

Numerical methods for stochastic simulation of biochemical systems

Mohsen Ben Hassine¹, Radhi Mhiri², Lamine Mili³

¹ ISET de Nabeul, Computer Engineering, Tunisia

mbh851@yahoo.fr

² Faculté de sciences de tunis, Electrical Engineering, Tunisia

radhi.mhiri@yahoo.fr

³ Virginia tech, Electrical and Computer Engineering, USA

lmili@vt.edu

Abstract: Stochastic simulation of complex biochemical networks is a topic of current interest, in spite of the importance of the Gillespie algorithm it requires substantial amount of computational effort to simulate a complex system, Many algorithms are available now to control the two most important parameters in simulation: speed and accuracy. It can be proved that the stochastic modelling approach provide a conceptual bridge between stochastic chemical kinetics (the CME and SSA) and conventional deterministic chemical kinetics (the RRE).

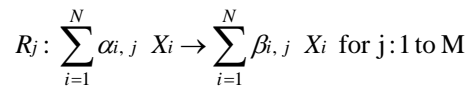
Keywords: SSA, CLE, Implicit tau, Explicit tau, Euler-Maruyama discretization, stiffness

1. Introduction

Through this paper we will outline the foundations of stochastic chemical kinetics beginning from the chemical master equation (CME), then the numerical realization of the stochastic process using a Monte Carlo strategy known as the stochastic simulation algorithm (SSA), followed by the approximate accelerated algorithm called the tau-leaping with its two essentials versions (explicit and implicit), next we can under certain conditions prove that the tau leaping can lead to the stochastic differential equation which called the chemical Langevin equation (CLE), and when the molecular populations of the reacting species become larger (in the thermodynamic limit) the CLE can in turn sometimes be approximated by an ordinary differential equation called the reaction rate equation (RRE). Finally we will provide a detailed description of the stochastic algorithms (which rarely found in literature), compare and discuss their efficiency with some examples.

2. The chemical master equation

We consider a well-stirred system of molecules of N chemical species $\{S_1, \dots, S_N\}$ interacting through M chemical reaction channels: R_1, \dots, R_M . The system is assumed to be confined to a constant volume, and to be in thermal equilibrium at some constant temperature. With $X_i(t)$ denoting the number of molecules of species S_i in the system at time t , we want to study the evolution of the state vector $X(t) = (X_1(t), \dots, X_N(t))$ given that the system was initially in some state $X(0) = x_0$. Each reaction channel R_j is assumed to be "elemental" in the sense that it describes a distinct physical event which happens essentially instantaneously. Elemental reactions are either unimolecular or bimolecular; more complicated chemical reactions (including trimolecular reactions) are actually coupled sequences of two or more elemental reactions. The system of biochemical reactions can be described as follow:



- R_j = reaction j ($j = 1, 2, \dots, M$)
- M = number of reactions
- N = number of species
- $X_i = X_1, X_2, \dots, X_N$ = molecules of species i in the system
- Let h_μ = Number of possible combinations of reactant molecules involved in reaction μ
- The rate constant c_μ
- a_μ = the probability of reaction μ occurring in time interval $(t; t + dt)$
- $a_\mu dt = h_\mu * c_\mu * dt$

By considering a discrete infinitesimal time interval $(t; t+dt)$ in which either 0 or 1 reaction occur, thus it exists only $M + 1$ distinct configurations at time t that can lead to the state X at time $t + dt$ and as such, we can write our resulted probability function at time $t + dt$ as a function of all possible precursor states at time t (Markov chain)

$$P(X, t + dt) = P(X, t) * P(\text{no state change over } dt) + \sum_{\mu=1}^M P(X - v_\mu, t) * a_\mu(X - v_\mu, t) dt$$

- $P(X, t + dt)$: Probability to have X at $t+dt$ time
- $P(X, t)$: Probability to have X at t time

$$P(\text{no state change over } dt) = 1 - \sum_{\mu=1}^M a_\mu(x) dt$$

- $P(X - v_\mu, t)$: Probability to be in state different from X , at t time
- v_μ is a stoichiometric vector defining the result of reaction μ on state vector X

