Stochastic Simulation of Chemical Reactions

Computational Models for Complex Systems

Paolo Milazzo

Dipartimento di Informatica, Università di Pisa
http://pages.di.unipi.it/milazzo
milazzo@di.unipi.it

Laurea Magistrale in Informatica
A.Y. 2019/2020

Introduction

We have seen that the dynamics of chemical reactions can be studied by analyzing the associated system of ODEs obtained through the application of the law of mass action

ODEs are continuous (both in variables values and time) and deterministic

- Sometimes chemical reactions exhibit random behaviors
- This led to the definition of stochastic simulation algorithms for chemical reactions

See also:

- Stochastic Simulation of Chemical Kinetics by Daniel T. Gillespie
  Freely accessible if you are within the UniPi subnet

Randomness in chemical reactions (1)

The occurrence of a chemical reaction in a chemical solution is actually difficult to be predicted in advance.

- the movement of molecules in the chemical solution is related with the interaction (elastic collisions) of the molecules with the fluid medium containing them
- so, if we ignore such low level interaction, we cannot predict exactly when a specific combination of reactants will meet (and react) in the chemical solution
- even in the case of a reaction with a single reactant (e.g. a dissociation $A \xrightarrow{k} B + C$) it is not possibile to predict exactly when the reactant will "decide" to react

Randomness in chemical reactions (2)

In the case of high concentrations of reactants, the law of large numbers allow us to ignore random aspects of chemical reactions

- This justifies the use of ODEs

But when numbers are small (one or a few molecules)

- random aspects become crucial
- and also the use of descrete variables becomes necessary

Example, think about $A \xrightarrow{k} B + C$ and assume that only one molecule A is present in the solution.

- ODEs would describe the concentration of A to pass slowly (and continuously) from 1 to 0
- in the real system the number of instances of A would pass from 1 to 0 with a discrete (instantaneous) change

Gillespie's Stochastic Approach

Gillespie's Stochastic Simulation Algorithm (SSA) is an exact procedure for simulating the time evolution of a chemical reacting system by taking proper account of the randomness inherent in such a system.

Given a set of reactions $\mathcal{R} = \{R_1, \ldots, R_n\}$, the SSA:

- assumes a stochastic reaction constant c_μ for each chemical reaction $R_\mu \in \mathcal{R}$
- such that $c_\mu dt$ is the probability that a particular combination of reactants of R_μ react in an infinitesimal time interval dt

Gillespie's Stochastic Approach

The constant c_μ is used to compute the propensity (or stochastic rate) of R_μ to occur in the whole chemical solution, denoted a_μ, as follows:

$$a_\mu = h_\mu c_\mu$$

where h_μ is the number of distinct molecular reactant combinations.

Let R_μ be

$$\ell_1 S_1 + \ldots + \ell_\rho S_\rho \xrightarrow{c} \ell'_1 P_1 + \ldots + \ell'_\gamma P_\gamma$$

In accordance with standard combinatorics, the number of distint reactant combinations of R_μ in a solution with X_i molecules of S_i, with $1 \leq i \leq \rho$, is given by

$$h_\mu = \prod_{i=1}^{\rho} \binom{X_i}{\ell_i}$$

Gillespie's Stochastic Approach

Example:

solution with X_1 molecules S_1 and X_2 molecules S_2
(remark: number of molecules, not concentrations...)

reaction $R_1 : S_1 + S_2 \rightarrow 2S_1$

- $h_1 = \binom{X_1}{1}\binom{X_2}{1} = X_1 X_2$
- $a_1 = X_1 X_2 c_1$

reaction $R_2 : 2S_1 \rightarrow S_1 + S_2$

- $h_2 = \binom{X_1}{2} = \frac{X_1(X_1-1)}{2}$
- $a_2 = \frac{X_1(X_1-1)}{2} c_2$

Note that propensity a_μ is similar, for suitable kinetic constants, to the mass action rates (note: here unitary volume is assumed):

- For R_1 with $k_1 = c_1$, the law of mass action gives $k_1[S_1][S_2] \approx a_1$
- For R_2 with $k_2 = c_2/2$, the law of mass action gives $k_2[S_1]^2 \approx a_2$

Gillespie's Stochastic Approach

Propensity a_μ is used in Gillespie's approach as the parameter of an exponential probability distribution modelling the time between subsequent occurrences of reaction R_μ.

