Mid Sem Assignment : Stochastic Simulation of Reaction Diffusion processes

October 2, 2014

1 Introduction

There are two fundamental approaches to the mathematical modelling of chemical reactions and diffusion. Deterministic models which are based on differential equations and stochastic models. Stochastic models can be simulated and can keep track of chemical species with low abundances during a reaction.
This report deals with stochastic modelling of reaction diffusion processes.
Every section includes :
(i) underlying reaction with a brief description
(ii)relevant algorithm
(iii)results
(iv)MATLAB program

2 Stochastic Simulation of Degradation

$$A \rightarrow \phi \tag{1}$$

(i) The reactant A decays with a rate constant k. ϕ represents a chemical species we do not include in our model.
kdt = probability that a randomly chosen molecule of A undergoes reaction in a time interval of $[t, t + dt)$.
Given A(t)= number of A molecules at time t
A(t)dt = gives the probability that exactly one reaction occurs in a time interval of $[t, t + dt)$.

(ii) Given that A(0) $= n_o$, the algorithm computes A(t) at any time $t > 0$. Algorithm:
1. A random number r uniformly distributed in the interval $(0, 1)$ is generated such that the probability that r is the subinterval $(a, b) \subset (0, 1)$ is equal to $b - a$ for any $a, b\epsilon(0, 1), a < b$.
2.If $r < A(t)k\mathrm{d}t$ then $A(t + dt) = A(t) - 1$ otherwise $A(t + dt) = A(t)$.
Step 1 is repeated and the algorithm is run several times in a loop to obtain the time evolution of number of A molecules and to observe its fluctuations.
Since r is a random number uniformly generated from the $(0, 1)$ interval the probability that $r < A(t)k\mathrm{d}t$ is equal to $A(t)k\mathrm{d}t$. Hence step 2 of the algorithm says that the probability that the chemical reaction occurs in the time interval $(t + dt)$ is equal to $A(t)k\mathrm{d}t$.

(iii)

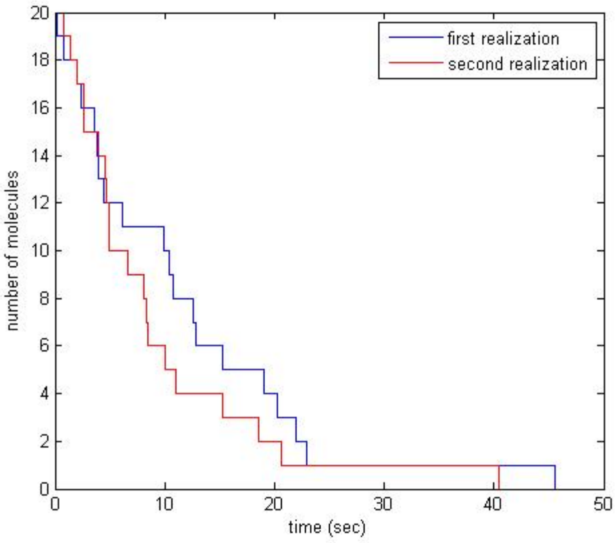

Figure 1: Two realizations of Step 1-2 for $k = 0.1sec^-1$ and $A(0) = 20$ with dt $= 0.005sec$

(iv)

```
%% section 2.1
%a1-b1

a=20;
k=0.1;
ds=0.005;
s=0;
store_a=zeros(10^5,1);
for i=1:10^5
    r=rand(1,1);             %generates ...
        random number - Step 1
    if r < a*k*ds            %checks for the ...
        probability of reaction occurring - ...
        Step2
        a = a - 1 ;          %executes the ...
            reaction - Step 2
        store_a(i,1)= a;     % stores the ...
            changes in number of A ...
            molecules if any
    else
        store_a(i,1)= a;
    end
    s=s+ds;
    store_a(i,2)=s;
end
stairs(store_a(:,2),store_a(:,1),'r')
```

3 Stochastic Simulation of degradation with τ

(i) In the above simulation, when $r > A(t)k\mathrm{d}t$ there is no change in the number of molecules of A because of which we observe many time intervals in Figure 1 wherein number of molecules remain same. Hence we end up generating many random numbers even though many a times the reaction doesn't occur. In order to reduce the computational intensity we compute τ such that from any time t, $t + \tau$ gives the time at which the next reaction occurs.
Deriving τ :
Let $f(A(t), s)\mathrm{d}s$ =the probability that next reaction occurs at during the time interval $[t + s, t + s+\mathrm{d}s)$, given that $A(t)$ molecules exist at t.
$g(A(t), s)$ = probability that no reaction occurs in $[t, t + s)$.
So, probability that reaction occurs only during the interval ds but not in $[t, t + s)$ is equal to

$$f(A(t), s)ds = g(A(t), s)A(t + s)kds \tag{2}$$

Since no reaction occurs in $[t, t + s)$, $A(t + s) = A(t)$. Implying

$$f(A(t), s)ds = g(A(t), s)A(t)kds \tag{3}$$

To compute probability $g(A(t), s)$ considering $\rho > 0$ and calculating the probability that no reaction occurs in the interval$[t, t + \rho+\mathrm{d}\rho)$ = probability of no reaction in $[t, t + \rho)*$ probability of no reaction in $[t + \rho, t + \rho+\mathrm{d}\rho)$. Hence,

$$g(A(t), \rho + d\rho) = g(A(t), \rho)[1 - A(t + \rho)kd\rho] \tag{4}$$

Since no reaction occurs in the interval
$[t, t + \rho), A(t + \rho) = A(t)$
$(g(A(t), \rho+\mathrm{d}\rho) - g(A(t), \rho)/\mathrm{d}\rho = -A(t + \rho)k\mathrm{d}\rho$
If $\mathrm{d}\rho \rightarrow 0$,
$\mathrm{d}g(A(t), \rho)/\mathrm{d}\rho = -A(t + \rho)k\mathrm{d}\rho$
Solving for the above equation with $g(A(t), 0) = 1$,
$g(A(t), \rho) = e^{-A(t)k\rho}$
Substituting into 3

$$f(A(t), ds) = A(t)ke^{-A(t)ks}ds \tag{5}$$

To find a τ such that $t + \tau$ is time when next reaction occurs, $F(.)$ is defined by $F(\tau) = e^{-A(t)k\tau}$.

