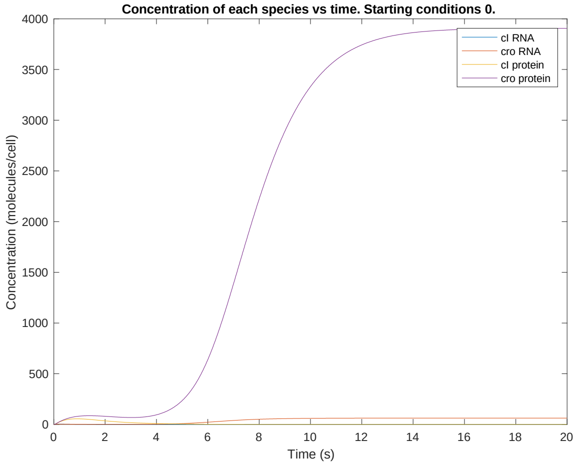
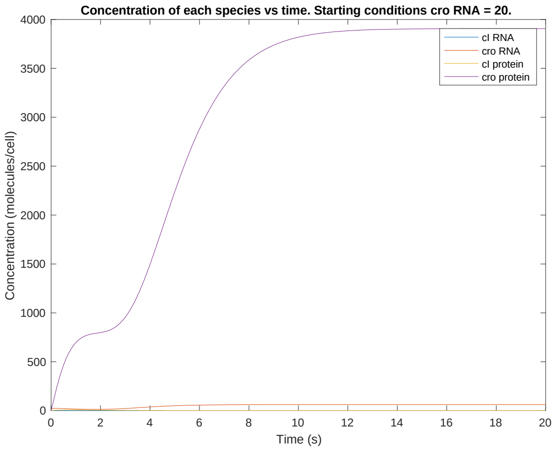
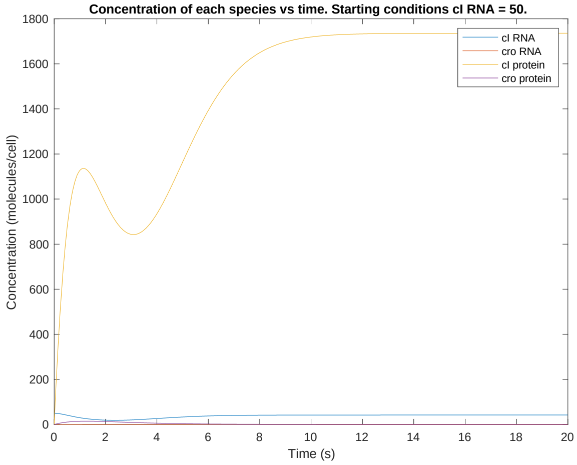
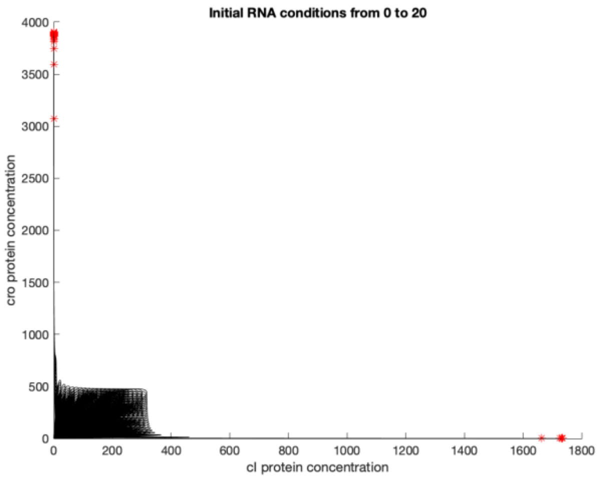


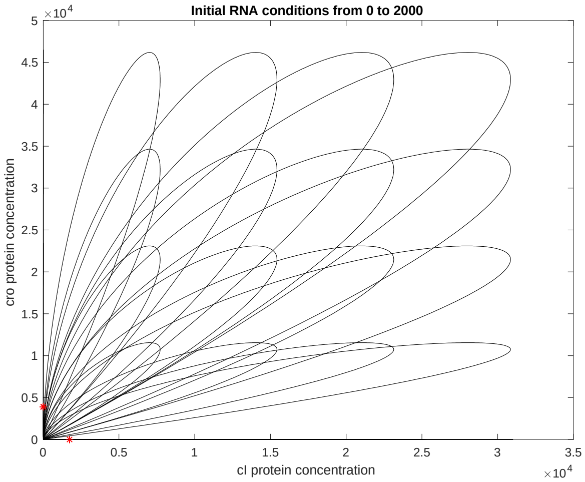




Figures 1-3 of a deterministic model of the cro-cI genetic network with the given constants and time step, and a total time of 20s. Concentrations given in molecules/cell. First figure has starting conditions of zero for all species and goes towards high cro protein and low cI (lysis) since constants are in favor of cro synthesis and cI degradation. The third figure has a high initial concentration of cI RNA and therefore tends toward a state of high cI and low cro (lysogeny).

2.





Figures 4-5 of cro protein vs cI protein with varying starting conditions. Endpoint of each line indicated by red star.

3. The above results show that, depending on the starting conditions, the system will end up in either a state of high cro and no cI or high cI and no cro. In a biological system, this represents lysis and lysogeny. The endpoint values of cro and cI concentrations are the same for both simulations since the stable point for lysis or lysogeny is dependent on the constants which don't change between the two. If the simulation was allowed to run for a longer time, the endpoints in each state would get closer together and eventually all end up in one of two places (lysis or lysogeny) because they would get closer to their final stable point. Since the constants in this specific model favor cro protein and RNA (lower degradation rate constant), the minimum requirements for the system to be in lysogeny are higher than for lysis. That is, we need a much higher starting concentration of cI RNA to get the system to lysogeny, while an equal starting concentration of cI and cro RNA would bring the system to lysis. In a typical infection, the starting concentration of all species is usually zero (unless already in a state of lysis or lysogeny), so in this model with these parameters, the phage would go to lysis. If all constants were equal in this deterministic model, the system would go to a saddle point of both lysis and lysogeny (which does not occur in nature). If all constants were equal in a stochastic model, the system would randomly go to either lysis or lysogeny (which is what occurs in nature). Figure 5 shows how our model has unbounded synthesis and degradation rates which make large loops ending up at the same point, which is inaccurate in nature.

