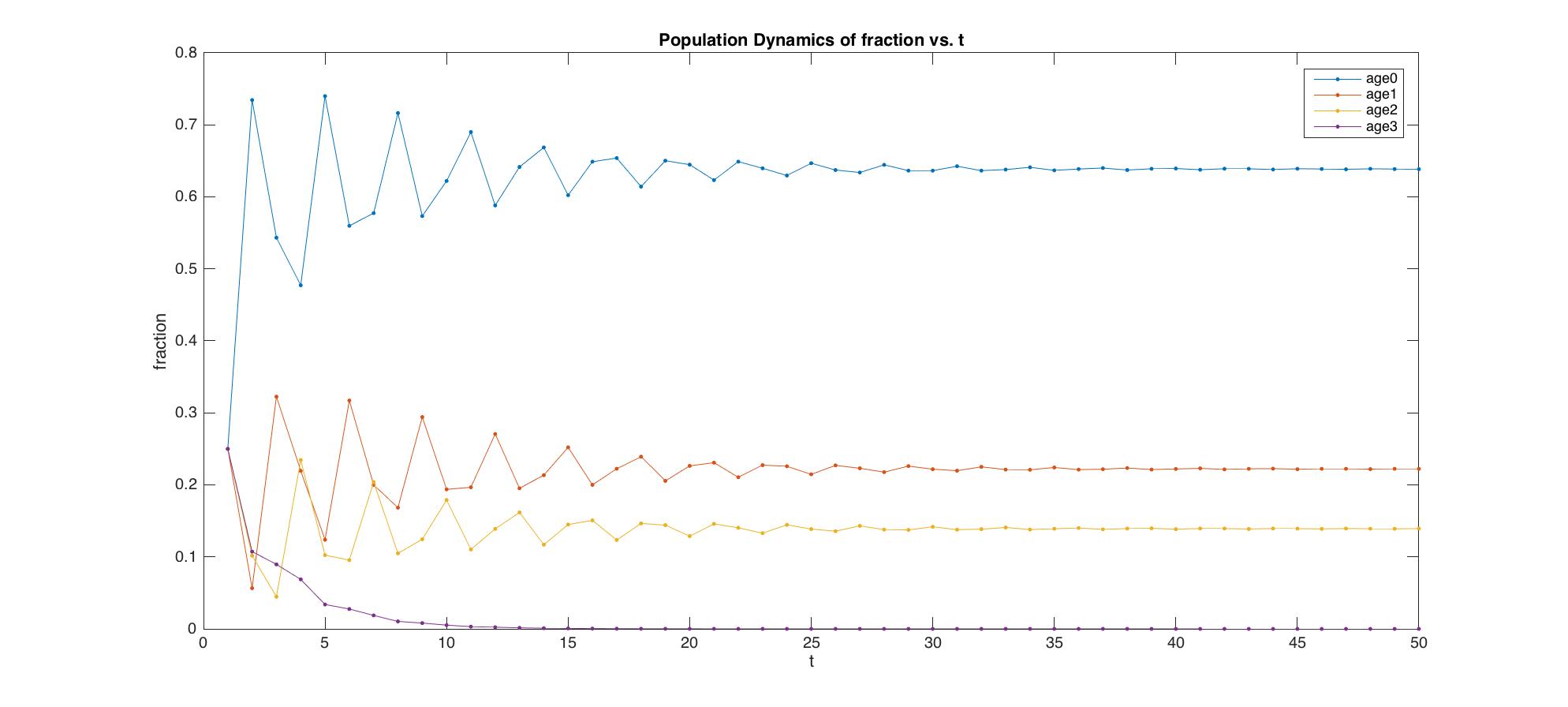
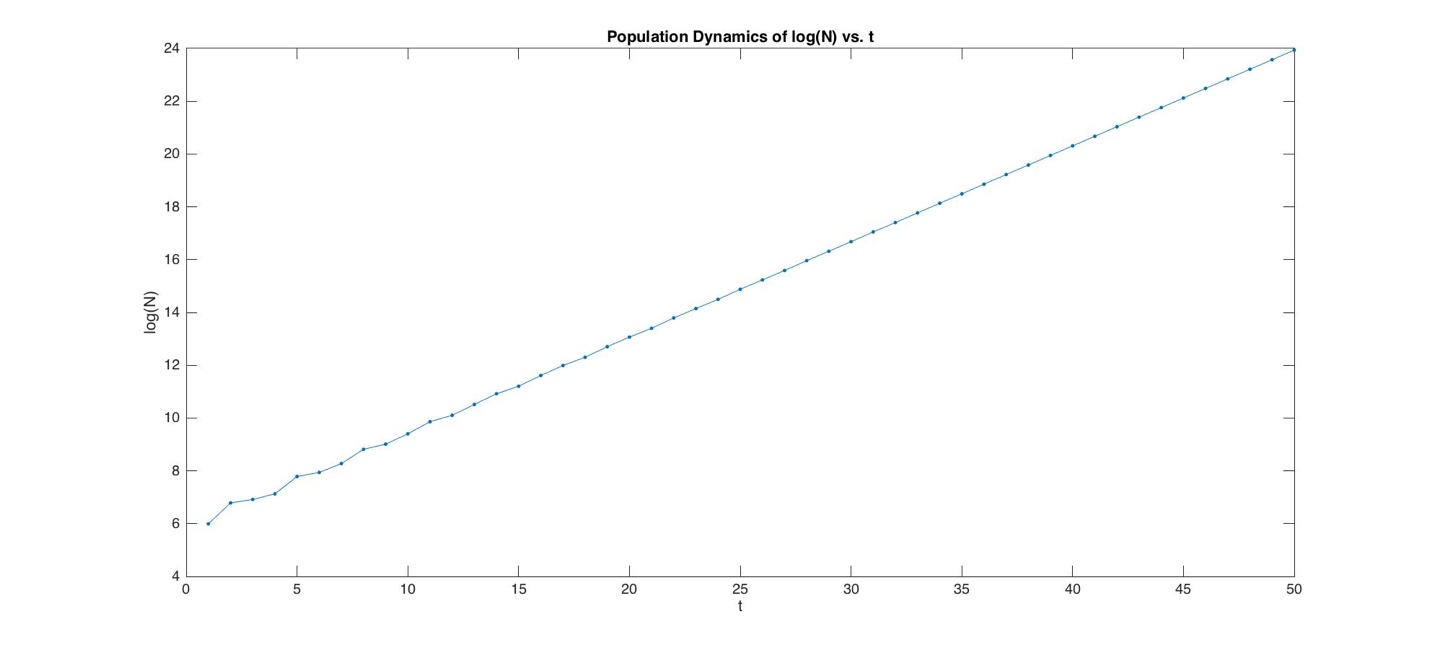
1. **Iterating Leslie Matrices and the Euler-Lotkerra Formula**