If we then note that $\lim_{dt \rightarrow 0} \frac{P(X; t + dt) - P(X; t)}{dt} = \frac{\partial P(X; t)}{\partial t}$

we can write now the chemical master equation that describes the stochastic dynamics of the system as (McQuarrie, 1967; Gillespie, 1992a):

$$(1) \quad \partial P(X;t) / dt = \sum_{\mu=1}^M a_{\mu}(X - v_{\mu})P(X - v_{\mu};t) - a_{\mu}(X)P(X;t)$$

This is the chemical master equation (CME). In principle, it completely determines the function $P(x, t | x_0, t_0)$. But the CME is really a set of nearly as many coupled ordinary differential equations as there are combinations of molecules that can exist in the system! So it is not surprising that the CME can be solved analytically for only a very few very simple systems, and numerical solutions are usually prohibitively difficult. To rise above this difficulty we can think at the reaction rate equation (RRE) of traditional deterministic chemical kinetics which can be described as:

$$dX_i(t) / dt = \sum_{j=1}^M v_{ij} a_j(X(t)) \quad (i = 1, \dots, N)$$

and proceed by probabilistic way as we will see later.

3. The Stochastic Simulation Algorithm (SSA)

Since the chemical Master equation is rarely of much use in computing the probability density function $P(x,t | x_0, t_0)$ of $X(t)$, we need another computational approach. One approach that has proven fruitful is to construct numerical realizations of $X(t)$, i.e., simulated trajectories of $X(t)$ -versus- t . This is not the same as solving the CME numerically, as that would give us the probability density function of $X(t)$ instead of samplings of that random variable. However, much the same effect can be achieved by either making a histogram or averaging the results of many realizations. The key to generating simulated trajectories of $X(t)$ is not the CME or even the function $P(x, t | x_0, t_0)$, but rather a new function, $P(\tau, j | x, t)$ (Gillespie, 1976). It is defined so that $P(\tau, j | x, t) d\tau$ is the probability, given $X(t) = x$, that the next reaction in the system will occur in the infinitesimal time interval $[t + \tau; t + \tau + d\tau]$ and will be an R_j reaction. Formally, this function is the joint probability density function of the two random variables "time to the next reaction" (τ) and "index of the next reaction" (j). To derive an analytical expression for $P(\tau, j | x, t)$, we begin by noting that if $P_0(\tau | x, t)$ is the probability, given $X(t) = x$, that no reaction of any kind occurs in the time interval $[t; t + \tau]$, then the laws of probability imply the relation [9]

$$p(\tau, j | x, t) d\tau = P_0(\tau | x, t) * a_j(x) d\tau$$

The laws of probability also imply:

$$P_0(\tau + d\tau | x, t) = P_0(\tau | x, t) * [1 - \sum_{j=1}^N a_j(x) d\tau]$$

An algebraic rearrangement of this last equation and passage to the limit $d\tau \rightarrow 0$ results in a differential equation whose solution is easily found to

$$P_0(\tau | x, t) = \exp(-a_0(x)\tau), \text{ where } a_0(x) = \sum_{j=1}^M a_j(x),$$

When we insert this result into the equation for p , we get:

$$p(\tau, j | x, t) = a_j(x) \exp(-a_0(x)\tau)$$

The Equation above is the mathematical basis for the stochastic simulation approach. It implies that the joint density function of τ and j can be written as the product of the τ -density function, $a_0(x) \exp(-a_0(x) \tau)$ and the j -density function, $a_j(x)/a_0(x)$. We can generate random samples from these two density functions by using the inversion method of Monte Carlo theory (Gillespie, 1992b): Draw two random numbers r_1 and r_2 from the uniform distribution in the unit-interval, and select τ and j according to:

$$\tau = 1/a_0(x) \ln(1/r_1)$$

j = the smallest integer satisfying

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

Thus we arrive at the following version of the stochastic simulation algorithm (SSA) (Gillespie, 1976; Gillespie, 1977).