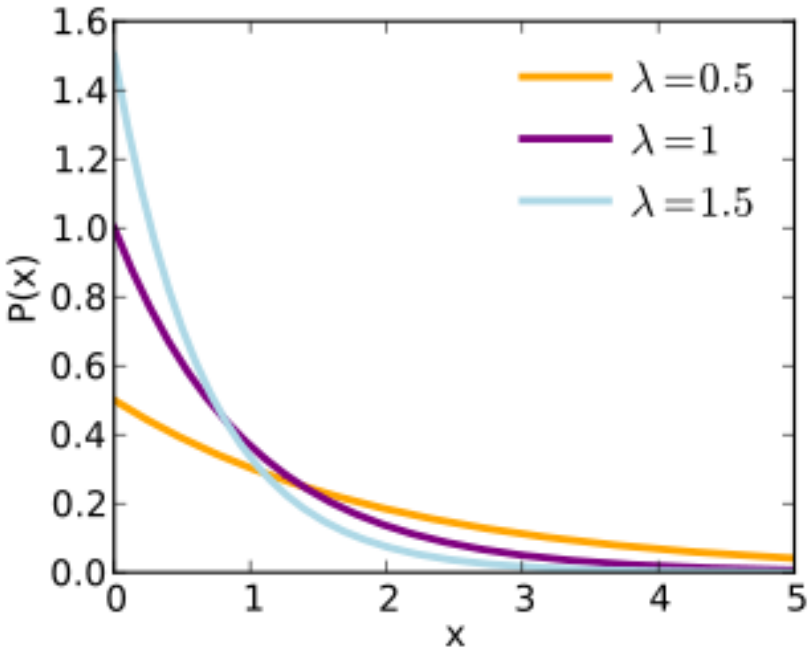
Exponential distrubution is a continuous probability distribution (on $[0, \infty]$) describing the timing between events in a Poisson process, namely a process in which events occur continuously and independently at a constant average rate (taken as parameter).

The probability density function f and the cumulative distribution function F of an exponential distribution with parameter λ are as follows:

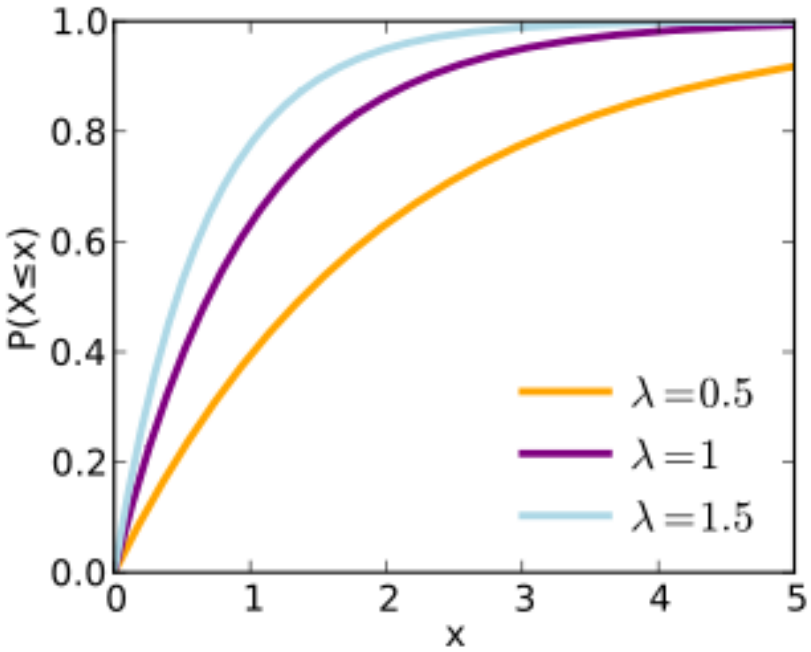
$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases} \qquad F(x) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Gillespie's Stochastic Approach

Probability density function



Cumulative distribution function



The mean of an exponentially distributed variable with parameter λ is $\frac{1}{\lambda}$.

Gillespie's Stochastic Approach

Two important properties of the exponential distribution hold:

- The exponential distribution is memoryless:

$$P(X > t + s \mid X > s) = P(X > t)$$

  This allows a simulation algorithm in which the exponential distribution is used to forget about the history of the simulation

- Let $X_1, \ldots, X_n$ be independent exponentially distributed random variables with parameters $\lambda_1, \ldots, \lambda_n$. Then

$$X = min(X_1, \ldots, X_n) \text{ is also exponentially distributed}$$

  with parameter $\lambda = \lambda_1 + \ldots + \lambda_n$. This allows a simulation algorithm to use a unique exponential distribution for the whole set of reactions to be simulated

Gillespie's Stochastic Simulation Algorithm (SSA)

Given:

- a set of molecular species $\{S_1, \ldots, S_n\}$
- initial numbers of molecules of each species $\{X_1, \ldots X_n\}$ with $X_i \in I\!N$
- a set of chemical reactions $\{R_1, \ldots R_M\}$

Gillespie's algorithm computes a possible evolution of the system

Gillespie's Stochastic Simulation Algorithm (SSA)

Gillespie's algorithm:

The state of the simulation:

- is a vector representing the multiset of molecules in the chemical solution (initially $[X_1, \ldots, X_n]$)
- a real variable t representing the simulation time (initially $t = 0$)

The algorithm iterates the following steps until t reaches a final value t_{stop}.

