

# INTRODUCING THE STOCHASTIC SIMULATION OF CHEMICAL REACTIONS

## *Using the Gillespie Algorithm and MATLAB*

JOAQUÍN MARTÍNEZ-URREAGA, JOSÉ MIRA, CAMINO GONZÁLEZ-FERNÁNDEZ  
*Universidad Politécnica de Madrid • José Gutiérrez Abascal-2 • 28006 Madrid, Spain*

There are two main approaches to numerically model and simulate the time evolution of chemical reacting systems. In the deterministic approach, the set of differential equations describing the time evolution of the concentrations is solved using either analytical or numerical methods such as Euler or Runge-Kutta. It is assumed that the complete time evolution of the reacting system is contained in the solution of the set of equations, *i.e.*, given a set of initial conditions, only one trajectory is possible. In this paper, a trajectory is a concentration-time curve. It corresponds to a reacting species in a given experiment and describes the time evolution of the reacting system in such an experiment.

In the stochastic approach, each individual reaction is considered a random event that can take place with a certain probability. Thus the time evolution of the concentrations depends on a series of consecutive probabilistic events. Given a set of initial conditions, there are many possible trajectories, each with its own probability and with the sum of probabilities adding up to one. These trajectories may be drawn by using the probabilistic rate law.

The increasing interest of stochastic methods has been pointed out by Schieber in this journal.<sup>[1]</sup> More recently, Scappin and Canu<sup>[2]</sup> have reviewed the use of stochastic models for simulating the dynamics of complex chemical systems and have shown that these models allow for easy identification of the main reaction paths in reacting systems involving hundreds of elementary steps.

In addition, several other authors<sup>[3,4]</sup> have pointed out that deterministic models cannot accurately simulate the dynamics of systems in which the time evolution depends on the behavior of a very small number of molecules. Interesting examples of such systems are individual cells in living or-

ganisms. McAdams and Arkin<sup>[4]</sup> have pointed out that

*... Even in clonal cell populations and under the most uniform experimental conditions, considerable variation is observed in the rates of development, morphology, and the concentration of each molecular species in each cell. These fluctuations ... play a fundamental role in the evolution of the living systems. ...*

These fluctuations may be predicted and explained by the stochastic models but not by the deterministic ones.

Traditionally, the deterministic methods are by far the most commonly used in modeling the time evolution of chemical



**Joaquín Martínez Urreaga** is Associate Professor in the Department of Industrial Chemical Engineering at the Universidad Politécnica de Madrid, Spain. He received his MSc (1982) and PhD (1988) from the Universidad de Zaragoza (Spain). His current fields of interest include teaching chemistry and research on interfaces and degradation in materials.

**José Mira** obtained a Masters Degree in Engineering in 1986 and in 1995 a PhD in Applied Statistics, both from the Universidad Politécnica de Madrid. He is presently an Associate Professor at that University.



**Camino González-Fernández** obtained her degree in Nuclear Engineering in 1987 and her PhD in 1993, both from the Universidad Politécnica de Madrid. She is Associate Professor in the Department of Statistics at the Universidad Politécnica de Madrid. Her current fields of interest include teaching and research on applied statistics.

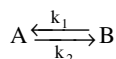
reacting systems. The above remarks, however, may justify the usefulness of introducing undergraduate students to the use of stochastic methods to model chemical reactions.

In order to achieve better understanding of the fundamentals of the stochastic simulation of chemical reactions, it is interesting that students develop their own software tools to carry out the simulation. We present here the basics of the stochastic simulation of a well-known, simple process—the AB equilibrium process—compared to the deterministic simulation of the same process. In the stochastic simulation, we follow the numerical method developed by Gillespie.<sup>[5]</sup> Both simulations are carried out with MATLAB, a numerical computation package of increasing use in chemical engineering education.