I coded in the Leslie matrix and run the simulation. Results are shown in Figure 1 and 2. The *lamda* I calculated based on the polyfit is **1.4371**. While the Euler-Lotka formula, in which *Ia* and *fa* are specified by numerical input gives a similar prediction to be **1.4624**.



% @Author: Baihan Lin

% @Date: Oct 2016

%% Question 1

% initiate projection matrix A.

A1 = [0 1 5 0.5;

0.5 0 0 0;

0 0.9 0 0;

0 0 0 0.95];

% simulation of population dynamics

n\_zero=[100; 100; 100; 100];

f\_zero=[0.25;0.25;0.25;0.25];

Tmax=50;

n\_vs\_t=zeros(4,Tmax);

n\_vs\_t(:,1)=n\_zero ;

f\_vs\_t=zeros(4,Tmax);

f\_vs\_t(:,1)=f\_zero ;

for t=2:Tmax;

n\_vs\_t(:,t)=A1\*n\_vs\_t(:,t-1);

n\_sum = sum(n\_vs\_t(:,t));

f\_vs\_t(:,t) = n\_vs\_t(:,t)/n\_sum;

end

N = sum(n\_vs\_t);

fig1 = figure;

set(gca,'FontSize',20);

plot(1:Tmax,log(N'),'.-','MarkerSize',14);

xlabel('t','FontSize',20);

ylabel('log(N)','FontSize',20);

title('Population Dynamics of log(N) vs. t');

fig2 = figure;

set(gca,'FontSize',20);

plot(1:Tmax,f\_vs\_t','.-','MarkerSize',14);

xlabel('t','FontSize',20);

ylabel('fraction','FontSize',20);

title('Population Dynamics of fraction vs. t');

legend('age0', 'age1', 'age2','age3');

%Fit log(N(t)) to a straight line

p=polyfit(25:Tmax,log(N(1,25:Tmax)),1) % 0.3626 5.8045

lamda\_estimate=exp(p(1)) % 1.4371

% Numerically solve lamda

Ia = [1 0.5 0.5\*0.9 0.5\*0.9\*0.95] ;

fa = [0 1 5 0.5];

lamda\_predicted = fzero(@(lambda) eulot(lambda,Ia,fa),1) % 1.4624 very close.

1. **E+G Ex. 2.12. Northern spotted owl females begin breeding at age *a* = 3, with an average of 0.24 female offspring till they die (*fa* = 0.24 for *a* >= 3). Survival prob from birth to age 3 is 0.0722. annual survival prob of adults is 0.942 (*pa* for age 3 <= *a* <= 49). Max age *A* = 50. New born age *a* = 0.**
2. *I3* = *p0p1p2* = 0.0722, but not individual *p* values, because any choice of individual p with same product will yield same long-term population growth rate (*lamda*). Why is this true?

In the Leslie matrix, since the fecundity for age 0~2 is zero, thus in equation **n**(t+1) = **L n**(t), the first three elements of row 1 are all zero, cancelling out terms n0(t+1), n1(t+1), n2(t+1), n0(t), n1(t), n2(t). In another word, these terms are free variables when solving this matrix, and stay zero in the eigenvectors, thus the eigenvalues are irrelevant to *p0*, *p1* and *p2.*

Thus, the first nontrivial **n** starts from n3(t+1) and n3(t), which is defined to be n3(t+1) = = , which is irrelevant from the individual values of *p0*, *p1* and *p2*.

1. Construct the projection matrix for the population.

From part (a), we found that we can treat age 0~2 as a single group in our owl age structure model. Thus the projection matrix would be the following:

% @Author: Baihan Lin

% @Date: Oct 2016

clear all; close all; clc;

%% Question 2

% initiate projection matrix A.

l = 0.942\*diag(ones(1, 48));

l(1) = 0.0722;

f = 0.24\*ones(1,48);

f(1) = 0;

A2 = [f;l(1:47,:)];

A =

1. Compute the long-term growth rate *lamda* for population.

From the code on the right, I tested that this projection matrix is power-positive based on Perron-Frobenius theorem (A^(n^2-2n+2)). Thus, we have a unique dominant eigenvalue *lamda* which is calculated to be **0.9388**, implying that the population is not safe, but in danger of extinction.

% test power-positive

n2 = 48;

Apos = A2^(n2^2-n2\*2+2);

Aispos = (Apos > 0);

ispos = prod(prod(Aispos)); %1

% calculate dominant eigenvalue lamda

[Vall,L2,Wall]=eig(A2);

L2 = diag(L2);

j2=find(abs(L2)==max(abs(L2)));

d\_lamda = L2(j2); % 0.9388

V = Vall(:,j2);

W = Wall(j2,:);

1. Compute the matrix of elasticity for projection matrix. Is this elasticity for fecundity value *fa* same for all ages *a*? give an explanation, and state 1 possible implication for management plans.