1. The time $t + \tau$ at which the next reaction will occur is randomly chosen with τ exponentially distributed with parameter $a_0 = \sum_{\nu=1}^{M} a_\nu$;
2. The reaction R_μ that has to occur at time $t + \tau$ is randomly chosen with probability $\frac{a_\mu}{\sum_{\nu=1}^{M} a_\nu}$ (that is $\frac{a_\mu}{a_0}$).

At each step t is incremented by τ and the multiset representing the chemical solution is updated by subtracting reactants and adding products.

Gillespie's Stochastic Simulation Algorithm (SSA)

Example: Let's consider the following reactions:

$$R_1: \quad A \xrightarrow{2} B + C \qquad R_2: \quad B + C \xrightarrow{0.1} A$$

and the following initial quantities: $A_0 = 10, B_0 = 5, C_0 = 20$ represented as $[10, 5, 20]$ in vector notation.

The initial state is $\mathbf{X} = [\mathbf{10}, \mathbf{5}, \mathbf{20}], \mathbf{t} = \mathbf{0}$.

The first iteration of Gillespie's algorithm is as follows:

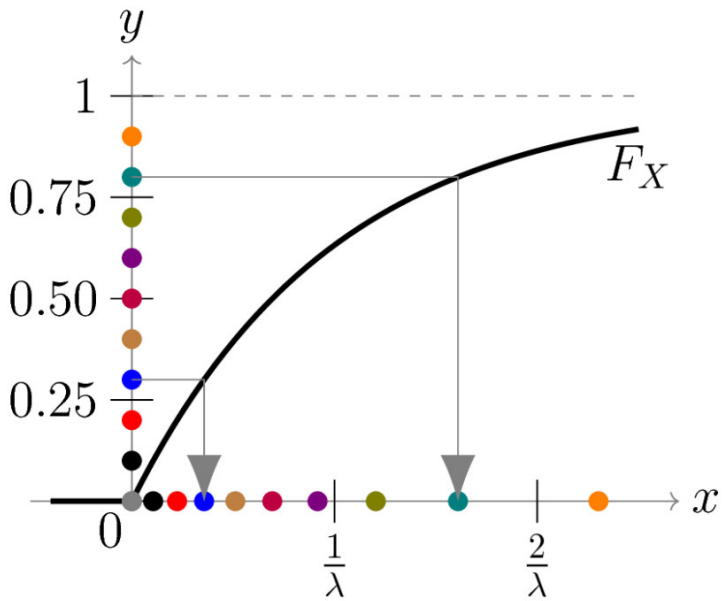
- compute propensities:
  - $a_1 = 10 \cdot 2 = 20 \qquad a_2 = 5 \cdot 20 \cdot 0.1 = 10 \qquad a_0 = a_1 + a_2 = 30$
- generate τ exponentially distributed with parameter $a_0 = 30$
  - the average τ is $1/a_0 = 1/30 \sim 0.0333$
- choose the reaction that has to occur: R_1 with probability $a_1/a_0 = 10/30$ and R_2 with probability $a_2/a_0 = 20/30$

If R_2 is chosen, the state of the simulation becomes $\mathbf{X} = [\mathbf{11}, \mathbf{4}, \mathbf{19}], \mathbf{t} = \tau$

Gillespie's algorithm: implementation details

Implementation detail: Generation of τ (exponentially distributed)

- A random number with any probability distribution can be computed from a random number with uniform distribution by applying the inversion sampling method
- The idea is to use the inverse of the cumulative distribution function



Given the cumulative distribution function F of a probability distribution *dist* and a uniformly distributed random variable U, the variable $X = F^{-1}(U)$ is a random variable with distribution *dist*

Gillespie's algorithm: implementation details

In the case of the exponential distribution, the cumulative distribution function (for $x \geq 0$) is $F(x) = 1 - e^{-\lambda x}$

Let's invert the function:

$$F(x) = 1 - e^{-\lambda x} \implies 1 - F(x) = e^{-\lambda x} \implies \ln(1 - F(x)) = -\lambda x$$
$$\implies -\frac{1}{\lambda}\ln(1 - F(x)) = x \implies \frac{1}{\lambda}\ln(\frac{1}{1 - F(x)}) = x$$