If τ is a random number belonging to the interval $(0, \infty)$, then $F(\tau)$is uniformly distributed between the interval $(0, 1)$.
Given $F(\tau)\epsilon(0, 1)$, a uniformly distributed random number r is generated by the algorithm such that $r = F(\tau)$.
From 5

$$\tau = 1/(A(t)k)ln[1/r] \tag{6}$$

(ii) The algorithm slightly modified from the previous section in that now we increment time such that $t = t + \tau$ wherein τ analogous to dt determines the interval after which the next reaction occurs.
Algorithm:
1. A random number r is generated uniformly distributed in the interval $(0, 1)$.
2. Compute time of next reaction $t + \tau$.
3. At $t + \tau$ change the number of molecules such that $A(t + \tau) = A(t) - 1$.
Steps 1-3 is repeated and the algorithm is run several times in a loop to obtain the time evolution of number of A molecules as done is previous section as well.

(iii)

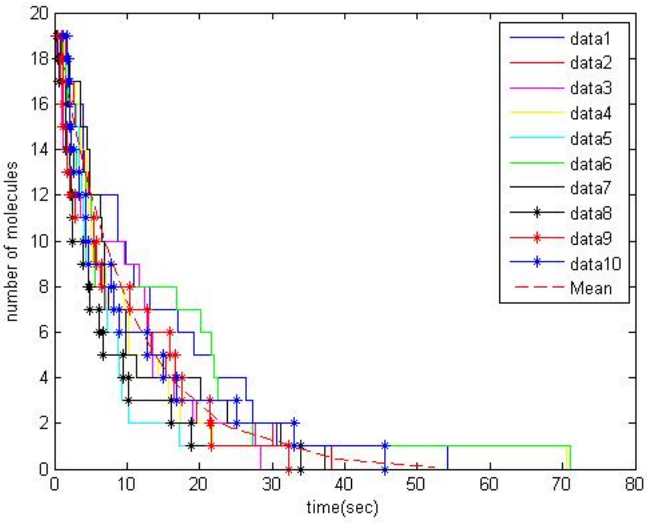

Figure 2: Ten realizations of Step 1-2 for $k = 0.1sec^-1$ and $A(0) = 20$

It is seen that no two realizations for a stochastic simulation are the same. The stochastic mean of $A(t)$ over many realizations is plotted as the dashed curve and was computed using the chemical master equation. The chemical master equation is given by:

$$dp_n/dt = k(n + 1)p_{n+1} - knp_n \tag{7}$$

The differential equation can be solved for a solution the mean of which is given by:

$$\sum_{n=0}^{n_o} np_n(t) = n_o e^{-kt} \tag{8}$$

The mean in 8 is also the solution of the deterministic equation given by $da/dt = -ka$ with initial condition of $a(0) = n_o$.
(iv)

```
%a2-c2
a=20;
k=0.1;
store_a=zeros(10^4,4);
t=0;
for i=1:10^4
    r=rand(1,1);
    tau=(1/(a*k))*reallog(1/r); ...
        %calculating the time for next reaction
    t = t + tau;                ...
        %incrementing the time step accordingly
    a= a -1;                         % carrying ...
         out the reaction
    if a ==0
        t = t + tau;
        store_a(i,2) = t;
        break
    end
    store_a(i,1) = a;
    store_a(i,2) = t;

end
figure(2);
stairs(store_a(1:20,2),store_a(1:20,1),'b*-')
plot(store_a(1:49,2),store_a(1:49,1),'*')
% mean by master equation
a=20;
n=exp(-k.*store_a(1:20,2));
m=a.*n;
plot(store_a(1:20,2),m,'r--')
```

4 Stochastic simulation of Production and Degradation

$$A \to \phi \tag{9}$$
$$\phi \to A \tag{10}$$

(i) 9 is similar to 1 representing degradation and 10 represents production of molecule A. The rate constant for the degradation reaction is k_1 and for the production reaction is k_2. The impact of ϕ on the production of A is incorporated into k_2.

(ii) Algorithm:
1. Two random numbers $r_1 and r_2$ are generated uniformly from the interval $(0, 1)$.
2. $\alpha_o = A(t)k_1 + k_2$ is calculated.
3. $\tau = 1/(A(t)k)ln[1/r]$ is calculated and so is the time for next reaction given by $t + \tau$.
4.The number of molecules are accordingly adjusted when $t = t + \tau$ such that

$$A(t + \tau) = A(t) + 1\, if\; r_2 < k_2/\alpha_o \tag{11}$$
$$A(t + \tau) = A(t) - 1\, if\; r_2 >= k_2/\alpha_o \tag{12}$$

α_o in Step 2 given the combined probability of 9 occurring given by $A(t)k_1dt$ and 10 occurring given by k_2dt. Step 3 from the previous section computes the time of next reaction. Once $t + \tau$ is computed, the second reaction occurs with a probability of $r_2 < k_2/\alpha_o$, and the first reaction occurs with (1 - probability of the second reaction). This is given in Step 4 of the algorithm.
To compute the stochastic mean and stochastic fluctuation, the chemical master equation as given in the previous section was constructed for the system of equations. The mean from the master equation for the given system of equations is computed to be the solution of $dM/dt = -k_1M + k_2$. Wherein M is the mean and its solution is given by $M(t)$ with an initial condition of $M(0) = 0$. The solution is plotted as the dashed curve in Figure 3.
The Stochastic mean can be thought as the steady state value of $A(t)$ as $t \to \infty$. In order to quantify the fluctuations, either the standard deviation can be calculated or a quantity called the stationary distribution is computed. Stationary distribution is given by :

$$\phi(n) = \lim_{t\to\infty} p_n(t)n = 0, 1, 2, 3, ... \tag{13}$$

The stationary distribution gives the probability that $A(t) = n$ after an infinitely long time. It can be either obtained from a long time simulation or by $\phi(n) = e^-k2/k1/n!(k2/k1)^n$ obtained from the chemical master equation at the steady state. The results are plotted in Figure 4. The histogram was obtained by running Step 1-4 over 10^5 sec and the black line from the equation of $\phi(n)$.