% calculate elasticity matrix

for x = 1:48

for y = 1:48

E(x,y) = A2(x,y)\*(V(x)\*W(y)/(dot(V, W)\*d\_lamda));

end

end

surf(real(E));

figure

set(gca,'FontSize',20);

plot(0:46, diag(E(2:48,1:47)),'.-','MarkerSize',20);

xlabel('ages','FontSize',10);

ylabel('survival rate elasticity','FontSize',10);

title('Elasticity for annual survival rate p\_a');

figure

set(gca,'FontSize',20);

plot(0:47, E(1,1:48),'.-','MarkerSize',20);

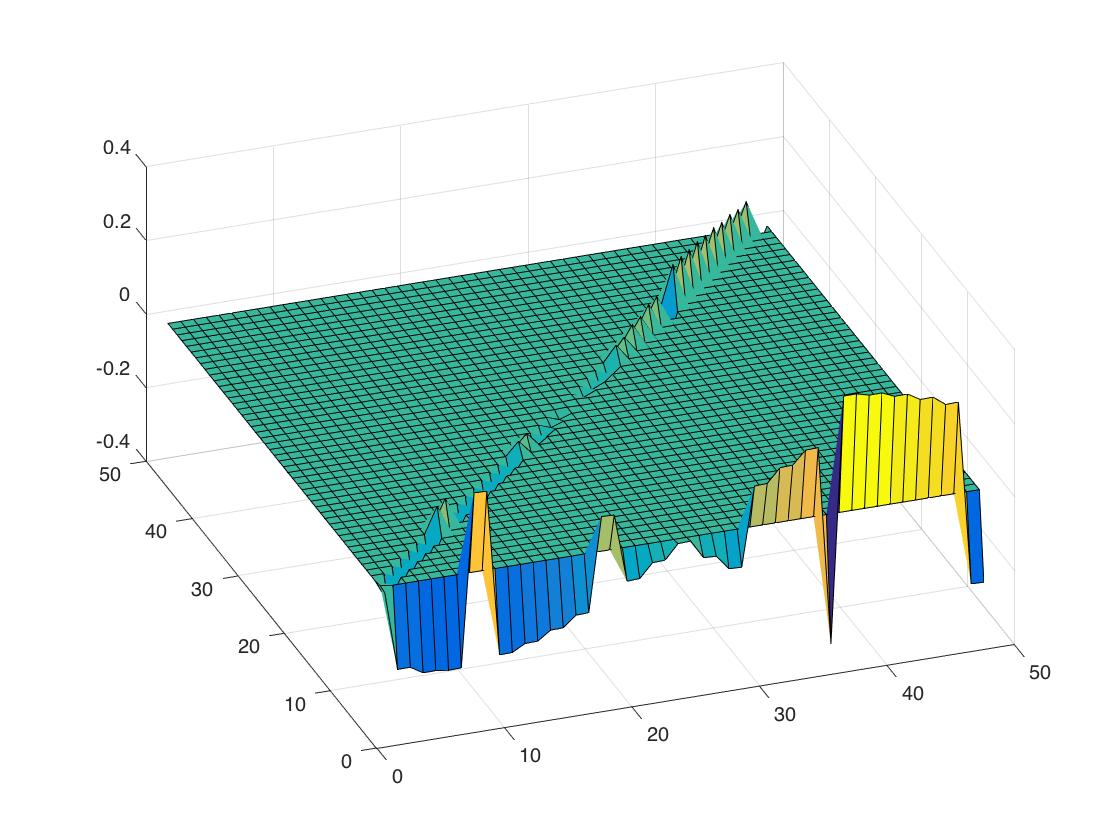
xlabel('ages','FontSize',10);

