

Department of Mathematics and Statistics

Prof. Joseph Mahaffy

Math 636, Mathematical Modeling

Homework Assignment

Stochastic Modeling

by Matteo Polimeno

12/5/2017

1. Problem 1

The transition matrix for red squirrels, gray squirrels, both or neither in that order is given by

$$T = \begin{bmatrix} 0.8797 & 0.0382 & 0.0527 & 0.0008 \\ 0.0212 & 0.8002 & 0.0041 & 0.0143 \\ 0.0981 & 0.0273 & 0.8802 & 0.0527 \\ 0.0010 & 0.1343 & 0.0630 & 0.9322 \end{bmatrix}$$

The equilibrium distribution of squirrels based on this transition matrix is given by the normalized eigenvector associated with the eigenvalue $\lambda = 1$, which is found to be

$$x_e = \begin{bmatrix} 0.1705 \\ 0.0560 \\ 0.3421 \\ 0.4313 \end{bmatrix}$$

This eigenvector shows that the predicted squirrel community for the given region of the Great Britain should be approximately 17% red squirrels, 6% grey squirrels, 34% both and 43% neither. Therefore the native red squirrel, which starts off as being 87% of the total squirrel population, will end up being 17%, while the grey squirrel goes from 2% to 6%. Interestingly, the total population at the equilibrium will be made of 43% different species of squirrels, whereas at $t = 0$ those were only 0.1% of the total.

Therefore, this model seems to suggest that the invasive grey species will displace the native red squirrel, however the latter will not go extinct nor will be outnumbered by the invasive grey squirrel, as we can see that, there is going to be a higher overall percentage of red

squirrels anyway at the equilibrium distribution.

2. Problem 2

2.1 Part a

From the transition model given, knowing that 100 frogs were released in the first habitat at $t = 0$, we find the expected distribution of the tagged frogs after 1,2,5 and 10 days to be, respectively

$$D_1 = \begin{bmatrix} 42 \\ 7 \\ 34 \\ 17 \end{bmatrix}, D_2 = \begin{bmatrix} 27.94 \\ 15.97 \\ 37.54 \\ 18.55 \end{bmatrix}, D_5 = \begin{bmatrix} 23.1737 \\ 20.6571 \\ 35.6359 \\ 20.5333 \end{bmatrix}, D_{10} = \begin{bmatrix} 23.0600 \\ 20.6914 \\ 35.4725 \\ 20.7762 \end{bmatrix}$$

2.2 Part b

The expected distribution after a long period of time is given by the following normalized eigenvector of the eigenvalue $\lambda = 1$

$$D_{long} = \begin{bmatrix} 23.06 \\ 20.69 \\ 35.47 \\ 20.78 \end{bmatrix},$$

which means that, after a long period of time, about 23% of the frogs will be living in the first habitat, around 20.7% will be leaving on the second habitat, a little over 35% will be living in the third habitat and around 20.8% will be leaving in the fourth habitat. Now, as we started with 100 frogs and obviously we can't have fractionary animals, we can approximate and say that, according to our simulation, 23 frogs will be living in the first habitat, 21 in the second habitat, 35 in the third and 21 in the fourth.

At a first glance, just by looking at the distribution above, it seems to be that the third habitat is the most suitable for the frogs, as a higher percentage of them will tend to move over there as time goes on, whereas the fourth and second habitats seem to be least suitable as they are the least populated by the frogs after a long period of time.

However, it seems reasonable to conclude that the least suitable habitat for the frogs is the first one. In fact, it's the one from which most frogs leave as time proceeds, going from 42 after the first day to around 23 after a long period of time, which seems to suggest that frogs

do not find such habitat to be suitable for them and tend to leave from it. Whereas, the second and the fourth habitat increase their number of frogs up until roughly 21, and then keep it that way as time goes on.

Also, the third habitat seems to be the most stable one in terms of distribution of frogs, which seems to oscillate between 37 and 35 of the total.

If we compare these results with our observations in Part a, we can see that the trend was clear already after the first day, and in fact the distribution of frogs among the habitats found after a long period of time is very much similar to the distribution found after 10 days.

