Coursework 1

To obtain a thorough understanding of the system behaviour, the principles underpinning this stochastic reaction simulation algorithm needs to be derived. From Gillespie's paper[1] (page 3-6), it can be concluded that a definitive probabilistic description of the stochastic time evolution of chemical reacting system is introduced, one that addresses two unknowns to move the system in time: (1) time for next reaction (2) the kind of reaction is to occur. This probabilistic approach is established based on a reaction probability density function $P(\tau, \mu)$, which describes the probability that after time τ, from the current time t, there will occur a next reaction of type μ. It is interesting to see that this joint probability $P(\tau, \mu)$ couples together a continuous variable τ in the time space, and a discrete variable μ.

To expand this joint probability $P(\tau, \mu)$, an infinitesimal $d\tau$ is multiplied so that it can be rewritten as:$P(\tau, \mu)d\tau = P_0(\tau) \cdot a_\mu d\tau$. I think this differential form is taken to ensure that the two variables can be separately expressed. $P_0(\tau)$ is the probability that no reaction will take place in the time interval (t,t+ τ). The term $a_\mu d\tau$ is the probability that reaction type μ will occur at the infinitesimal interval $d\tau$. I believe the joint probability can be analytically expressed as a product of these terms if the independence of these two events is assumed.

The definition of propensity of a reaction a_μ stems from stochastic reaction kinetics. a_μ can be written as the product of stochastic reaction constant c_μ and function h_μ, equivalent to the reaction rate constant k_μ that defines the deterministic kinetics. The formulation of function h_μ can be generalized according to the reaction in such a way: let m be the number of identical reactant molecules, X to be the population (total number of molecules) for that reactant, then it can be defined such that :$h_\mu = \frac{X(X-1)(X-2)\ldots\ldots(X-m+1)}{m!}$. It is also interesting to see as m increases, the constant factor that stochastic approach differs from deterministic approach grows larger, and stochastic simulation deviates more from deterministic models in practical sense. Together $a_\mu d\tau = c_\mu h_\mu d\tau$ gives the probabilistic description for which reaction to occur at the next reaction time interval.

The only term left to be expressed is $P_0(\tau)$. The probability that no reaction occur before t+ τ, can be thought as the complement to the probability that all reactions take place during the interval $d\tau$, which is simply $(1 - \sum_{v=1}^{M} a_v d\tau)$. From that it can derived that $P_0(\tau)$ follows the exponential distribution where $P_0(\tau) = \exp(-\sum_{v=1}^{M} a_v)$. Thus finally the joint probability can be rewritten as: $P(\tau, \mu) = a_\mu \exp(-a_0\tau)$, where a_0 is the sum of all propensities. Again, it is worth noting that the dependence on all reaction constant c_j $(j = 1,2,3 \ldots)$ and populations of all species X_i $(i = 1,2,3 \ldots)$ confirms the theoretical backbone that drives this probabilistic expression for stochastic approach to reactions.

To implement this stochastic approach, a randomly generated number-pair (μ, τ) is required which satisfies the prescribed joint probability density function. Splitting the joint probability, $P(\tau, \mu) = P(\tau) \cdot P(\mu)$, and $P(\tau) = a_0 \exp(-a_0\tau)$, $P(\mu) = \frac{a_\mu}{a_0}$. Now, a pair of r1 and r2 can be drawn from a uniform random number generator so that r1 can be used to satisfy $P(\tau)$, and r2 satisfies $P(\mu)$. Thus the next reaction time will be expressed as a random variable satisfying its own probability density function: $\tau = \frac{1}{a_0} \ln(\frac{1}{r_1})$, and propensity a_μ becomes $r_2 a_0$.

The algorithm itself will be based on the probability density functions, and thereby choosing the next reaction and increment the time and reaction counter forward, repeating all the steps until time or reaction counter reaches a set point.

All the above findings will be used to explain some of the below questions.

Q1(a):

