Lukas Fu - HW1 problem 1

Lukas Fu

Problem 3

The dynamics of a population described by the Ricker map can be modelled by the following equation:

$$\eta_{\tau+1} = R\eta_{tau}e^{-\alpha\eta_{tau}} \quad (1)$$

The dynamics are simulated for 300 steps, of which only the last 100 (steps 200-300) are included. The decision to exclude the initial 200 steps is to exclude the unstable parts so that the stable 100 steps (the interesting steps where the period-doubling cascade is present) is included.

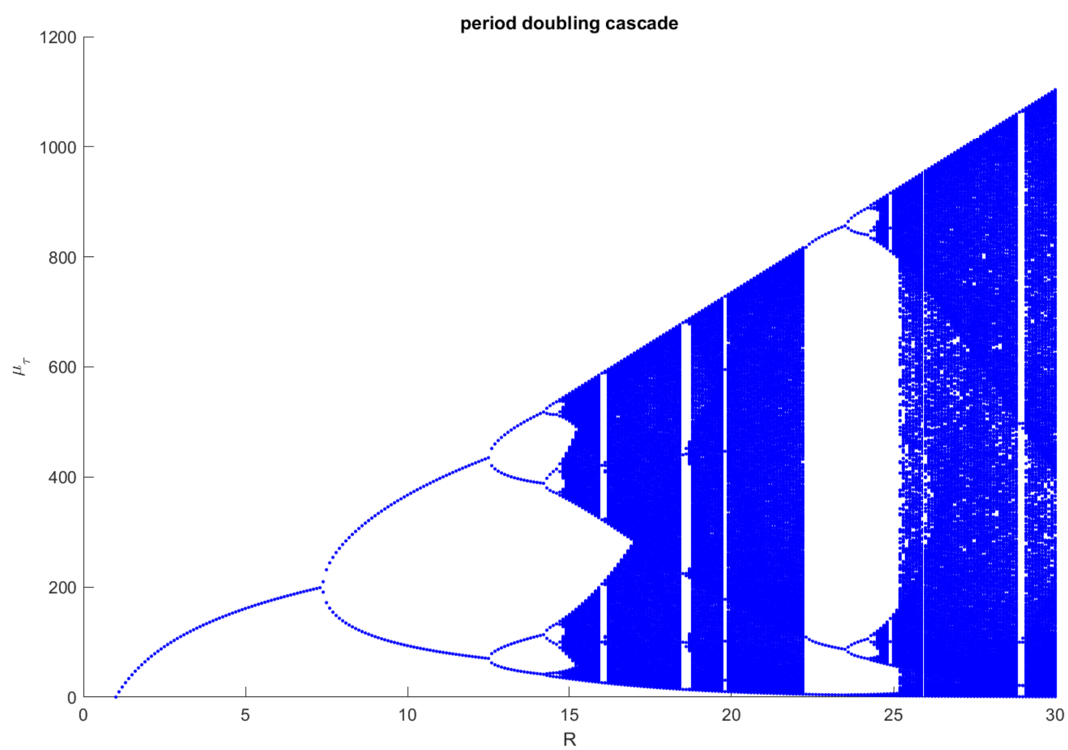
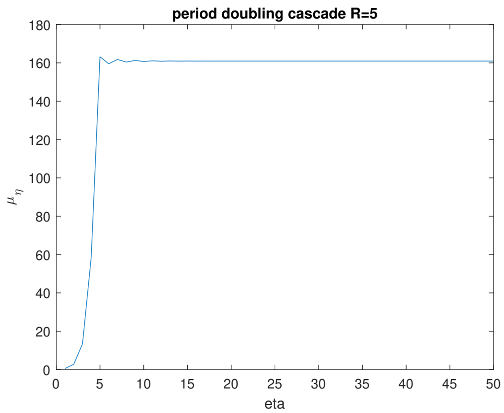
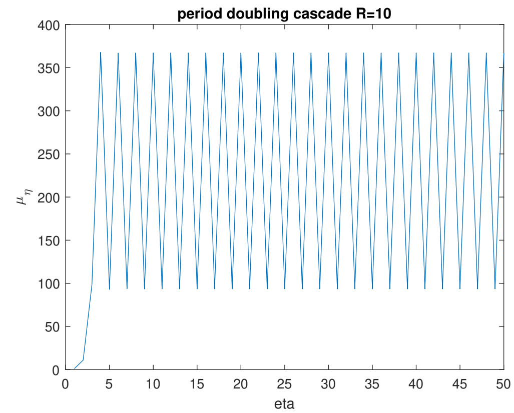


