Simulation Algorithms

A Brief Introduction

Kai Zhao

CMACS Workshop 2012, Lehman College

January 12, 2012

Outline

- The Problem
- ODE
- Gillespie Algorithm

The Problem

We have a set of reactions.
Eg. DNA transcription

- Gene $D \to$ mRNA M;
- mRNA $M \to$ Protein P;
- Both M and P can degrade.

Describe in chemical reaction equations:

$$D \xrightarrow{k_1} D + M$$

$$M \xrightarrow{k_2} M + P$$

$$M \xrightarrow{k_3} \Phi$$

$$P \xrightarrow{k_4} \Phi$$

The Problem

- We know a set of the chemical reactions;
- We know the concentration of each molecules at initial time $t = 0$;
- We want to know the concentration of each molecule at time t.

Simulation Algorithms in BioNetGen

Two Simulation Algorithms are used in BioNetGen:

i simulation_ode()
An Ordinary Differential Equation (ODE) solver
  - Explicit Euler Method
  - Implicit Euler Method

ii simulation_ssa()
Stochastic Simulation Algorithm (SSA)
  - Gillespie Algorithm

Simulation Algorithms in BioNetGen

ODE

- When the numbers of molecules are large, any two reactions can happen at the same time.
- ODE (Euler method) represents the collection of reactions occurring simultaneously all through the reaction volume.
- Concentration Description

SSA

- When the numbers of molecules are small, reactions happen in some random order, rather than simultaneously.
- SSA (Gillespie Algorithm) represents one reaction occurring at a time.
- Probability Description

Simulating by Ordinary Differential Equation (ODE) Solver

Chemical Reactions as Differential Equations

The chemical reactions we have:

$$D \xrightarrow{k_1} D + M$$

$$M \xrightarrow{k_2} M + P$$

$$M \xrightarrow{k_3} \Phi$$

$$P \xrightarrow{k_4} \Phi$$

Translate to differential equations:

$$\frac{d[D]}{dt} = ?$$

$$\frac{d[M]}{dt} = ?$$

$$\frac{d[P]}{dt} = ?$$

Chemical Reactions as Differential Equations

The chemical reactions we have:

$$D \xrightarrow{k_1} D + M$$

$$M \xrightarrow{k_2} M + P$$

$$M \xrightarrow{k_3} \Phi$$

$$P \xrightarrow{k_4} \Phi$$

Translate to differential equations:

$$\begin{aligned}
\frac{d[D]}{dt} &= k_1[D] - k_1[D] = 0 \\
\frac{d[M]}{dt} &= k_1[D] + k_2[M] - k_2[M] - k_3[M] \\
&= k_1[D] - k_3[M] \\
\frac{d[P]}{dt} &= k_2[M] - k_4[P]
\end{aligned}$$

Initial Value Problem

- The Initial Value Problem (IVP)
  - Differential Equations
  - Initial Condiitons

$$\begin{cases} y'(t) = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

- Problem
  What is the value of y at time t?

Numerical Differentiation

- Definition of Differentiation:

$$\frac{df}{dx} = f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- Problem
  We do not have an infinitesimal h.
- Solution
  Use a small h as an approximation.

Forward Difference & Backward Difference

- Forward Difference

$$\frac{df}{dx} \approx f'(x)_{approx} = \frac{f(x+h) - f(x)}{h}$$

- Backward Difference

$$f'(x)_{approx} = \frac{f(x) - f(x-h)}{h}$$

Numerical Differentiation: Example

- Compute the derivative of function

$$f(x) = e^x$$

- At point $x = 1.15$

$$f'(1.15) = f(1.15) \approx 3.1581$$

- Forward Difference, $h = 0.001$:

$$f'(1.15) \approx \frac{f(1.151) - f(1.15)}{0.001} = 3.1598$$

Explicit Euler Method

- Consider Forward Difference

$$y'(t) \approx \frac{y(t+\Delta t) - y(t)}{\Delta t}$$

- which implies

$$y(t+\Delta t) \approx y(t) + \Delta t \cdot y'(t)$$

Explicit Euler Method

- Split time t into n slices of equal length Δt

$$\begin{cases} t_0 = 0 \\ t_i = i \cdot \Delta t \\ t_n = t \end{cases}$$

- The **Explicit Euler Method Formula**

$$y(t_{i+1}) = y(t_i) + \Delta t \cdot y'(t_i)$$

Explicit Euler Method: Algorithm

Input: f, y_0, t_0, t, dt
Output: y_c
$t_c \leftarrow t_0$
$y_c \leftarrow y_0$
while $t_c < t$ **do**
  $y_c \leftarrow y_c + dt \cdot f(t_c, y_c)$
  $t_c \leftarrow t_c + dt$
end
return y_c

Implicit Euler Method

- Consider Backward Difference

$$y'(t) \approx \frac{y(t) - y(t - \Delta t)}{\Delta t}$$

- which implies

$$y(t) \approx y(t - \Delta t) + \Delta t \cdot y'(t)$$

Implicit Euler Method

- Split the time into slices of equal length

$$y(t_{i+1}) \approx y(t_i) + \Delta t \cdot y'(t_{i+1})$$

- The above differential equation should be solved to get the value of $y(t_{i+1})$
  - Extra computation
  - Sometimes worth because implicit method is more accurate

A Simple Example

- Try to solve IVP

$$\begin{cases} y'(t) = e^{-t} + t \\ y(0) = 1 \end{cases}$$

- What is the value of y when $t = 0.5$?
- The analytical solution is

$$y = -e^{-t} + \frac{1}{2}t^2 + 2$$

A Simple Example

- Using explicit Euler method

$$y_{i+1} = y_i + dt \cdot (e^{t_i} + t_i)$$

- We choose different dts to compare the accuracy:

$$\begin{cases} dt_1 = 0.05 & \Rightarrow t = 0, 0.05, 0.1, \ldots, 0.5 \\ dt_2 = 0.025 & \Rightarrow t = 0, 0.025, 0.05, \ldots, 0.5 \\ dt_3 = 0.0125 & \Rightarrow t = 0, 0.0125, 0.025, \ldots, 0.5 \end{cases}$$

A Simple Example

t	exact	error $dt_1 = 0.05$	error $dt_2 = 0.025$	error $dt_3 = 0.0125$
0.1	1.10016	0.00014	0.00006	0.00003
0.2	1.20126	0.00050	0.00024	0.00011
0.3	1.30418	0.00107	0.00052	0.00025
0.4	1.40968	0.00182	0.00089	0.00044
0.5	1.51846	0.00274	0.00135	0.00067

At some given time t, error is proportional to dt.

Euler Method: Instability

- For some equations called *Stiff Equations*, Euler method requires an extremely small dt to make result accuracy

$$y'(t) = -k \cdot y(t), k > 0$$

- Explicit Euler method Formula

$$y_{i+1} = y_i - \Delta t \cdot t \cdot y_i = (1 - \Delta t \cdot k) y_i$$

- The choice of Δt matters!

Euler Method: Instability

- Assume $k = 5$

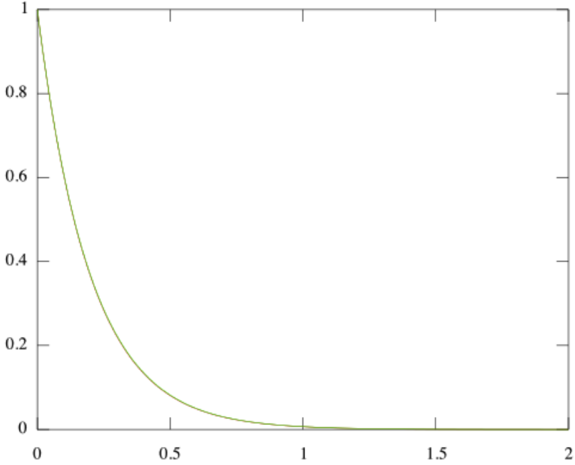
$$\begin{cases} y'(t) = -5y(t) \\ y(0) = 1 \end{cases}$$

