Fall 2007 Math 636

Markov Chain Theory

The simplest of stochastic processes is one where the current state decides the next state of the system. If a discrete stochastic process depends only on the current state (so is independent of its past history), then this is called a *Markov process*. A *Markov chain* is a model that follows a series of steps using a Markov process at each step.

Consider a system that has n possible states and assume over a fixed time period there is a certain probability t_{ij} that the system moves from state j into state i. These transition probabilities form a *transition matrix*, $T = (t_{ij})$, where the columns sum to one. If we define a probability vector, $\mathbf{x} = (x_1, ..., x_n)^T$, with nonnegative entries summing to one, then a general Markov model for transitions has the form of a discrete dynamical system given by

$$\mathbf{x}_{n+1} = T\mathbf{x}_n.$$

Since the columns of T sum to one, then the dominant eigenvalue is $\lambda_1 = 1$, and it's associated eigenvector (normalized) provides the equilibrium distribution, provided some power of T has all positive entries. (It is easy to see that $\lambda_1 = 1$ is an eigenvalue by considering looking at $\mathbf{x} = [1, ..., 1]$, since $\mathbf{x}T = \mathbf{x}$. The Gerschgoren Circle theorem, which states that all eigenvalues of a matrix, (t_{ij}), lie inside a circle radius $C_j = \sum_{j \neq i} t_{ij}$ with center at t_{jj}, shows all others have magnitude less than 1.)

Princeton Forest Ecosystem

A complex model for the successional dynamics for the Princeton forest ecosystem was created by Horn [2,3]. Transitional probabilities were found for five dominant species of trees based on which species replaced a resident species of tree that dies. The Figure 1 shows the transition probabilities for the five dominant species of trees.

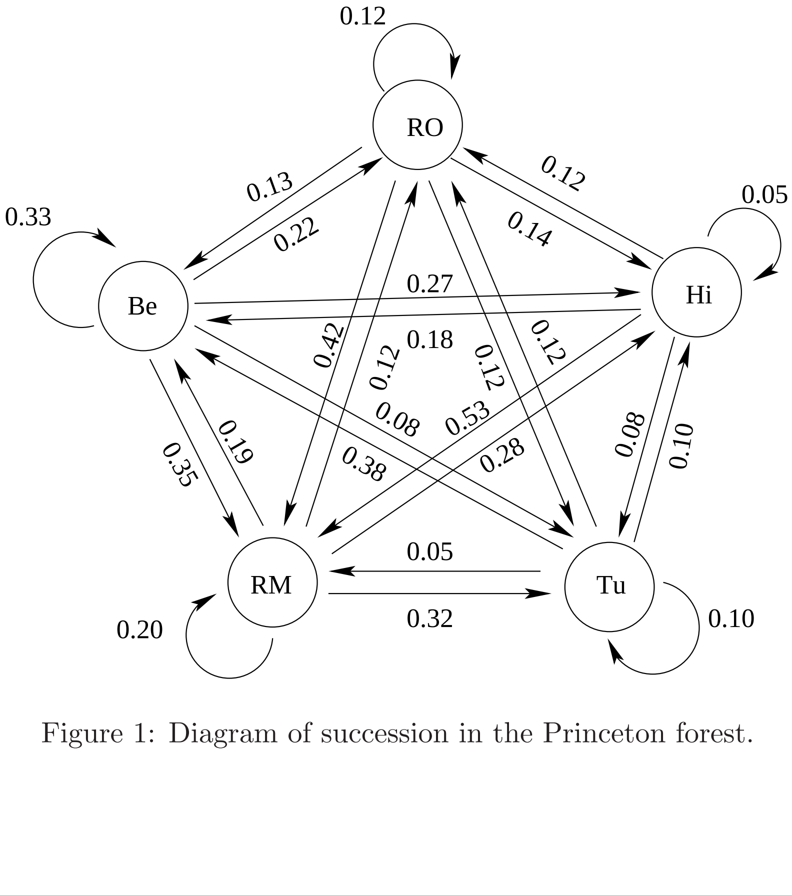


Figure 1: Diagram of succession in the Princeton forest.