4. To describe how the state of a bacteriophage changes when a bacterium is under stress, we need to introduce a new term to the differential equation for cI protein. This term should be negative since it reduces the amount of cI protein. It should have a constant, s, which increases with increasing bacterial stress, multiplied by the concentration of cI protein so that the cI protein concentration does not fall below zero. In this model with the given parameters, increasing s until the endpoint of cro protein concentration becomes higher than cI, the minimum s that is able to switch the system from lysogeny to lysis is 4.82. A high s could increase cI degradation to a point where there is no longer a stable state for lysogeny.

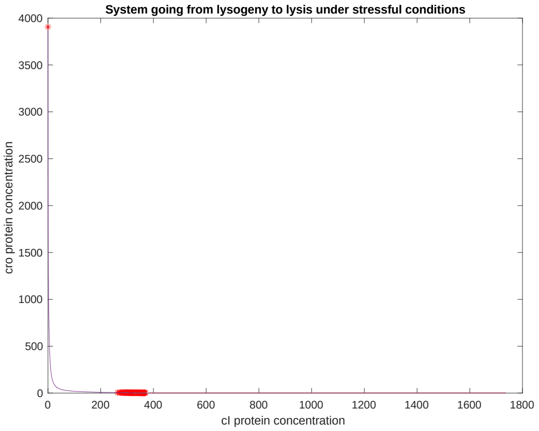


Figure 6 showing system starting in lysogeny with multiple stress constants, s, ranging from 4 to 5 in intervals of 0.01. The final line is the system going from lysogeny to lysis with s=4.82

Part B

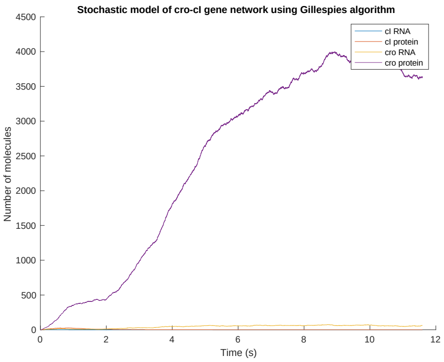
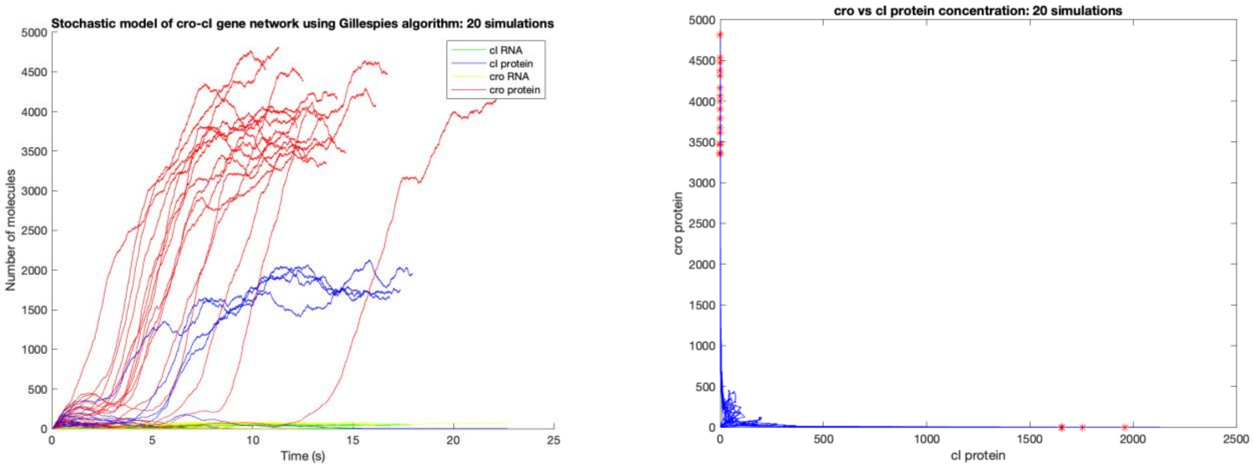


Figure 7 showing one run of a stochastic model using Gillespie's algorithm with 50,000 steps.

1. When the stochastic simulation was modeled using Gillespie's algorithm, we observe a less smooth curve since at each time point one species is moving up by one. We also see that the curve does not follow an exact path, it moves up and down many times and its slope is constantly changing, but most simulations end up at the same place of high cro protein and low cI. We also see that RNA concentrations are very low compared to protein concentrations because it takes a small amount of RNA to produce a large amount of protein. The average of this curve, in most cases, should be close to the Mass-Action Kinetics model, but the way it gets there is much more random.