So, we obtain $F^{-1}(Y) = \frac{1}{\lambda}\ln(\frac{1}{1-Y})$. Since Y is uniformly distributed, also $1 - Y$ is uniformly distributed. This allows us to simplify the definition of F^{-1} as follows: $F^{-1}(Y) = \frac{1}{\lambda}\ln(\frac{1}{Y})$

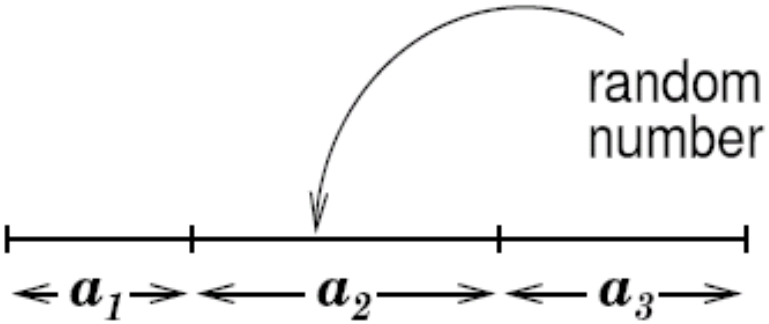
- τ exponentially distributed with parameter a_0 can be computed as $\tau = \frac{1}{a_0}\ln(\frac{1}{Y})$ with Y obtained from a standard random number generator

Gillespie's algorithm: implementation details

Implementation detail: Choice of reaction R_μ (with probability $\frac{a_\mu}{a_0}$)

Idea:

1. generate a random number N uniformly distributed in $[0, a_0)$, that is $N = n \cdot a_0$ with $n \in [0, 1)$ obtained from a standard number generator
2. Start summing $a_1, a_2, \ldots$
3. as soon as the sum becomes greather than N, the number of completed iterations gives you μ



Formally:

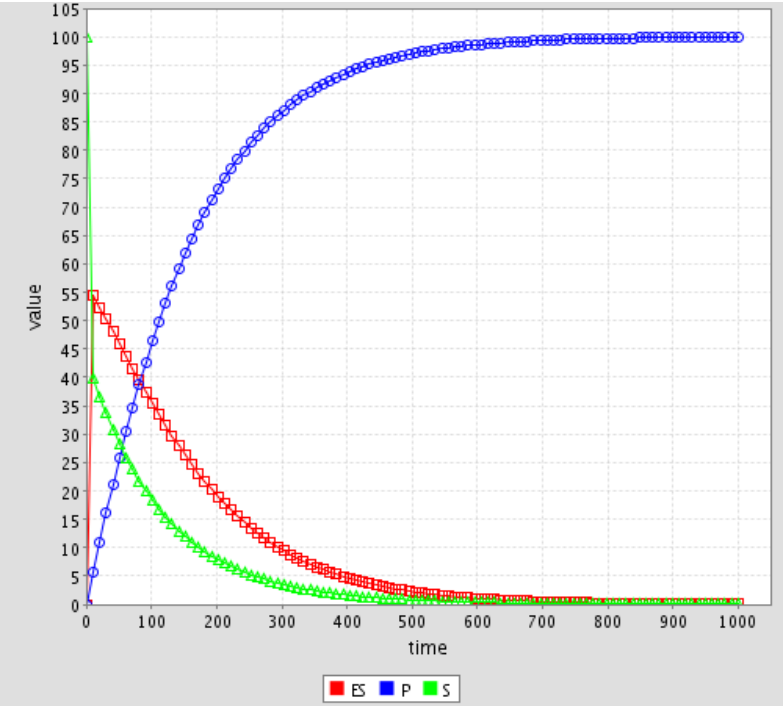
- μ is the smallest integer k satisfying $\sum_{i=1}^{k} a_i > n a_0$ with n uniformly distributed in $[0, 1)$