Assume that the ordering of the probability state vector is Red Oak, Hickory, Tulip tree, Red Maple, and Beech (in that order), then the transition matrix is given by

$$T = \begin{pmatrix} 0.12 & 0.14 & 0.12 & 0.12 & 0.13 \\ 0.12 & 0.05 & 0.08 & 0.28 & 0.27 \\ 0.12 & 0.10 & 0.10 & 0.05 & 0.08 \\ 0.42 & 0.53 & 0.32 & 0.20 & 0.19 \\ 0.22 & 0.18 & 0.38 & 0.35 & 0.33 \end{pmatrix}.$$

The normalized eigenvector associated with $\lambda_1 = 1$ is

$$\mathbf{x_e} = \begin{pmatrix} 0.128 \\ 0.197 \\ 0.080 \\ 0.298 \\ 0.297 \end{pmatrix}.$$

This eigenvector shows that the predicted climax forest community should be approximately 12.8% Red Oak, 19.7% Hickory, 8.0% Tulip tree, 29.8% Red Maple, and 29.7% Beech.

Stochastic Models – Gillespie's Method

Many chemical reaction systems are very complex. It can be hard to create detailed ordinary differential equation systems or the number of reacting molecules is small. This is particularly true for biochemical reactions happening inside cells. We will develop the basic scheme for Gillespie's method [1], which creates a stochastic approach to simulating chemical reactions by considering molecules in the reactions as a kind of random walk process. This process is governed by differential-difference equation, called the *master equation.*

The traditional models for chemical kinetics use systems of ordinary differential equations of the form:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, ...x_n), \\ \dot{x}_2 &= f_2(x_1, x_2, ...x_n), \\ &\vdots \\ \dot{x}_n &= f_n(x_1, x_2, ...x_n). \end{aligned}$$

Usually, these are highly nonlinear systems determined by structures and rate constants for M chemical reactions. The models are continuous and deterministic. Biological situations commonly have small numbers of specific molecules and significant fluctuations. Ordinary differential equations may not accurately follow the "average" molecular populations. This may be particularly significant for certain threshold switches.

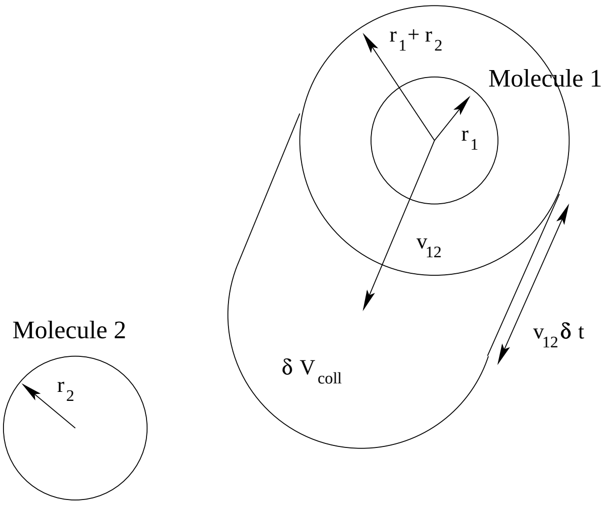
Stochastic Formulation of Chemical Kinetics

Assume idealized spherical molecular species, S_1 and S_2, in thermal, but not necessarily chemical equilibria. A collision occurs when the center to center distance decreases to

$$r_{12} = r_1 + r_2.$$

We calculate the rate of collisions in a fixed volume. Estimate the number os S_2 molecules whose centers lie inside

$$\delta V_{coll} = \pi r_{12}^2 + v_{12}\delta t.$$



(If we let $\delta t \to 0$, then this becomes an ordinary differential equation model.)

Assume that the molecules are distributed randomly and uniformly in volume, V. This implies that the probability that the center of an arbitrary S_2 molecule is inside δV_{coll} at time t is the ratio $\frac{\delta V_{coll}}{V}$.