2.3 Part c

Below is the Matlab code implemented to run a MonteCarlo simulation of this experiment run 1000 times.

```
co%Monte Carlo Simulation for Problem 2, Part c

%Distribution matrix for the frogs population (not necessary, but listed to
%see what data we are working with)
bratakoi = [0.42 0.16 0.19 0.16;
0.07 0.38 0.24 0.13;
0.34 0.19 0.51 0.27;
0.17 0.27 0.06 0.44];

count1 = zeros(10,1000); %initializing frogs count
count2 = zeros(10,1000);
count3 = zeros(10,1000);
count4 = zeros(10,1000);

btk = ones(1000,100,10); %initializing and setting all 100 frogs in
%habitat 1 for 10 days and 1000 simulations

for k = 1:10 %number of days
for j = 1:1000 %number of simulation
for i = 1:100 %each individual frog goes through loop
if btk(j,i,k) == 1 %if frog is in habitat 1, function hbt_1
btk(j,i,k) = hbt_1; %all frogs released in habitat 1
%go through this function when k=1
elseif btk(j,i,k) == 2 %if frog is in habitat 2, function hbt_2
btk(j,i,k) = hbt_2; %kicks in when k=2
elseif btk(j,i,k) == 3
btk(j,i,k) = hbt_3; %same
else
btk(j,i,k) = hbt_4; %same
end

if btk(j,i,k) == 1 %counts # of frogs that ends up in which habitat
count1(k,j) = count1(k,j)+1;
elseif btk(j,i,k) == 2
count2(k,j) = count2(k,j)+1;
elseif btk(j,i,k) == 3
count3(k,j) = count3(k,j)+1;
else
count4(k,j) = count4(k,j)+1;
end
end
end
btk(:,:,k+1) = btk(:,:,k); %sets the next day simulation to start with previous
end

%array btk ends up being 10 arrays, one per day, of where the 100 frogs end
```

```
%up. Individual frogs are the columns, number of simulations are the rows.

mean1 = [mean(count1(1,:)) mean(count2(1,:))... %Distribution vector
mean(count3(1,:)) mean(count4(1,:))]' /100; %with mean of frogs per habitat of day 1.

std1 = [std(count1(1,:)) std(count2(1,:))... %Std deviation
std(count3(1,:)) std(count4(1,:))]' /100; %of frogs per habitat of day 1.

mean2 = [mean(count1(2,:)) mean(count2(2,:))... %Same for day 2.
mean(count3(2,:)) mean(count4(2,:))]' /100;
std2 = [std(count1(2,:)) std(count2(2,:))...
std(count3(2,:)) std(count4(2,:))]' /100;

mean3 = [mean(count1(5,:)) mean(count2(5,:))... %Day 5.
mean(count3(5,:)) mean(count4(5,:))]' /100;
std3 = [std(count1(5,:)) std(count2(5,:))...
std(count3(5,:)) std(count4(5,:))]' /100;

mean4 = [mean(count1(10,:)) mean(count2(10,:))... %Day 10.
mean(count3(10,:)) mean(count4(10,:))]' /100;
std4 = [std(count1(10,:)) std(count2(10,:))...
std(count3(10,:)) std(count4(10,:))]' /100;
```

The following are the means and standards deviation for the distribution of the frogs after 1,2,5 and 10 days respectively

$$\mu_1 = \begin{bmatrix} 43.0870 \\ 6.8770 \\ 33.9890 \\ 16.0470 \end{bmatrix}, \sigma_1 = \begin{bmatrix} 4.8260 \\ 2.5535 \\ 4.6539 \\ 3.7639 \end{bmatrix},$$

$$\mu_2 = \begin{bmatrix} 29.0390 \\ 15.9640 \\ 37.6650 \\ 17.3320 \end{bmatrix}, \sigma_2 = \begin{bmatrix} 4.6044 \\ 3.5422 \\ 4.9080 \\ 3.8197 \end{bmatrix},$$

$$\mu_5 = \begin{bmatrix} 24.3480 \\ 20.6410 \\ 35.8830 \\ 19.1280 \end{bmatrix}, \sigma_5 = \begin{bmatrix} 4.2866 \\ 4.1409 \\ 4.9762 \\ 3.8772 \end{bmatrix},$$

$$\mu_{10} = \begin{bmatrix} 24.4670 \\ 20.5220 \\ 35.7660 \\ 19.2450 \end{bmatrix}, \sigma_{10} = \begin{bmatrix} 4.4979 \\ 4.0007 \\ 4.8298 \\ 3.7645 \end{bmatrix}.$$

As we can see, the mean values obtained with the Monte Carlo simulation are all within the

standard deviation, if compared to our previous results from Part a.

3. Problem 3

3.1 Part a

P_n is a vector that varies discretely with time. Let L be our Leslie Matrix with $P_n = [P_1, P_2, P_3, P_4]^T_n$ the population vector at time n divided in 4 age classes (0-1,1-2,2-3,3-4). Then the Leslie Model is

$$P_{n+1} = LP_n,$$

where L is our Leslie matrix defined as

$$L = \begin{bmatrix} 0 & 1.5 & 2.2 & 3.4 \\ 0.4 & 0 & 0 & 0 \\ 0 & 0.7 & 0 & 0 \\ 0 & 0 & 0.75 & 0 \end{bmatrix}. \quad (3.1)$$

To find the steady-state percentage for each age group with build a model and run it. The dominant eigenvalue is found to be $\lambda=1.2465$, which implies that the population growth is about 25% over a 1-year period. The associated eigenvector is found to be

$$P_e = \begin{bmatrix} 0.6213 \\ 0.1994 \\ 0.1120 \\ 0.0674 \end{bmatrix}, \quad (3.2)$$

this shows that the steady-state distribution should be approximately 62% in the 0-1 age group, about 20% in the 1-2 age group, about 11% in the 2-3 age group and finally a little less than 7% in the 3-4 age group.

The time that it takes for the population to double after reaching its steady-state distribution is found to be

$$t = 3.1460,$$

thus this population, according to this model, will double in size after roughly 3 years and two months.

3.2 Part b

The survival rates s_2 and s_4 is reduced by a fraction α , so that the survival rate of 1-2 years old is 0.7α and the survival rate of 2-3 years old is 0.75α . We compute the value of α in

MatLab and we find it to be

$$\alpha = 0.4324.$$

Then, for this value of α and 550 mature (3-4 years old) animals, we determine the total population to be

$$P_{tot} = 20066,$$

whereas the population for each age group is found to be

$$P_e = \begin{bmatrix} 12860 \\ 5144 \\ 1557 \\ 505 \end{bmatrix}. \quad (3.3)$$

As for the total number of animals harvested each year under this conditions, this is found to be roughly equal to 834.