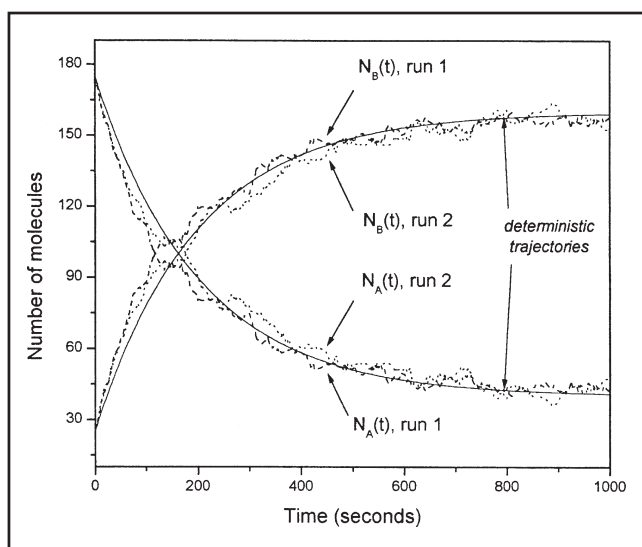
This example may prove useful for studying how the predictions of the stochastic model relate to the deterministic predictions (and to real-life experiences). To extend the stochastic simulation to other chemical processes, the students can either develop the corresponding MATLAB software, taking as a starting point the MATLAB software supplied in this paper, or they can use commercial simulation software. Two of these commercial programs (freely downloadable from the Internet) are noted in this paper.

## SIMULATION OF THE AB EQUILIBRIUM PROCESS

We have chosen as our example the process



because



**Figure 1.** Deterministic (solid) and stochastic (dash, dot) trajectories for the AB equilibrium process. The stochastic trajectories were obtained in two consecutive runs. Initial conditions were

$$N_A(0) = 175; N_B(0) = 25; k_1 = 4 \times 10^{-3} \text{ s}^{-1}; k_2 = 1 \times 10^{-3} \text{ s}^{-1}.$$

- It describes various real processes, such as the hydrolysis of lactone to  $\gamma$ -hydroxybutyric acid in strong hydrochloric acid.<sup>[6]</sup>
- It has been previously treated by different authors.<sup>[6,7]</sup>
- It is simple enough to be modeled by the undergraduate students. In order to make the modeling easier, we will assume an isothermal process at constant volume.

### Deterministic Simulation

The differential rate laws can be written as

$$\frac{dN_A(t)}{dt} = k_2 N_B(t) - k_1 N_A(t) \quad (1)$$

$$\frac{dN_B(t)}{dt} = k_1 N_A(t) - k_2 N_B(t) \quad (2)$$

where  $N_A(t)$  and  $N_B(t)$  are the numbers of molecules after a given reaction time  $t$ , and  $k_1$  and  $k_2$  are the direct and reverse reaction rate constants.

Once the initial values of  $N_A(t)$  and  $N_B(t)$ , ( $N_A(0)$  and  $N_B(0)$ ) are specified, it is assumed that the solution of these differential equations describes the complete time evolution of the reacting system. Figure 1 shows an example of the time evolution of  $N_A(t)$  and  $N_B(t)$  predicted by this approach for  $N_A(0) = 175$ ,  $N_B(0) = 25$ ,  $k_1 = 0.004 \text{ s}^{-1}$  and  $k_2 = 0.001 \text{ s}^{-1}$ . These trajectories were calculated using the MATLAB programs listed in Appendices 1 and 2. In the deterministic approach, given a set of initial conditions, all runs will give the same trajectory.

The equilibrium values of  $N_A$  and  $N_B$  ( $N_{A,eq}$  and  $N_{B,eq}$ ) may be easily calculated by the students. At equilibrium,

$$k_1 N_{A,eq} = k_2 N_{B,eq} \quad (3)$$

Using the mass balance, we obtain

$$N_{A,eq} = [N_A(0) + N_B(0)] / [1 + (k_1 / k_2)] \quad (4)$$

In our case,  $N_{A,eq} = 200/5 = 40$  and  $N_{B,eq} = 160$ .