ODEs vs SSA

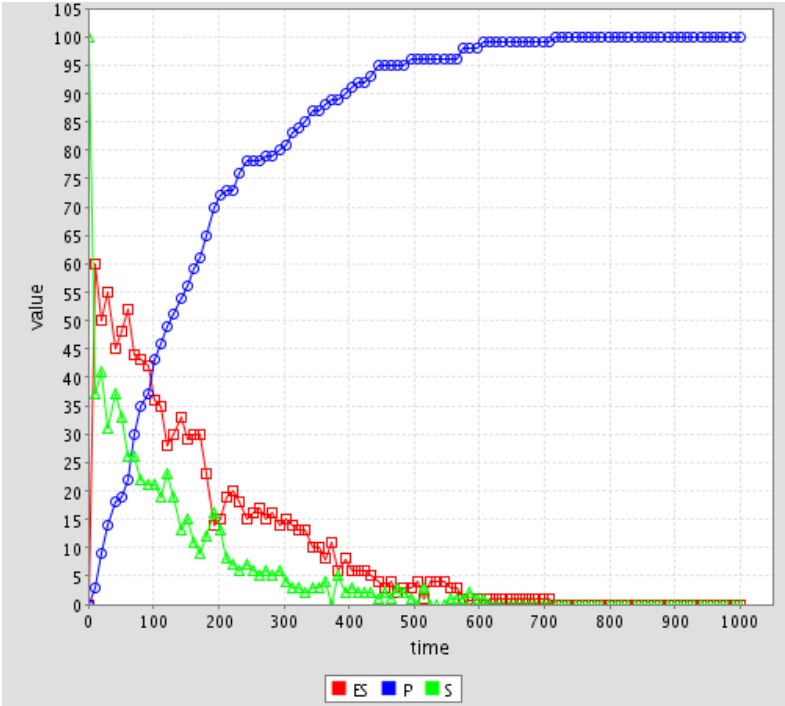
Let us compare the deterministic and stochastic approach with some examples of (bio)chemical reactions:

First example: Enzymatic activity: $E + S \underset{10.0}{\overset{0.3}{\rightleftharpoons}} ES \overset{0.01}{\rightarrow} E + P$

Starting with: $100E$ and $100S$.



ODEs



SSA

ODEs vs SSA

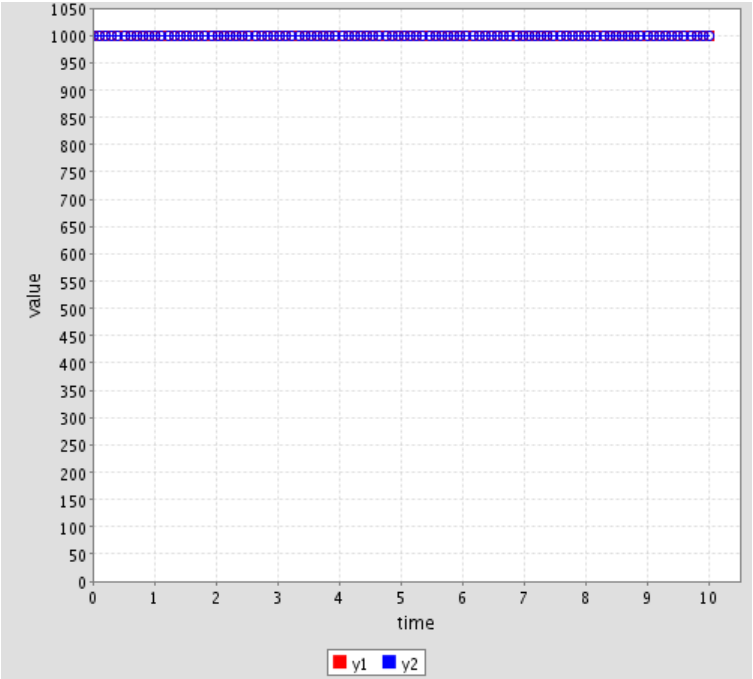
Second example: Lotka/Volterra reactions:

$$Y_1 \xrightarrow{10} 2Y_1$$

$$Y_1 + Y_2 \xrightarrow{0.01} 2Y_2$$

$$Y_2 \xrightarrow{10} Z$$

Starting with $1000\,Y_1$ and $1000\,Y_2$



ODEs

ODEs vs SSA

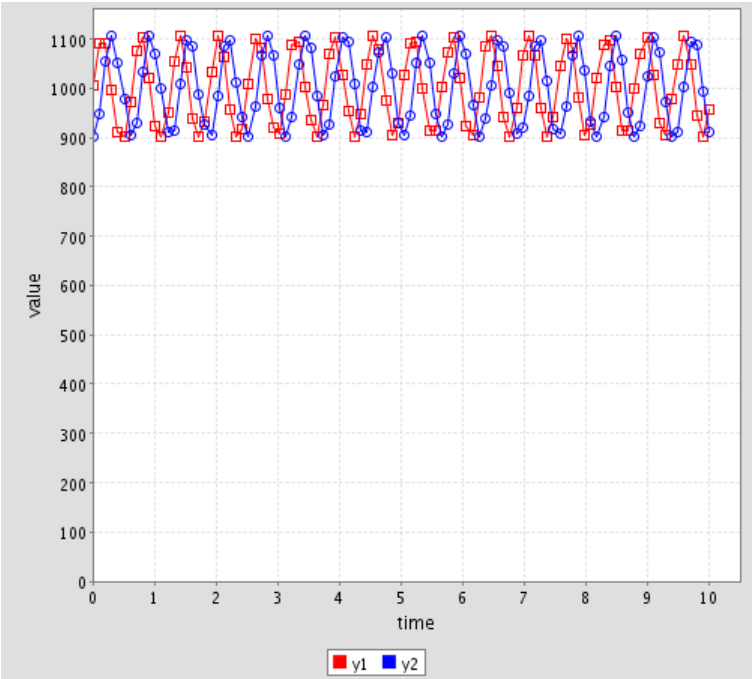
Second example: Lotka/Volterra reactions:

$$Y_1 \xrightarrow{10} 2Y_1$$

$$Y_1 + Y_2 \xrightarrow{0.01} 2Y_2$$

$$Y_2 \xrightarrow{10} Z$$

Starting with $1000\,Y_1$ and $900\,Y_2$ (slight perturbation).



ODEs

ODEs vs SSA

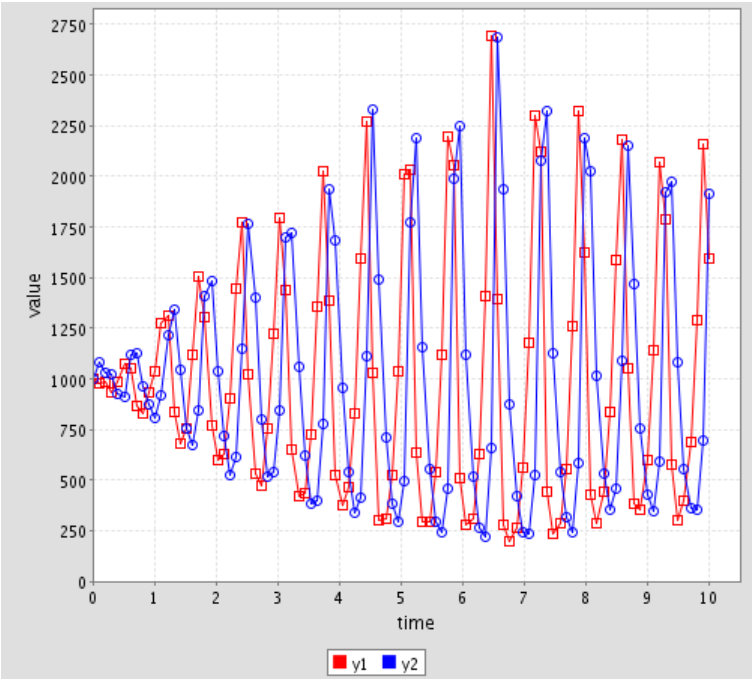
Second example: Lotka/Volterra reactions:

$$Y_1 \xrightarrow{10} 2Y_1$$

$$Y_1 + Y_2 \xrightarrow{0.01} 2Y_2$$

$$Y_2 \xrightarrow{10} Z$$

Starting with $1000\,Y_1$ and $1000\,Y_2$



SSA

ODEs vs SSA

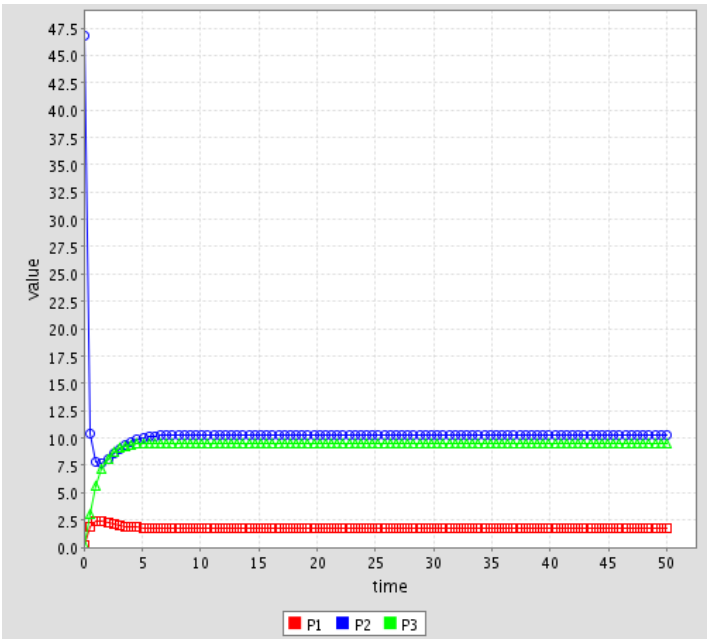
Third example: Negative feedback loop:

$$G_1 \xrightarrow{10} G_1 + P_1 \qquad P_1 + G_2 \underset{2}{\overset{10}{\rightleftharpoons}} P_1 G_2 \qquad P_1 \xrightarrow{1}$$