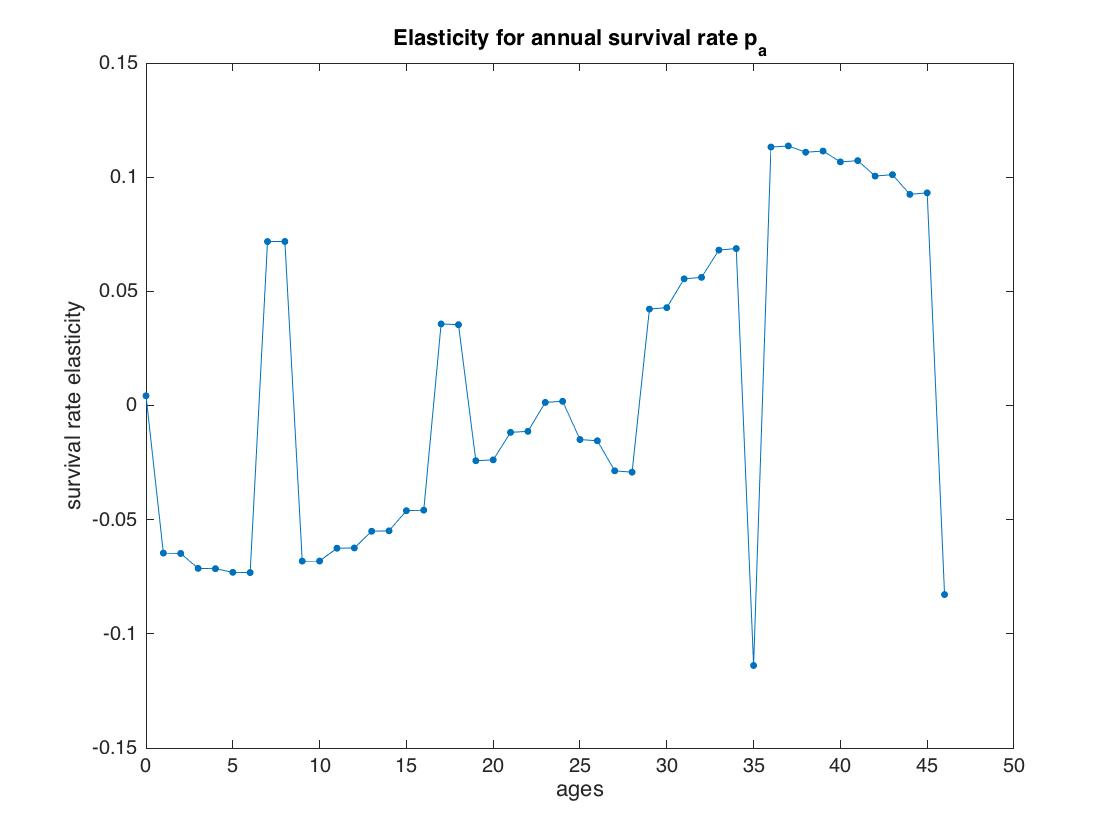
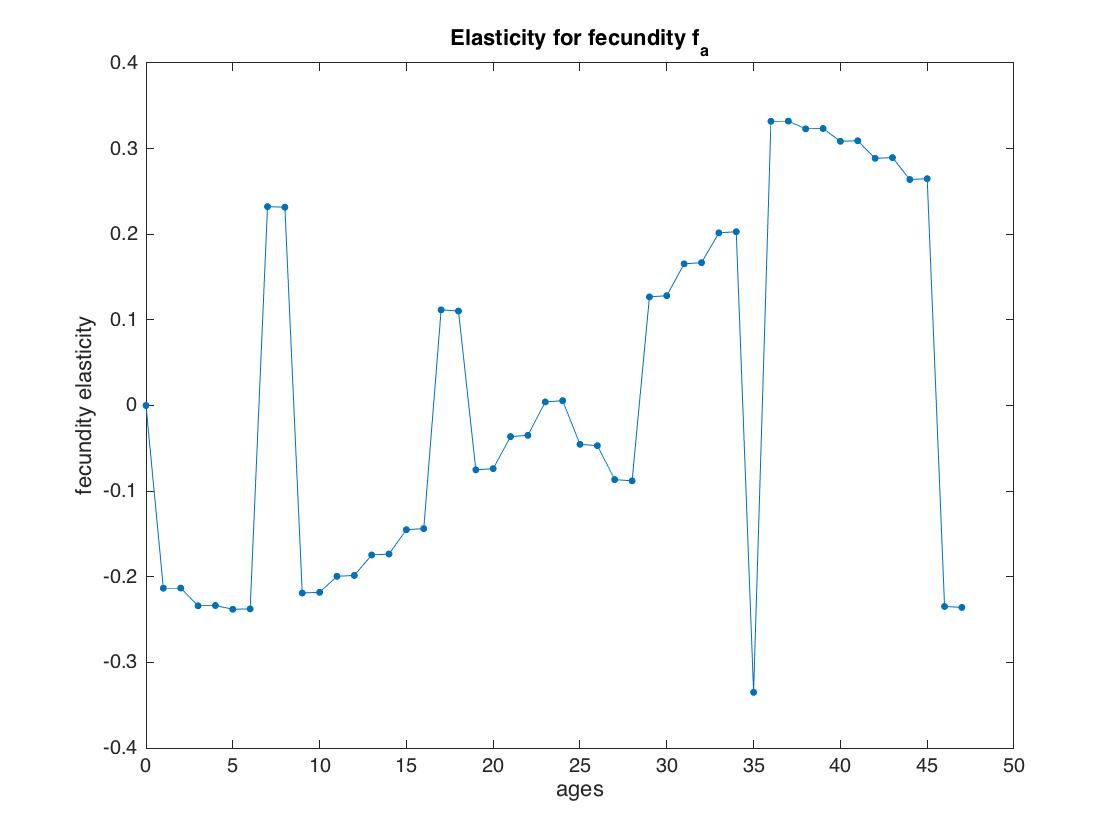
ylabel('fecundity elasticity','FontSize',10);

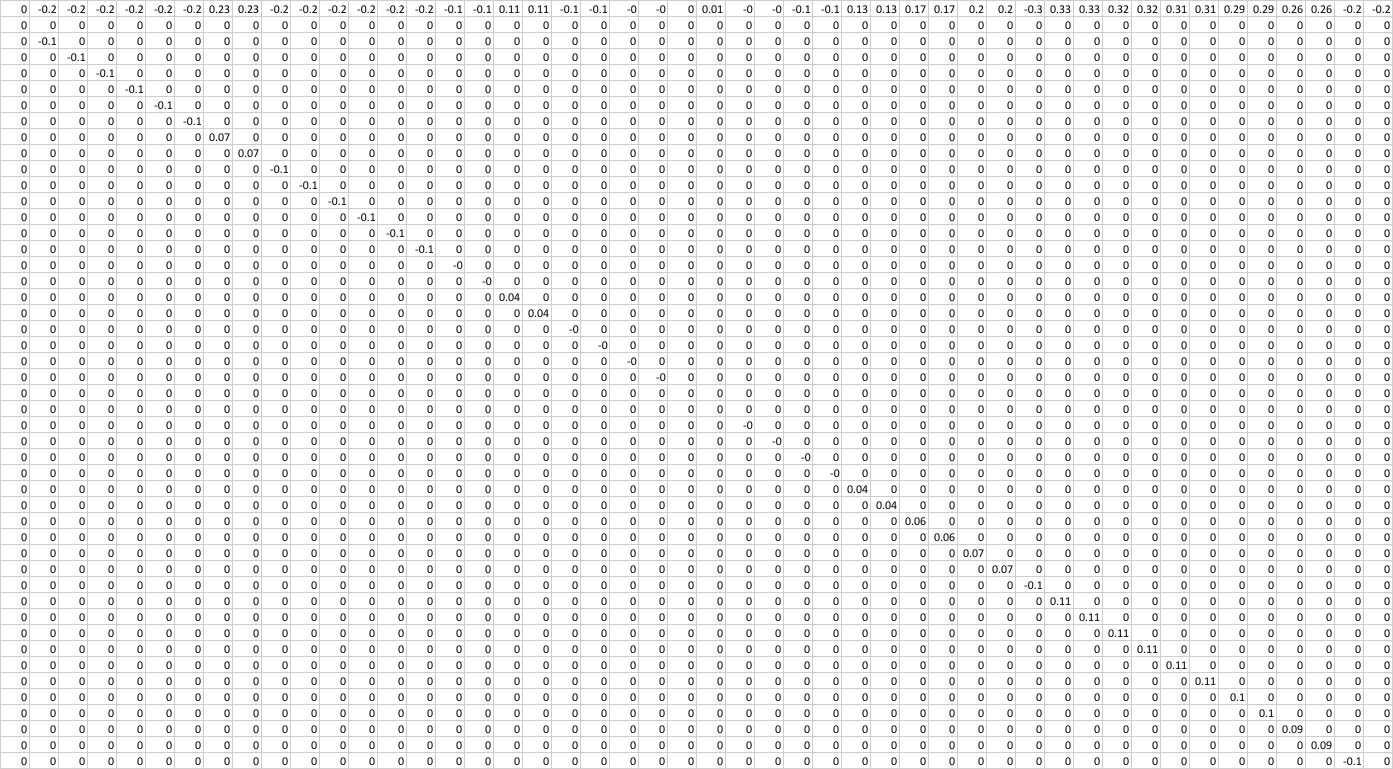
title('Elasticity for fecundity f\_a');