4. Problem 4

4.1 Part a

Using the data and information given, we found the average survival rates s_{12} , s_{23} and s_{33} to be

$$s_{12} = 0.3564,$$

$$s_{23} = 0.7339 = s_{33}.$$

As for the average birth rates b_2 and b_3 we found

$$b_2 = 0.8118,$$

$$b_3 = 2.0038.$$

4.2 Part b

From the results found in Part a, we can build the following Leslie matrix

$$L = \begin{bmatrix} 0 & 0.8118 & 2.0038 \\ 0.3564 & 0 & 0 \\ 0 & 0.7339 & 0.7339 \end{bmatrix}. \quad (4.1)$$

Using this matrix we found the population for each of the age classes over the next three years and the following are results in vector form starting with the population after 1 year

$$P_1 = \begin{bmatrix} 365.2366 \\ 110.8404 \\ 173.9343 \end{bmatrix}, \quad (4.2)$$

so the population for the first age group (0-1 years old) after one year will be about 365 birds, for the second age group (1-2 years old) will be about 111 birds and for the third group (older than 2 years old) will be about 174 birds.

Now the population vector for the distribution after 2 years

$$P_2 = \begin{bmatrix} 438.5098 \\ 130.1703 \\ 208.9962 \end{bmatrix}, \quad (4.3)$$

so the population for the first age group (0-1 years old) after two years will be about 439 birds, for the second age group (1-2 years old) will be about 130 birds and for the third group (older than 2 years old) will be about 209 birds.

Now the population vector for the distribution after 3 years

$$P_3 = \begin{bmatrix} 524.4588 \\ 156.2849 \\ 248.9143 \end{bmatrix}, \quad (4.4)$$

so the population for the first age group (0-1 years old) after two years will be about 524 birds, for the second age group (1-2 years old) will be about 156 birds and for the third group (older than 2 years old) will be about 249 birds.

4.3 Part c

The eigenvalues for the model are found to be

$$\lambda_1 = -0.2303 + 0.4560i,$$

$$\lambda_2 = -0.2303 - 0.4560i,$$

$$\lambda_3 = 1.1946,$$

so we have a pair of complex conjugates (both with negative real part) and a real positive eigenvalue, which is the dominant one.

The eigenvectors for λ_1 , λ_2 and λ_3 , respectively, are found to be

$$V_1 = \begin{bmatrix} 0.7631 \\ -0.2400 - 0.4752i \\ 0.0095 + 0.3662i \end{bmatrix}, \quad V_2 = \begin{bmatrix} 0.7631 \\ -0.2400 + 0.4752i \\ 0.0095 - 0.3662i \end{bmatrix}, \quad V_3 = \begin{bmatrix} 0.8721 \\ 0.2602 \\ 0.4145 \end{bmatrix}.$$

The limiting age population in each of the age classes is found to be

$$P_e = \begin{bmatrix} 0.5638 \\ 0.1682 \\ 0.2680 \end{bmatrix}.$$

Therefore, given our dominant eigenvalue of $\lambda_3 = 1.1946$, we see that, once the population reaches this limiting distribution, the approximate annual growth rate is 19%.

To find the time that it takes for the population to double we simply compute

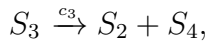
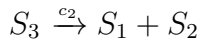
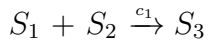
$$\frac{\log(2)}{1.1946} = 3.8985,$$

therefore the population doubles after roughly 3 years and 11 months.

5. Problem 5

5.1 Part a

Michaelis-Menten enzymes follow simple first order kinetics, i.e. the rate of the reaction depends only on one of the reactants, the substrate. If we let S_1 be the substrate, S_2 be the enzyme, S_3 be the enzyme-substrate complex and S_4 be the resultant product, then we can write a chemical reaction for the transformation of S_1 in S_4 as



where c_1 , c_2 , c_3 are the rate constants for each reaction.

Let a cell of *E.Coli* have $S_1=500$ molecules of substrate, $S_2=50$ molecules of enzyme and $S_3 = S_4 = 0$, when the reaction begins. The rate constants satisfy:

$$c_1 = 0.002, c_2 = 6 \cdot 10^{-5}, c_3 = 0.08.$$

We develop a Gillespie Algorithm for these Michaelis-Menten enzyme reactions and present 2 distinct simulations of the time series for $t \in [0, 200]$, using the same rate constants listed above. Below is the code built

```

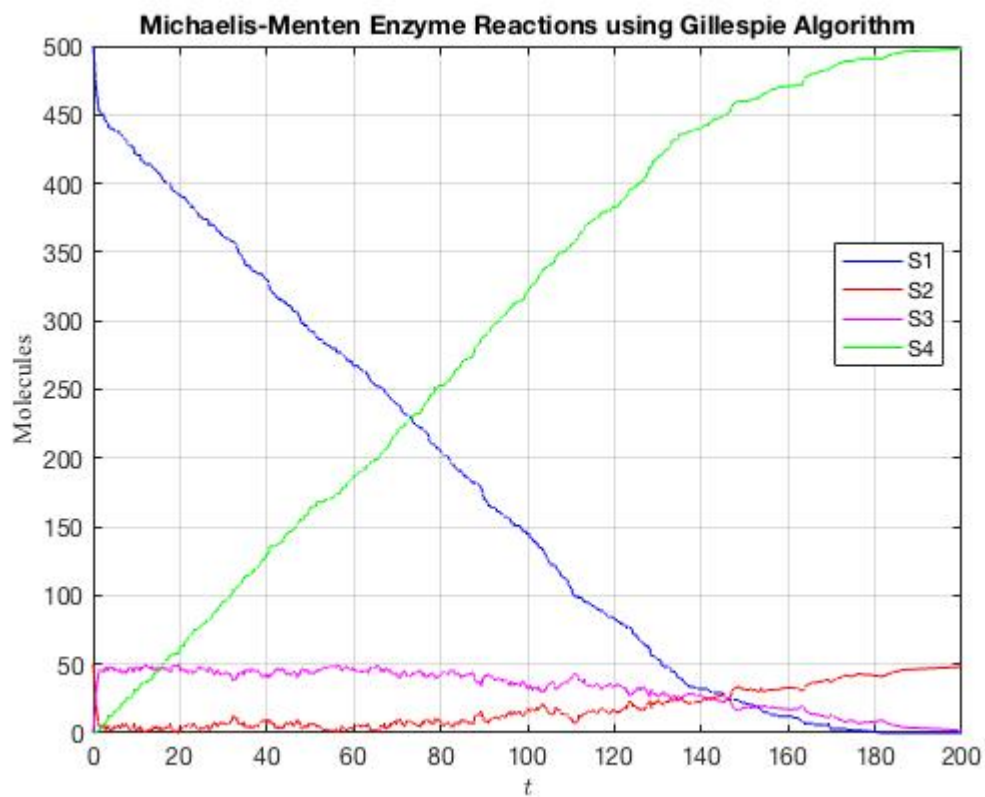
% LV_gill.M
% Simple implementation of the Stochastic Simulation Algorithm
% (or Gillespie's algorithm) for the Lotka-Volterra system.
%
% rand('state',100) % Can fix rand # pattern
% stoichiometric matrix for rxs
V = [-1 1 0;-1 1 1;1 -1 -1;0 0 1];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Parameters and Initial Conditions
X = zeros(4,1);
X(1) = 500; % initial molecules of S1 (substrate)
X(2) = 50; % initial molecules of S2 (enzyme)
X(3) = 0; % initial molecules of S3 (enzyme-substrate complex)
X(4) = 0; % initial molecule of S4 (product)

Y1(1) = X(1); % store # of molecules
Y2(1) = X(2);
Y3(1) = X(3);
Y4(1) = X(4);

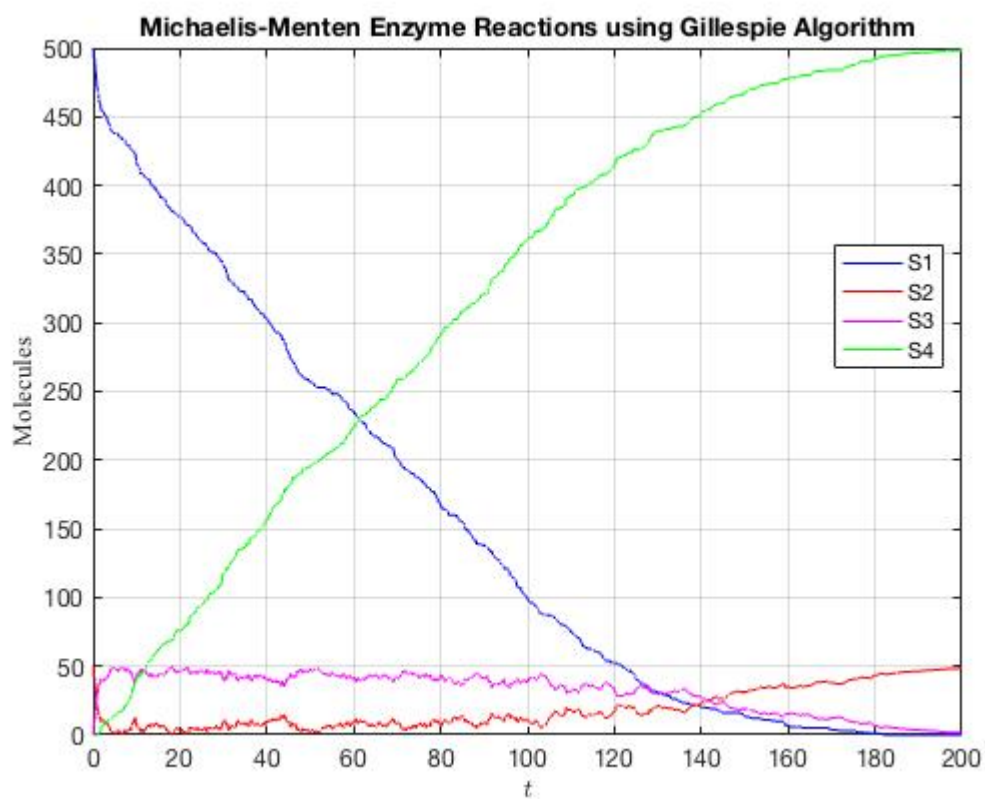
% set chem rx coefficients
c(1) = 0.002; c(2) = 0.00006; c(3) = 0.08;
t = 0; % initial time
T(1) = t;
tfinal = 200; % final time
i = 1;
while t < tfinal
% rx combination functions
a(1) = c(1)*X(1)*X(2);
a(2) = c(2)*X(3);
a(3) = c(3)*X(3);
asum = sum(a); % total a
% generate rand # and find rx occurring
j = min(find(rand<cumsum(a/asum)));
% 2nd rand # for time until rx
tau = log(1/rand)/asum;
X = X + V(:,j); % Stochastic matrix adjusts X
t = t + tau;
i = i + 1;
T(i) = t;
Y1(i) = X(1);
Y2(i) = X(2);
Y3(i) = X(3);
Y4(i) = X(4);
figure(101)
plot(T,Y1,'b-',T,Y2,'r-',T,Y3,'m-',T,Y4,'g-');grid;
xlim([0,200]);
fontlabs = 'Times New Roman'; % Font type used in labels
xlabel('$t$', 'FontSize',14,'FontName',fontlabs,...
'interpreter','latex');
ylabel('Molecules', 'FontSize',14,'FontName',fontlabs);
set(gca,'FontSize',12);
legend('S1','S2','S3','S4','location','best');
title('Michaelis-Menten Enzyme Reactions using Gillespie Algorithm');