(iii)

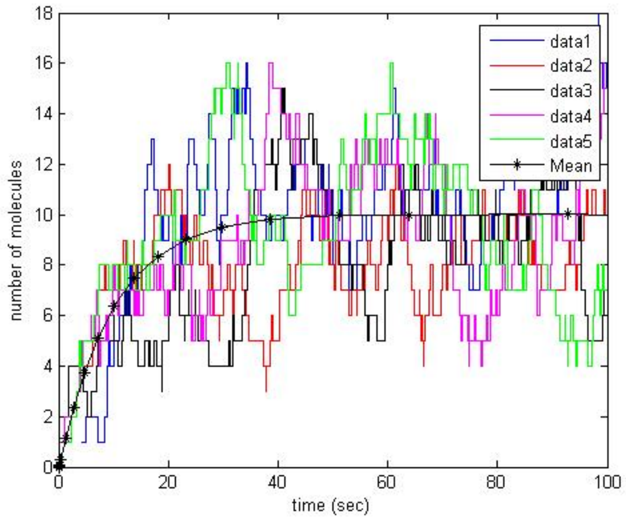

Figure 3: Five realizations of Step 1-4 for $k_1 = 0.1sec^{-}1, k_2 = 1sec^{-}1$ and $A(0) = 0$

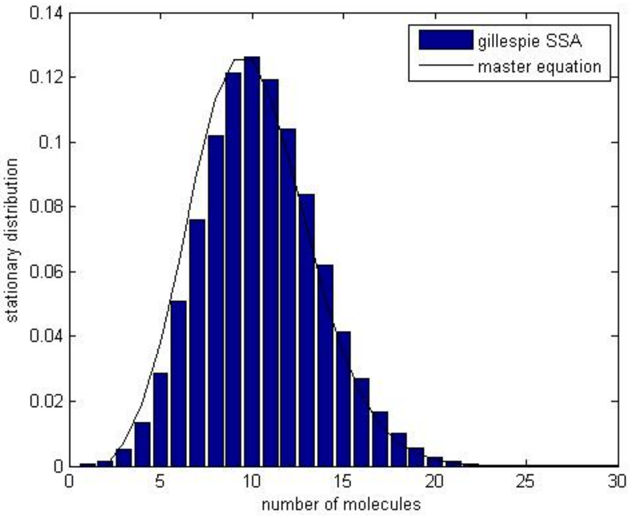

Figure 4: Stationary distribution $\phi(n)$ obtained by long time simulation (gillespie) and from chemical master equation for $k_1 = 0.1sec^{-}1, k_2 = 1sec^{-}1$ and $A(0) = 0$

(iv)

```
% a3-d3
for x =1:5
    a=0;
    k1=0.1;
    k2=1;
    t=0;
    for i =1:10^5
        r1=rand(1,1);
        r2=rand(1,1);
        alpha=a*k1 + k2;
        tau=(1/alpha)*reallog(1/r1);

        if r2 ≥ (k2/alpha) % happens only ...
            at t=t + tau
            a= a -1;
        else
            a =a + 1;
        end
        t=t + tau;
        store_a(i,x)=a;
        store_t(i,x)=t;

    end
end

%plots
figure(3);
stairs(store_t(:,1),store_a(:,1))
hold on
stairs(store_t(:,2),store_a(:,2),'r')
hold on
stairs(store_t(:,3),store_a(:,3),'k')
hold on
stairs(store_t(:,4),store_a(:,4),'m')
hold on
stairs(store_t(:,5),store_a(:,5),'g')
%mean plot
[T,Y] = ode23(@ode,[0 10000],0);
hold on
plot(T,Y,'k*-')

% histogram
max(store_a(:,1))
hist(store_a(:,1),26)
bar(hist(store_a(:,1),26)./10^5)

%stationary distribution
 x=[0 1];
 k1=0.1;
 k2=1;
for n=3:30
x(n)=(exp(-k2/k1)/factorial(n))*(k2/k1)^n;
end
x(1)=[];
x(2)=0;
y=x./sum(x);
figure(4);
plot(y)

%% ode file solving for the mean
function dy = ode(t,y)
dy = zeros(1,1);
k1=0.1;
k2=1;
dy(1) = -k1*y(1) + k2;
```

5 Gillespie

$$A + A \to \phi \tag{14}$$
$$A + B \to \phi \tag{15}$$
$$\phi \to A \tag{16}$$
$$\phi \to B \tag{17}$$

(i)The rate constants for the above series of second order chemical reactions are k_1, k_2, k_3andk_4 respectively. In order to simulate the number of molecules of A and B at any later time t the following algorithm was performed starting with $A(0) = n_o, B(0) = m_o$ at time $t = 0$. The probability of a reaction occurring is defined as propensity of the reaction times dt. The propensity of the first reaction is defined as $\alpha_1 = A(t)A((t) - 1)k_1$, for the second reaction as $\alpha_2 = A(t)B(t)k_2$, for the third as k_3 and fourth as k_4. The combined propensity of either of the four reactions happening is given by the summation over the propensity of each reaction.

(ii)Algorithm:
1. Two uniformly distributed random numbers r_1andr_2 are generated from (0,1) interval.
2. The combined propensity is computed and called α_o.
3. The time when the next reaction occurs is calculated give by τ as in the previous section such that $\tau = 1/alpha_o ln[1/r]$.
4.The number of molecules at $t + \tau$ are updated such that :

$$A(t+\tau) = A(t) - 2\,if\ 0 <= r_2 < alpha_1/\alpha_o$$
$$A(t+\tau) = A(t) - 1\,if$$
$$alpha_1/\alpha_o <= r_2 < alpha_1 + alpha_2/\alpha_o$$
$$A(t+\tau) = A(t) + 1\,if$$
$$alpha_1+alpha_2/\alpha_o <= r_2 < alpha_1+alpha_2+alpha_3/\alpha_o$$
$$A(t+\tau) = A(t)\,if$$
$$alpha_1 + alpha_2 + alpha_3/\alpha_o <= r_2 < 1$$

$$B(t+\tau) = B(t)\; if\; 0 <= r_2 < alpha_1/\alpha_o$$
$$B(t+\tau) = B(t) - 1\; if$$
$$alpha_1/\alpha_o <= r_2 < alpha_1 + alpha_2/\alpha_o$$
$$B(t+\tau) = B(t)\; if$$
$$alpha_1 + alpha_2/\alpha_o <= r_2 < alpha_1 + alpha_2 + alpha_3/\alpha_o$$
$$B(t+\tau) = B(t) + 1\; if$$
$$alpha_1 + alpha_2 + alpha_3/\alpha_o <= r_2 < 1$$