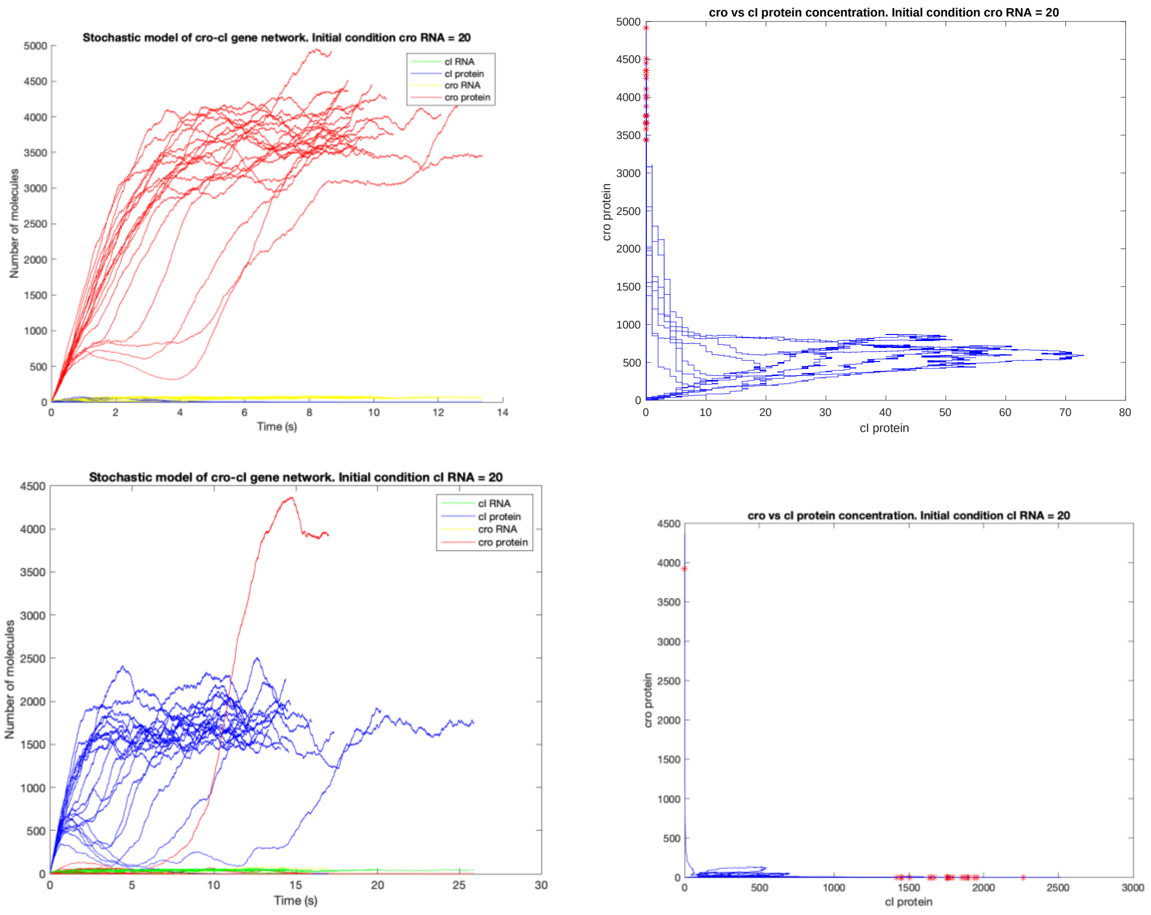
2.



Figures 8-9 showing 20 simulations of Gillespie's algorithm modeling cro-cI gene network. Figure 8 shows number of molecules vs time, and figure 9 shows cro protein molecules vs cI protein molecules with the endpoint of each simulation shown as a red star.

From these simulations, we can see that almost every run ends up in a state of high cro and low cI (most red stars end up on the y axis but much less end up on the x axis) but the path that each simulation takes is very different. On the plot to the right, we can see that most runs ended up with a concentration of high cro protein and low cI protein, but in this example, 3-4 simulations ended up with a high cI protein concentration and low cro protein concentration due to random chance. The simulations that did end up with high cI and low cro reach a steady state at a lower concentration than the others. This is due to the given constants that favor cro production and push the steady state for cI to a lower concentration.

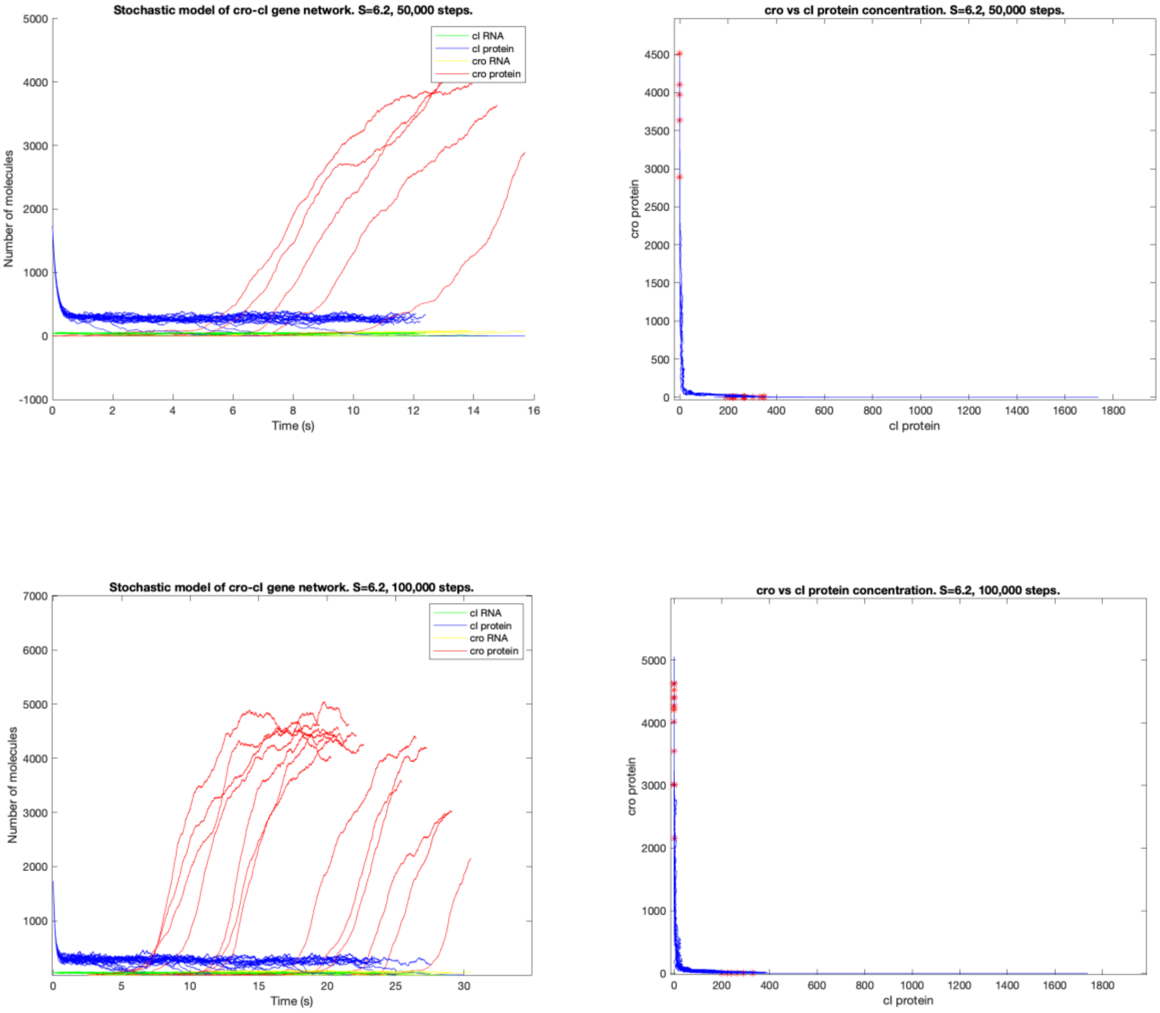
3.