```

The following figure shows two different simulations of the time series $t \in [0, 200]$ for all 4 chemical species



(a)



(b) 12

Figure 5.1: a) First Simulation; b) Second Simulation

In Figure 5.1a we see that the concentration of the substrate S_1 (blue line) steadily decreases as time goes on, resembling an exponential decay, from its initial value of 500 molecules at $t = 0$ to asymptotically approaching zero. The almost exact opposite "route" is followed by the product S_4 , (green line) whose concentration that starts obviously at zero molecules and steadily increases, levelling off at around 500 molecules. The concentrations of the two chemical species crosses each other at around $t = 70$.

As for the enzyme concentration S_2 (red line), it start at 50 molecules, almost instantaneously drops close to zero and seems to oscillate between zero and 10 molecules up until $t = 80$, when it appears to more steadily increases all the way until $t = 200$ when it comes back to 50 molecules. This is in accord with enzyme kinetics, as the enzyme regenerates after the formation of the product, getting ready to catalyse a new reaction. The enzyme-substrate complex follows almost exactly the opposite "route" of the enzyme, and we can see that its concentration S_3 (magenta line) starts at zero and almost immediately peaks at around 50 molecules, where it stays pretty much constant, with some minor fluctuations, up until $t = 80$ when it seems to be more steadily decreasing up until $t = 200$ when it reaches zero. The enzyme concentration and the enzyme-substrate complex concentration cross their paths at around $t = 143$, and they cross the substrate concentration almost immediately after, the enzyme at $t = 145$, while the enzyme-substrate complex at $t = 150$.

All of this is in accordance with what we know about enzyme kinetics, as the enzyme is regenerated at the end of the reaction and with the law of concentration of mass, as we start with a total of 550 molecules and we end up with a total of 550 and 50 molecules.

In Figure 5.1b we see that the concentration of the substrate S_1 (blue line) steadily decreases as time goes on, resembling an exponential decay, from its initial value of 500 molecules at $t = 0$ to asymptotically approaching zero. The almost exact opposite "route" is followed by the product S_4 , (green line) whose concentration that starts at zero molecules and then steadily increases to level off at around 500 molecules. The concentrations of the two chemical species cross each other at around $t = 61$.

As for the enzyme concentration S_2 (red line), it start at 50 molecules, and drops close to zero at around $t = 5$, then seems to oscillate between zero and 10 molecules up until $t = 80$, when it appears to more steadily increases all the way until $t = 200$ when it comes back to 50 molecules. This is in accord with enzyme kinetics, as the enzyme regenerates after the formation of the product, getting ready to catalyse a new reaction. The enzyme-substrate complex follows almost exactly the opposite "route" of the enzyme, and we can see that its concentration S_3 (magenta line) starts at zero and peaks at around 50 molecules at around $t = 5$, then it stays pretty much constant, with some minor fluctuations, up until $t = 80$ when it seems to be more steadily decreasing up until $t = 200$ when it reaches zero. The enzyme concentration and the enzyme-substrate complex concentration cross their paths at around $t = 143$, after having crossed the path of the concentration of the substrate, the enzyme at $t = 139$, while the enzyme-substrate complex at $t = 130$ respectively.

All of this is in accordance with what we know about enzyme kinetics, as the enzyme is regenerated at the end of the reaction and with the law of concentration of mass, as we start with a total of 550 molecules and we end up with a total of 550 and 50 molecules.