We average this ratio over velocity distributions of S_1 and S_2. The average probability that a particular 1-2 pair will collide in a small time interval δt is given by:

$$\overline{\frac{\delta V_{coll}}{V}} = V^{-1}\pi r_{12}^2 \overline{v}_{12}\delta t,$$

where $\overline{v}_{12} = \sqrt{8kT/\pi r_{12}}$ (Maxwellian velocity distribution). If there are X_1 molecules of S_1 and X_2 molecules of S_2, then the probability of any 1-2 collisions is

$$X_1 X_2 V^{-1}\pi r_{12}^2 \overline{v}_{12}\delta.$$

These collisions are a *stochastic Markov process.*

Stochastic Reaction Constant c_μ

We apply the above stochastic Markov process to reactive collisions, then

$X_1 X_2 c_1 dt =$ probability that an R_1 reaction will occur inside the volume, V, in the time interval $(t, t + dt)$. More generally, we suppose that V contains a spatially homogeneous mixture of X_i molecules of species S_i, $(i = 1, .., N)$. Further, these N species interact through M specified chemical reaction channels, R_μ, $(\mu = 1, ..M)$. Assume there exists M constants, c_μ, $(\mu = 1, ..M)$, depending on physical properties of the molecules and the temperature, then

$c_\mu dt =$ average probability that a particular combination of R_μ reactant molecules will react in the time interval $(t, t + dt)$. This equation is the fundamental hypothesis of the stochastic formulation of chemical kinetics and is valid for “well-mixed” systems.

This mean stochastic approach is closely related to the rate constants, k_i, in deterministic equations:

$$k_i = \frac{V c_i \langle X_i X_{i+1}\rangle}{\langle X_i\rangle\langle X_{i+1}\rangle},$$

where $\langle X\rangle =$ average ensemble and $\langle XY\rangle \simeq \langle X\rangle\langle Y\rangle$. It follows that $k_i \simeq V c_i$. (The V remains in this formulation whereas the ordinary differential equation (ODE) models use concentrations.) There are a number of differences between this formulation and the ODE models, especially due to the discrete nature and other properties, but the models are considered closely related.

Master Equation Approach

We begin with the "Grand Probability function,"

$$\frac{d}{dt}P(X_1,...X_N;t) = \sum_{\mu=1}^{M}[B_\mu - a_\mu P(X_1,...X_N;t)].$$

Its derivation is very similar to the derivation of the birth only process discussed earlier in class. In general, this equation is harder to use than deterministic equations.

We can look at the discrete time version of this grand probability function with time step dt. The equation is given by:

$$P(X_1,...X_N;t+dt) = P(X_1,...X_N;t)\left[1-\sum_{\mu=1}^{M}a_\mu dt\right]+\sum_{\mu=1}^{M}B_\mu dt.$$

The quantity $a_\mu dt = c_\mu dt\times$ (number of distinct R_μ molecular combinations in the state $(X_1,...X_N)$ = probability that an R_μ reaction will occur in V during $(t,t+\delta t)$ given that the system is in the state $(X_1,...X_N)$) at time t. The terms $B_\mu dt$ represent the probabilities that the system is one R_μ reaction removed from the state $(X_1,...X_N)$.

Stochastic Simulation Algorithm

We want to move away from the Master equation to determine how to simulate the stochastic time evolution of the chemical reactions. Given that the system is in the state $(X_1,...X_N)$ at time t, to simulate the model we need to answer two questions:

1. When will the next reaction occur?
2. What kind of reaction is it?

We introduce the probability function

$P(\tau,\mu)$ $\equiv$ probability that given state $(X_1,...X_N)$ at time t, the next reaction in V occurs in the infinitesimal time interval $(t+\tau, t+\tau+d\tau)$ and this reaction is an R_μ reaction.

This is the reaction probability density function on the space of the continuous variable τ $(0\le\tau<\infty)$ and the discrete variable μ $(\mu=1,2,...,M)$. These variables are valuable for answering the two questions posed above.

For the algorithm, we need analytical expressions for $P(\tau,\mu)$. Begin by defining the following for each reaction, R_μ

h_μ $\equiv$ number of distinct R_μ molecular reactant combinations available in the state $(X_1,...,X_N)$ $(\mu=1,...,M)$.

For example, $S_1+S_2\rightarrow$ *anything* gives $h_\mu = X_1X_2$, while $2S_1\rightarrow$ *anything* gives $h_\mu = \frac{X_1(X_1-1)}{2}$. In general, h_μ is a combinatorial function of $X_1,...,X_N$. It follows that

$a_\mu dt \equiv h_\mu c_\mu dt$ $=$ probability that an R_μ reaction will occur in V in $(t,t+dt)$ given that the system is in the state $(X_1,...,X_N)$ at time t $(\mu=1,...,M)$.