### Stochastic Simulation

As mentioned above, the stochastic simulation of a chemical reacting system is rather different from the deterministic one. Each reaction is a random event that can take place with a given probability, which is a function of the reaction rate constants and the number of molecules. There are many possible trajectories, which we can draw by using the probabilistic rate law. Thus the development of the stochastic simulation requires a deep foundation of the probability theory. A complete description of the stochastic treatment of the AB equilibrium process can be found in a text by Steinfeld, Francisco, and Hase.<sup>[7]</sup>

In this paper we describe the development of a MATLAB application for the generation of stochastic trajectories by using the Gillespie algorithm.<sup>[5]</sup> Gillespie developed an el-

egant and efficient algorithm that uses Monte Carlo techniques to carry out the numerical stochastic simulation of any given chemical reacting system and demonstrated that this simulation gives an accurate description of the time evolution of the system.

In the Gillespie algorithm, the probability of each reaction is obtained by multiplying the reaction rate constant by the number of combinations of molecules that can lead to the reaction. For the AB process, the numbers of combinations are

$$\binom{N_A}{1} = N_A \quad \text{and} \quad \binom{N_B}{1} = N_B$$

respectively. In order to develop this algorithm, we first must define

► dt

a time interval so small that either only one reaction or no reaction at all can occur in the interval  $(t, t+dt)$  (*i.e.*,  $dN_A(t) = N_A(t+dt) - N_A(t)$  can only take the values -1, 0, and 1).

►  $k_1, k_2$

$k_1$  is defined so that  $k_1 dt$  is the probability that any A molecule will react to give a B molecule in  $(t, t+dt)$  and similarly for  $k_2$ .

►  $W_+[N_A(t)], W_-[N_A(t)]$

two positive functions such that  $W_+[N_A(t)]dt$  and  $W_-[N_A(t)]dt$  are the probabilities that, given that the number of molecules of A at time  $t$  is  $N_A(t)$ , at time  $t+dt$  the number of molecules  $N_A(t+dt)$  is equal to  $N_A(t)+1$  and  $N_A(t)-1$ , respectively.  $W_+[N_A(t)]dt$  and  $W_-[N_A(t)]dt$  are conditional probabilities (conditional on the value of  $N_A(t)$ ) and play an essential role in defining the stochastic model. In our example, taking into account the above definitions of  $k_1$  and  $k_2$ , these functions are

$$W_+[N_A(t)] = k_2 N_B(t) \quad (5)$$

$$W_-[N_A(t)] = k_1 N_A(t) \quad (6)$$

After defining the initial conditions,  $N_i(0)$  and the constant  $k_i$ , the Gillespie algorithm generates time steps of variable length, depending on the probabilities of the reactions and on the random nature of the process (a random number is used to generate the time steps). Subsequently, a second random number is generated to determine which of the two possible reactions occurs, taking into account the reaction probabilities. Next, the  $N_i(t)$  values are updated according to the stoichiometry and the process is repeated.

In order to determine the above reaction probabilities and time steps, we need to define

►  $a[N_A(t)]$

a non-negative function such that  $a[N_A(t)]dt$  is the probability that the number of molecules of A, which

takes the value  $N_A(t)$  at time  $t$ , suffers a unitary increment (positive or negative) in the differential interval  $(t, t+dt)$ . It verifies

$$a[N_A(t)] = W_+[N_A(t)] + W_-[N_A(t)] \quad (7)$$

►  $w_{\pm}[N_X(t)]$

probability that the process, which has suffered an increment of one on either sense, does it positively or negatively (+1 for  $w_+$  and -1 for  $w_-$ ). These are also conditional probabilities (conditional on the fact that a reaction has taken place). Evidently

$$w_+[N_X(t)] + w_-[N_X(t)] = 1 \quad (8)$$

$$w_{\pm}[N_X(t)] = \frac{W_{\pm}[N_X(t)]}{a[N_X(t)]} \quad (9)$$

► u

random time step. It is the random variable "time to the next reaction given that the number of molecules of A at time  $t$  is  $N_A(t)$ ."