The stationary distribution is now two dimensional given that there are two chemical species A and B and is given by $\phi(n,m) = \lim_{t\to\infty} p_{n,m}(t)$. This is obtained by long time simulation of Step 1-4 of the algorithm as depicted in Figure 5. To obtain the stationary distribution along one of the dimensions, the other dimension can be summed over such that $\phi(n) = \sum_{m=0}^{\infty} \phi(n,m)$ as plotted in Figure 5.

(iii)

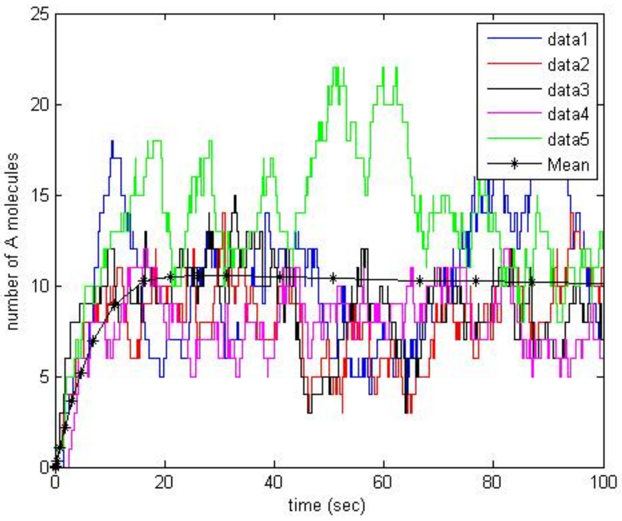

Figure 5: Five realizations with time evolution of number of A molecules with $A(0) = 0, B(0) = 0, k_1 = 10^{-}3sec^{-}1, k2 = 10^{-}2sec^{-}1, k3 = 1.2^{-}2sec^{-}1, k3 = 1.2sec^{-}1, k4 = 1sec^{-}1$

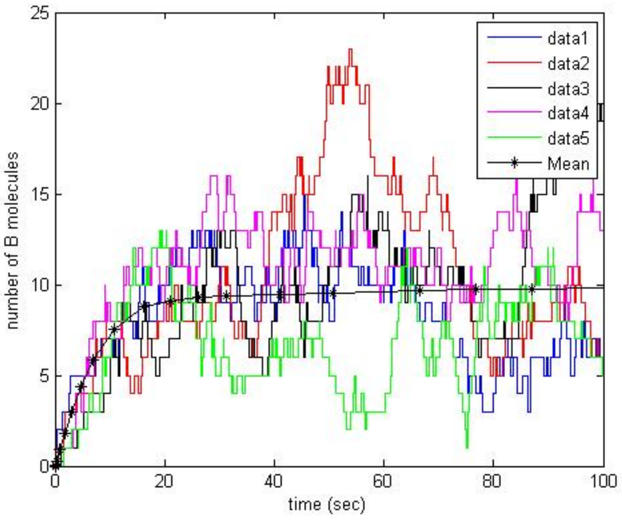

Figure 6: Five realizations with time evolution of number of B molecules with $A(0) = 0, B(0) = 0, k_1 = 10^{-}3sec^{-}1, k2 = 10^{-}2sec^{-}1, k3 = 1.2^{-}2sec^{-}1, k3 = 1.2sec^{-}1, k4 = 1sec^{-}1$

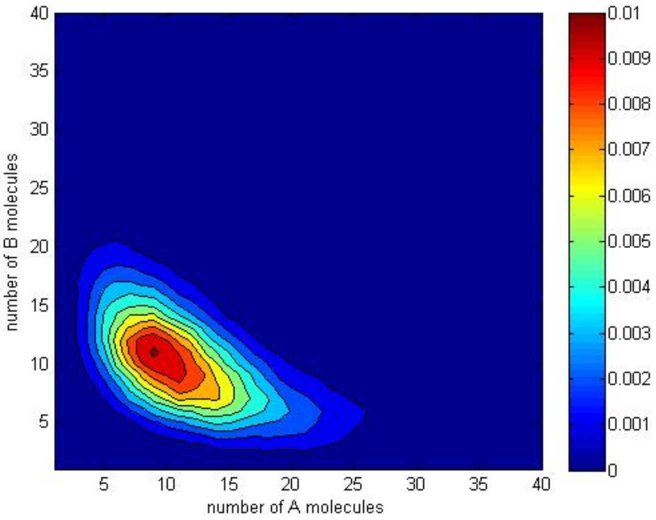

Figure 7: Stationary distribution$\phi(n,m)$ from long time simulation with $A(0) = 0, B(0) = 0, k_1 = 10^{-}3sec^{-}1, k2 = 10^{-}2sec^{-}1, k3 = 1.2^{-}2sec^{-}1, k3 = 1.2sec^{-}1, k4 = 1sec^{-}1$

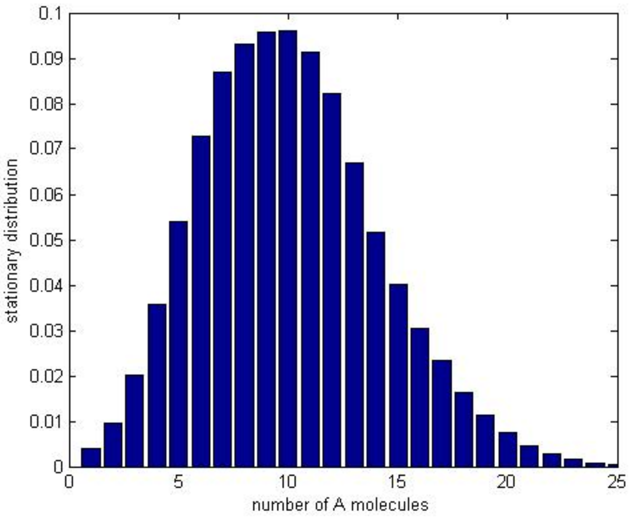

Figure 8: Stationary distribution$\phi(n)$ with $A(0) = 0, B(0) = 0, k_1 = 10^{-}3sec^{-}1, k2 = 10^{-}2sec^{-}1, k3 = 1.2^{-}2sec^{-}1, k3 = 1.2sec^{-}1, k4 = 1sec^{-}1$

