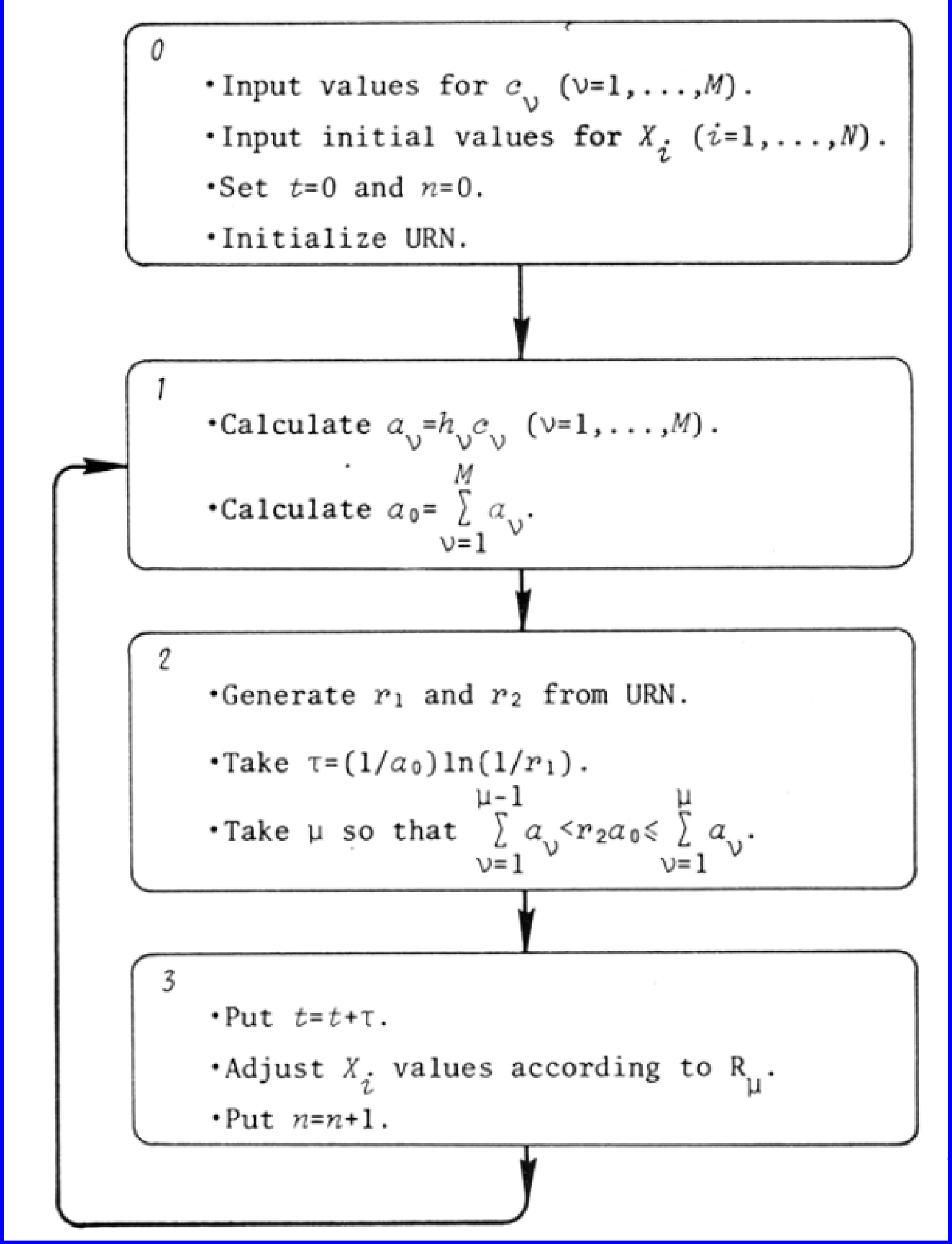


Step 3 Update the System State

- Update $t = t + \tau$
- Update $X = [X_1, X_2, \ldots X_C]$ according to the reaction stoichiometry
  - Subtract substrates and add products for the indicated rth reaction.
  - For each c, $X_c = X_c - S(r,c) + P(r,c)$
- In matrix format:
  - X(end+1,:) = X(end,:) – S(r,:) + P(r,:)
- Update reaction step counter (RC = RC+1).

Step 3 is to determine how each of C chemicals are affected

Schematic of SSA



0

- Input values for c_ν $(\nu=1,\ldots,M)$.
- Input initial values for X_i $(i=1,\ldots,N)$.
- Set $t=0$ and $n=0$.
- Initialize URN.