1. Initialize the time $t = t_0$ and the system's state $x = x_0$.
2. With the system in state x at time t , evaluate all the $a_j(x)$ and their sum $a_0(x)$.
3. Generate values for τ and j .
4. Effect the next reaction by replacing $t \leftarrow t + \tau$ and $x \leftarrow x + v_j$.
5. Record $(x; t)$ as desired. Return to Step 2, or else end the simulation.

Algorithm 1: The SSA method

```

Init(Tstart, Tfinal,C,X,V,a0,T)
WHILE((T<=Tfinal)&& (a0>0))
compute(a)
a0←sum(a)
IF (a0>0)
r1←rand(1)
r2←rand(1)
tau←(-1/a0)*log(r1)
j←1
cpt←a(1)
WHILE(cpt<r2*a0)
J←j+1
Cpt←cpt+a(j)
END
T←T+tau
X←X+V(:,j)
    
```

END
END

4. The Tau leaping

The SSA and the CME are logically equivalent to each other; yet even when the CME is completely intractable, the SSA is quite straightforward to implement. The problem with the SSA is that it is often very slow. The source of this slowness can be traced to the factor $1/a_0(x)$ in the τ computing formula : $a_0(x)$ can be very large if the population of one or more reactant species is large, and that is often the case in practice [4,8].

One approximate accelerated simulation strategy is tau-leaping (Gillespie, 2001). It advances the system by a pre-selected time τ which encompasses more than one reaction event. In its simplest form, tau-leaping requires that τ be chosen small enough that the Leap Condition is satisfied: The expected state change induced by the leap must be sufficiently small that no propensity function changes its value by a significant amount. The main idea of this method is to select many reactions in the same tau time to accelerate the speed of our algorithm. We recall that the Poisson random variable $P(a, \tau)$ is by definition the number of events that will occur in time τ given that $a dt$ is the probability that an event will occur in any infinitesimal time dt , where 'a' can be any positive constant. Therefore, if $X(t) = x$, and if we choose τ small enough to satisfy the Leap Condition, so that the propensity functions stay approximately constant during the leap, then reaction R_j should fire approximately $P_j(a_j(x)*\tau)$ times in $[t; t+\tau]$: Thus, to the degree that the Leap Condition is satisfied, we can leap by a time τ simply by taking (Gillespie, 2001; Gillespie & Petzold, 2003):

$$X(t + \tau) = X + \sum_{j=1}^M v_j * p_j(a_j(x(t))\tau)$$

It will result in a faster simulation than the SSA to the degree that the total number of reactions leapt over the tau time is large compared to M . In order to use this simulation technique efficiently, we obviously need a way to estimate the largest value of τ that is compatible with the Leap Condition. One possible way of doing that (Gillespie & Petzold, 2003), is to estimate the largest value of τ for which no propensity function is likely to change its value during τ by more than $\varepsilon*a_0(x)$, where ε ($0 < \varepsilon \ll 1$) is some pre-chosen accuracy control parameter.

The main problem for the tau leaping is: how to choose the tau time that respect the two constraints , τ must be so small that the propensity functions stay approximately constant during the leap, and τ must be so large that many reactions can fire many times ?. Gillespie and petzold (in 2003) showed that the largest value of τ that satisfies these conditions can be estimated as follows [1, 5]:

$$(2) \begin{cases} f_{ij}(x) \equiv \sum_{i=1}^N \partial a_j(x) / \partial x_i v_{ij}. & j, j' = 1..M \\ \mu_j(x) \equiv \sum_{j'=1}^M f_{jj'}(x) / a_{j'} v_{jj'} \\ \sigma_j^2(x) \equiv \sum_{j'=1}^M f_{jj'}^2(x) a_{j'}(x) \end{cases}$$

The first Expression is the a jacobian matrix that compute the relative change of the propensities functions depending on the species concentrations, the second is the average of this change, and the third is the variance. Tau time must be taken according to the next formula :

$$(3) \quad \tau = \min_{j \in [1, M]} \left\{ \varepsilon a_0(x) / |\mu_j(x)|, (\varepsilon a_0(x))^2 / \sigma_j^2(x) \right\}$$