Figures 10-13 showing 20 simulations of Gillespie’s algorithm modeling cro-cI gene network. First two modeled with the initial condition of 20 molecules of cro RNA and second two modeled with initial condition of 20 molecules of cI RNA with all other species set to zero.

These results show that if the system starts with a bias toward one state, the system will most likely go towards that state. The first two plots show that every simulation ended at the high cro low cI (lysis) state, and the second two plots show that every simulation except one ended up in the high cI low cro (lysogeny) state. This shows that it is still possible but unlikely for the system to go towards the opposite state once it already has an initial amount of RNA present. In a biological system, this means that if a phage were to infect a bacterium that was already in lysogeny or lysis, the phage would go toward that state.

4.



Figures 14-17 showing 20 simulations of Gillespie's algorithm modeling cro-cI gene network starting in the lysogenic state with new stress-induced cI protein degradation constant, s, set to 6.2. The first two simulations are modeled for 50,000 steps and the second two are modeled for 100,000 steps.

These plots show that a stress induced degradation rate can pull the system out of lysogeny after a certain amount of time depending on how large the constant is. The length of the simulation matters because it takes time for cI to get low enough for cro to take over and enter lysis. The first plot shows a set of 20 simulations with 50,000 steps and only a few make it out of lysogeny. The second plot shows 100,000 steps, and almost all simulations make it out of lysogeny and into lysis.

5. The results from stochastic models are different from deterministic models because they're outcomes are much less predictable. A deterministic model gives the exact same results every time and can stop in saddle points, whereas a stochastic model has an infinite number of possibilities and won't stop in saddle points because exact values are extremely unlikely to occur. The stochastic model is much better for modeling this biological system because with very small numbers of molecules, the fate of a system will depend greatly on the random chance of chemical reactions occurring at the right time and orientation. This is most important in the beginning of infection when the phage can go into either lysis or lysogeny, and there are very few molecules to start with. This decision is dependent entirely on random chance of chemical reactions. The results of stochastic simulations are often difficult to understand because they do not produce definitive values. For example, trying to find the concentrations consistent with lysogeny is difficult because each time we run a simulation that ends in lysogeny, the values of each species is different.