The matrix is too big to show below, but attached in the next page.

No, it is not the same, as shown here (a surface plot of this elasticity matrix), and also next page on fecundity and survival rate individually. I think the reason is that in this example the juvenile survival rate seems to be the bottleneck, and later on, due to the accumulated effect from survival rate, the older age group seems to be also the limiting factor to keep up with the extinction. Thus, the management plan should focus on increasing the survival rate of juvenile and increasing fecundity rate and survival rate of the older group.



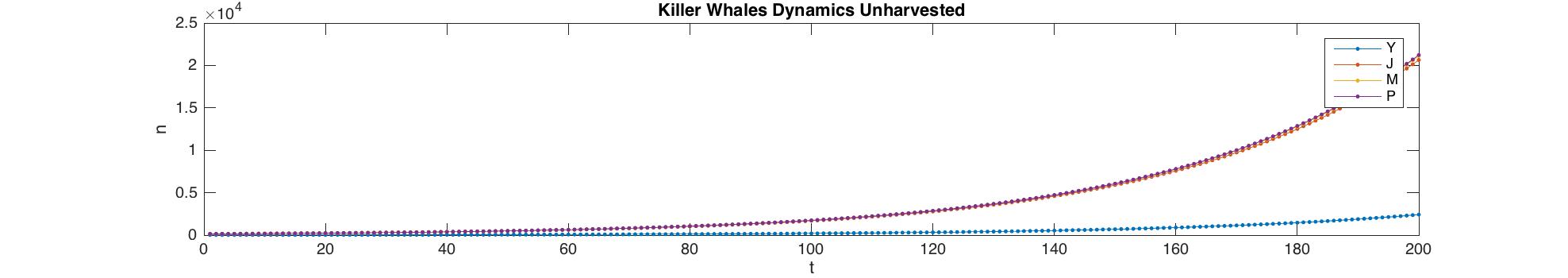




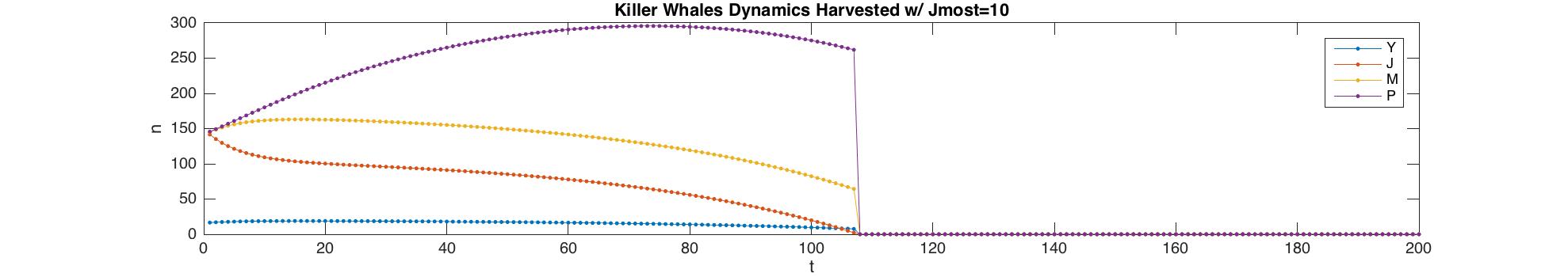
1. **E+G Ex. 2.15.**

From attached code behind, I calculated *lamda* = 1.0254, and *w* = [0.0663;0.5663;0.5793;0.5825].

Then I simulated the unharvested case in 200 steps (Figure 3).