- Analytical Solution is

$$y(t) = e^{-5t}$$

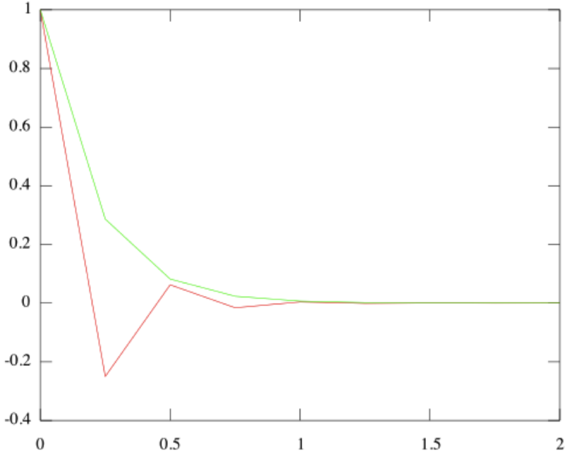
- Try Explicit Euler Method with different dts.

Explicit Euler Method: Instability



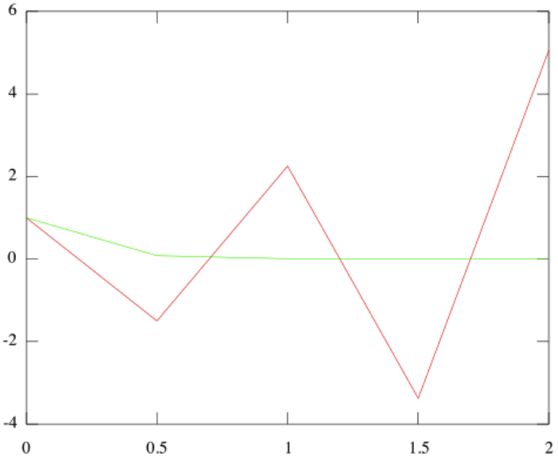
$dt = 0.002$ works.

Explicit Euler Method: Instability



$dt = 0.25$ oscillates, but works.

Explicit Euler Method: Instability



$dt = 0.5$, instability.

Explicit Euler Method: Instability

For large dt, explicit Euler Method does not guarantee an accurate result.

t	exact	err % $dt = 0.5$	err % $dt = 0.25$	err % $dt = 0.002$
0.4	0.135335	6.389056	2.847264	0.010017
0.8	0.018316	82.897225	1.853096	0.019933
1.2	0.002479	906.714785	1.393973	0.02975
1.6	0.000335	10061.73321	1.181943	0.039469
2	0.000045	111507.9831	0.663903	0.04909

Implicit Euler Method

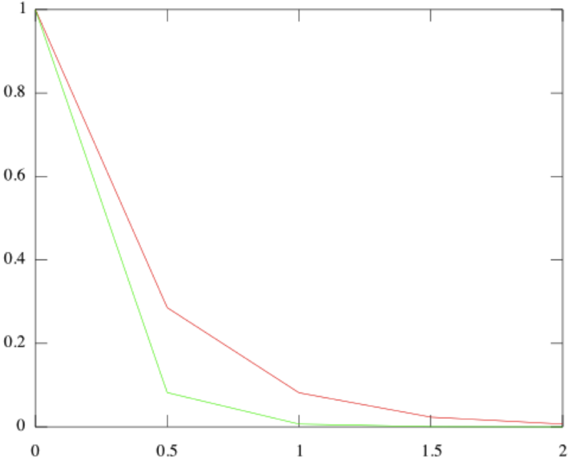
- Implicit Euler Method Formula

$$y_{i+1} = y_i - dt \cdot 5 \cdot y_{i+1}$$

- which implies:

$$y_{i+1} = \frac{y_i}{1 + 5dt}$$

Impilcit Euler Method: Instability



$dt = 0.5$, Oscillation eliminated.