►  $p_0[N_A(t), u]$

complementary distribution function for u. Probability that the number of molecules of X, which takes the value  $N_A(t)$  in time  $t$ , does not suffer any changes in  $(t, t+u)$ . It can be shown<sup>[5]</sup> that

$$p_0[N_A(t), u] = \exp\{-a[N_A(t)u]\} \quad (10)$$

The distribution of u is an exponential with mean  $1/a[N_A(t)]$ . Using the Monte Carlo method, we can generate a suitable value of the random number u using<sup>[5,7]</sup>

$$u = \left\{ 1/a[N_A(t)] \right\} \log(1/r) \quad (11)$$

where r is a random number of the uniform distribution between 0 and 1. Note that the random time step decreases in average as the probability that any chemical change takes place in the time interval increases.

Thus the algorithm of generation of stochastic trajectories can be written

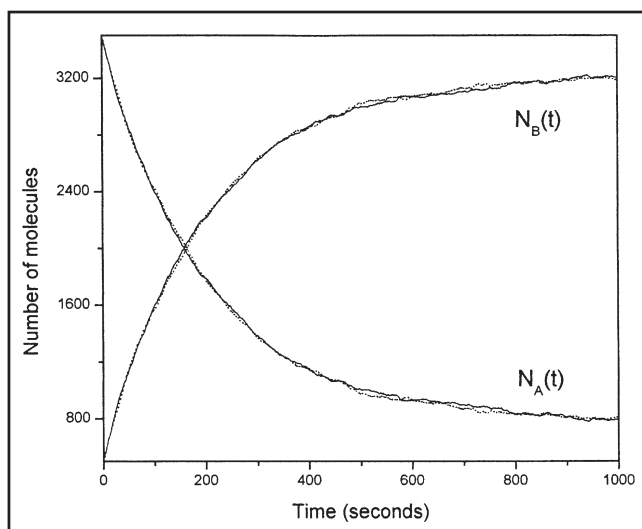
1. Initialize  $t=0$ . Introduce the initial values  $N_A(0)$  and  $N_B(0)$ ,  $k_1$  and  $k_2$ . Define the total number of reactions Z.
2. Generate a value of u: first a random number r is generated from the uniform distribution in (0,1) and then  $u = \{1/a[N_A(t)]\} \log(1/r)$ .
3. Generate a second number r' from the uniform distribution in (0,1). This random number determines which reaction will occur, based on conditional probabilities. If  $w_-[N_A(t)] > r'$ , then take  $v = -1$  and if not,  $v = 1$ .
4. Update the process:  $t = t + u$ ;  $N_A(t+u) = N_A(t) + v$
5. If the total number of reactions  $i < Z$ , go back to step

2. If  $i \geq Z$ , then stop.

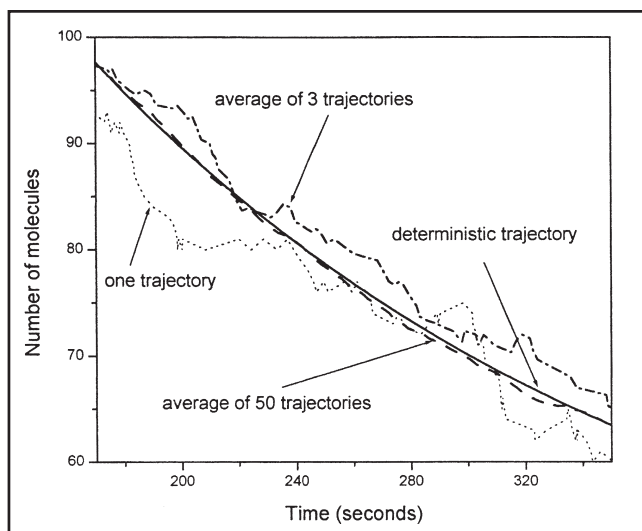
An example of the MATLAB program (stochasticab.m), which implements the above algorithm for the simulation of the AB equilibrium process, is listed in Appendix 3. We emphasize the practical importance of using a random number generator as good as possible to achieve an accurate simulation, including those processes having a wide range of rate constant magnitudes. In this work we have used the random number generation algorithm provided by MATLAB, which is applied extensively in statistical research.