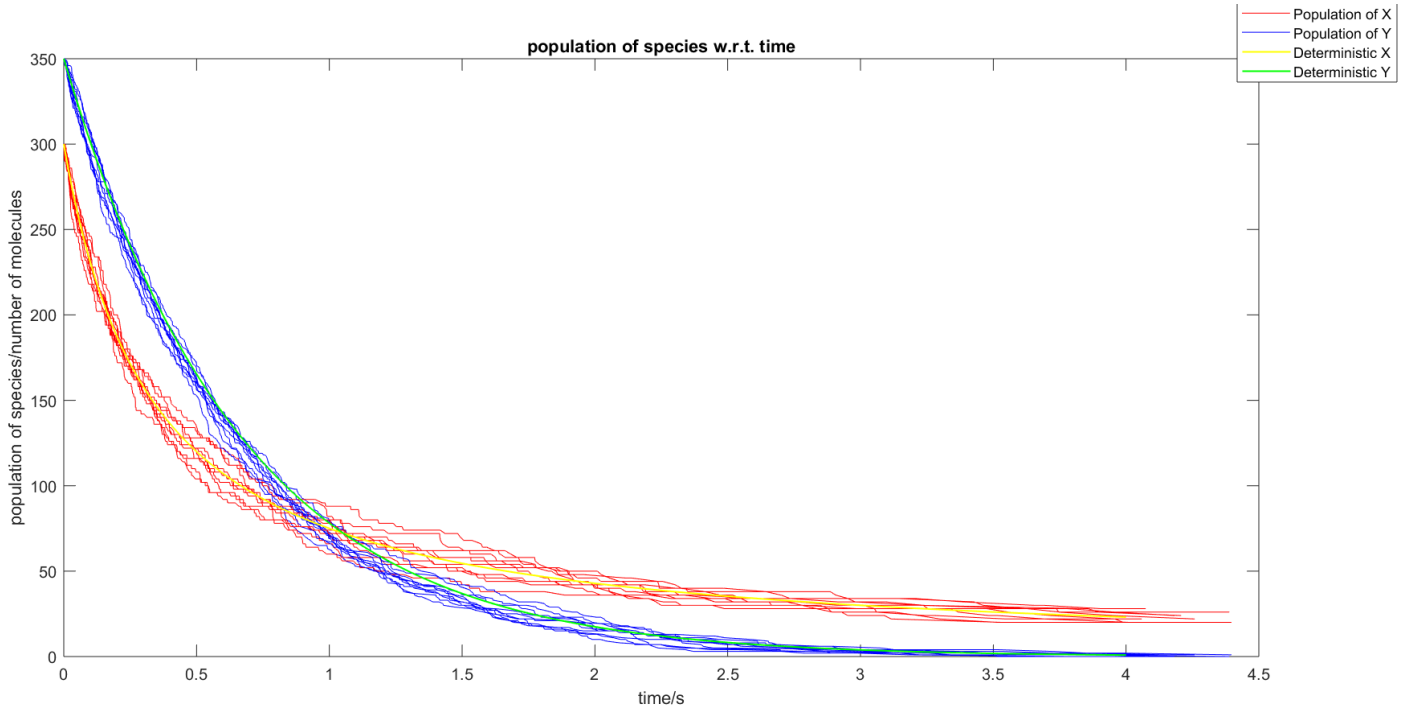

Fig.1. 10 realisations of the system, with deterministic solutions underpinning

As can be seen from figure 1, the deterministic equations successfully capture the general decaying trend of the system of chemical reactions, however there are some quantitative deviation from the actual realisation. Out of the 10 realisations, the green line (deterministic Y) only manage to overlap roughly one or two blue lines (actual stochastic Y). The deviation is more obvious for the red (X). It can be seen that at the time interval t=0.5s to t=2.5s the yellow line (deterministic X) grossly misrepresent the actual reaction, at some point during 1 to 1.5s there is even a population deviation as high as around 20 molecules. This deviation can be explained by the following: chemical reactions take place when collision between molecules occur and such collision take place randomly in a system under thermal equilibrium. The simple deterministic solution from reaction-rate ODE formulated assumes reactions are a continuous and deterministic process. Thus the inability to take the stochastic and discrete nature of chemical reaction system into account causes random deviation of the realisations to the deterministic lines.

It is interesting to note that yellow line (deterministic X) is less capable at capturing the system behaviour than the green. This phenomenon could be interpreted as the reaction process for X is inherently more random than that for Y. There could be many reasons behind this: (1) X exhibits more randomness than Y as the initial population for X is 300, less than the 350 for Y. The smaller population X is more susceptible to stochastic event like this chemical reaction which is modelled to be stochastic and discrete. (2) The reaction for X involves two identical reactant molecules of X, while for Y there is only one (simple isomerization). Using the stochastic kinetics formulation discussed above, the function h that defines the stochastic reaction for X and Y can be derived: $h_x = \frac{(X)(X-1)}{2!}, and\ h_y = \frac{Y}{1!}$, as number of identical reactant molecules $m_x = 2, m_y = 1$. In general, the stochastic formulation is

larger than the deterministic reaction rate constant k_μ roughly by a factor of (m_μ!): $k_\mu \approx \frac{h_\mu c_\mu}{m_\mu!}$, where c_μ is the stochastic reaction constant. Therefore, in this case, the deviation of stochastic simulation of reaction for X from the deterministic solution for X is at least twice as much as the deviation of that for Y.

Q1(b):

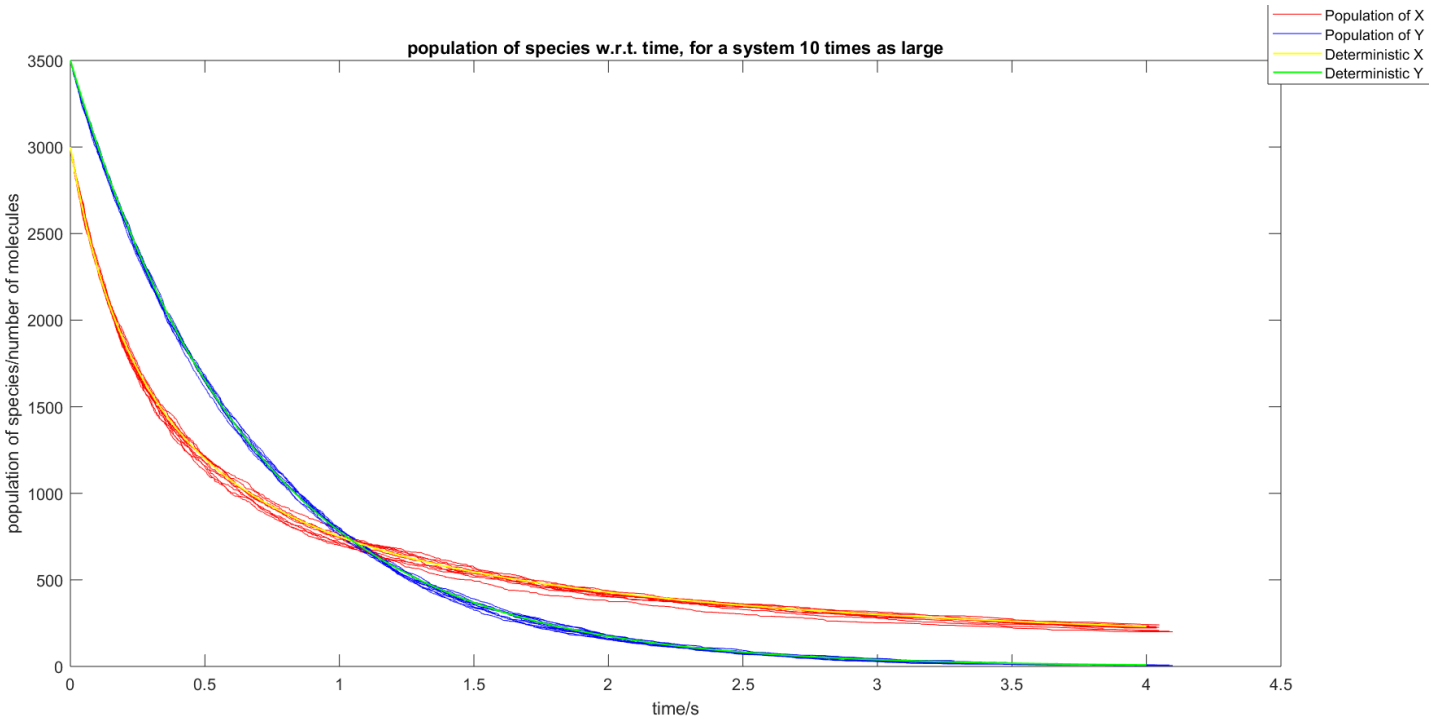

Fig.2. 10 realisations of the enlarged system,
with deterministic solutions underpinning