More later (in 2006) Cao and Gillespie have proposed a new tau-selection procedure as follows:

$$(4) \quad \tau' = \min_i \left\{ \max \{ \varepsilon x_i / g_i, 1 \} |\mu_i(x)|, \max \{ (\varepsilon x_i / g_i, 1)^2 / \sigma_i^2(x) \} \right\}$$

Algorithm 2: The Tau leaping Method (with formula 3)

```

Init(Tstart, Tfinal,C,X,V,a0,T,eps)
WHILE((T<=Tfinal)&& (a0>0))
compute(a)
a0 ← sum(a)
IF (a0>0)
jac ← compute-jacobian(aj(xi))
fii ← jac*v
mu ← fii*a
sigcarre ← (fii.^2)*a
errglob ← eps*a0
tau ← min(errglob/mu(x),errglob^2/sigcarre)
T ← T+tau
M ← Generate_M_Poissonv(a(j)*tau)
X ← X+V*M
END
END

```

Although the tau leaping method is so attractive in the sense that it accelerate the time running of the SSA But it is not as foolproof as the SSA. If one takes leaps that are too large, some species populations might be driven negative. moreover this method is not

adequate to stiff systems as we will see next. Several strategies have been proposed to get around this problem. Tian and Burrage, and independently Chatterjee et al, proposed approximating the unbounded Poisson random numbers K_j with bounded binomial random numbers [7,18]. But it turns out that it is usually not the unboundedness of the Poisson k_j 's that produces negative populations.

5. The implicit Tau-leaping method

Stiffness can be defined roughly as the presence of widely varying time-scales in a dynamical system, the fastest of which is stable. It poses special problems for the numerical solution of both deterministic ordinary differential equations (ODEs) and stochastic differential equations (SDEs), particularly in the context of chemical kinetics. A stiff system has (at least) two time scales. There is a long (slow) time scale for the quasi-equilibrium phase, and a short (fast) time scale for the transient phase following a perturbation. The more different these two time scales are, the "stiffer" the system is said to be. A system of ODEs is said to be stiff if its solutions show strongly damped behaviour as a function of the initial conditions (fast evolution in a few time) [1, 11]. The restriction of the explicit Euler method to time steps τ that are on the order of the short (fast) time scale makes the method very slow for stiff systems. So it is natural to ask if there are other solution methods for which the time steps are not restricted by stability, but only by the need to resolve the solution curve. It is now widely recognized that a general way of doing this is provided by "implicit" methods (Ascher & Petzold, 1998), the simplest of which is the implicit Euler method with the next form:

$$X_{n+1} = X_n + \tau f(T_{n+1}, X_{n+1})$$

In contrast to the explicit Euler formula:

$$X_{n+1} = X_n + \tau f(T_n, X_n)$$

If we consider $g(X_{n+1}) = X_{n+1} - X_n - \tau f(T_{n+1}, X_{n+1})$

Usually, the most efficient way to numerically solve $g(X_{n+1}) = 0$ is by the Newton iteration:

$$X_{n+1} - X_n = -g(X_n) / g'(X_n)$$

Algorithm 3: The Newton algorithm for NLAE

Init(max_iterations , nlae_tolerance ,h)

Initialise $X_0 = X_n$

FOR k = 1 **TO** max_iterations

$$x^{(k)} = x^{(k-1)} - g \left(1 - h \frac{\theta_f}{\theta_x} \right)^{-1}$$

$g \leftarrow X_{k-1} - X_n - h * f$
 if $|g| < \text{nlae_tolerance}$ exit
 calculate
END
 $X_{n+1} \leftarrow X_{k-1}$

An implicit scheme which takes large steps (on the time scale of the slow mode) will do just fine if the fluctuations of the slow manifold are negligibly small. To get the implicit formula of the chemical system (Rathinam *et al.*, 2003) we can modify the explicit method as follows:

$$X(t + \tau) = X + \sum_{j=1}^M v_j * p_j(a_j(x(t))\tau) \quad (* \text{ explicit scheme } *)$$

↓

$$X(t + \tau) = X + \sum_{j=1}^M v_j * p_j(a_j(x(t + \tau))\tau) \quad (* \text{ implicit scheme } *)$$