(iv)

```
for z=1:5 %Number of realizations
    a=0;    %initialising
    b=0;
    k1=.001;
    k2=0.01;
    k3=1.2;
    k4=1;
    t=0;
    alpha1=0;
    alpha2=0;
    alpha3=0;
    alpha4=0;
    alpha=0;
    for i =1:10^6 % number of steps/ runs ...
        for each realization
        r=rand(2,1);
        alpha1=a*(a-1)*k1;
        alpha2=a*b*k2;
        alpha3=k3;
        alpha4=k4;
        alpha=alpha1+alpha2+alpha3+alpha4; ...
            %Step 2 of algorithm
        tau=(1/alpha)*reallog(1/r(1));     ...
            %Step 3 of algorithm
        mu=r(2)*alpha ;                      ...
            %Step 4 of algorithm
        if mu < alpha1 % happens only at ...
            t=t + tau
            a= a - 2;
        elseif mu < alpha1+alpha2
            a= a - 1;
            b= b - 1;
        elseif mu < alpha1+alpha2+alpha3
            a = a + 1;
        else
            b = b + 1 ;
        end

        t=t + tau;
        store_a(i,1)=a;
        store_t(i,1)=t;
        store_b(i,1)=b;
    end
end
figure(5);
stairs(store_t(:,1),store_a(:,1))
hold on
stairs(store_t(:,2),store_a(:,2),'r')
hold on
stairs(store_t(:,3),store_a(:,3),'k')
hold on
stairs(store_t(:,4),store_a(:,4),'m')
hold on
stairs(store_t(:,5),store_a(:,5),'g')

figure(6);
stairs(store_t(:,1),store_b(:,1))
hold on
stairs(store_t(:,2),store_b(:,2),'r')
hold on
stairs(store_t(:,3),store_b(:,3),'k')
hold on
stairs(store_t(:,4),store_b(:,4),'m')
hold on
stairs(store_t(:,5),store_b(:,5),'g')
%mean
[T,Y] = ode23(@ode_2,[0 1000],[0 0]);
hold on
plot(T,Y(:,1),'k*-')
plot(T,Y(:,2),'k*-')

%stationary distribution by long time ...
    simulation
z(:,1)=store_a(:,1);
z(:,2)=store_b(:,1);
stat=zeros(40,40);
for i= 1:length(store_b(:,1))
    if (((store_a(i,1)>0)&(store_a(i,1)<40))
    &((store_b(i,1)>0)&(store_b(i,1)<40)))

        stat(store_a(i,1),store_b(i,1))=
        stat(store_a(i,1),store_b(i,1))+1;

    end;
end
stat=stat./177559;
figure(7)
contourf(stat)
% histogram

%summing over b
hist(store_a(:,1),max(store_a(:,1)))
figure(8)
bar(hist(store_a(:,1),max(store_a(:,1)))./177559)

hist(store_b(:,1),max(store_b(:,1)))
bar(hist(store_b(:,1),max(store_b(:,1)))./10^8)
 if ((A>0)&(A<101))
        pa(A)=pa(A)+1;
end;
pa=pa./10^6;
bar(pa);
```

6 Smoluchowski equation and diffusion

(i) Diffusion is the random migration of particles or molecules arising from motion due to thermal energy. As a result the trajectory of a particle is not straight and but executes a random walk. The Smoluchowski equations predict the position of the diffusing particle at later time t given its initial position. To do this a random number generator with a zero mean and unit variance and normally distributed is used. Diffusive spreading of the molecule depends on the $diffusion\, constant\ D$ which depends on the size of the molecule, absolute temperature and viscosity of the solution. Choosing $\triangle t = 0.1sec$ starting from $[X(0), Y(0), Z(0)] = [0, 0, 0]$ the algorithm below was run for $10mins$.

(ii)Algorithm:
1. Three normally distributed random numbers r_1, r_2andr_3 are generated.
2. The positions of the particle at time $t + \triangle t$ are computed by
$X(t + \triangle t) = X(t) + \sqrt{2D\triangle t}r_1$
$Y(t + \triangle t) = Y(t) + \sqrt{2D\triangle t}r_2$
$Z(t + \triangle t) = Z(t) + \sqrt{2D\triangle t}r_3$
The $XandY$ coordinates of the diffusing particle are plotted as in Figure 6.
To obtain the probability distributions, assuming $\psi(x, y, z)dxdydz$ be the probability that $X(t)\epsilon[x, x + dx), Y(t)\epsilon[y, y + dy), Z(t)\epsilon[z, z + dz)$ at time t. Then ψ evolves according to :
$\frac{\partial\psi}{\partial t} = D\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} + \frac{\partial^2\psi}{\partial z^2}$
Assuming the random walk starting at origin with $\psi(x, y, z, 0) = \delta(x, y, z)$ where δ is the Dirac distribution at the origin, the time evolution is $\psi(x, y, z, t) = \frac{1}{4D\pi t}^{3/2} e^{-\frac{x^2+y^2+z^2}{4Dt}}$. Figure 6 is probability distribution obtained for two dimensions obtained by integrating over z, such that $\psi(x, y, t) = \frac{1}{4D\pi t} e^{-\frac{x^2+y^2}{4Dt}}$.