Appendix

Part A (1)

```
totaltime = 20;
deltat = 0.01;
numsteps = totaltime/deltat;
t = zeros(numsteps, 1);
cIrna = zeros(numsteps,1);
crorna = zeros(numsteps,1);
cIprot = zeros(numsteps,1);
croprot = zeros(numsteps,1);

for (i=1:numsteps-1)
    dcIprot_dt = (50*cIrna(i)) - (1.2*cIprot(i));
    dcIrna_dt = 50*(1-((croprot(i)^2)/(10+croprot(i)^2))) - (1.2*cIrna(i));
    dcroprot_dt = (50*crorna(i)) - (0.8*croprot(i));
    dcrorna_dt = 50*(1-((cIprot(i)^2)/(10+cIprot(i)^2))) - (0.8*crorna(i));

    cIrna(i+1) = cIrna(i) + dcIrna_dt*deltat;
    crorna(i+1) = crorna(i) + dcrorna_dt*deltat;
    cIprot(i+1) = cIprot(i) + dcIprot_dt*deltat;
    croprot(i+1) = croprot(i) + dcroprot_dt*deltat;

    t(i+1) = t(i) + deltat;
end
figure(1)
plot(t,cIrna, 'DisplayName','cI RNA')
hold on
plot(t, crorna, 'DisplayName', 'cro RNA')
plot(t, cIprot, 'DisplayName', 'cI protein')
plot(t, croprot, 'DisplayName', 'cro protein')
legend()
xlabel('Time (s)')
ylabel('Concentration (molecules/cell)')
title('Concentration of each species vs time. Starting conditions 0.')
hold off

cIrna = zeros(numsteps,1);
```

```
crorna = zeros(numsteps,1);
cIprot = zeros(numsteps,1);
croprot = zeros(numsteps,1);

crorna(1) = 20;

for (i=1:numsteps-1)
    dcIprot_dt = (50*cIrna(i)) - (1.2*cIprot(i));
    dcIrna_dt = 50*(1-((croprot(i)^2)/(10+croprot(i)^2))) - (1.2*cIrna(i));
    dcroprot_dt = (50*crorna(i)) - (0.8*croprot(i));
    dcrorna_dt = 50*(1-((cIprot(i)^2)/(10+cIprot(i)^2))) - (0.8*crorna(i));

    cIrna(i+1) = cIrna(i) + dcIrna_dt*deltat;
    crorna(i+1) = crorna(i) + dcrorna_dt*deltat;
    cIprot(i+1) = cIprot(i) + dcIprot_dt*deltat;
    croprot(i+1) = croprot(i) + dcroprot_dt*deltat;

    t(i+1) = t(i) + deltat;
end
figure(2)
plot(t,cIrna, 'DisplayName','cI RNA')
hold on
plot(t, crorna, 'DisplayName', 'cro RNA')
plot(t, cIprot, 'DisplayName', 'cI protein')
plot(t, croprot, 'DisplayName', 'cro protein')
legend()
xlabel('Time (s)')
ylabel('Concentration (molecules/cell)')
title('Concentration of each species vs time. Starting conditions cro RNA =
20.')
hold off

cIrna = zeros(numsteps,1);
crorna = zeros(numsteps,1);
cIprot = zeros(numsteps,1);
croprot = zeros(numsteps,1);

cIrna(1) = 50;

for (i=1:numsteps-1)
    dcIprot_dt = (50*cIrna(i)) - (1.2*cIprot(i));
    dcIrna_dt = 50*(1-((croprot(i)^2)/(10^2+croprot(i)^2))) - (1.2*cIrna(i));
    dcroprot_dt = (50*crorna(i)) - (0.8*croprot(i));
    dcrorna_dt = 50*(1-((cIprot(i)^2)/(10^2+cIprot(i)^2))) - (0.8*crorna(i));

    cIrna(i+1) = cIrna(i) + dcIrna_dt*deltat;
    crorna(i+1) = crorna(i) + dcrorna_dt*deltat;
    cIprot(i+1) = cIprot(i) + dcIprot_dt*deltat;
    croprot(i+1) = croprot(i) + dcroprot_dt*deltat;

    t(i+1) = t(i) + deltat;
end
figure(3)
plot(t,cIrna, 'DisplayName','cI RNA')
hold on
```

```
plot(t, crorna, 'DisplayName', 'cro RNA')
plot(t, cIprot, 'DisplayName', 'cI protein')
plot(t, croprot, 'DisplayName', 'cro protein')
legend()
xlabel('Time (s)')
ylabel('Concentration (molecules/cell)')
title('Concentration of each species vs time. Starting conditions cI RNA =
50.')
hold off
```

Part A (2)

```
totaltime = 20;
deltat = 0.01;
numsteps = totaltime/deltat;

hold on

for (j=0:20)
    for (k=0:20)

        cIrna = zeros(numsteps,1);
        crorna = zeros(numsteps,1);
        cIprot = zeros(numsteps,1);
        croprot = zeros(numsteps,1);

        crorna(1) = j;
        cIrna(1) = k;

        for (i=1:numsteps-1)
            dcIprot_dt = (50*cIrna(i)) - (1.2*cIprot(i));
            dcIrna_dt = 50*(1-((croprot(i)^2)/(10^2+croprot(i)^2))) -
(1.2*cIrna(i));
            dcroprot_dt = (50*crorna(i)) - (0.8*croprot(i));
            dcrorna_dt = 50*(1-((cIprot(i)^2)/(10^2+cIprot(i)^2))) -
(0.8*crorna(i));

            cIrna(i+1) = cIrna(i) + dcIrna_dt*deltat;
            crorna(i+1) = crorna(i) + dcrorna_dt*deltat;
            cIprot(i+1) = cIprot(i) + dcIprot_dt*deltat;
            croprot(i+1) = croprot(i) + dcroprot_dt*deltat;

        end
        croprot(end)
        cIprot(end)
        figure(1)
        plot(cIprot,croprot,'k')
        plot(cIprot(end),croprot(end),'r*')
    end
end
title('Initial RNA conditions from 0 to 20')
xlabel('cI protein concentration')
ylabel('cro protein concentration')
hold off
```

```
for (j=0:500:2000)
    for (k=0:500:2000)

        cIrna = zeros(numsteps,1);
        crorna = zeros(numsteps,1);
        cIprot = zeros(numsteps,1);
        croprot = zeros(numsteps,1);

        crorna(1) = j;
        cIrna(1) = k;

        for (i=1:numsteps-1)
            dcIprot_dt = (50*cIrna(i)) - (1.2*cIprot(i));
            dcIrna_dt = 50*(1-((croprot(i)^2)/(10^2+croprot(i)^2))) -
(1.2*cIrna(i));
            dcroprot_dt = (50*crorna(i)) - (0.8*croprot(i));
            dcrorna_dt = 50*(1-((cIprot(i)^2)/(10^2+cIprot(i)^2))) -
(0.8*crorna(i));

            cIrna(i+1) = cIrna(i) + dcIrna_dt*deltat;
            crorna(i+1) = crorna(i) + dcrorna_dt*deltat;
            cIprot(i+1) = cIprot(i) + dcIprot_dt*deltat;
            croprot(i+1) = croprot(i) + dcroprot_dt*deltat;

        end
        figure(2)
        plot(cIprot,croprot, 'k')
        hold on
        plot(cIprot(end),croprot(end),'r*')

    end
end
title('Initial RNA conditions from 0 to 2000')
xlabel('cI protein concentration')
ylabel('cro protein concentration')
hold off
```