The right part of the last formula is simply the Poisson function expressed at time $t + \tau$, the problem is that this function $p_j(a_j(x(t + \tau)))$ is unknown at time t , then we must give it an approximate value. The idea is to use the equation

$$p_j(a_j(x(t + \tau))) = a_j(x(t + \tau))\tau + p_j(a_j(x(t + \tau))) - a_j(x(t + \tau))\tau$$

the expression in the right is the sum of two terms the first is the propensity function at time $t + \tau$ and the next is a zero mean expression, let change this formula by :

$$p_j(a_j(x(t + \tau))) = a_j(x(t + \tau))\tau + (p_j(a_j(x(t))) - a_j(x(t))\tau) \quad , \text{ thus we can}$$

write the implicit scheme as:

$$(5) \quad X(t + \tau) \approx X(t) + \sum_{j=1}^M v_j a_j(X(t + \tau))\tau + \sum_{j=1}^M v_j [P_j(a_j(X(t)), \tau - a_j(X(t))\tau)]$$

Or (Y. Cao and L. Petzold 2005) with trapezoidal form

$$(6) \quad X(t + \tau) = X + \sum_{j=1}^M v_j (p(a_j(x(t))) - \tau/2 * a_j(x) + \tau/2 * a_j(x(t + \tau)))$$

Algorithm 4 : The implicit Tau-leaping method (formula 5)

Init(Tstart, Tfinal, C, X, V, a0, T, eps, N)
WHILE((T <= tfinal) && (a0 > 0))
 compute(a)


```

a0 ← sum(a)
IF (a0 > 0)
jac ← compute-jacobian(aj(xi))
fii ← jac*v
mu ← fii*a
sigcarre ← (fii.^2)*a
errglob ← eps*a0
tau ← min(errglob/mu(x), errglob^2/sigcarre)
T ← T+tau
M ← Generate_M_Poissonv(a(j)*tau)
FOR j=1:N DO
cst1 ← (M(i)-a(i)*tau)*v(:,i)
END
v0 ← X;
compute(fp,f0)
I ← Newton( max_iterations , nlae_tolerance ,tau,v0,fp,f0)
X ← i;
END
END

```

6. The Chemical Langevin Equation and the Reaction Rate Equation

If the molecular populations of the considered reacting species are larger, we can go further in the approximation of our equations as follows:

The term $a_j(x)\tau$ becomes so large that $a_j(x)\tau \gg 1$, the physical significance of this last condition is that each reaction channel is expected to fire many more times than once in the next τ time. This approximation stems from the purely mathematical fact that the Poisson random variable $P(a, \tau)$, which has mean and variance $a\tau$, can be well approximated when $a\tau \gg 1$ by a normal random variable with the same mean and variance. Denoting the normal random variable with mean m and variance σ^2 by $N(m, \sigma^2)$, and as we know that $N(m, \sigma^2) = m + \sigma N(0,1)$, we can deduce that (Gillespie, 2000; Gillespie, 2002) [10]:

$$(7) \quad P_j(a_j(x), \tau) \approx N_j(a_j(x)\tau, a_j(x)\tau) = a_j(x)\tau + (a_j(x)\tau)^{1/2} N_j(0,1)$$

$$\text{Thus: } X(t+\tau) \approx X(t) + \sum_{j=1}^M v_j a_j(x)\tau + \sum_{j=1}^M v_j \sqrt{a_j(x)} N_j(0,1) \sqrt{\tau}$$