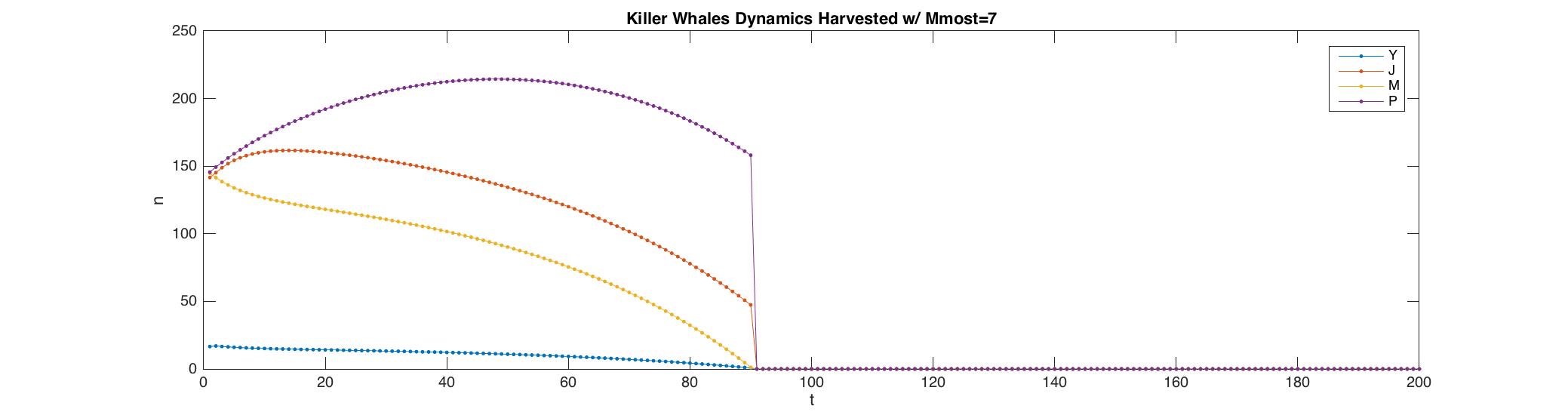
To simulated the harvesting on juveniles, I looped to harvest 1 more juveniles each step and break when it first extinct within 200 steps. The maximum juvenile to take without extinction is **9,** because if we go to 10, it goes to extinction (Figure 4).



To simulated the harvesting on reproductive adults,

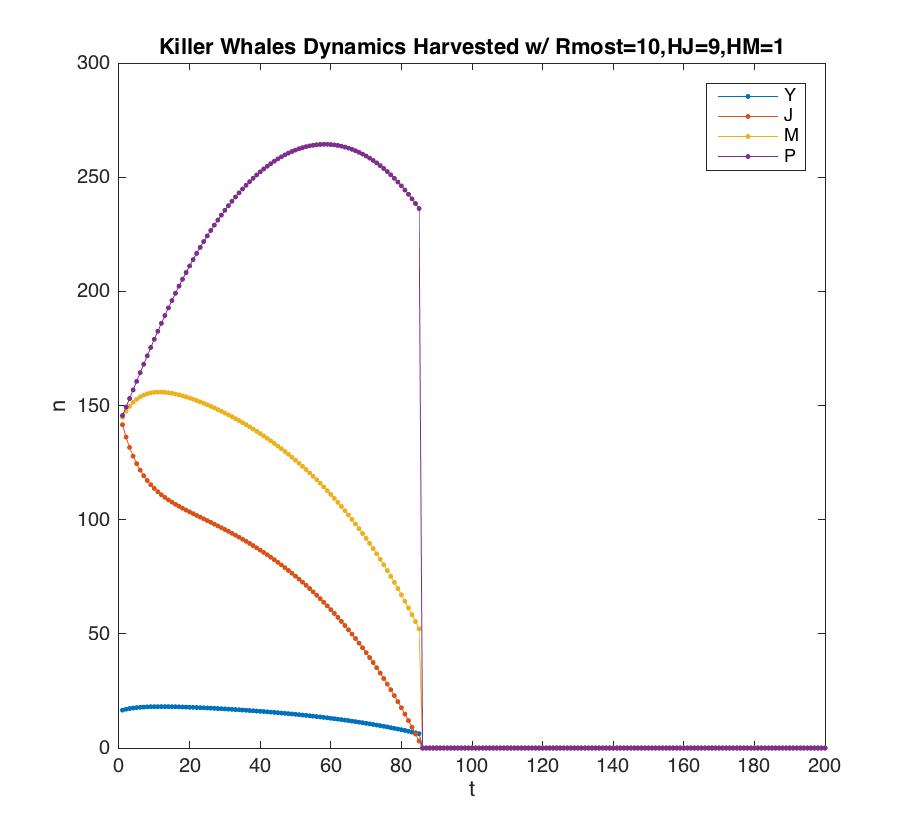
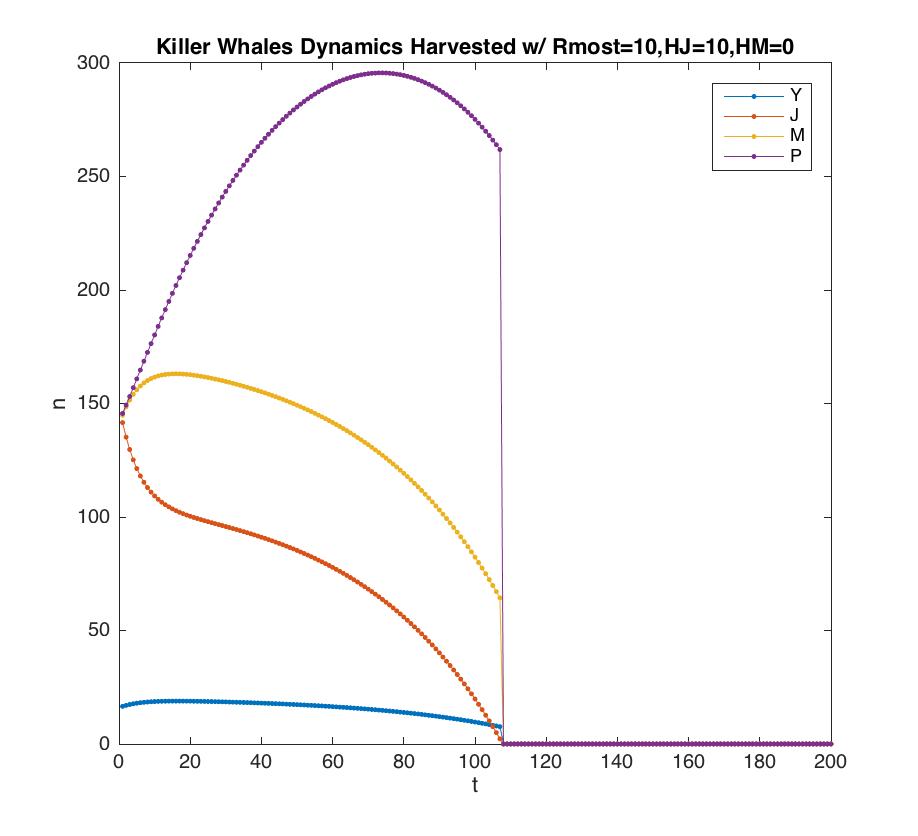
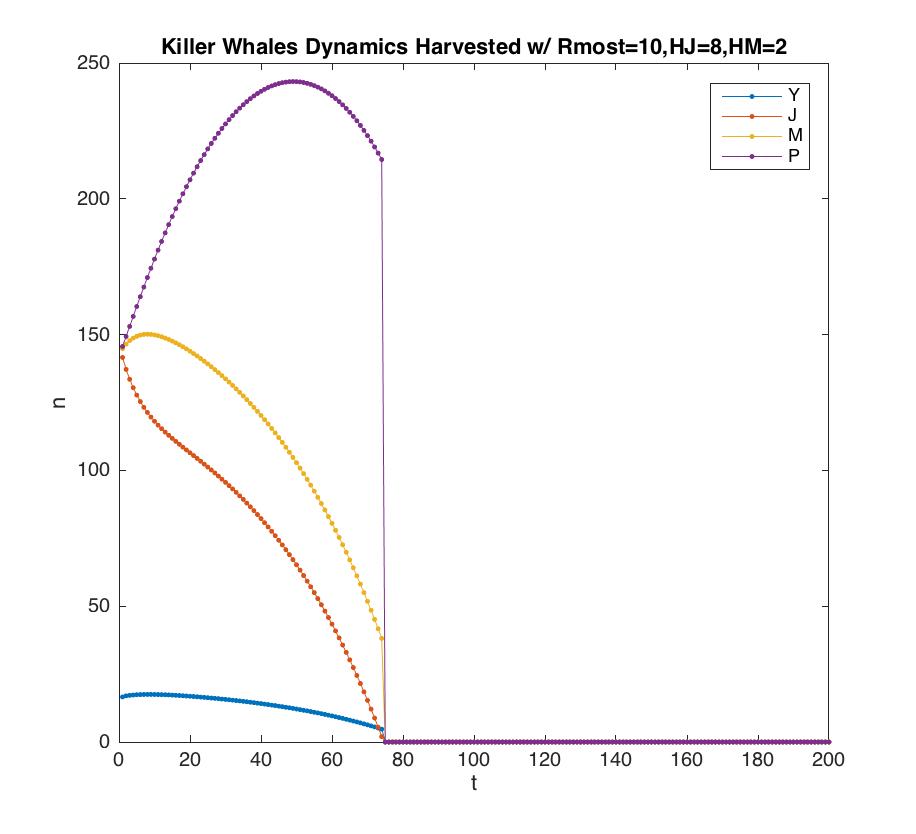
1) If we consider juvenile not as adult, only counting mature:

I looped to harvest 1 more juveniles until maximum and 1 more mature, each step and break when it first extinct within 200 steps. The maximum mature to take without extinction is **6,** because if we go to 7, it goes to extinction (Figure 5).



2) If we consider juvenile as adult as well, not just mature:

I looped to harvest 1 more juveniles until maximum and 1 more mature, each step and break when it first extinct within 200 steps. The maximum juvenile and mature to take without extinction is **9,** because if we go to 10, it goes to extinction (Figure 6, 7, 8).



% @Author: Baihan Lin

% @Date: Oct 2016

%% Question 3

% initiate projection matrix A.

A3 = [0 0.0043 0.1132 0;

0.9755 0.9111 0 0;

0 0.0736 0.9534 0

0 0 0.0452 0.9804]

N = 250;

% calculate dominant eigenvalue lamda

[V3, L3]=eig(A3);

L3 = diag(L3);

j3=find(abs(L3) == max(abs(L3)));

dominant\_eigenvalue=L3(j3) % 1.0254

w3 = V3(:,j3); %[0.0663;0.5663;0.5793;0.5825]

% simulation of unharvested case

n\_zero=w3\*N;

Tmax=200;

n\_vs\_t=zeros(4,Tmax);

n\_vs\_t(:,1)=n\_zero ;

for t=2:Tmax;

n\_vs\_t(:,t)=A3\*n\_vs\_t(:,t-1) ;

end

fig1 = figure;

set(gca,'FontSize',20);

plot(1:Tmax,n\_vs\_t','.-','MarkerSize',14);

xlabel('t','FontSize',20);

ylabel('n','FontSize',20);

title('Killer Whales Dynamics Unharvested');

legend('Y', 'J', 'M','P');

% simulation of harvested case

n\_zero=w3\*N;

Tmax=200;

n\_vs\_t\_harvested=zeros(4,Tmax);

n\_vs\_t\_breakJ=zeros(4,Tmax);

n\_vs\_t\_breakM=zeros(4,Tmax);

n\_vs\_t\_breakR=zeros(4,Tmax);

n\_vs\_t\_harvested(:,1)=n\_zero ;

n\_vs\_t\_breakJ(:,1)=n\_zero ;

n\_vs\_t\_breakM(:,1)=n\_zero ;

n\_vs\_t\_breakR(:,1)=n\_zero ;

Jmost = 0;

Mmost = 0;

RmostSet = [];

RmostJ = [];

RmostM = [];

% find the smallest J possible to get extinction.