Part A (4)

```
totaltime = 500;
deltat = 0.01;
numsteps = totaltime/deltat;
cIrna = zeros(numsteps,1);
crorna = zeros(numsteps,1);
cIprot = zeros(numsteps,1);
croprot = zeros(numsteps,1);

%Initial conditions of lysogeny steady state:
cIrna(1) = 41.666;
crorna(1) = 0.00020737;
cIprot(1) = 1736.1;
croprot(1) = 0.013;
```

```
for (s=4:0.01:5)

for (i=1:numsteps-1)
    dcIprot_dt = (50*cIrna(i)) - (1.2*cIprot(i)) - (s*cIprot(i));
    dcIrna_dt = 50*(1-((croprot(i)^2)/(10^2+croprot(i)^2))) - (1.2*cIrna(i));
    dcroprot_dt = (50*crorna(i)) - (0.8*croprot(i));
    dcrorna_dt = 50*(1-((cIprot(i)^2)/(10^2+cIprot(i)^2))) - (0.8*crorna(i));

    cIrna(i+1) = cIrna(i) + dcIrna_dt*deltat;
    crorna(i+1) = crorna(i) + dcrorna_dt*deltat;
    cIprot(i+1) = cIprot(i) + dcIprot_dt*deltat;
    croprot(i+1) = croprot(i) + dcroprot_dt*deltat;
end
figure(1)
plot(cIprot,croprot)
hold on
plot(cIprot(end),croprot(end),'r*')
if (croprot(end) > cIprot(end))
    s
    break
end
end

title('System going from lysogeny to lysis under stressful conditions')
xlabel('cI protein concentration')
ylabel('cro protein concentration')
```

Part B (1)

```
steps = 50000;
%Define population vector cI RNA, cI prot, cro RNA, cro prot:
x = zeros(steps,4);
t = zeros(steps,1);
%Define how each reaction changes population vector:
R1 = [0, 1, 0, 0];
R2 = [0, -1, 0, 0];
R3 = [1, 0, 0, 0];
R4 = [-1, 0, 0, 0];
R5 = [0, 0, 0, 1];
R6 = [0, 0, 0, -1];
R7 = [0, 0, 1, 0];
R8 = [0, 0, -1, 0];

for (i=1:steps)
    v1 = 50*x(i,1);
    v2 = 1.2*x(i,2);
    v3 = 50*(1-((x(i,4)^2)/(10^2+x(i,4)^2)));
    v4 = 1.2*x(i,1);
    v5 = 50*x(i,3);
    v6 = 0.8*x(i,4);
    v7 = 50*(1-((x(i,2)^2)/(10^2+x(i,2)^2)));
    v8 = 0.8*x(i,3);
    rates=[v1, v2, v3, v4, v5, v6, v7, v8];
    vtot = sum(rates);
    tau = -log(rand())/vtot;
    t(i+1) = t(i) + tau;
    rxn = find(cumsum(rates) > (vtot*rand()) ,1);
    switch rxn
        case 1
            x(i+1,:) = x(i,:) + R1;
        case 2
            x(i+1,:) = x(i,:) + R2;
        case 3
            x(i+1,:) = x(i,:) + R3;
        case 4
            x(i+1,:) = x(i,:) + R4;
        case 5
            x(i+1,:) = x(i,:) + R5;
        case 6
            x(i+1,:) = x(i,:) + R6;
        case 7
            x(i+1,:) = x(i,:) + R7;
        case 8
            x(i+1,:) = x(i,:) + R8;
    end


end


hold on

plot(t, x(:,1), 'DisplayName', 'cI RNA')
plot(t, x(:,2), 'DisplayName', 'cI protein')
plot(t, x(:,3), 'DisplayName', 'cro RNA')
plot(t, x(:,4), 'DisplayName', 'cro protein')




legend()
xlabel('Time (s)')
ylabel('Number of molecules')
title('Stochastic model of cro-cI gene network using Gillespies algorithm')
```

Part B (2)

```
steps = 50000;

for (j=1:20)
%Define population vector cI RNA, cI prot, cro RNA, cro prot:
x = zeros(steps,4);
t = zeros(steps,1);
%Define how each reaction changes population vector:
R1 = [0, 1, 0, 0];
R2 = [0, -1, 0, 0];
R3 = [1, 0, 0, 0];
R4 = [-1, 0, 0, 0];
R5 = [0, 0, 0, 1];
R6 = [0, 0, 0, -1];
R7 = [0, 0, 1, 0];
R8 = [0, 0, -1, 0];
```

```
for (i=1:steps)
    v1 = 50*x(i,1);
    v2 = 1.2*x(i,2);
    v3 = 50*(1-((x(i,4)^2)/(10^2+x(i,4)^2)));
    v4 = 1.2*x(i,1);
    v5 = 50*x(i,3);
    v6 = 0.8*x(i,4);
    v7 = 50*(1-((x(i,2)^2)/(10^2+x(i,2)^2)));
    v8 = 0.8*x(i,3);
    rates=[v1, v2, v3, v4, v5, v6, v7, v8];
    vtot = sum(rates);
    tau = -log(rand())/vtot;
    t(i+1) = t(i) + tau;
    rxn = find(cumsum(rates) > (vtot*rand()) ,1);
    switch rxn
        case 1
            x(i+1,:) = x(i,:) + R1;
        case 2
            x(i+1,:) = x(i,:) + R2;
        case 3
            x(i+1,:) = x(i,:) + R3;
        case 4
            x(i+1,:) = x(i,:) + R4;
        case 5
            x(i+1,:) = x(i,:) + R5;
        case 6
            x(i+1,:) = x(i,:) + R6;
        case 7
            x(i+1,:) = x(i,:) + R7;
        case 8
            x(i+1,:) = x(i,:) + R8;
    end


end


hold on
    figure(1)
    plot(t, x(:,1), 'green')
    plot(t, x(:,2), 'blue')
    plot(t, x(:,3), 'yellow')
    plot(t, x(:,4), 'red')

    figure(2)
    plot(x(:,2), x(:,4), 'blue')
    plot(x(end,2),x(end,4), 'r*')
end

figure(1)
legend('cI RNA', 'cI protein', 'cro RNA', 'cro protein')
xlabel('Time (s)')
ylabel('Number of molecules')
```

```
title('Stochastic model of cro-cI gene network using Gillespies algorithm: 20
simulations')

figure(2)
xlabel('cI protein')
ylabel('cro protein')
title('cro vs cI protein concentration: 20 simulations')
```