As can be seen from Fig.1 and 2, deterministic solutions capture the 10-time larger system better than that of the previous system, as the realisations deviate less from the green and yellow deterministic lines now. The deviation of red (population of X) from the deterministic solution remains greater than that of blue (population of Y), as the same happens in the previous system in figure 1.

Again, this shows the fact that larger system is less susceptible to random changes, thus deterministic solutions capture system behaviour better. If X and Y are to be picked out as separate systems, Y with its initial population 3500 greater than X's 3000 is the larger system, thus deterministic solution fits Y better.

Q2(a):

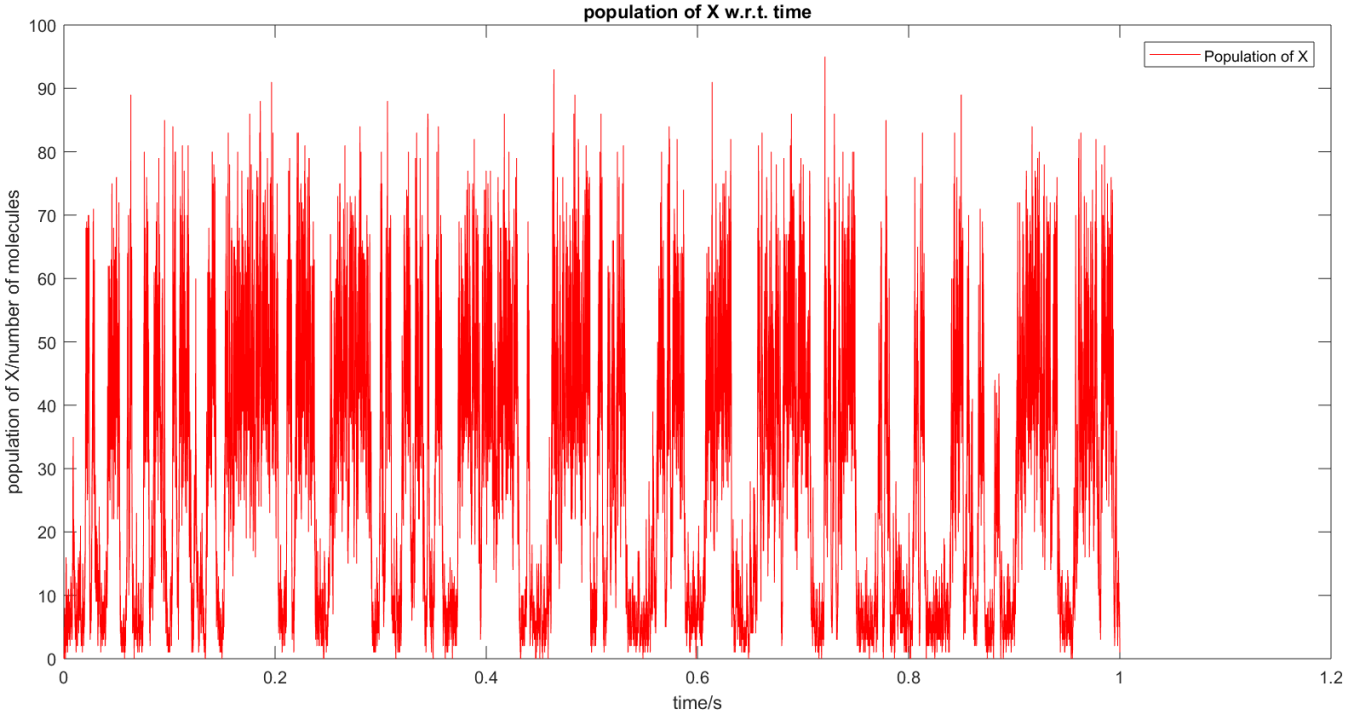

Fig.3. 1st realisation of the system in Q2

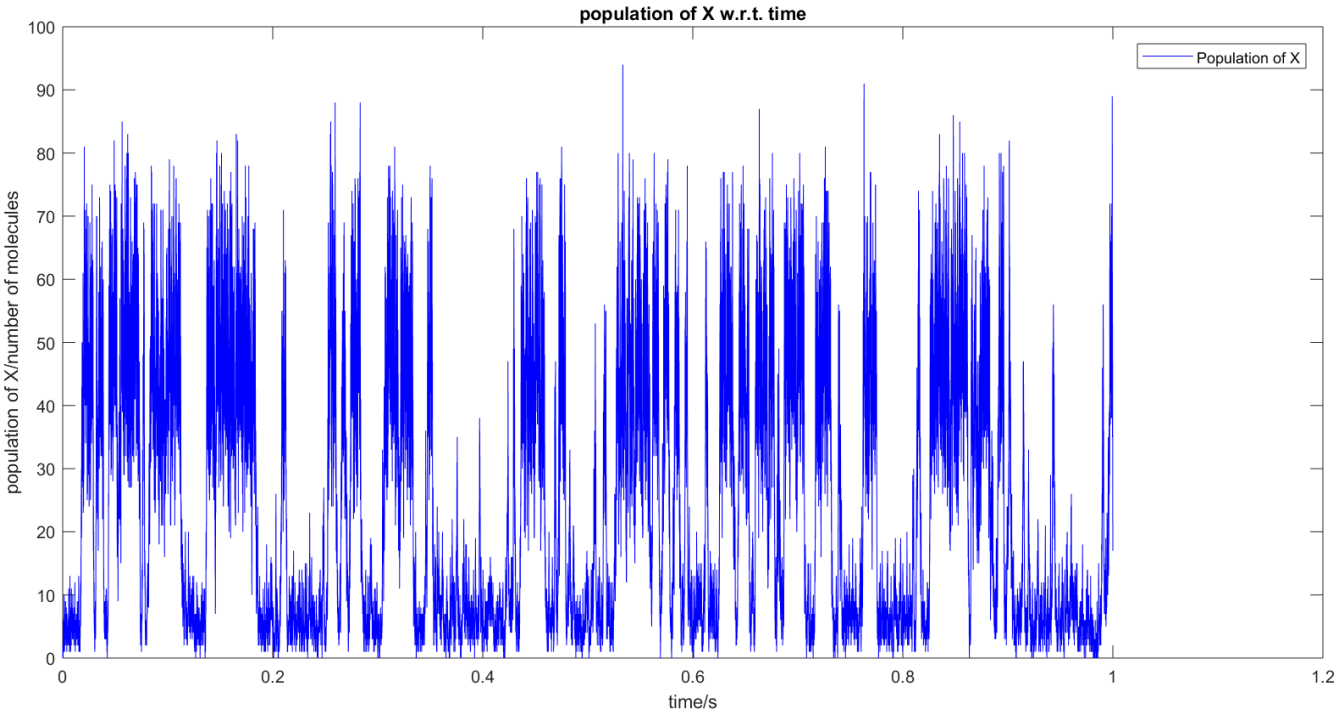

Fig.4. 2nd realisation for system in Q2

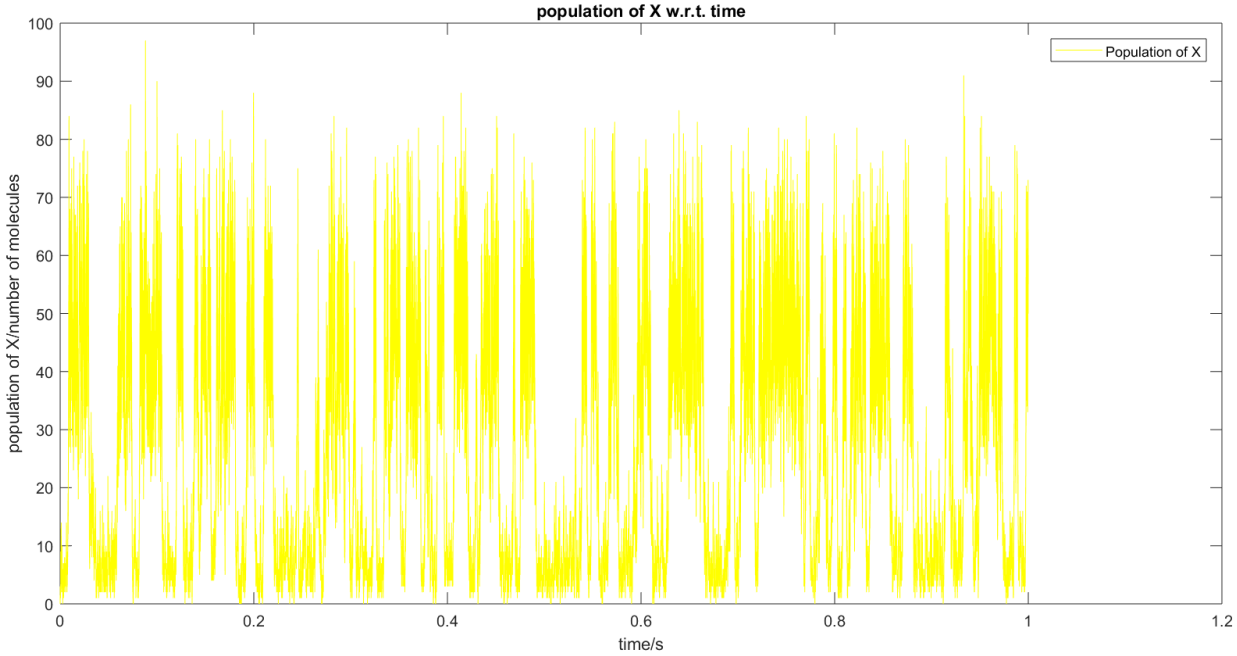

Fig.5. 3rd realisation of system in Q2

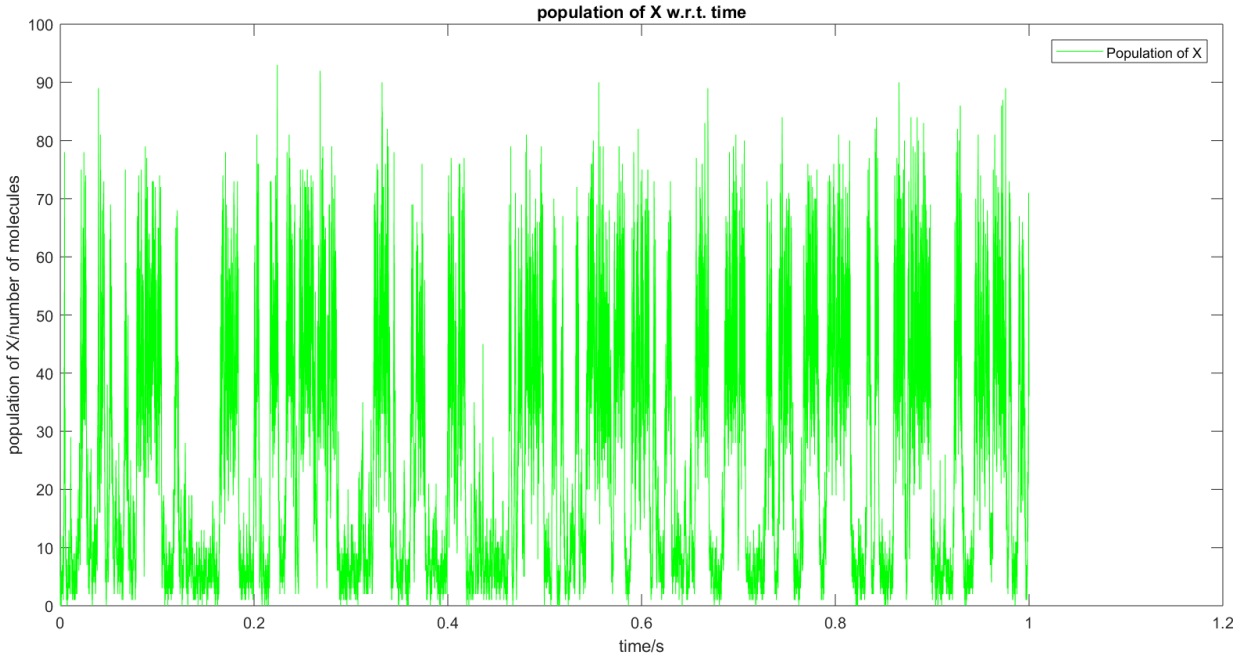