We write the probability density function as the product of $P_0(\tau)$, which is the probability that given that the system is in the state $(X_1,...,X_N)$ at time t and no reaction occurs in the time interval $(t,t+\tau)$, and the subsequent probability that an R_μ reaction occurs in the interval $(t+\tau,t+\tau+d\tau)$:

$$P(\tau,\mu)d\tau = P_0(\tau)\cdot a_\mu d\tau.$$

The expression

$$P_0(\tau) = \exp\left(-\sum_{\mu=1}^{M} a_\mu \tau\right),$$

is the exponential waiting time for a reaction to occur. It follows that the reaction probability density function satisfies:

$$P(\tau, \mu) = \begin{cases} a_\mu \exp(-a_0\tau) & \text{if } 0 \leq \tau < \infty \text{ and } \mu = 1, ..., M \\ 0 & \text{otherwise,} \end{cases}$$

where $a_\mu = h_\mu c_\mu$ $(\mu = 1, ...M)$ and

$$a_0 \equiv \sum_{\nu=1}^{M} a_\nu \equiv \sum_{\nu=1}^{M} h_\nu c_\nu.$$

This probability is key to the *Stochastic Simulation Algorithm.*

The actual simulation algorithm is a Monte Carlo simulation that uses two random numbers at each step of the process. The random numbers, r_1 and r_2, selected from the unit interval, give the waiting time, τ, for a reaction to happen and define specifically which reaction, μ, occurs. We chose these variables as follows:

$$\begin{aligned} \tau &= \frac{1}{a_0} \ln\left(\frac{1}{r_1}\right), \\ \mu &= \text{integer satisfying } \sum_{i=1}^{\mu-1} a_i \leq r_2 a_0 \leq \sum_{i=1}^{\mu} a_i. \end{aligned}$$

Specific Simulation Algorithm

We are now ready to write the specific stochastic simulation algorithm for the time evolution of a chemically reacting system.

Step 0 (Initialization): Input M reaction constants $c_1, ..., c_M$ and N initial molecular populations numbers $X_1, ..., X_N$. Set $t = 0$ and reaction number $n = 0$. Initialize the random number generator.

Step 1: Calculate and store the M quantities $a_1 = h_1 c_1, ..., a_M = h_M c_M$ for the current populations, where h_i is a function of $X_1, ..., X_N$. Calculate and store $a_0 = \sum_{\mu=1}^{M} a_\mu$.

Step 2: Generate random numbers r_1 and r_2. Compute

$$\begin{aligned} \tau &= \frac{1}{a_0} \ln\left(\frac{1}{r_1}\right), \\ \mu &= \text{integer satisfying } \sum_{i=1}^{\mu-1} a_i \leq r_2 a_0 \leq \sum_{i=1}^{\mu} a_i. \end{aligned}$$

Step 3: Increase t by τ (add waiting time) and adjust molecular populations based on the reaction R_μ. (For example, if $S_1 + S_2 \to 2S_1$, then X_1 increases by one and X_2 decreases by one.) Increase the reaction counter by one, $n \to n + 1$.

Repeat Steps 1–3 until the reaction reaches the time desired. This simulation should be run multiple times with averages and standard deviations computed.

Discussion of the Simulation

Advantages

1. This method is exact and mathematically rigorous, designed to simulate stochastic events in the spatially homogeneous master equation.
2. Not approximations of continuous changes with finite time steps, so allows sudden molecular changes.
3. Easily coded independent of how complicated and coupled the chemical equations.
4. Minimal computer memory required because of the Markov process.
5. Can easily obtain averages and variation to collect statistics on the reactions.

Disadvantages

1. Uses lots of computer time, so need high speed processors.
2. Only a limited number of molecules and reactions are possible from a practical standpoint.
3. Need high quality random number generators because of the huge number of random numbers being used.
4. Statistical averages are computationally expensive.