Part B (3)

```
steps = 50000;

for (j=1:20)
%Define population vector cI RNA, cI prot, cro RNA, cro prot:
x = zeros(steps,4);
%Set initial concentration of cro RNA to 20 molecules
x(1,3) = 20;
t = zeros(steps,1);
%Define how each reaction changes population vector:
R1 = [0, 1, 0, 0];
R2 = [0, -1, 0, 0];
R3 = [1, 0, 0, 0];
R4 = [-1, 0, 0, 0];
R5 = [0, 0, 0, 1];
R6 = [0, 0, 0, -1];
R7 = [0, 0, 1, 0];
R8 = [0, 0, -1, 0];

for (i=1:steps)
    v1 = 50*x(i,1);
    v2 = 1.2*x(i,2);
    v3 = 50*(1-((x(i,4)^2)/(10^2+x(i,4)^2)));
    v4 = 1.2*x(i,1);
    v5 = 50*x(i,3);
    v6 = 0.8*x(i,4);
    v7 = 50*(1-((x(i,2)^2)/(10^2+x(i,2)^2)));
    v8 = 0.8*x(i,3);
    rates=[v1, v2, v3, v4, v5, v6, v7, v8];
    vtot = sum(rates);
    tau = -log(rand())/vtot;
    t(i+1) = t(i) + tau;
    rxn = find(cumsum(rates) > (vtot*rand()) ,1);
    switch rxn
        case 1
            x(i+1,:) = x(i,:) + R1;
        case 2
            x(i+1,:) = x(i,:) + R2;
        case 3
            x(i+1,:) = x(i,:) + R3;
        case 4
            x(i+1,:) = x(i,:) + R4;
        case 5
            x(i+1,:) = x(i,:) + R5;
        case 6
            x(i+1,:) = x(i,:) + R6;
        case 7
            x(i+1,:) = x(i,:) + R7;
        case 8
            x(i+1,:) = x(i,:) + R8;
    end


end


hold on
    figure(1)
    plot(t, x(:,1), 'green')
    plot(t, x(:,2), 'blue')
    plot(t, x(:,3), 'yellow')
    plot(t, x(:,4), 'red')

    figure(2)
    plot(x(:,2), x(:,4), 'blue')
    plot(x(end,2),x(end,4), 'r*')
end

figure(1)
legend('cI RNA', 'cI protein', 'cro RNA', 'cro protein')
xlabel('Time (s)')
ylabel('Number of molecules')
title('Stochastic model of cro-cI gene network. Initial condition cro RNA =
20')

figure(2)
xlabel('cI protein')
ylabel('cro protein')
title('cro vs cI protein concentration. Initial condition cro RNA = 20')



for (j=1:20)
%Define population vector cI RNA, cI prot, cro RNA, cro prot:
x = zeros(steps,4);
%Set initial concentration of cI RNA to 20 molecules
x(1,1) = 20;
t = zeros(steps,1);
%Define how each reaction changes population vector:
R1 = [0, 1, 0, 0];
R2 = [0, -1, 0, 0];
R3 = [1, 0, 0, 0];
R4 = [-1, 0, 0, 0];
R5 = [0, 0, 0, 1];
R6 = [0, 0, 0, -1];
R7 = [0, 0, 1, 0];
R8 = [0, 0, -1, 0];

for (i=1:steps)
    v1 = 50*x(i,1);
    v2 = 1.2*x(i,2);
    v3 = 50*(1-((x(i,4)^2)/(10^2+x(i,4)^2)));
    v4 = 1.2*x(i,1);
    v5 = 50*x(i,3);
    v6 = 0.8*x(i,4);
    v7 = 50*(1-((x(i,2)^2)/(10^2+x(i,2)^2)));
    v8 = 0.8*x(i,3);
    rates=[v1, v2, v3, v4, v5, v6, v7, v8];
    vtot = sum(rates);
    tau = -log(rand())/vtot;
    t(i+1) = t(i) + tau;
    rxn = find(cumsum(rates) > (vtot*rand()) ,1);
    switch rxn
        case 1
            x(i+1,:) = x(i,:) + R1;
        case 2
            x(i+1,:) = x(i,:) + R2;
        case 3
            x(i+1,:) = x(i,:) + R3;
        case 4
            x(i+1,:) = x(i,:) + R4;
        case 5
            x(i+1,:) = x(i,:) + R5;
        case 6
            x(i+1,:) = x(i,:) + R6;
        case 7
            x(i+1,:) = x(i,:) + R7;
        case 8
            x(i+1,:) = x(i,:) + R8;
    end


end


hold on
    figure(3)
    plot(t, x(:,1), 'green')
    hold on
    plot(t, x(:,2), 'blue')
    plot(t, x(:,3), 'yellow')
    plot(t, x(:,4), 'red')

    figure(4)
    plot(x(:,2), x(:,4), 'blue')
    plot(x(end,2),x(end,4), 'r*')
end

figure(3)
legend('cI RNA', 'cI protein', 'cro RNA', 'cro protein')
xlabel('Time (s)')
ylabel('Number of molecules')
title('Stochastic model of cro-cI gene network. Initial condition cI RNA =
20')

figure(4)
xlabel('cI protein')
ylabel('cro protein')
```

```
title('cro vs cI protein concentration. Initial condition cI RNA = 20')
```