(iii)

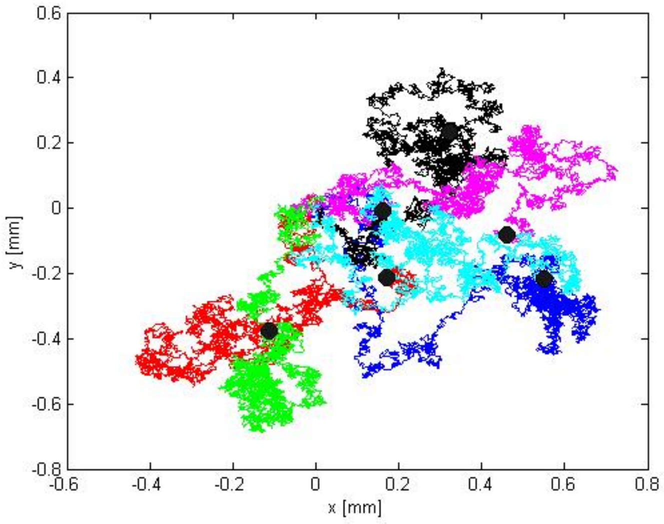

Figure 9: Six trajectories of a random particle for $D = 10^{-4}mm^2sec^{-1}$ and $\triangle t = 0.1sec$

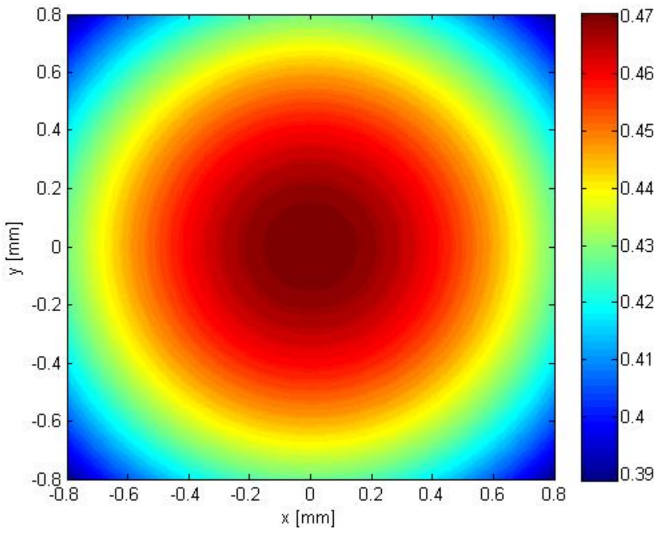

Figure 10: Probability distribution $\psi(x, y, t)$ given by at $t = 10min$

(iv)

```
%a6-b6

for i = 1: 6 % number of trajectories
    dt=0.1;
    D=0.0001;
    x=0;
    y=0;
    z=0;
    for n= 1:6000
         r=randn(3,1); %updating ...
             coordinates of the particle
        x = x + (r(1)*sqrt(2*D*dt));
        y = y + (r(2)*sqrt(2*D*dt));
        z = z + (r(3)*sqrt(2*D*dt));
    store_x(n,i)=x;
    store_y(n,i)=y;
    store_z(n,i)=z;
    end
end

figure(9);
plot(store_x(:,1),store_y(:,1))
hold on
plot(store_x(:,2),store_y(:,2),'r')
hold on
plot(store_x(:,3),store_y(:,3),'k')
hold on
plot(store_x(:,4),store_y(:,4),'m')
hold on
plot(store_x(:,5),store_y(:,5),'g')
hold on
plot(store_x(:,6),store_y(:,6),'c')

hold on

plot(store_x(end,1),store_y(end,1),'ko',
'MarkerSize',10,'MarkerFaceColor',[0.1,0.1,0.1])
hold on
plot(store_x(end,2),store_y(end,2),'ko',
'MarkerSize',10,'MarkerFaceColor',[0.1,0.1,0.1])
hold on
plot(store_x(end,3),store_y(end,3),'ko',
'MarkerSize',10,'MarkerFaceColor',[0.1,0.1,0.1])
hold on
plot(store_x(end,4),store_y(end,4),'ko',
'MarkerSize',10,'MarkerFaceColor',[0.1,0.1,0.1])
hold on
plot(store_x(end,5),store_y(end,5),'ko',
'MarkerSize',10,'MarkerFaceColor',[0.1,0.1,0.1])
hold on
plot(store_x(end,6),store_y(end,6),'ko',
'MarkerSize',10,'MarkerFaceColor',[0.1,0.1,0.1])
hold off

%prob distribution function in two dimensions
x=[-0.8:0.01:0.8];
y=[-0.8:0.01:0.8];
for i=1:length(x)
    for j=1:length(y)
        prob(i,j)=(1/4*D*6000*pi)*
        exp(-(x(i)^2+y(j)^2)/4*D*6000);
    end
end
figure(10)
contourf(x,y,prob)
shading flat;
```

7 Diffusion with boundary

(i) In the previous section the diffusing molecules is unbounded and free to do a random walk without any boundary restrictions. The algorithm given below has boundary conditions applied demonstrating a $reflective\ boundary\ condition$ or a zero flux boundary condition. It is used when there is no chemical interaction between the boundary and the diffusing molecule. Hence choosing $D = 10^{-4}mm^2sec{-}1, L = 1mm, X(0) = 0.4mmand\delta t = 0.1sec$, a system of 1000 molecules are released at position $x = 0.4mm$ at time $t = 0$.

(ii)Algorithm: 1.A normally distributed random number r_1 is generated (with zero mean and unit variance).
2. The position of the molecule at time $t + \triangle t$ is computed by $X(t + \triangle t) = X(t) + \sqrt{2D\triangle t}r_1$.
3. If $X(t + \triangle t)$ is less then 0, then $X(t + \triangle t) = -X(t) - \sqrt{2D\triangle t}r_1$ $X(t + \triangle t)$ is greater then L, then
$X(t + \triangle t) = 2L - X(t) - \sqrt{2D\triangle t}r_1$
The above algorithm is run for $4minsor\frac{4*60}{\triangle t}steps$. Ten realizations for a single molecules given the above boundary conditions is shown in Figure 7 . A rather interesting result is the spatial histogram depicting the density of the molecules Figure 7 . This is obtained by simulating 1000 molecules till $t = 4mins$ and diving the $[0, L]$ domain into 40 bins of length $h = L/40 = 25\mu m$. The number of molecules in each of the bins is calculated and the distribution is plotted.