Gillespie Algorithm for Lotka Reactions

The paper by Gillespie [1] showed the Lotka chemical reactions (developed by Lotka in 1920) that result in the famous Lotka-Volterra predator prey model. The chemical reactions are written:

$$\begin{array}{rcl} \bar{X} + Y_1 & \overset{c_1}{\longrightarrow} & 2Y_1 \\ Y_1 + Y_2 & \overset{c_2}{\longrightarrow} & 2Y_2 \\ Y_2 & \overset{c_3}{\longrightarrow} & Z \end{array}$$

The resulting differential equations are given by

$$\begin{aligned} \frac{dY_1}{dt} &= c_1XY_1 - c_2Y_1Y_2, \\ \frac{dY_2}{dt} &= c_2Y_1Y_2 - c_3Y_2, \end{aligned}$$

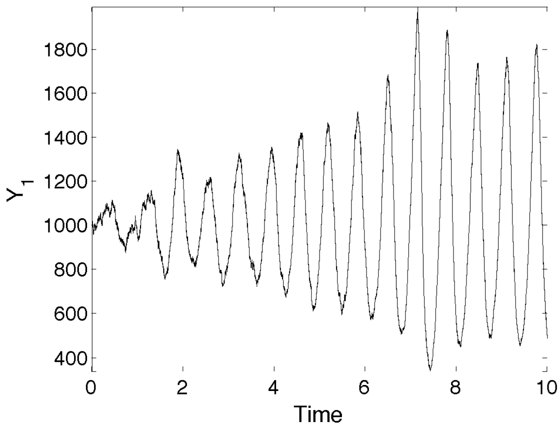
which is the classic Lotka-Volterra model. In their simulation, they used the parameters, $c_1X = 10$, $c_2 = 0.01$, and $c_3 = 10$. It is easy to show that the nonzero equilibrium is $Y_{1e} = Y_{2e} = 1000$, and these values were used as the starting values for the simulation.

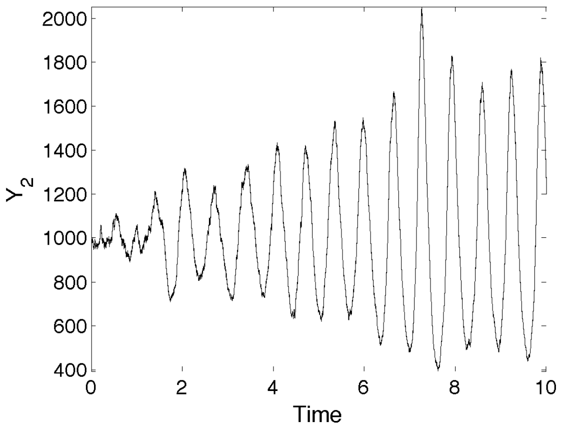
John Aven wrote the following MatLab code to perform the Gillespie algorithm for this model.