This equation is called the Langevin leaping formula. It evidently expresses the state increment $X(t+\tau) - X(t)$ as the sum of two terms: a deterministic drift term proportional to τ , and a fluctuating diffusion term proportional to $\sqrt{\tau}$. It is important to keep in mind that this equation is an approximation, which is valid only to the extent that τ is small enough that no propensity function changes its value significantly during τ , yet large enough that every reaction fires many more times than once during τ . The next formula is underscored by the fact that $X(t)$ therein is now a continuous (real-

valued) random variable instead of a discrete (integer-valued) random variable; we lost discreteness when we replaced the integer-valued Poisson random variable with a real-valued normal random variable. The Langevin leaping formula gives faster simulations than the tau-leaping formula not only because the condition $a_j(x) \tau \gg 1$, implies that very many reactions get leapt over at each step, but also because the normal random numbers that are required can be generated much more easily than the Poisson random numbers (Press et al., 1986). If we subtract x from both sides and then divide through by dt , the result can be shown to be the following (approximate) stochastic differential equation, which is called the chemical Langevin equation (CLE) (Gillespie, 2000; Gillespie, 2002):

$$(8) \quad dX(t)/dt \approx \sum_{j=1}^M v_j a_j(X(t)) + \sum_{j=1}^M v_j \sqrt{a_j(X(t))} \Gamma_j(t)$$

The $\Gamma_j(t)$ here are statistically independent “Gaussian white noise” processes. Molecular systems become “macroscopic” in what physicists and chemists call the thermodynamic limit. This limit is formally defined as follows: The system volume and the species populations X_i all approach ∞ , but in such a way that the species concentrations X_i all remain constant. To discern the implications of those formulas in the thermodynamic limit, we evidently need to know the behaviour of the propensity functions in that limit. It turns out that all propensity functions grow linearly with the system size as the thermodynamic limit is approached. It follows that, as the thermodynamic limit is

approached, the deterministic drift term ($\sum_{j=1}^M v_j a_j$) grows like the size of the system,

while the fluctuating diffusion term grows like the square root of the size of the system, and likewise for the CLE. This establishes the well known rule-of-thumb in chemical kinetics that relative fluctuation effects in chemical systems typically scale as the inverse square root of the size of the system. In the full thermodynamic limit, the size of the second term on the right side will usually be negligibly small compared to the size of the drift term, in which case the CLE can be reduced to the RRE (drift term). Thus we have derived the RRE as a series of limiting approximations to the stochastic theory that underlies the CME and the SSA

$$(9) \quad \text{RRE:} \quad dX_i(t)/dt = \sum_{j=1}^M v_{ij} a_j(X(t)) \quad (i = 1..N)$$

Now, if we return to the CLE with its numerical form (7), readers familiar with numerical methods for stochastic differential equations (SDEs) will recognize it as an Euler-Maruyama discretization of the continuous time problem

$$dX(t) = \sum_{j=1}^M v_j * a_j(x(t)) dt + \sum_{j=1}^M v_j \sqrt{a_j(x(t))} dw_j(t)$$

where the $W_j(t)$ are independent scalar Brownian motions, the next code presents how we can simulate the Brownian motion.

Algorithm 5 : The Brownian motion

```

Init(Tfinal,step,w,dw)
N←Tfinal/step
FOR j = 1:N
alea←randn;
dW(j) ← sqrt(step)*alea
W(j) ← W(j-1) + dW(j);
END

```

The next code presents code presents how we can implement the CLE with an Euler-Maruyama discretization

Algorithm 6 : The Chemical langevin equation

```

Init(Tstart, Tfinal,C,X,SM,a0,T,eps,N,tau,M,Som)
WHILE ((T<=Tfinal)&& (a0>0))
compute(a)
a0←sum(a);
FOR j=1:M
Ad(j)← (tau*a(j) + sqrt(abs(tau*a(j)))*randn)*SM(:,j)
Som←som+Ad(j)
END
X ← X + som
T←T+tau
END

```

7. Application and Results

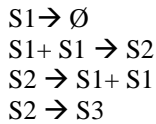
In this section we will see how we can decide or choose the best method for a given system? . As we have seen before, there are two criteria for the choice: the population size and the stiffness of the system. The population size considered can be large or small (or 'mixt' where some species are of large size and others are not) , the system can be stiff or non stiff . Let the variable 'Nat' be a set of two values 'S' and 'NS', Nat={'S', 'NS' }, and the variable 'pop' is a set of three values 'Allbig', 'Allsmall' and 'mixt', pop={'Allbig', 'Allsmall', 'mixt'}, the next table presents the choice of the best method depending on these variables (not always true !).