Fig.6. 4th realisation of system in Q2

The population-time plots from figure 3 to 6 demonstrates four different time responses, yet one unchanging theme throughout the randomness still can be picked out: from the starting point of zero, all systems quickly develop into fierce oscillations about some positive set points, frequency of each oscillation is very high, causing a severe overlapping of lines when visualised on plot. Interestingly,

majority of oscillations for all four realisations occur about two constant set points: one is at population level of about 5 molecules, and the other is about 45 molecules. These values can be visually confirmed across figure 3 to 6, by the two clusters of heavy overlapping sections, one concentrates along the line X=5, the other along X=45. Particularly, along X=5 there are slightly more overlapping occurring than along X=45.

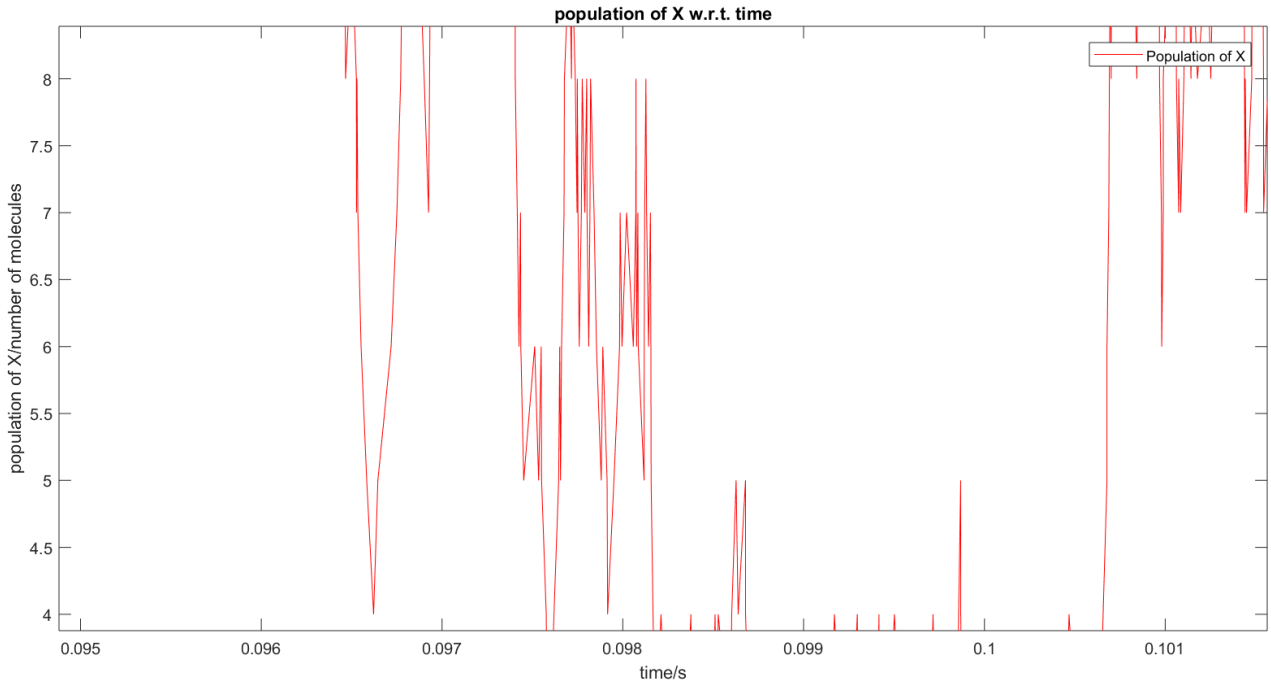

Fig.7. Zoom in of one overlapping section of the 1st realisation plot in fig.3.

Take the 1st realisation in figure 3 as an example, it can be clearly seen that oscillations appear to be sharp and rapid, and sometimes each oscillation spans across multiple molecule number levels along the y-axis, even though each possible reaction can only change the molecule number by 1. However, a closer look at the lines can answer this seemingly contradicting observation. Zooming in and focusing on the big oscillation appearing between t=0.096s and t=0.097s, from figure 7, it is finally clear that the big oscillation amplitude actually consists of many line segments, each representing a reaction and its time τ_μ. When the reaction time τ_μ for successive reactions are close, and they all increase the molecule number by one, they combine and in effect create a "straight" upward line that if viewed in low resolution, appears to be a single big oscillation. These findings imply that for this system of reactions, the next reaction time τ is usually very small.