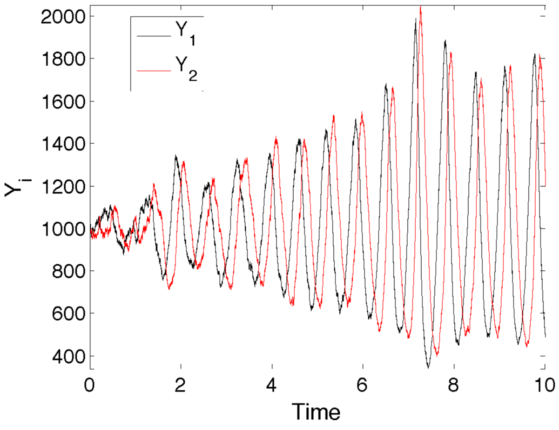
```
    function    Chem_React
%
%   This code simulates the Chemical Reaction on page 2350 (Eqs.38a-38c) of
%   Gilespie's Paper using the Stochastic Queuing routine from Figure 2 of
%   the same paper. We recreate the images in Figures 8a and 8c (not
%   exactly the same since simulations are intrinsicly stochastic).
%
%
%       Written 11-24-2007 by John L. Aven
clc clear all close all
% -----      Initialize Molecular Populations       ----- %
Y_1_i = 1000.0; Y_2_i = 1000.0;
% -----   FoodStuff Population(Quantity)   -----  %
X = 10.0 ^ (1.0);
% -----   The Number of Reactions   -----   %
M = 3;
% -----  Initialize Chemical Reaction Coefficients  ----- %
c1 = 10.0 / X; c2 = 0.01; c3 = 10.0;
%   --   Place coefficients in Vector C   --    %
C = [c1;c2;c3];
tau = 0.1;                      %The Waiting Time.
% Initialization of Evolutionary Variables;
Y = [Y_1_i;Y_1_i];
time = 0.0; n = 1;
t = 0;
t_max = 10.0;

while time <= t_max
%     C(1) = 10.0/X;                            % Uncomment if you allow X
                                                % to change with time.
    A = C .* Reaction_Combination_Functions([X;Y(:,n)]);
    A_sum = sum(A);         % COEFFICIENT a_o
    u = rand(2);            % The two uniform (0,1) random variables.
    tau = ( 1 / A_sum ) * log ( 1 / u(1) );     % The next waiting time.
    time = time + tau                          % Update the time
    t(n+1) = time ;                             % Store time for plotting
    seek_value = u(2) * A_sum;                  % next reaction threshold
    %  ---  Find the Reaction to Enact  --  %
    seek_truth = 1;
    seek_test  = A(1);
    seek_mu    = 1;
    while (seek_truth == 1)
        if (seek_value < seek_test)
            mu = seek_mu;
            seek_truth = 0;
        elseif (seek_mu < M)
            seek_mu = seek_mu + 1;
            seek_test = seek_test + A(seek_mu);
        else
            mu = M;
            seek_truth = 0;
        end
    end
    n = n + 1;                              % Update the reaction counter
    if (mu == 1)        % Reaction 1 (X + Y1 --> 2*Y1)
        Y(1,n) = Y(1,n-1) + 1.0;
%         X = X - 1.0;                      % Uncomment if X can change
                                            % with time
        Y(2,n) = Y(2,n-1);
    elseif (mu == 2)    % Reaction 2 (Y1 + Y2 --> 2*Y2)
        Y(1,n) = Y(1,n-1) - 1.0;
        Y(2,n) = Y(2,n-1) + 1.0;
    elseif (mu == 3)    % Reaction 2 (Y2 --> Z)
        Y(1,n) = Y(1,n-1);
        Y(2,n) = Y(2,n-1) - 1.0;
    end
end

figure(1)
plot(t,Y(1,:),'k')
axis tight
xlabel('Time','FontSize',20)
ylabel('Y_1','FontSize',20)

figure(2)
plot(t,Y(2,:),'k')
axis tight
xlabel('Time','FontSize',20)
ylabel('Y_2','FontSize',20)

figure(3)
plot(t,Y(1,:),'k')
hold on
plot(t,Y(2,:),'r')
axis tight
xlabel('Time','FontSize',20)
ylabel('Y_i','FontSize',20)
legend('Y_1','Y_2')

figure(4)
plot(Y(1,:),Y(2,:),'k.','MarkerSize',1)
axis tight
xlabel('Y_1','FontSize',20)
ylabel('Y_2','FontSize',20)

title('Dot Plot')

figure(5)
plot(Y(1,:),Y(2,:),'k','MarkerSize',1)
axis tight
xlabel('Y_1','FontSize',20)
ylabel('Y_2','FontSize',20)
title('Continuous Plot')
```

```
size(Y)

% ----------------------------------------------------------------------- %
function [H] = Reaction_Combination_Functions(Var)
% The Reaction Function for all Reactions in a Vectorized Form.


Var2 = [Var(2:3);1.0];

H = Var .* Var2;
```

Below are some graphic outputs for $t = 10$ sec simulations. These simulations require hundreds of thousands of time steps because of the small size of the time steps from the algorithm. Below are the time series simulations for Y_1 and Y_2.