Simulating by Stochastic Simulation Algorithm

Gillespie Algorithm: Overview

Iteration:

- Determine next reaction time
- Determine which reaction happens
- Update number of molecules

Chemical Reaction as a Stochastic Process

- Poisson Process
  - An event can happen in a time interval with probability p
  - Counts the number of events and the times these events occur
  - Event: a reaction takes place
  - Time: the time step the system moves forward
- The simulation process is a Poisson process. The times that an event occurs follow an exponential distribution.

Exponential Distribution

- Assume in time interval Δt, the probability of an event happens is p
  - p only depends on Δt
  - The probability of two or more events happen is ignored because it is too small
- The time that next event happens:
  - Event does not happen in time interval $\tau_1 + \tau_2$: $P(t > \tau_1 + \tau_2)$
  - Event does not happen in time interval τ_1: $P(t > \tau_1)$, AND Event does not happen in time interval τ_2: $P(t > \tau_2)$.

  Memorylessness:

$$P(t > \tau_1 + \tau_2) = P(t > \tau_1) \cdot P(t > \tau_2)$$

Exponential Distribution

$$P(t > \tau = \frac{m}{n}) = P(t > \frac{m-1}{n})P(t > \frac{1}{n})$$
$$P(t > \tau = \frac{m}{n}) = P^m(t > \frac{1}{n})$$

$$P(t > 1 = \frac{n}{n}) = P^n(t > \frac{1}{n})$$
$$P(t > \tau) = P^{m/n}(t > 1) = P^{\tau}(t > 1)$$

Let $\lambda = -\ln P(t > 1)$, then

$$P(t > \tau) = e^{-\lambda\tau}$$

Exponential Distribution

With $P(t > \tau) = e^{-\lambda\tau}$, we can get:
Cumulative Distribution Function (CDF):

$$F(t) = 1 - e^{-\lambda t}$$

Probability Distribution Function (PDF):

$$f(t) = \lambda e^{-\lambda t}$$

Conclusion: The time follows an exponential distribution.

Determine Time

- Assume the probability of a reaction happens in time Δt is $p^* \Delta t$, when Δt is extremely small:

$$F(\Delta t) = 1 - e^{-\lambda \Delta t} = p^* \Delta t$$

- As Δt is extremely small, with Taylor expansion:

$$F(\Delta t) = 1 - e^{-\lambda \Delta t} \approx \lambda \Delta t = p^* \Delta t$$

$$\Rightarrow \lambda = p^*$$

- The time of next reaction follows an exponential distribution:

$$f(t) = p^* e^{-p^* t}$$

Choose a Reaction

- Denote the probability of ONLY Reaction R_i:

$$\mathrm{Rct}_{i1} + \mathrm{Rct}_{i2} \xrightarrow{k_i} \mathrm{Pdt}_{i1} + \mathrm{Pdt}_{i2}$$

happens in time period Δt as $p_i \Delta t$.

  - If Rct_{i1} and Rct_{i2} are different.

$$p_i = c_i \cdot \mathrm{Rct}_{i1}\mathrm{Rct}_{i2}$$

  - If Rct_{i1} and Rct_{i2} are identical.

$$p_i = c_i \cdot \mathrm{Rct}_{i1}(\mathrm{Rct}_{i1} - 1)$$

where

  - Rct_{i1} and Rct_{i2} are the numbers of molecules in the volume
  - c_i is a reaction rate description, proportional to k_i

The Gillespie Algorithm

- Initialize number of each molecules from initial concentrations; set $t \leftarrow 0$.
- Iterate until t is larger than end time
  - At time t, determine a time interval τ that a reaction will happen. Draw a random number from an exponential distribution with parameter p^*:
  $$f(\tau) = p^* e^{-p^* \tau}$$
  $$p^* = \sum_i p_i$$
  - Choose the reaction that happens at in time $(t, t + \tau)$ Reaction R_i has probabiltity $\frac{p_i}{p^*}$
  - Update molecule numbers based on reaction chosen.
  - Update $t \leftarrow t + \tau$.

The END

Thanks!