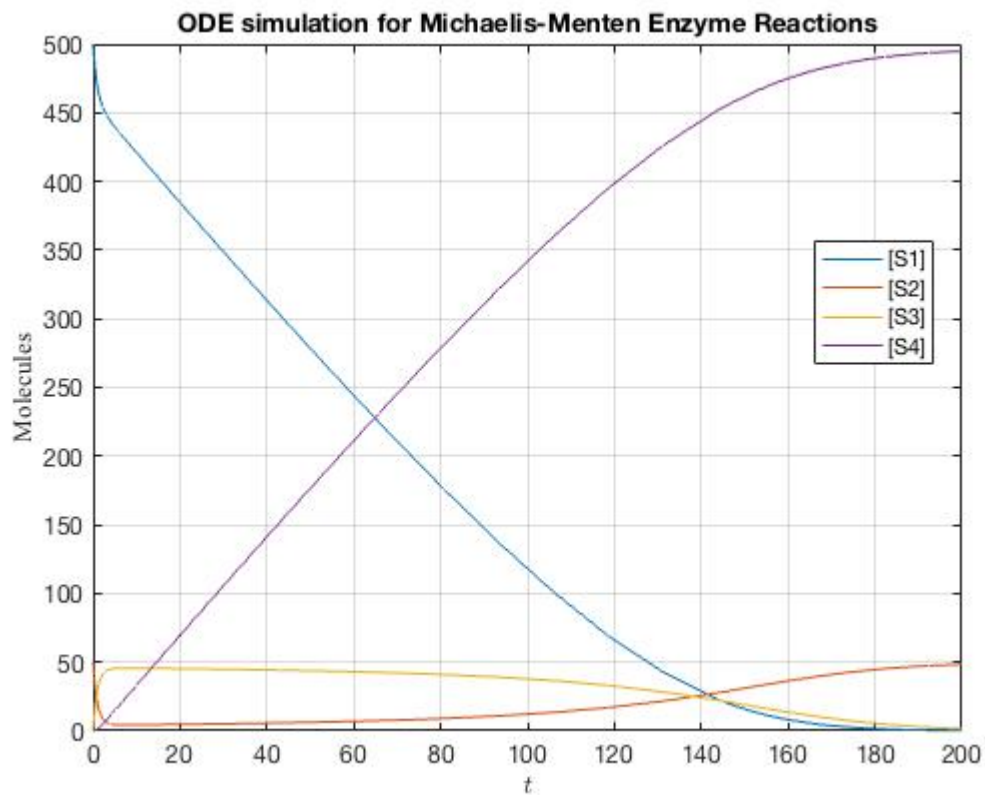
5.2 Part b

We can rewrite the reactions of Part a) as system of 4 differential equations, as follows

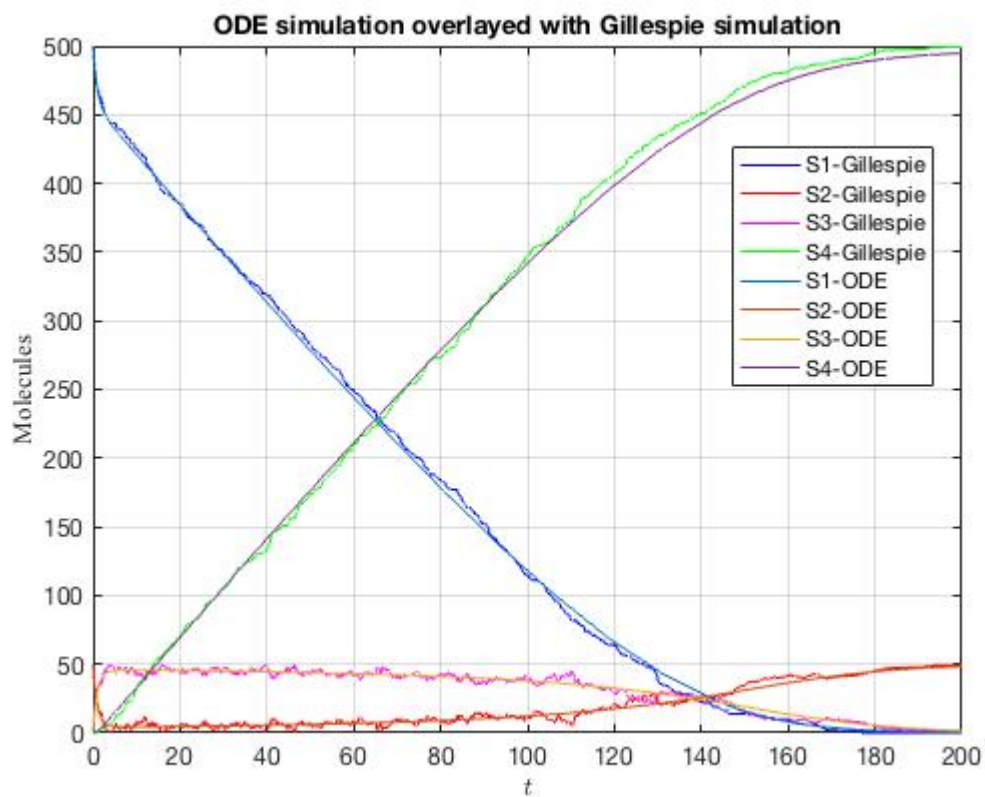
$$\begin{aligned}\frac{d[S_1]}{dt} &= -c_1[S_1][S_2] + c_2[S_3] \\ \frac{d[S_2]}{dt} &= -c_1[S_1][S_2] + (c_2 + c_3)[S_3] \\ \frac{d[S_3]}{dt} &= c_1[S_1][S_2] - (c_2 + c_3)[S_3] \\ \frac{d[S_4]}{dt} &= c_3[S_3],\end{aligned}$$

where $[S_1]$, $[S_2]$, $[S_3]$, $[S_4]$ are the concentration of the substrate, the enzyme, the enzyme-substrate complex and the product respectively.