POP/NAT	S	NS
Allbig	RRE (or Implicit)	RRE (or CLE)
Allsmall	SSA	SSA

Mixt	Implicit	Explicit
------	----------	----------

We can add that even the explicit or the implicit method can use in its internal code an appeal to SSA techniques or adaptative tau selection (ex. interlaced implicit→ down-shifting) for the tau choice, in the limit where the species size becomes so small that the tau leaping formula can't give a good choice. In general there is no way to predict the nature of the system !, because if the system presents dynamical behaviour such that in one time period it is stiff but in another time period it is non stiff, we can not resolve the system efficiency with one method, we must necessarily use an adaptive simulation strategy to automatically switch between explicit and implicit tau leaping methods and even mix the three methods in some stiff systems.

To see the effect of stiffness in simulation we present the Decaying-Dimerizing Reactions. The system is presented as follows [11]:



With parameters vector: $C=(1, 10, 1000, 0.1)$, initial conditions vector $X0=(400, 798, 0)$ and final time=0.2 sec , the curves obtained are the next :

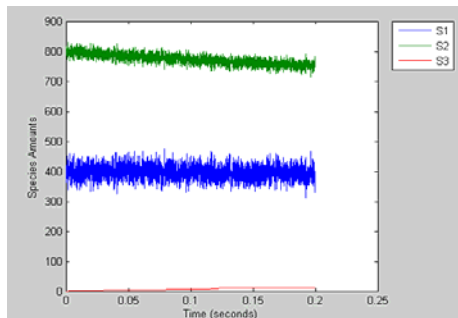


Fig1: SSA

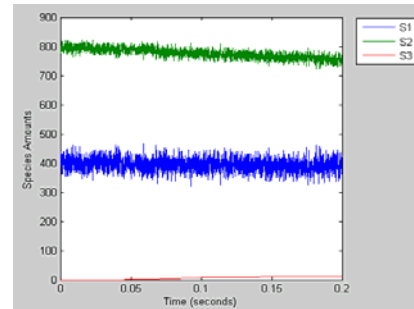


Fig2: Explicit tau

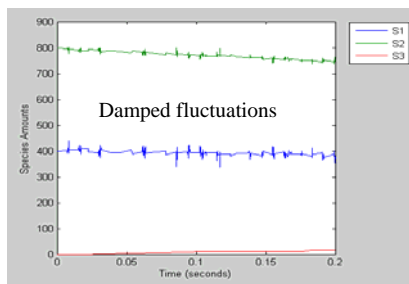
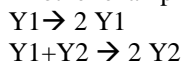


Fig3: Implicit tau

Another example is the Lotka volterra or predator-prey model, presented as follows :



$Y_2 \rightarrow \emptyset$

With parameters vector: $C=(0.1, 0.002, 5e-4)$, initial conditions vector $X_0=(100,50)$ and final time=10 sec, the curves obtained are the next:

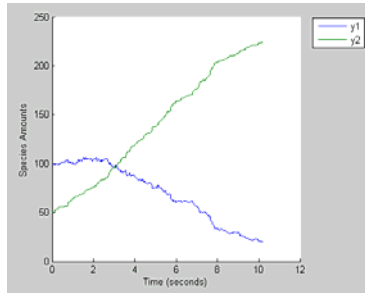


Fig4: SSA

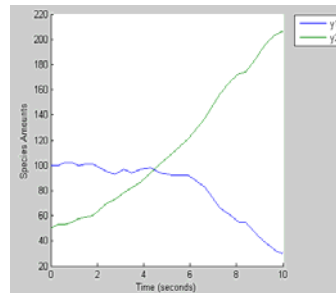


Fig5: Explicit tau

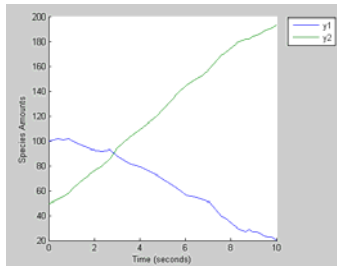


Fig6: Implicit tau

The time running of each method was presented in the next table (Computer: Pentium 4 with 2.1Ghz as frequency, System: windows XP, Soft: Mathworks Simbiology)