1

- Calculate $a_\nu=h_\nu c_\nu$ $(\nu=1,\ldots,M)$.
- Calculate $a_0=\sum_{\nu=1}^{M} a_\nu$.

2

- Generate r_1 and r_2 from URN.
- Take $\tau=(1/a_0)\ln(1/r_1)$.
- Take μ so that $\sum_{\nu=1}^{\mu-1} a_\nu < r_2 a_0 \leqslant \sum_{\nu=1}^{\mu} a_\nu$.

3

- Put $t=t+\tau$.
- Adjust X_i values according to R_μ.
- Put $n=n+1$.

Advantages of Gillespie

- Exact simulation; doesn't make linear approximation of probability = a_0 Dt.
- This means that time steps can be longer;

  Dt (simple time step) <<τ(Gillespie time step)
- No need to predetermine time steps
- Time step responds to changes in rates as numbers of particles change.
- *No conditions: always is accurate.*

Disadvantages of Gillespie

- Gets very slow when any one reaction is very fast relative to others. (stiff systems)
- Fast reaction occurs with fast rates or if there are a lot of a reactant.
- Thus, can be ineffective even though it is accurate in theory.
- Tau-leap algorithm deals with these situations

Four Ways to Simulate a Poisson Process:

- Take tiny time steps (<<1 events)
  - **“simple” algorithm**
  - Always accurate if time step is small enough but very expensive. Safe time step may depend on state.
- Match time step to next event (1 event)
  - **Gillespie’s “exact” algorithm**
  - Always accurate, but moderately expensive
- Take medium time steps (0 to ~20 events)
  - **“Tau-leap” algorithm**
  - Accurate if system doesn’t change significantly in a time step; less expensive
- Take large time steps (> 10 events)
  - **Gillespie’s “Chemical Langevin Equation” algorithm**
  - Accurate if system doesn’t change significantly in a time step, but > 10 events occur. Least expensive

Tau-leap method

- When there are lots of some type of chemical, you don't need to take so few steps for reactions involving this chemical, since your system doesn't change much anyway.

- Tau-leap:
  - for a given size of time step,
  - determine the number of each reaction that occur in that time step using the Poisson distribution
  - update all reactions simultaneously

- Assumes: system (rates) doesn't change much in one time step (like all ODEs).

D. Gillespie and L. Petzold, 2003, J. Chem. Phys. **119**, 8229.

Expected Number of Events:

- Like Gillespie:
  - The rate that each reaction r occurs is a_r
  - It is calculated in the exact same way.

- Unlike Gillespie:
  - For each r, ask: how many reactions will take place in time Dt?
  - Expected value is $\lambda_r = a_r * Dt$.
  - Recall: Poisson distribution: what is the probability of getting n when the expected value is λ ?

$$P(n) = \frac{\lambda^n e^{-\lambda}}{n!}$$

Algorithm for actual number of events:

- From last slide, have calculated the expected value
- Pick p_r , a random number between 0 and 1, uniform distribution.
- Psum = 0
- In a loop starting at n = 0

  Calculate P(n) given λ_r from $P(n) = \frac{\lambda_r{}^n e^{-\lambda_r}}{n!}$

  Psum = Psum + P(n)

  if Psum > p_r , then n_r = n; exit loop
- n gives how many of this reaction occurs.
- Do this for each reaction!

Update the System State:

- On last two slides, for each reaction:
  - Calculated estimated number of reactions, λ_r
  - Calculated the actual number of reactions, n_r
- Then, update values of each chemical c by using the reaction stoichiometry multiplied by n_r
- In matrix format:

  X(end+1,:) = X(end,:)+ n*(-S + P);

  Where n is a vector of the n_r.

How to Determine the Time Step?

- Choice 1: fix the time step throughout the simulation (easy, but inaccurate and/or inefficient)
- Choice 2: adjustable time step depends on system state (accurate and efficient)
  - To do this, want largest time step that isn't expected to change the system state much.
  - Must decide this BEFORE finding the random numbers
  - Thus must use the EXPECTED fractional change
    - Change =λ*(-S + P);
    - Current = X(end,:)
    - Consider: Max(abs(change./current))
      - If > RelTol, make step smaller.