## DISCUSSION

Figure 1 shows the trajectories calculated by the above stochastic algorithm in two consecutive runs, as well as the deter-



**Figure 2.** Stochastic trajectories (two runs) with  $N_A(0) = 3500$  and  $N_B(0) = 500$ ;  $k_1$  and  $k_2$  are the same as in Figure 1.



**Figure 3.** Deterministic and stochastic (single and averages) trajectories  $[N_A(t)]$  for the AB process. The initial conditions are the same as in Figure 1.

ministic trajectories, using the same initial values in all cases. These trajectories can be used to carry out a comparative study on the two simulation approaches—stochastic and deterministic.

We can see that there are clear differences. The stochastic trajectories show important fluctuations. Moreover, two consecutive runs predict different trajectories, although the initial conditions are the same ones, *i.e.*, we cannot assure the value of  $N_A(t)$  at each time point. On the other hand, the deterministic simulation will always predict the same trajectory, given a set of initial conditions, and it does not present fluctuations in the time evolution. When these results are analyzed, some interesting questions arise. For instance, do these fluctuations (also called stochastic noise) have some physical meaning? Is it possible that we cannot predict with certainty the value of  $N_A(t)$  at each time point in a real-life experience?

The students should know that the fluctuations are a real consequence of the probabilistic nature of each chemical reaction. Some interesting real experiments showing stochastic effects have been presented by de Levie.<sup>[3]</sup> We can't see these fluctuations in most real-life experiments, however. Which are the factors that determine the importance of the fluctuations? In order to develop an answer, the students can repeat the simulations and vary the input conditions.

Figure 2 shows two new trajectories obtained through the stochastic algorithm, but using a much larger number of initial molecules than in Figure 1, namely  $N_A(0) = 3500$  and  $N_B(0) = 500$ . It can be seen that the fluctuations are only important when the process starts with a small number of molecules. As the initial number of molecules increases, the fluctuations decrease and the stochastic trajectory approaches the deterministic one. This result was explained by Gillespie<sup>[5]</sup> showing that the relative fluctuations in  $N_A(t)$  around the mean value of  $N_A(t)$  ( $\langle N_A(t) \rangle$ ), which can be obtained from repeated runs) are approximately of the order of  $(\langle N_A(t) \rangle)^{1/2} / \langle N_A(t) \rangle$ .

This is an important result, as it explains how the fluctuations are not important in most real-life experiments. When we work with  $10^{20}$  molecules, the relative fluctuations ( $\sim 10^{-10}$ ), *i.e.*, the uncertainties in the value of  $N_A$ , are absolutely negligible. In that case, a deterministic model allows an adequate representation of most processes (with some exceptions—see below).

In many microscopic systems, however, the intrinsic fluctuations are important. For instance, some biochemical reactions taking place in individual cells of living organisms depend on ten or less molecules. In that case, the fluctuations can play a fundamental role in the behavior of the system. Deterministic models cannot adequately describe such behavior.

Figure 3 can be used to explain the relationship between deterministic and stochastic trajectories. It can be seen how



the average of stochastic trajectories approaches the deterministic trajectory as the number of averaged trajectories increases (*i.e.*, the deterministic trajectories can be explained as the average behavior of the whole set of possible stochastic trajectories).