Figure 1: Bifurcation diagram showing the period-doubling cascade

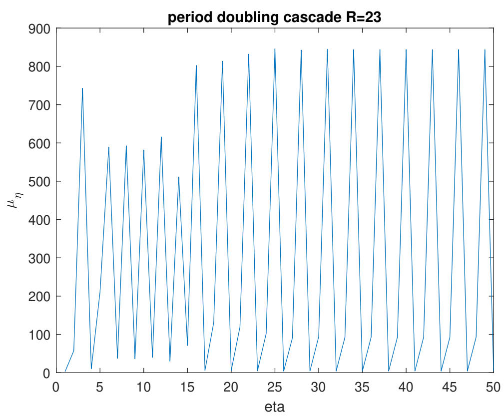
Looking at the bifurcation diagram, the amount of branches present at some value of R would be cycle points. The stable (one-point), two-point and four-point cycles can clearly be seen before the chaotic behaviour starts, for example at $R = 5$, $R = 10$ and $R = 13$ respectively. The three-point cycle occurs at a higher value of R, around $R = 23$. The following figures show the snapshots of each point cycle.



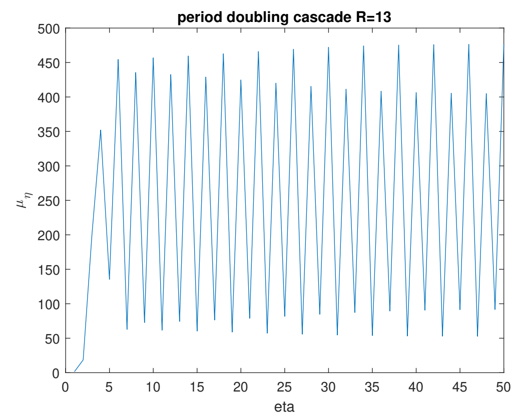
(a) Stable (one-point) cycle



(b) Two-point cycle



(c) Three-point cycle



(d) Four-point cycle

Figure 2: Snapshots of one, two, three and four-point cycles for $R = 5; 10; 23; 13$ respectively.

To more accurately pinpoint the bifurcation points, the dynamics are simulated for a smaller interval of R to manually zoom in on the bifurcation point. The first point of bifurcation, going from a stable cycle to a two-point cycle occurs at $R = 7.33$, and the second point of bifurcation going from a two-point cycle to a four-point cycle occurs at $R = 12.25$.

To find R_∞ we plot the amount of periods that take place for each value of R. In figure 3 it can be observed that the first time period quantity heavily increases happens for $R = 14.77$. Additionally the first and second doubling, going from stable to two-point to four-point cycle can be seen for $= 7.33$ and $R = 12.25$, as well as the eight and sixteen-point cycles closely before $R_\infty = 14.77$.