Part B (4)

```
%first two graphs with 50000 steps:
steps = 50000;

for (j=1:20)
%Define population vector cI RNA, cI prot, cro RNA, cro prot:
x = zeros(steps,4);
%Set initial conditions to lysogeny state
x(1,1) = 41.66;
x(1,2) = 1736.1;

%Define stress constant s based on previously acquired value
s = 6.2;

t = zeros(steps,1);
%Define how each reaction changes population vector:
R1 = [0, 1, 0, 0];
R2 = [0, -1, 0, 0];
R3 = [1, 0, 0, 0];
R4 = [-1, 0, 0, 0];
R5 = [0, 0, 0, 1];
R6 = [0, 0, 0, -1];
R7 = [0, 0, 1, 0];
R8 = [0, 0, -1, 0];

for (i=1:steps)
    v1 = 50*x(i,1);
    v2 = 1.2*x(i,2);
    v3 = 50*(1-((x(i,4)^2)/(10^2+x(i,4)^2)));
    v4 = 1.2*x(i,1);
    v5 = 50*x(i,3);
    v6 = 0.8*x(i,4);
    v7 = 50*(1-((x(i,2)^2)/(10^2+x(i,2)^2)));
    v8 = 0.8*x(i,3);
    %New rate for degradation term
    v9 = s*x(i,2);
    rates=[v1, v2, v3, v4, v5, v6, v7, v8, v9];
    vtot = sum(rates);
    tau = -log(rand())/vtot;
    t(i+1) = t(i) + tau;
    rxn = find(cumsum(rates) > (vtot*rand()) ,1);
    switch rxn
        case 1
            x(i+1,:) = x(i,:) + R1;
        case 2
            x(i+1,:) = x(i,:) + R2;
        case 3
            x(i+1,:) = x(i,:) + R3;
        case 4
            x(i+1,:) = x(i,:) + R4;
        case 5
            x(i+1,:) = x(i,:) + R5;
        case 6
```

```
            x(i+1,:) = x(i,:) + R6;
        case 7
            x(i+1,:) = x(i,:) + R7;
        case 8
            x(i+1,:) = x(i,:) + R8;
        case 9
            x(i+1,:) = x(i,:) + R2;

    end


end


hold on
    figure(1)
    plot(t, x(:,1), 'green')
    plot(t, x(:,2), 'blue')
    plot(t, x(:,3), 'yellow')
    plot(t, x(:,4), 'red')

    figure(2)
    plot(x(:,2), x(:,4), 'blue')
    plot(x(end,2),x(end,4), 'r*')
end

figure(1)
legend('cI RNA', 'cI protein', 'cro RNA', 'cro protein')
xlabel('Time (s)')
ylabel('Number of molecules')
title('Stochastic model of cro-cI gene network. S=6.2, 50,000 steps.')

figure(2)
xlabel('cI protein')
ylabel('cro protein')
title('cro vs cI protein concentration. S=6.2, 50,000 steps.')

%second two plots with 10000 steps:
steps = 100000;

for (j=1:20)
%Define population vector cI RNA, cI prot, cro RNA, cro prot:
x = zeros(steps,4);
%Set initial conditions to lysogeny state
x(1,1) = 41.66;
x(1,2) = 1736.1;

%Define stress constant s based on previously acquired value
s = 6.2;

t = zeros(steps,1);
%Define how each reaction changes population vector:
R1 = [0, 1, 0, 0];
R2 = [0, -1, 0, 0];
```

```
R3 = [1, 0, 0, 0];
R4 = [-1, 0, 0, 0];
R5 = [0, 0, 0, 1];
R6 = [0, 0, 0, -1];
R7 = [0, 0, 1, 0];
R8 = [0, 0, -1, 0];

for (i=1:steps)
    v1 = 50*x(i,1);
    v2 = 1.2*x(i,2);
    v3 = 50*(1-((x(i,4)^2)/(10^2+x(i,4)^2)));
    v4 = 1.2*x(i,1);
    v5 = 50*x(i,3);
    v6 = 0.8*x(i,4);
    v7 = 50*(1-((x(i,2)^2)/(10^2+x(i,2)^2)));
    v8 = 0.8*x(i,3);
    %New rate for degradation term
    v9 = s*x(i,2);
    rates=[v1, v2, v3, v4, v5, v6, v7, v8, v9];
    vtot = sum(rates);
    tau = -log(rand())/vtot;
    t(i+1) = t(i) + tau;
    rxn = find(cumsum(rates) > (vtot*rand()) ,1);
    switch rxn
        case 1
            x(i+1,:) = x(i,:) + R1;
        case 2
            x(i+1,:) = x(i,:) + R2;
        case 3
            x(i+1,:) = x(i,:) + R3;
        case 4
            x(i+1,:) = x(i,:) + R4;
        case 5
            x(i+1,:) = x(i,:) + R5;
        case 6
            x(i+1,:) = x(i,:) + R6;
        case 7
            x(i+1,:) = x(i,:) + R7;
        case 8
            x(i+1,:) = x(i,:) + R8;
        case 9
            x(i+1,:) = x(i,:) + R2;

    end


end


hold on
    figure(3)
    plot(t, x(:,1), 'green')
    hold on
    plot(t, x(:,2), 'blue')
    plot(t, x(:,3), 'yellow')
    plot(t, x(:,4), 'red')
    figure(4)
    plot(x(:,2), x(:,4), 'blue')
    plot(x(end,2),x(end,4), 'r*')
end

figure(3)
legend('cI RNA', 'cI protein', 'cro RNA', 'cro protein')
xlabel('Time (s)')
ylabel('Number of molecules')
title('Stochastic model of cro-cI gene network. S=6.2, 100,000 steps.')

figure(4)
xlabel('cI protein')
ylabel('cro protein')
title('cro vs cI protein concentration. S=6.2, 100,000 steps.')
```

I did the project completely on my own; I discussed some problems with a few classmates but ultimately figured out each problem on my own and wrote the code and writing myself; I did not share any of my work with others. I did make extensive use of the given readings, lectures, lecture slides, Wikipedia and mathworks.com for MATLAB implementation stuff. Signed: Jack Vaska.