Method/ system	Decaying-Dimerizing	predator-prey
SSA	93,49	0.16
Explicit	0.654	0.013
Implicit	3.061	0.04

Through many applications that we have made with some systems, we can conclude the next results:

- The SSA is until now the more straightforward algorithm and its stability is guaranteed, nevertheless its slowness in large population systems case make it useless
- The explicit and implicit tau leaping efficiency depends on the efficiency of the tau selection formula
- An efficient generic algorithm is until now a topic of future research, such algorithm that can rise above speed and accuracy

8. Conclusion

Through this paper we have seen some numerical methods (based on Gillespie and al. group research) to solve stochastic differential equations in biochemical systems from the chemical master equation to the RRE, we have proved that there is a logical relation between both, and that each chemical system have its own and best method to be simulated. It remains until now, that there is no way to find a generic algorithm that can solve all the kind of systems (see slow scales algorithms and the new algorithm Weighted Stochastic Simulation Algorithm WSSA [16,18]), especially the stiff systems (most cases of biochemical systems), which remains the challenge for the future research [1]. For further information refer to our web site : <http://automatictn.wikispaces.com>, biosystèmes section)

References

- [1] Cao, Y., Daniel T. Gillespie and Linda Petzold. The Adaptive Explicit-Implicit Tau-Leaping Method with Automatic Tau Selection. Draft copy
- [2] Cao, Y., H. Li and L. Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *J. Chem. Phys.*, 121:4059-67, 2004.
- [3] Cao, Y., L. Petozld, M. Rathinam and D. Gillespie. The numerical stability of leaping methods for stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 121:12169-78, 2004.
- [4] Rathinam, M., L. Petzold, Y. Cao, D. Gillespie. Consistency and stability of tau leaping schemes for chemical reaction systems. *SIAM Multiscale Modeling*, 4:867-895, 2005.
- [5] Cao, Y., D. Gillespie and L. Petzold. Efficient Stepsize Selection for the Tau-Leaping Method. *J. Chem. Phys.*, 124:044109, 2006.
- [6] Cao, Y., D. Gillespie and L. Petzold. The slow-scale stochastic simulation algorithm. *J. Chem. Phys.*, 122:014116, 2005.
- [7] Tian, T., and K. Burrage. Binomial leap methods for simulating stochastic chemical kinetics. *J. Chem. Phys.*, 121:10356-64, 2004.
- [8] Cao, Y., D. Gillespie and L. Petzold. Avoiding negative populations in explicit tau leaping. *J. Chem. Phys.*, 123:054104, 2005.
- [9] Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* 81(25): 2340-2361.
- [10] Gillespie, D. T. (2000). The chemical Langevin equation. *J. Chem. Phys.* 113,297-306.
- [11] Rathinam, M., Cao, Y., Petzold, L. & Gillespie, D. (2003). Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method. *J. Chem. Phys.* 119, 12784-12794.
- [12] McQuarrie, D. A. (1967). Stochastic approach to chemical kinetics. *J. Appl. Prob.* 4: 413-478.
- [13] McAdams, H. H., Arkin, A., 1997. Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. U. S. A.* 94, 814-819.
- [14] Gibson, M. A. & Bruck, J. (2000). Exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem.* 105, 1876-1889.

- [15] Cai, X., X. Wang Stochastic Modeling and Simulation of Gene Networks, A review of the state-of-the-art, research on stochastic simulations , IEEE signal processing magazine january 2007
- [16] El Samad, H., M. Khammash, Stochastic modelling of gene regulatory networks, international journal of robust and nonlinear control, int. j. robust nonlinear control 2005; 15:691–711
- [17] Chatterjee, A., D. G. Vlachos and M. A. Katsoulakis. Binomial distribution based tau-leap accelerated stochastic simulation. J. Chem. Phys., 122:024112, 2005.
- [18] Dan T. Gillespie, Min Roh and Linda R. Petzold, Refining the Weighted Stochastic Simulation Algorithm, J. Chem. Phys. 129,165101 (2008)