Gillespie

- Step 0: set up S,P,K
- Step 1: Calculate

  a_r

  a_0 =sum(a_r)
- Step 2: Use 2 random numbers to pick when the next event occurs and which reaction it is.
- Step 3: update system for the one reaction only

Tau-leap

- Step 0: set up S,P,K
- Step 1: Calculate

  a_r

  $\lambda_r = a_r \Delta \mathrm{t}$

  (adjust time step until OK)
- Step 2: use R random numbers to determine number of events, n_r , for each reaction.
- Step 3: update system for all R reactions

Conditions for tau-leap

- *Condition*: Require Dt to be *small* enough that so the system doesn’t change appreciably during that step (same as any deterministic ODE solver)

Four Ways to Simulate a Poisson Process:

- Take tiny time steps (<<1 events)
  - **“simple” algorithm**
  - Always accurate if time step is small enough but very expensive. Safe time step may depend on state.
- Match time step to next event (1 event)
  - **Gillespie’s “exact” algorithm**
  - Always accurate, but moderately expensive
- Take medium time steps (0 to ~20 events)
  - **“Tau-leap” algorithm**
  - Accurate if system doesn’t change significantly in a time step; less expensive
- Take large time steps (> 10 events)
  - **Gillespie’s “Chemical Langevin Equation” algorithm**
  - Accurate if system doesn’t change significantly in a time step, but > 10 events occur. Least expensive

Chemical Langevin Equations

$$\frac{dX_i(t)}{dt} = \sum_{j=1}^{M} \nu_{ji} a_j(\mathbf{X}(t)) + \sum_{j=1}^{M} \nu_{ji} a_j^{1/2}(\mathbf{X}(t)) \Gamma_j(t)$$

$$a_j(\mathbf{x}) = c_j h_j(\mathbf{x}) .$$

- A way to simulate chemical reactions where numbers of elements are
  - large enough to use continuous model,
  - but small enough to consider stochastic process
- Name and method are motivated by the Langevin Equation for diffusion.

Gillespie DT. 2000. *The Journal of Chemical Physics* 113:297-306

Gillespie exact SSA, tau-leap, Chemical Langevin Algorithmic Notes

1. All three methods are reaction-based.
2. Use S, P, and K for stoichiometry and rate constants
3. SSA is more accurate
4. A combination of Tau-leap and CLE, (in which normal distribution replaces Poisson when expected number > 15) is faster for certain systems (but can be slower for others).

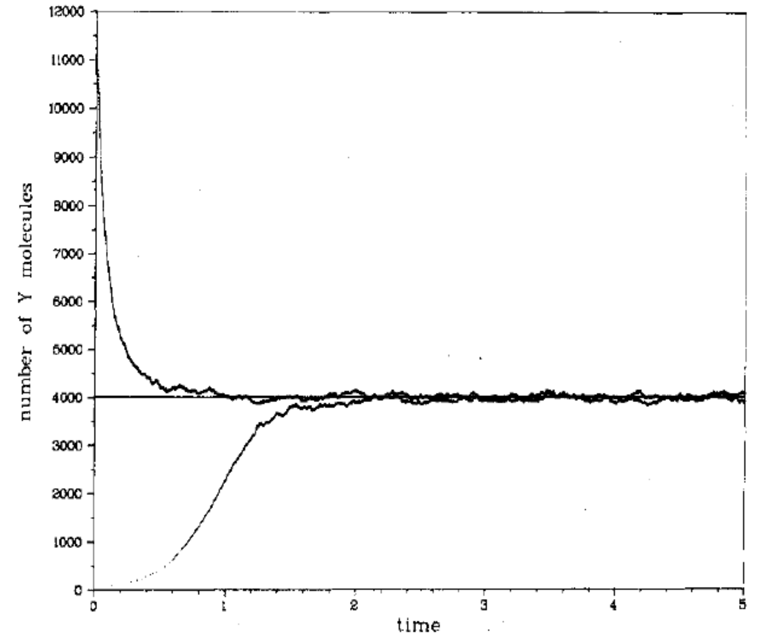
Assignment 9

Using SSA and tau-leap algorithms to reproduce Fig7 in the following paper: Gillespie DT (1977) The Journal of Physical Chemistry 81: 2340-2361. Please provide the running time to compare the efficiency of the two algorithms.

$$\overline{X} + Y \xrightarrow{c_1} 2Y$$

$$2Y \xrightarrow{c_2} Z$$

Figure 7. A 40 rpd plot of two stochastic simulation runs for reactions 29, with $c_1X = 5$ and $c_2 = 0.00125$, corresponding to the deterministic steady state $Y = 4000$; one run starts at $Y = 40$, and the other run starts at $Y = 12000$.


number of Y molecules
time