The behaviour of this set of chemical reactions can be further understood by considering the Brusselator example model covered in the paper[2]. The Brusselator model using this stochastic algorithm can create positively "stable" oscillations (unlike Lotka's "stable about zero" behaviour), regardless of the starting position. And from the viewpoint of chemical system, the Brusselator essentially consists of two intermediate species whose population could either increase by 1 molecule or decrease by 1 molecule for each occurrence of reaction through a set of four coupled reactions. Examining the set of reactions for system in Q2, there also exists one intermediate X whose population can be increased by 1 through the 1st reaction with stochastic constant c1, or decreased by 1 through c2 reaction, or increased by 1 through the forward reaction c3, or decreased by 1 through the

backward reaction c4. In short, I believe these reactions in Q2 will also form a system that acts like an oscillator similar to the Brusselator model. Therefore the behaviour displayed in four realisations matches the experimental results for the Brusselator model in Gillespie's paper and makes sense.

Q2(b)

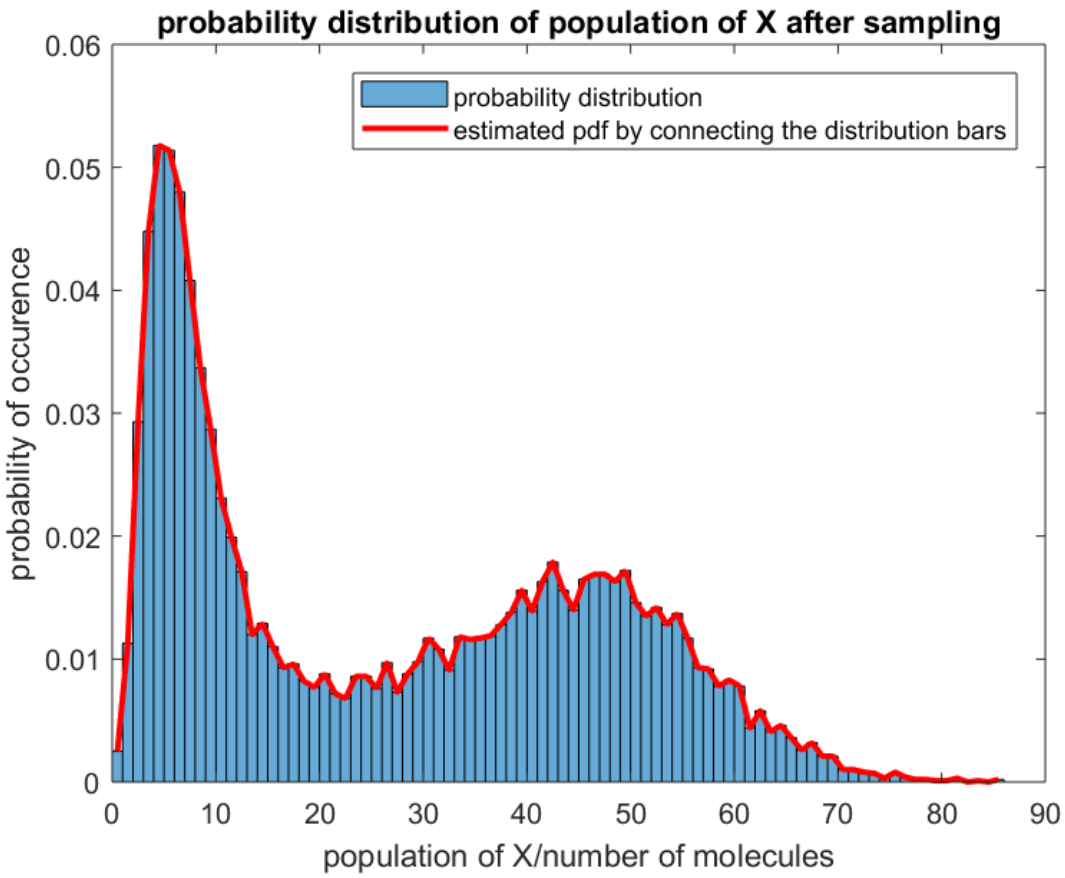

Fig.8. Distribution of sampled population of X

The distribution plot on figure 8 exhibits two general peaks, one at about X=5 and the other vaguely in between X=40 and X=50. This observation can be immediately explained by the population-time plot obtained previously, in which the system oscillates around two set points: X=5 and X=45. For a population value, the more the system oscillates around it, the more probability that population value holds. The peak at X=5 is higher as more overlapping occurs along X=5 in the population-time plots in figure 3-6.

The general trend of this distribution with two separate peaks prompts a reasonable hypothesis: the distribution is generated by a superposition of two or even more distributions. As each reaction has a distinctive stochastic constant and reaction kinetics, the system in Q2 has a total of four possible reactions, each may contribute to a probability distribution of the population. Therefore when plotting the system distribution, it may be viewed as all individual reaction's contribution superposing into one mixed distribution, which may have features from individual distribution.

Q2(c)

As the population data is sampled with a sampling time step dt=0.001s, data from Q(2)b constitutes a large sample with sample size N equal to 10002 (obtained from length(X_pop), see appendix B). However it is worth noting that this is not the actual population data, as the real population data should be obtained from data points with time increment t=t+τ, instead of the sampling time increment t=t+dt.

(i) The sample mean $\bar{x}$ can be calculated to be 26.2070 (number of molecules). The population mean μ can be estimated from the sample mean.
(ii) The population variance s^2 can be estimated from the sample variance σ^2through the unbiased estimation: $s^2 = \frac{N}{N-1}\sigma^2$. The population standard deviation s is calculated to be s=20.0479 (number of molecules).
Now, given this is a large enough sample, suffice to say that a confidence interval for estimating the population mean μ can be established, without the prior knowledge of the population distribution and population variance. Therefore, a 95% confidence interval can be calculated: $\left[\bar{x} - z\frac{s}{\sqrt{N}}, \bar{x} + z\frac{s}{\sqrt{N}}\right]$, where z=1.96 for a 95% confidence level. The confidence interval that the population mean μ lies within is: [25.814,26.600].