The figure below shows a graph of the ODE simulation (Figure 5.2a) by itself and a graph of the ODE simulation overlayed with one from the Gillespie stochastic simulation algorithm for the same reactions.



(a)



(b) 15

Figure 5.2: a)ODE simulation; b) ODE vs. Gillespie Simulation

Figure 5.2 shows how the Gillespie Algorithm and the ODE simulation compare to each other. The overall trend appears to be basically the same, with the ODE simulation giving smoother graphs, whereas the Gillespie Algorithm has more "irregular" paths. Obviously the ODE simulation assume the reaction to be happening in a continuous manner, such that there are no corners or irregularities in the graphs, whereas the Gillespie Algorithm uses a stochastic approach and allows small fluctuations to happen among the concentrations of the different chemical species, giving, in this regard, a more realistic picture of the evolution of the reaction. In fact, it's unlikely that the concentration of the various species will just follow a smooth path from the beginning to the end of the reaction, as there are many factors that could alter the stability of the system at any given moment and the Gillespie Algorithm takes into account the different probabilities that the species interact with each other at any given moment.

That being said, both approaches give the same general trend for the overall reaction.

5.3 Part c

Using the quasi-steady state approximation, the Michaelis-Menten enzyme reactions can be re-written as the following system of differential equations

$$\begin{aligned}\frac{ds_1}{dt} &= -\frac{V_m s_1}{K_m + s_1} \\ \frac{ds_4}{dt} &= \frac{V_m s_1}{K_m + s_1},\end{aligned}$$

where s_1 and s_4 are the concentration of the substrate and the product respectively, and V_m and K_m are kinetic constants.

So, our first estimate of the two parameters is done by applying some basic knowledge about Michaelis-Menten enzyme kinetics:

$$\begin{aligned}V_m &= c_2[S_2]_{t=0} \rightarrow V_m = 4, \\ K_m &= \frac{c_2 + c_3}{c_1} \rightarrow K_m = 40.03,\end{aligned}$$

where $[S_2]$ is the concentration of the enzyme at the beginning of the reaction (i.e. 50 molecules in our case), and c_1 , c_2 , c_3 are the kinetic constants given in Part a.

So from here, we build a MatLab code to optimise V_m and K_m using `fminsearch` and we find the optimal values to be

$$\begin{aligned}V_m &= 4.8224, \\ K_m &= 88.9313.\end{aligned}$$

The sum of square errors is found to be

$$SSE = 12588$$

Below is a graph for the simulations of the systems in Parts b and c

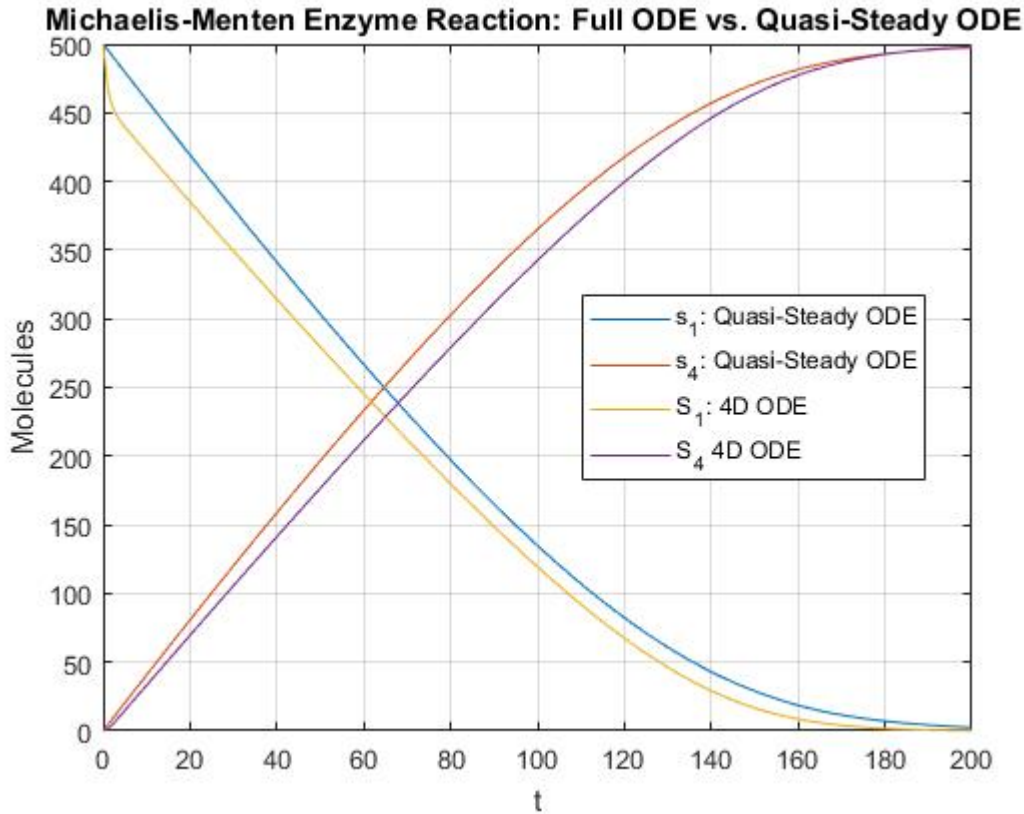


Figure 5.3: 4D-ODE Simulation vs. Quasi-Steady State Simulation

Figure 5.3 shows that for both simulation the overall trend is the same, as the concentration on the product steadily decreases until it gets to zero (or very close to it), while the concentration of the product steadily increases until it gets to 500 (or very close to it), as expected.