% thus the maximum J not to extinct is Jmost - 1

for x = 0:1:100

h = [0;x;0;0];

for t=2:Tmax;

check = prod(((A3\*n\_vs\_t\_harvested(:,t-1)-h)>=0));

n\_vs\_t\_harvested(:,t)=check\*(A3\*n\_vs\_t\_harvested(:,t-1)-h);

end

if (prod((n\_vs\_t\_harvested(:,Tmax)==0))==1)

Jmost = x; %10, the maximum J not to extinct is Jmost - 1 = 9

n\_vs\_t\_breakJ(:,:)=n\_vs\_t\_harvested(:,:);

break;

end

end

figure;

set(gca,'FontSize',20);

plot(1:Tmax,n\_vs\_t\_breakJ','.-','MarkerSize',14);

xlabel('t','FontSize',20);

ylabel('n','FontSize',20);

title(strcat('Killer Whales Dynamics Harvested w/ Jmost=',num2str(Jmost)));

legend('Y', 'J', 'M','P');

% find the smallest M possible to get extinction.

% thus the maximum M not to extinct is Mmost - 1

for x = 0:1:100

h = [0;0;x;0];

for t=2:Tmax;

check = prod(((A3\*n\_vs\_t\_harvested(:,t-1)-h)>=0));

n\_vs\_t\_harvested(:,t)=check\*(A3\*n\_vs\_t\_harvested(:,t-1)-h);

end

if (prod((n\_vs\_t\_harvested(:,Tmax)==0))==1)

Mmost = x; %7, the maximum M not to extinct is Mmost - 1 = 6

n\_vs\_t\_breakM(:,:)=n\_vs\_t\_harvested(:,:);

break;

end

end

figure;

set(gca,'FontSize',20);

plot(1:Tmax,n\_vs\_t\_breakM','.-','MarkerSize',14);

xlabel('t','FontSize',20);

ylabel('n','FontSize',20);

title(strcat('Killer Whales Dynamics Harvested w/ Mmost=',num2str(Mmost)));

legend('Y', 'J', 'M','P');

% find the smallest reproductive adults possible to get extinction.

% thus the maximum R not to extinct is Rmost - 1

for x = 0:1:Jmost

for y = 0:1:100

h = [0;x;y;0];

for t=2:Tmax;

check = prod(((A3\*n\_vs\_t\_harvested(:,t-1)-h)>=0));

n\_vs\_t\_harvested(:,t)=check\*(A3\*n\_vs\_t\_harvested(:,t-1)-h);

end

if (prod((n\_vs\_t\_harvested(:,Tmax)==0))==1)

RmostSet = [RmostSet x+y];

RmostJ = [RmostJ x];

RmostM = [RmostM y];

break;

end

end

end

Rmost = max(RmostSet); % 10, maximum R not to extinct is Rmost - 1 = 9

JRmost = RmostJ(find(RmostSet == Rmost));

MRmost = RmostM(find(RmostSet == Rmost));

% plot each possible J and M breakpoints.

for k = 1:length(JRmost)

x = JRmost(k);

y = MRmost(k);

h = [0;x;y;0];

for t=2:Tmax;

check = prod(((A3\*n\_vs\_t\_breakR(:,t-1)-h)>=0));

n\_vs\_t\_breakR(:,t)=check\*(A3\*n\_vs\_t\_breakR(:,t-1)-h);

end

figure;

set(gca,'FontSize',20);

plot(1:Tmax,n\_vs\_t\_breakR','.-','MarkerSize',14);

xlabel('t','FontSize',20);

ylabel('n','FontSize',20);

title(strcat('Killer Whales Dynamics Harvested w/ Rmost=',num2str(Rmost),',HJ=',num2str(x),',HM=',num2str(y)));

legend('Y', 'J', 'M','P');

end

1. **MATLAB programming tools or tricks**
2. **clear all; close all; clc;**

I like to add this line to all my codes. It ensures no variables and figures get into the way of this round of analysis. I personally find this a good habit to make sure you have a clear workspace before the current programs run.

1. **diary('diary.txt'); diary on;**

I like to add this line to my code, this function diary save text of MATLAB session to a file. Diary function causes a copy of all subsequent command window input and most of the resulting command window output to be appended to the named file. I personally find this a good habit to keep a record of all the output. This is also very helpful when debugging.

1. **prompt = 'What is your folder?: '; path = input(prompt,'s');**

I like to use this line to create some interaction in my MATLAB session. This is very helpful for other people to run your code in a more general base. This facilitates coding collaboration.

1. **Project Warmup, E+G Ex. 1.1**
2. **Complete citation**

Yu, J., Xiao, J., Ren, X., Lao, K. and Xie, X.S., 2006. Probing gene expression in live cells, one protein molecule at a time. *Science*, 311(5767), pp.1600-1603.

1. **Purpose of the model**

This article tries to stochastically model the gene expression of a fusion protein of a fast-maturing yellow fluorescent protein (YFP) and a membrane-targeting peptide under a repressed condition. And they tried to use this model to explain the mechanism of the in vitro real-time production of single protein molecules in individual *Escherichia coli* cells.

1. **State variables**

Because it is in a single protein molecule level model, the state variables are the numbers of Tsr-Venus, *lac* repressor, *lac* operator, mRNA, ribosome molecules.

1. **Simplifying assumptions**

When they assigned the rate limiting step for protein production, they made the assumption that one mRNA molecule per gene expression burst, when this is not always true. They also derive the autocorrelation function based on the following three assumptions: (1) the transcription initiation events are temporally uncorrelated (i.e. a Poisson processes); (2) the number of protein molecules translated from one mRNA follows a geometric distribution; (3) there is one rate-limiting step in the post-translational assembly of the protein, and transcription/translation times are short.