(iii) The skewness k is calculated to be k=0.3605

From figure 8, according to the probability distribution plot, the mode of population data should be X=5, as indicated by the highest probability. From figure 3 to 6, the original population plot, the population level that most overlapping sections cluster along is X=5, thus the mode should also be 5. However, mean calculated above takes value between 25.814 and 26.600, which is much higher than the mode. At X=26 in the probability plot the probability value is only about 0.01, much less than the peak value of more than 0.05, and on the population-time plots there are few overlapping at X=26. All these observations lead to the conclusion that the estimated population mean is not a good representative of the population.

The estimated standard deviation is quite high at 20.0479, considering the mean is only around 26. But this actually makes sense because the mean cannot represent the population, most data points are either at X=5, or within the interval from X=40 to 50. Both cases are quite far from X=26, therefore explaining the high standard deviation.

The skewness k is calculated to be a small but positive number 0.3605. Consider the mean μ is around 26, greater than the mode which is 5, and slightly more data points lies on the left of the mean than the right (only slight more as one must consider that although peak at X=5 left of the mean is high, it is also sharp, while at X=40-50, right of the mean, the peak is lower but much broader). The positive but small nature of the skewness does make sense.

Reference:
[1],[2]: Gillespie, D. T. (1977). *"Exact Stochastic Simulation of Coupled Chemical Reactions"*. The Journal of Physical Chemistry.

Appendix A

This appendix contains all matlab codes for Q1.

```
%set the number of kinetic constants m and number of species n
m=2;
n=2;
%initialize Cj and Xi:
c=[];
X=[];
for i=[1:m]
    c(i)=0;
end
for i=[1:n]
    X(i)=0;
end
%populate the c and X according to Q1:
c(1)=0.01;
c(2)=1.5;
X(1)=300;
X(2)=350;
%specify the tmax and nmax according to Q1:
tmax=4;
nmax=100000;
%initialize t and cnt:
t=0;
cnt=0;
%set a 10-seed list for MT19937 PRNG:
seeds=[9:100:9+100*9];
%define the propensity functions for Q1:
H1=@(x) 0.5*(x-1)*x;

%the following codes are for Q1(a)
%begin the main loop:
for i=seeds
    s=RandStream('mt19937ar','Seed',i);
    RandStream.setGlobalStream(s);       %initialize the PRNG
    a=[];                                %initialize the propensity
    c(1)=0.01;
    c(2)=1.5;
    X(1)=300;
    X(2)=350;
    X_pop=[X(1)];
    Y_pop=[X(2)];
    t=0;
    t_list=[0];
    cnt=0;
    while (t<tmax) && (cnt<nmax)
        a(1)=c(1)*H1(X(1));
        a(2)=c(2)*X(2);
        a0=sum(a);
        r1=rand(s);
        r2=rand(s);
        tau=(1/a0)*log(1/r1);
        u_list=[1:length(a)];    %initialize all possible u values
        for u=u_list
            sum1=0;
            sum2=0;
            for j=[1:u]
                sum1=sum1+a(j);
```

```
            end
            for j=[1:u-1]
                sum2=sum2+a(j);
            end
            if (r2*a0<=sum1) && (r2*a0>sum2)
                u_react=u
                break
            end
        end
        %these conditionals only apply to Q1 reactions:
        if u_react==1
            X(1)=X(1)-2;
        elseif u_react==2
            X(2)=X(2)-1;
        end
        %update the time and reaction times
        t=t+tau;
        cnt=cnt+1;
        X_pop(end+1)=X(1);
        Y_pop(end+1)=X(2);
        t_list(end+1)=t;
    end
    figure(1)
    title('population of species w.r.t. time')
    xlabel('time/s')
    ylabel('population of species/number of molecules')
    plot(t_list,X_pop,'r','DisplayName','Population of X')
    hold on
    plot(t_list,Y_pop,'b','DisplayName','Population of Y')
end
%define and plot the deterministic expression for X and Y population:
X_det=@(x) 1./(1/300+0.01.*x);
Y_det=@(y) 350.*exp(-1.5.*y);
time=[0:0.01:4];
plot(time,X_det(time),'y','DisplayName','Deterministic X','LineWidth',1)
plot(time,Y_det(time),'g','DisplayName','Deterministic Y','LineWidth',1)
legend('-DynamicLegend')
hold off

%the following codes are for Q1(b)
%loop again for a different set of parameters:
for i=seeds
    s=RandStream('mt19937ar','Seed',i);
    RandStream.setGlobalStream(s);        %initialize the PRNG
    a=[];                                 %initialize the propensity
    c(1)=0.001;
    c(2)=1.5;
    X(1)=3000;
    X(2)=3500;
    X_pop=[X(1)];
    Y_pop=[X(2)];
    t=0;
    t_list=[0];
    cnt=0;
    while (t<tmax) && (cnt<nmax)
        a(1)=c(1)*H1(X(1));
        a(2)=c(2)*X(2);
        a0=sum(a);
        r1=rand(s);
        r2=rand(s);
        tau=(1/a0)*log(1/r1);
```

```
        u_list=[1:length(a)];    %initialize all possible u values
        for u=u_list
            sum1=0;
            sum2=0;
            for j=[1:u]
                sum1=sum1+a(j);
            end
            for j=[1:u-1]
                sum2=sum2+a(j);
            end
            if (r2*a0<=sum1) && (r2*a0>sum2)
                u_react=u
                break
            end
        end
        %these conditionals only apply to Q1 reactions:
        if u_react==1
            X(1)=X(1)-2;
        elseif u_react==2
            X(2)=X(2)-1;
        end
        %update the time and reaction times
        t=t+tau;
        cnt=cnt+1;
        X_pop(end+1)=X(1);
        Y_pop(end+1)=X(2);
        t_list(end+1)=t;
    end
    figure(2)
    title('population of species w.r.t. time, for a system 10 times as
large')
    xlabel('time/s')
    ylabel('population of species/number of molecules')
    plot(t_list,X_pop,'r','DisplayName','Population of X')
    hold on
    plot(t_list,Y_pop,'b','DisplayName','Population of Y')
end
%define and plot the deterministic expression for X and Y population:
X_det_new=@(x) 1./(1/3000+0.001.*x);
Y_det_new=@(y) 3500.*exp(-1.5.*y);
time=[0:0.01:4];
plot(time,X_det_new(time),'y','DisplayName','Deterministic
X','LineWidth',1)
plot(time,Y_det_new(time),'g','DisplayName','Deterministic
Y','LineWidth',1)
legend('-DynamicLegend')
hold off
```