However, a couple of differences can be noticed right away: first of all, according to the system build in Part b, the concentration of the product almost instantaneously drops from its starting value of 500 to 450 molecules, whereas in the quasi-steady state simulation the path of the substrate is smoother, with no abrupt drops. Interestingly, the substrate and the product cross each other at the same time in both simulation (around $t = 70$), however, at that point in time, substrate and product have different concentrations in the two simulations: in the 4D-ODE simulation of Part b, when they intersect they both have around 225 molecules, whereas in the quasi-steady state simulation when they intersect they both have about 250 molecules, which corresponds to half the initial concentration of the substrate and half the final concentration of the product.

As for the product, both simulations have the same overall trend, but the quasi-steady state simulation shows a faster increase in its concentration if compared to the simulation run in Part b.

5.4 Part d

To convert from molecules to mole, we divide the number of molecules by the Avogadro number, which is given by

$$N_A \approx 6.022 \cdot 10^{23} \text{ molecules/mole}.$$

We are given that the volume of the *E.Coli* is

$$V = 7 \cdot 10^{-19} L$$

Therefore the conversion from *molecules/E.Coli* to *Moles/Liter* is achieved by doing the following units multiplication

$$\left(\frac{\text{molecules}}{E.coli} \cdot \frac{E.Coli/N_A}{7 \cdot 10^{-19} L} \right).$$

So, since we start with 500 molecules of substrate $[S_1]$, 50 of enzyme $[S_2]$ and obviously zero of enzyme-substrate complex as well as product, then, by converting the units, we get, for the substrate

$$[S_1] = \left(\frac{500 \text{ molecules}}{E.Coli} \right) \cdot \left(\frac{E.Coli / (6.022 \cdot 10^{23} \text{ molecules/mole})}{7 \cdot 10^{-19} L} \right) = 1.1865 \cdot 10^{-3} \text{ moles/L}$$

so 500 molecules of substrate are about 1 millimol.

Now for the product we get

$$[S_2] = \left(\frac{50 \text{ molecules}}{E.Coli} \right) \cdot \left(\frac{E.Coli / (6.022 \cdot 10^{23} \text{ molecules/mole})}{7 \cdot 10^{-19} L} \right) = 1.1865 \cdot 10^{-4} \text{ moles/L},$$

thus 50 molecules of the enzyme correspond to a little more than 0.1 millimols.

Now we convert the kinetic constants used for Part a c_1 , c_2 , c_3 to the proper units.

Since Michaelis-Menten enzyme reactions follow simple first order kinetic, the units of the constants are in $msec^{-1}$ (milliseconds), therefore to convert them in sec^{-1} we just need to multiply the given values by a factor of

$$\frac{1 msec}{0.001 sec}.$$

So,

$$c_1 = \left(0.002 msec^{-1} \cdot \frac{1 msec}{0.001 sec} \right) = 2 \cdot 10^{-6} sec^{-1},$$

$$c_2 = \left(6 \cdot 10^{-5} msec^{-1} \cdot \frac{1 msec}{0.001 sec} \right) = 6 \cdot 10^{-8} sec^{-1},$$

$$c_3 = \left(0.08 msec^{-1} \cdot \frac{1 msec}{0.001 sec} \right) = 8 \cdot 10^{-5} sec^{-1}.$$

Let's take a look back at our system of differential equations for the quasi-steady state simulation

$$\begin{aligned}\frac{ds_1}{dt} &= -\frac{V_m s_1}{K_m + s_1} \\ \frac{ds_4}{dt} &= \frac{V_m s_1}{K_m + s_1}.\end{aligned}$$

Obviously, as derivatives express rates of change, both $\frac{ds_1}{dt}$ and $\frac{ds_4}{dt}$ have units of *concentration/time*.

So we see that K_m needs to have the units of concentration, as it is added to s_1 (which is the concentration of the substrate) in our quasi-steady state system. So to convert to standard units from *molecules/E.Coli*, we do

$$K_m = \left(88.9313 \frac{\text{molecules}}{\text{E.Coli}}\right) \cdot \left(\frac{\text{E.Coli}/(6.022 \cdot 10^{23} \text{molecules/mole})}{7 \cdot 10^{-19} \text{L}}\right) = 2.1097 \cdot 10^{-4} \text{moles} \cdot \text{L}^{-1}.$$

On the other hand, V_m needs to have units of *concentration/time* since it is multiplied by a concentration and then the numerator is divided by another concentration in our differential equations for the quasi-steady state system. So,

$$\begin{aligned}V_m &= \left(4.8224 \frac{\text{molecules}}{\text{E.Coli} \cdot \text{msec}}\right) \cdot \left(\frac{\text{E.Coli}/(6.022 \cdot 10^{23} \text{molecules/mole})}{7 \cdot 10^{-19} \text{L}}\right) \cdot \left(\frac{1 \text{msec}}{0.001 \text{sec}}\right) = \\ &1.144 \cdot 10^{-8} \frac{\text{moles} \cdot \text{L}^{-1}}{\text{sec}}.\end{aligned}$$