(iii)

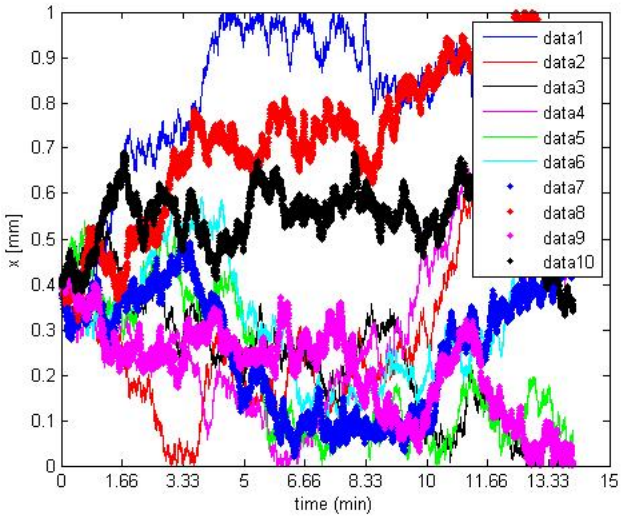

Figure 11: Ten trajectories a random particle with boundary conditions such that $D = 10^{-4}mm^2sec{-}1, L = 1mm, X(0) = 0.4mm\ and\ \triangle t = 0.1sec$

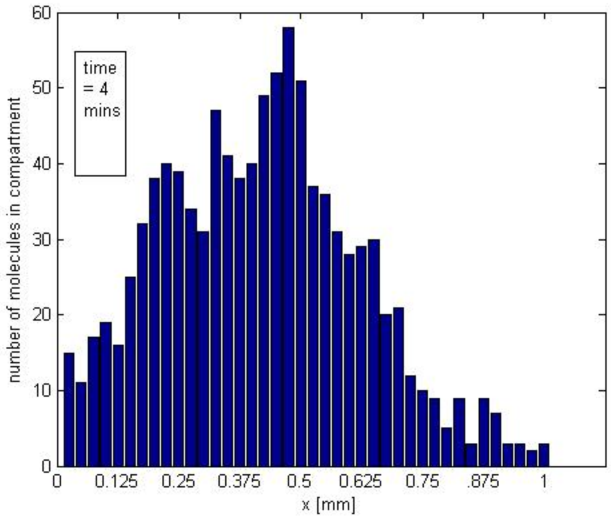

Figure 12: Number of molecules in bins of length $h = 25\mu m$ at $t = 4min$

(iv)

```
%a7-b7

for i = 1: 1000 % number of molecules
    D=0.0001;
    L=1;
    dt=0.1;
    x=0.4;
    for n= 1:2400 % % run for 4 mins
         r=randn(1,1);
```

```
        x = x + (r*sqrt(2*D*dt));
        if x < 0              %adding ...
            boundary conditions in the if loop
            x = -x - (r*sqrt(2*D*dt));
        elseif x > L
            x= 2*L -x - (r*sqrt(2*D*dt));
        end
        store_x(n,i)=x;

    end
end
figure(11);
plot(store_x(:,1))
hold on
plot(store_x(:,2),'r')
plot(store_x(:,3),'k')
plot(store_x(:,4),'m')
plot(store_x(:,5),'g')
plot(store_x(:,6),'c')
plot(store_x(:,7),'b.')
plot(store_x(:,8),'r.')
plot(store_x(:,9),'m.')
plot(store_x(:,10),'k.')
hold off
Xplot(:,1)=store_x(2400,:);
figure(12);
h=hist(Xplot(:,1),40);
bar(h)
```

8 Compartment based approach to diffusion

(i) Instead of simulating 1000 molecules between $x = 0\, and\, 1mm$ and dividing them into 40 compartments of $0.025\mu m$ length each, the time evolution of 40 compartments is carried out directly in this section. This is done by dividing the $[0, L]$ domain into 40 bins and assigning number of molecules to each compartment. Subsequently the time evolution is carried out using the gillespie algorithm using rate constant $d = D/h^2$.
Hence the series of reactions between compartments looks as follows :
$A_1 \leftrightarrow A_2 \leftrightarrow A_3 \leftrightarrow .... \leftrightarrow A_k$
such that $i = 1, 2, 3, 4, ...k\, and\, k = 40$ and $A_i \leftrightarrow A_{i+1}$ means that $A_i \rightarrow A_{i+1}$ and $A_{i+1} \rightarrow Ai$.

(ii) Algorithm:
Starting with initial condition $A_i(t) = a_{o,i}$. 1. Two random numbers uniformly distributed in $(0, 1)$ are generated.
2. The propensity functions of each of the reactions is computed including the forward and backward reactions such that $\alpha_i = A_i(t)d$ and the combined propensity is then given by $\alpha_o = \sum_{i=1}^{K-1} \alpha_i + \sum_{i=2}^{K} \alpha_i$
3.The time for next reaction $t + \tau$ is computed as done in the previous sections.
4. If $r_2 < \sum_{i=1}^{K-1} \frac{\alpha_i}{\alpha_o}$, then finding $j\epsilon 1, 2, 3....K - 1$ such that
$r_2 >= \frac{1}{\alpha_o} \sum_{i=1}^{i=j-1} \alpha_i$ and
$r_2 < \frac{1}{\alpha_o} \sum_{i=1}^{i=j} \alpha_i$
then the number of molecules are computed as :

$$A_j(t + \tau) = A_j(t) - 1 \qquad (18)$$
$$A_{j+1}(t + \tau) = A_{j+1}(t) + 1 \qquad (19)$$
$$A_i(t + \tau) = A_i(t) \qquad (20)$$

for $i \neq j, i \neq j + 1$
Else if $r_2 >= \sum_{i=1}^{K-1} \frac{\alpha_i}{\alpha_o}$, then finding $j\epsilon 2, 3....K$ such that
$r_2 >= \frac{1}{\alpha_o}(\sum_{i=1}^{i=K-1} \alpha_i + \sum_{i=1}^{i=j-1} \alpha_i)$ and
$r_2 < \frac{1}{\alpha_o}(\sum_{i=1}^{i=K-1} \alpha_i + \sum_{i=2}^{i=j} \alpha_i)$
then the number of molecules are computed as :

$$A_j(t + \tau) = A_j(t) - 1 \qquad (21)$$
$$A_{j-1}(t + \tau) == A_{j-1}(t) + 1 \qquad (22)$$
$$A_i(t + \tau) = A_i(t) \qquad (23)$$

for $i \neq j, i \neq j - 1$
In order to obtain a plot similar to Figure 7, the initial conditions is given by
$A_{16}(0) = 500, A_{17}(0) = 500\, and\, A_i(0) = 0$ for $i \neq 16, i \neq 17$ as $x = 0.4mm$ lies at the boundary of the 16^{th} and 17^{th} compartments. Figure 8 gives the distribution of number molecules in each compartment at time $t = 4min$. Figure 8 is obtained by simulating the time evolution of 1 particle starting with $A_{16}(0) = 1$. The time evolution of the particles across the compartments in plotted a distance measure as 1 compartment = $0.025\mu m$.