Finally, students can see that both the stochastic and deterministic approaches predict the same final state for the AB equilibrium process, but this is not evident in all cases, even at a macroscopic scale. There are processes with more than one possible stable final state. These processes will evolve toward one of the possible final states, each evolution having a probability that depends on the initial conditions. This is an infrequent behavior in nature, which can be explained and predicted using a stochastic approach.<sup>[5]</sup>

## EXTENDING THE SIMULATION TO OTHER REACTIONS: COMMERCIAL PROGRAMS

Once the students understand the fundamentals of the stochastic simulation of chemical reactions, the application to the simulation of other chemical processes can be carried out using commercial simulation programs or even by developing new MATLAB software similar to the programs presented here. The key point of this development is obtaining the functions  $W_+[N_A(t)]$  and  $W_-[N_A(t)]$ , which represent the sources and wells of A molecules in each chemical process.

The commercial programs save user time since he/she only has to provide a suitable mechanism and the set of initial data. Moreover, most of these programs include a built-in collection of developed (and interesting) examples. Of the several commercial packages available, we will focus on two high-quality programs that can be freely downloaded from the Internet.

**Chemical Kinetics Simulator** (CKS 1.01 currently available in versions for OS 2.2.x and higher, Apple Macintosh and Power Macintosh, and Microsoft Windows, 3.1/Windows, 95/Windows NT) was developed at IBM's Almaden Research Center in San Jose, California, and can be downloaded from its homepage.<sup>[8]</sup> It is an easy-to-use program (with an excellent tutorial) that allows the accurate stochastic simulation of chemical reactions, including those in which changes in volume, pressure, or temperature are expected. For instance, it may work with explosions. The simulations included in the package may also be useful as learning tools. Some examples of these simulations are the copolymerization of two monomers, a catalytic process in a batch reactor, and the simulation of gas phase reactions in a CVD reactor.

**StochSim** is a stochastic simulator with a marked focus on biochemical processes. In this case the examples included simulate, for instance the Michaelis-Menten enzyme kinetics and the Lotka Volterra process. It was written by Carl Firth at the University of Cambridge. The currently available version 1.4<sup>[9]</sup> consists of a platform-independent core simula-

tion engine encapsulating the stochastic algorithm and a separate graphical user interface. The stochastic algorithm used in this program is rather different from the Gillespie algorithm; here each molecule is represented as a separate software object. This is advantageous for simulating processes in which the physical and chemical properties of the reacting molecules change in the course of the reaction.<sup>[10]</sup>

## CONCLUSIONS

Stochastic models are playing an increasing role in the simulation of chemical and biochemical processes, as they allow adequate prediction of the so-called stochastic effects, including the intrinsic fluctuations of the system. These fluctuations can play a fundamental role in the evolution of the living systems and, in general, in the behavior of many microscopic systems.

In this paper the Gillespie algorithm is proposed as a suitable tool for introducing undergraduate students to the basics of the stochastic simulation of chemical reactions. Application of the Gillespie algorithm to a simple and well-known reaction, the AB equilibrium process, is presented. Using this algorithm, the students can develop their own MATLAB programs to carry out the stochastic simulations of the AB process and then use the results to analyze the main differences between the stochastic and the deterministic modeling of a chemical reaction.

Two examples of MATLAB programs are presented. Students can also easily adapt these two programs to other reaction schemes. Finally, two commercial simulation programs (freely downloadable from the Internet) are proposed as additional tools for extending the stochastic simulation to other chemical processes.

## ACKNOWLEDGMENTS

The authors are grateful to the referees for their helpful comments and suggestions.