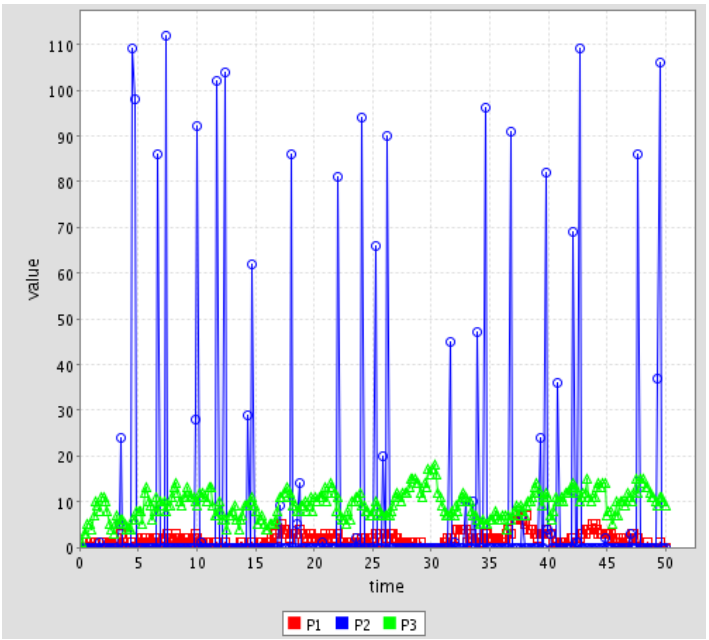
$$G_2 \xrightarrow{10000} G_2 + P_2 \qquad P_2 + G_3 \underset{20}{\overset{0.1}{\rightleftharpoons}} P_2 G_3 \qquad P_2 \xrightarrow{100}$$

$$G_3 \xrightarrow{10} G_3 + P_3 \qquad P_3 + G_1 \underset{20}{\overset{10}{\rightleftharpoons}} P_3 G_1 \qquad P_3 \xrightarrow{1}$$

Starting with $G_i = 1$ and $P_i = 0$



ODEs



SSA

Computational cost of Gillespie's algorithm

The computational cost of this detailed stochastic simulation algorithm may become extremely high for large models

- the key issue is that the time elapsing between two reactions can become extremely small

The algorithm becomes very inefficient when:

- there are large number of molecules
- kinetic constant are high

that is, when reaction rates increase...

Computational cost is the main disadvantage of stochastic simulation with respect to ODEs

Computational cost of Gillespie's algorithm

Several variants of Gillespie's algorithm aimed at reducing the computational cost have been proposed.

Exact approaches (improving the computation of each step):

- Gibson and Bruck proposed the use of some efficient data structures (indexed binary tree priority queues) to improve the choice of the reaction R_μ to occur at each step
- Cao et al. and McCollum et al. proposed dynamical ordering strategies for reaction propensities $a_1, a_2, \ldots$ in order to probabilistically reduce the time needed to choose R_μ at each step

See lecture notes for more details...

Computational cost of Gillespie's algorithm

Several variants of Gillespie's algorithm aimed at reducing the computational cost have been proposed.

Approximate approaches (reducing the number of steps):

- Gillespie proposed the τ-leaping method: the idea is to allow several reactions to take place in a single (longer) time step, under the condition that reaction rates do not change too much during that time.
- Gillespie et al. proposed the slow-scale Stochastic Simulation Algorithm (ssSSA) which separates fast reactions from slow reactions. At each step fast reactions are dealt with by assuming that they reach a dynamic equilibrium (so, only their steady state is computed). Slow reaction are simulated one by one as in the standard SSA.
- Hybrid simulation is a technique which combines ODEs with stochastic simulation: ODEs are applied to molecules occurring in big numbers, stochastic simulation to molecules occurring in small numbers

Implementations

Tools and libraries for the numerical simulation of chemical reaction systems often implement also (some variants of) Gillespie's algorithm:

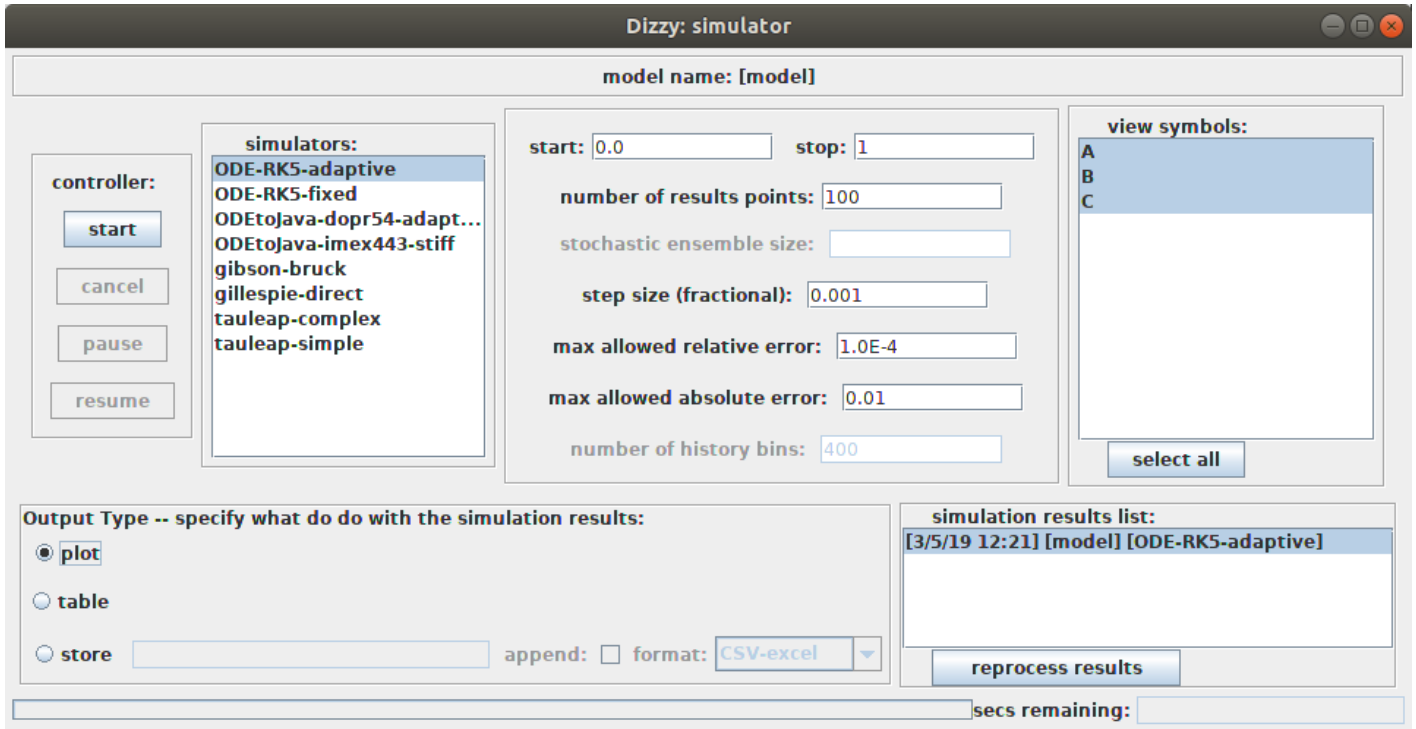
- COPASI (`http://copasi.org/`)
  Professional (free) tool
- libRoadRunner (`http://libroadrunner.org/`)
  Library for C++/Python
- Dizzy (available on the course web page)
  Multiplatform simulation tool. Unmantained, but very simple...
- Many other tools...
  See `sbml.org/SBML_Software_Guide/SBML_Software_Summary` for a list

In addition:

- StochKit (`https://sourceforge.net/projects/stochkit/`) is a state-of-the-art C++ library (with a Python binding) for the stochastic simulation of chemical reactions

Implementations

In Dizzy, stochastic simulation algorithms (Gillespie SSA, Gibson&Bruck, and τ-leaping) can be selected from the simulation window



Lessons learnt

Summing up:

- Stochastic (and discrete) simulation methods are more accurate than ODEs for the analysis of reaction systems, in particular when molecules are present in small quantities
- Gillespie's algorithm is THE stochastic simulation algorithm
- Gillespie's algorithm perform one step for each occurrence of a reaction in the chemical solution
  - performance problems...
- Approximate variants of Gillespie's algorithm improve performances (to some extent) with still a good accuracy

Exercises

- Implement Gillespie's algorithm in your favorite programming language
  - inspect its execution with a profiler... where does it spend the majority of its execution time?
  - test some implementation strategies to improve the computation of each single step (e.g. McCollum dynamical ordering strategy, or your own strategy). How much do they improve performances?