(iii)

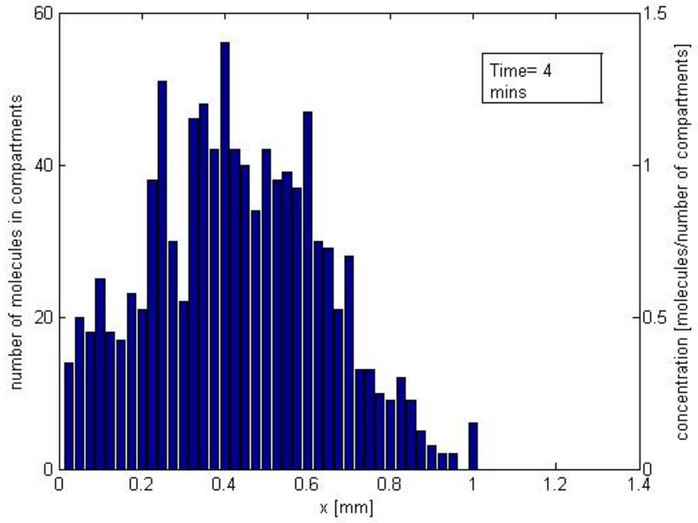

Figure 13: Distribution of molecules in each of the compartments with rate constant $d = D/h^2 = 0.16 sec^{-1}$, number of compartments $K = 40$, and $initial condition A_16(0) = 500, A_17(0) = 500 and A_i(0) = 0\, for\, i \neq 16, i \neq 17$

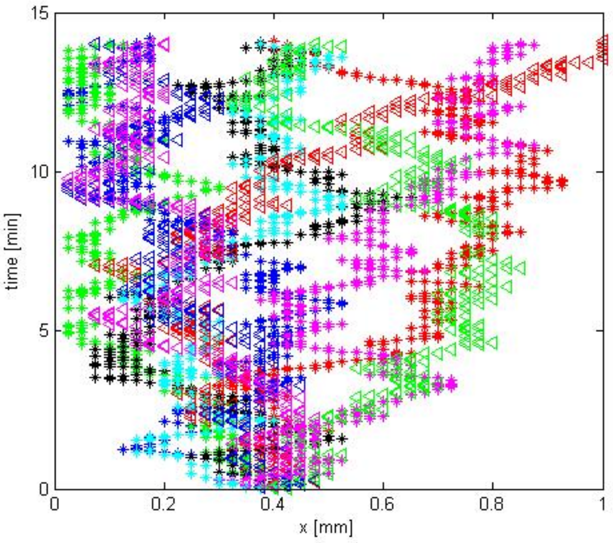

Figure 14: Ten realizations for an individual molecule using the above algorithm

(iv)

```
%initialisation
    clear;
    clc;
    t=0;
    D=0.0001;
    h=0.025; %1/number of compartments
    k=D/h^2;
    time= 14*60 ; % corresponds to number ...
        of steps taken by each molecules
    a = zeros(40,1);
    a(16,1)=1;
    a(17,1)=0;
    alpha_f=zeros(40,1);
    alpha_b=zeros(40,1);
     n=1;

while t < time

     %propensities in forward direction
     for i=1:39
     alpha_f(i,n)= a(i,n)*k;
     end

     %propensities in backward direction
     for i=2:40
     alpha_b(i,n)= a(i,n)*k;
     end

     %computing alpha summation of propensities
     alpha = 2*k*1-a(1,n)*k-a(40,n)*k;

     %generating 2 random numbers
      r=rand(2,1);

     %computing the time of next reaction ...
         occurence
     tau=(1/alpha)*reallog(1/r(1));

     %determining which forward reaction occurs
     mu=r(2)*alpha ;


     a(:,n+1) = a(:,n); %updating molecules ...
         in the next column
     j=0;
     while j < 39    % Step 4 of algorithm
        if sum(alpha_f(1:j,n))≤ mu
            j = j+1;
        else
            break
        end

     end

     if mu < sum(alpha_f(1:j,n)) %Step 4 of ...
         algorithm
        a(j,n+1)=a(j,n)-1;
        a(j+1,n+1)=a(j+1,n)+1;

     else
        forw=sum(alpha_f(:,n)); %Step 5 of ...
            the algorithm
        z=1;
            while z<40
                if  (forw + ...
                    sum(alpha_b(1:z,n)))≤ mu
                    z=z+1;
                else
                    a(z,n+1) = a(z,n)-1;
                    a(z-1,n+1) = a(z-1,n)+1;
                    break
                end
            end

     end
     t=t + tau;
     store_t(n,1)=t;
     n=n+1;
end

%plotting for histogram
figure(13);
y1=a(:,n);
x=[1*0.025:1*0.025:40*0.025];
y2=a(:,n).*0.025;
[hAx] =plotyy(x,y1,x,y2,@bar);
xlabel('x [mm]')
ylabel(hAx(1),'number of molecules in ...
    compartments') % left y-axis
ylabel(hAx(2),'concentration ...
    [molecules/number of compartments]')

%plotting 10 realizations
figure(14);
hold on
a(:,n)=[];
for j=1:n-1
    for i=1:40
        if a(i,j)==1
            plot(i*0.025,store_t(j,1)./60,'m<')
            hold on
        end
    end
end
```