Appendix B

This appendix contains all matlab codes for Q2.

```
%set the number of kinetic constants m and number of species n for Q2:
m=4;
n=1;
%initialize Cj and Xi:
c=[];
X=[];
for i=[1:m]
    c(i)=0;
end
for i=[1:n]
    X(i)=0;
end
%initialize the propensity functions according to Q2:
H3=@(x) 0.5*(x-1)*x;
H4=@(y) (1/6)*y*(y-1)*(y-2);
%specify the tmax and nmax according to Q2:
tmax=1;
nmax=1e+08;
%set a 4-seed list for MT19937 PRNG:
seeds=[9:100:9+100*3];

%the following codes are for Q2(a)
col=['r','b','y','g'];
counter=0;
for i=seeds
    counter=counter+1;
    s=RandStream('mt19937ar','Seed',i);
    RandStream.setGlobalStream(s);       %initialize the PRNG
    a=[];                                %initialize the propensity
    c(1)=2.168e+04;
    c(2)=4.95e+03;
    c(3)=602;
    c(4)=25.85;
    X(1)=0;
    X_pop=[X(1)];
    t_list=[0];
    t=0;
    cnt=0;
    while (t<tmax) && (cnt<nmax)
        a(1)=c(1);
        a(2)=c(2)*X(1);
        a(3)=c(3)*H3(X(1));
        a(4)=c(4)*H4(X(1));
        a0=sum(a);
        r1=rand(s);
        r2=rand(s);
        tau=(1/a0)*log(1/r1);
        u_list=[1:length(a)];    %initialize all possible u values
        for u=u_list
            sum1=0;
            sum2=0;
            for j=[1:u]
                sum1=sum1+a(j);
            end
            for j=[1:u-1]
```

```
                sum2=sum2+a(j);
            end
            if (r2*a0<=sum1) && (r2*a0>sum2)
                u_react=u
                break
            end
        end
        %these conditionals only apply to Q2 reactions:
        if u_react==1
            X(1)=X(1)+1;
        elseif u_react==2
            X(1)=X(1)-1;
        elseif u_react==3
            X(1)=X(1)+1;
        elseif u_react==4
            X(1)=X(1)-1;
        end
        %update the time and reaction times
        t=t+tau;
        cnt=cnt+1;
        X_pop(end+1)=X(1);
        t_list(end+1)=t;
    end
    figure(counter)
    plot(t_list,X_pop,col(counter),'DisplayName','Population of X')
    title('population of X w.r.t. time')
    xlabel('time/s')
    ylabel('population of X/number of molecules')
    legend('Population of X')
end

%the following codes are for Q2(b)
seeds=[1999];
tmax=10;
for i=seeds
    s=RandStream('mt19937ar','Seed',i);
    RandStream.setGlobalStream(s);        %initialize the PRNG
    a=[];                                 %initialize the propensity
    c(1)=2.168e+04;
    c(2)=4.95e+03;
    c(3)=602;
    c(4)=25.85;
    X=[];
    X(1)=0;
    X_pop=[X(1)];
    t=0;
    cnt=0;
    tsample=0;
    dt=0.001;
    while (t<tmax) && (cnt<nmax)
        a(1)=c(1);
        a(2)=c(2)*X(1);
        a(3)=c(3)*H3(X(1));
        a(4)=c(4)*H4(X(1));
        a0=sum(a);
        r1=rand(s);
        r2=rand(s);
        tau=(1/a0)*log(1/r1);
        u_list=[1:length(a)];    %initialize all possible u values
        for u=u_list
            sum1=0;
```

```
            sum2=0;
            for j=[1:u]
                sum1=sum1+a(j);
            end
            for j=[1:u-1]
                sum2=sum2+a(j);
            end
            if (r2*a0<=sum1) && (r2*a0>sum2)
                u_react=u
                break
            end
        end
        %these conditionals only apply to Q2 reactions:
        if u_react==1
            X(1)=X(1)+1;
        elseif u_react==2
            X(1)=X(1)-1;
        elseif u_react==3
            X(1)=X(1)+1;
        elseif u_react==4
            X(1)=X(1)-1;
        end
        %update the time and reaction times
        t=t+tau;
        cnt=cnt+1;
        %sample over time t and record corresponding populations
        while tsample<t
            X_pop(end+1)=X(1);
            tsample=tsample+dt;
        end
    end
    figure(5)
    histogram(X_pop,'BinWidth',1,'Normalization','probability')
    hold on
    [N,edges] =
histcounts(X_pop,'BinWidth',1,'Normalization','probability');
    edges = edges(2:end) - (edges(2)-edges(1))/2;      %to manually compute
the mid point of distribution bars
    plot(edges, N,'r','LineWidth',2)
    title('probability distribution of population of X after sampling')
    xlabel('population of X/number of molecules')
    ylabel('probability of occurence')
    legend('probability distribution','estimated pdf by connecting the
distribution bars')
    hold off
end

%the following codes are for Q2(c)
%estimated mean, use sample mean as an estimation for population mean:
mean=sum(X_pop)/(length(X_pop));
%estimated standard deviation:
sum_v=0;
for i=[1:length(X_pop)]
    sum_v=sum_v+(X_pop(i)-mean)^2;
end
%use an unbiased estimator for population variance s
std=sqrt(sum_v/(length(X_pop)-1));
%estimate skewness;
sum_s=0;
for i=[1:length(X_pop)]
    sum_s=sum_s+((X_pop(i)-mean)/std)^3;
```

```
end
K=length(X_pop)/((length(X_pop)-1)*(length(X_pop)-2))*sum_s;
```