## REFERENCES

1. Schieber, J.D., "Applied Stochastic for Engineering," *Chem. Eng. Ed.*, **27**(4), 170 (1993)
2. Scappin, M., and P. Canu, "Analysis of Reaction Mechanisms through Stochastic Simulation," *Chem. Eng. Sci.*, **56**, 5157 (2001)
3. De Levie, R., "Stochastics: The Basis of Chemical Dynamics," *J. Chem. Ed.*, **77**(6), 771 (2000)
4. McAdams, H.H., and A. Arkin, "It's a Noisy Business! Genetic Regulation at the Nanomolar Scale," *Trends in Genetics*, **15**(2), 65 (1999)
5. Gillespie, D., *Markov Processes: An Introduction for Physical Scientists*, Academic Press, Inc., New York, NY (1984)
6. Fahidy, T.Z., "Solving Chemical Kinetics Problems by the Markov Chain Approach," *Chem. Eng. Ed.*, **27**(1), 42 (1993)
7. Steinfeld, J.I., J.S. Francisco, and W.L. Hase, *Chemical Kinetics and Dynamics*, Prentice Hall, Englewood Cliffs, NJ (1989)
8. CKS home page: <<http://www.almaden.ibm.com/st/msim/>> (Available Sept. 2002)
9. Download StochSim: <<ftp://ftp.cds.caltech.edu/pub/dbray/>> (Available Sept. 2002)

10. StochSim homepage: <<http://www.zoo.cam.ac.uk/comp-cell/StochSim.html>> (Available Sept. 2002)

---

## APPENDIX 1

### Program *deterab.m* for deterministic simulation

```
% This program performs the deterministic simulation of the
AB chemical % process
% k1 y k2 are the direct and reverse reaction rate constants
% na0 and nb0 are the initial numbers of molecules
% tfinal is the total reaction time
k1=4.e-3;
k2=1.e-03;
h=k1;
l=k2;
na0=175;
nb0=25;
tfinal=1000;
timestep=1;
time=0;timestep:tfinal;
[T,Y]=ode45('fisomer',time,[na0],[],na0,nb0,k1,k2);
matrix=[T Y na0+nb0-Y];
save figure 1.dat matrix/ascii;%save results
plot(T(1:500),Y(1:500),T(1:500),na0+nb0-Y(1:500));
% plot curves
```

---

## APPENDIX 2

### Auxiliary function *fisomer* for deterministic simulation

```
function F = fisomer(time,Y,flag,na0,nb0,k1,k2)
h=k1;
l=k2;
F=(-h*Y)+(1*(na0+nb0-Y));
```

---

## APPENDIX 3

### Program *stochasticab.m* for stochastic simulation

```
% This program performs the stochastic simulation of the
```

AB process  
% k1 y k2 are the direct and reverse reaction rate constants  
% na0 and nb0 are the initial numbers of molecules  
% Z is the total number of reactions  
% numtray is the number of trajectories to be generated

```
k1=5.e-3;
k2=1.e-03;
h=k1;
l=k2;
na0=175;
nb0=25;
Z=500;
numtray=2;

% in this example we generate and plot just two trajectories
unif=rand(numtray,N,2);

na=na0;
nb=nb0;
t=0;
x(1:numtray,1)=ones(numtray,1)*na0;
xb(1:numtray,1)=ones(numtray,1)*nb0;
for m=1:numtray;
    na=na0;
    nb=nb0;
    t=0
    for i=2:N;
        Wplus(i)=1*nb;
        Wminus(i)=h*na;
        a=Wplus(i)+Wminus(i);
        wplus(i)=1*nb/((h*na)+(1*nb));
        wminus(i)=(h*na)/((h*na)+(1*nb));
        u(i)=(1/a)*log(1/unif(m,i,1));
        if wminus(i)>unif(m,i,2)
            v=-1;
        else
            v=1;
        end;
        na=na+v;
        nb=nb-v;
        t=t+u(i);
        x(m,i)=na;
        xb(m,i)=nb;
        time(m,i)=t;
    end;
% close loop for each trajectory
end;
% close loop for number of trajectories
matrix(:,(i-1)*3+1)=time(m,:);
plot(time(1,:),x(1,:),time(2,:),x(2,:),time(1,:),xb(1,:),time(2,:),xb(2,:))
% plot two trajectories
matrix=[time(1,:) x(1,:) xb(1,:) time(2,:) x(2,:) xb(2,:)'];
save figure2.dat matrix/ascii;

% save results
□
```