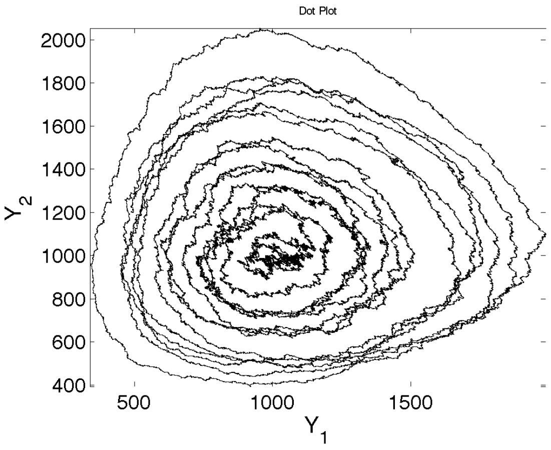


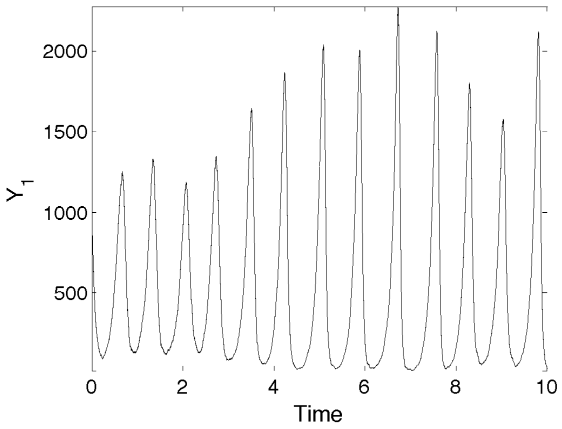


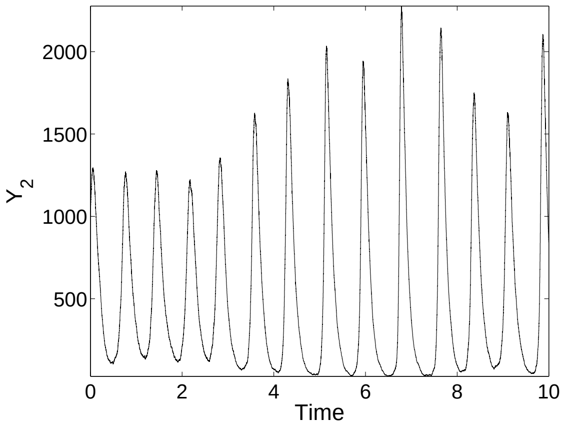
The Phase portrait is given by:

The simulations were repeated with $c_2 = 0.02$, which gave sharper oscillations from the interaction terms.

The Phase portrait is given by:



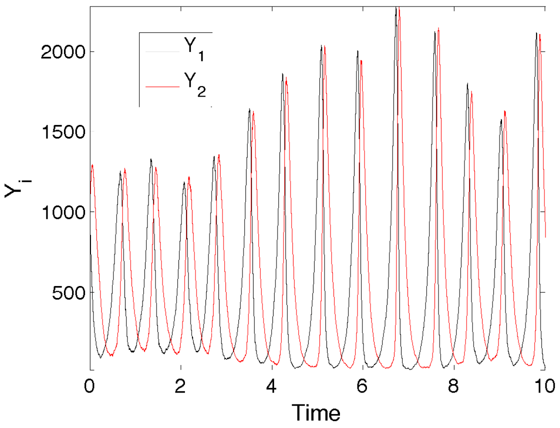




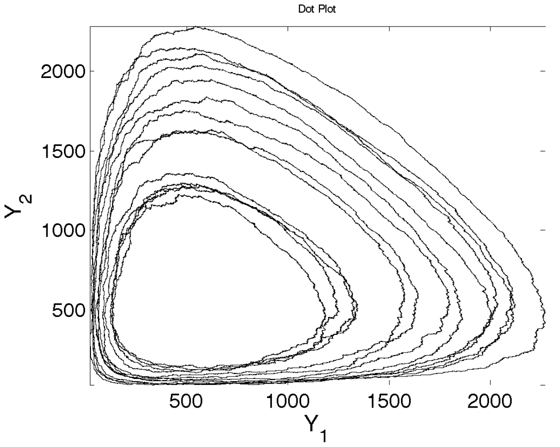
[1] Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions, *J. Phys. Chem.*, **81**, 2340-2361.

[2] Horn, H. S. (1975). Forest succession, *Scientific American*, **232**, 90-98.

[3] Horn, H. S. (1975). Markovian properties of forest succession. In M. L. Cody and J. M. Diamond, ed., *Ecology and Evolution of Communities*, 196-211, University Press, Cambridge, MA.



Y_1
Y_2
Y_i
2000
1500
1000
500
0
2
4
6
8
10
Time



Dot Plot
Y_2
2000
1500
1000
500
500
1000
1500
2000
Y_1