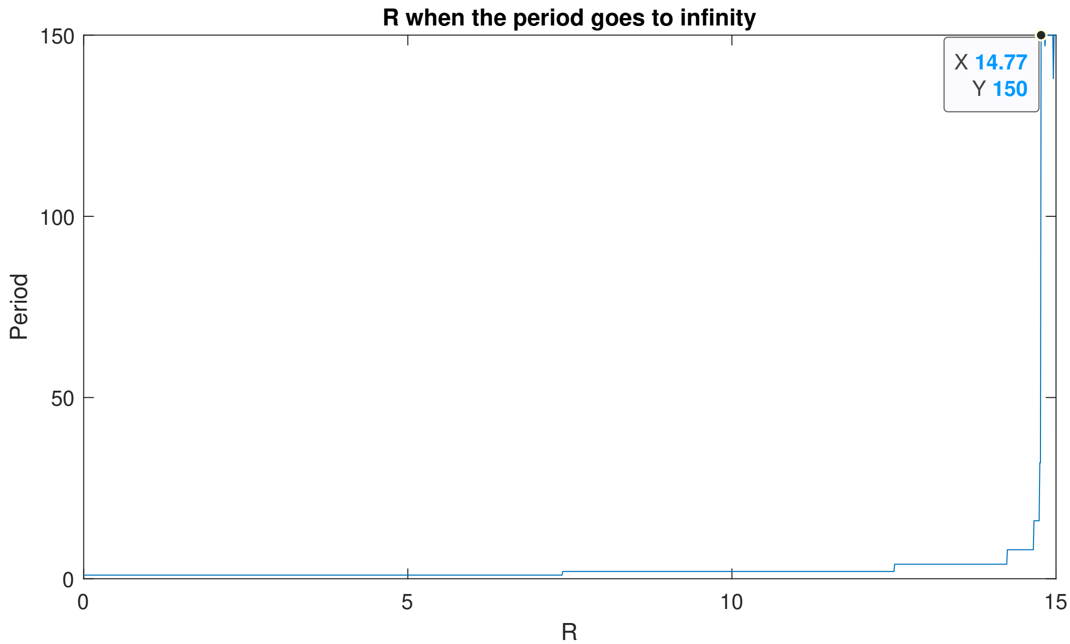


Figure 3: Values of R plotted against the period.

Code

Problem 3

```
%% a
clf
T=1000000;
K=2000;
Rs = ones(K,1);
figure
hold on

for R = 1:0.1:30
    vec = runDynamics(T,R);
    plot(Rs*R,vec(T+1-K:end),'b.')
end
xlabel('R')
ylabel('\mu_\tau')
title('period doubling cascade')


%% b
Rlist=[5,10,13,23];
for i =1:4
    figure
    vec = runDynamics(50,Rlist(i));
    plot(vec)
    xlabel('eta')
    ylabel('\mu_\eta')
    title('period doubling cascade R='+string(Rlist(i)))
end
vec1=vec(2+delay:end);
vec2=vec(1+delay:end-1);

%% c
clf
T=500;
Rs = ones(100,1);
xt=[];yt=[];
for R = 12.4:0.01:12.5
    vec = runDynamics(T,R);
    xt=[xt;Rs*R];yt=[yt;vec(201:300)];
end

plot(xt,yt,'b.')
%%
%7.33=1->2   12.25=2->4      14.15=4->8   14.6=8->16     14.738=16->32
x=[7.33,12.25,14.15,14.6,14.738];
x2=x(2:5)-x(1:4);
x=fliplr(x);
y=[1,2,3,4];

T=6000;
for R=12.35:0.001:12.44
```

```
    clf
    hold on
    vec = runDynamics(T,R);
    vec2=vec(end-50:end);
    plot(vec2,'b')
    %xlim([900,1000])
    %ylim([432,433])
    [val,loc]=findpeaks(vec2);
    %plot(901+loc,vec2(loc),'rx')
    if abs(max(val)-min(val))>=0.000001
        R
        break;
    end
    title(R)
    pause(0.01)
end
%7.33 --> stable oscillation  12.3973 (no longer)

%% d
T=1000000;
Rs = ones(100,1);
Rmin=0;
Rmax=15;
dt=0.01;
period = zeros(length(Rmin:dt:Rmax),1);
R=Rmin:dt:Rmax;
clf
hold on
Rf=0;%14.82
for iR = 1:length(R)
    vec = runDynamics(T,R(iR));
    loglog(Rs*R(iR)-Rf,vec(T+1-100:T),'b.')
    period(iR)=getPeriod(vec(T+1-150:T),0.001);
end
set(gca, 'XScale', 'log');
xlim([-16,0])
ylim([300,500])
clf
plot(R,period);
xlabel('R')
ylabel('Period')
title('R when the period goes to infinity')
%% functions
function period = getPeriod(vec,dt)
    ex = [vec(1)];
    for iT =2:length(vec)
        bool = abs(ex-vec(iT))<=dt;
        if max(bool)
        else
            ex=[ex;vec(iT)];
        end
    end
    period=length(ex);
end
```

```
function [a,N0] = getVariables()
    a = 0.01;
    N0 = 900;
end
function val = getNext(R,N)
    [a,~] = getVariables;
    val = R*N*exp(-a*N);
end
function vec = runDynamics(t,R)
    [~,N0] = getVariables;
    vec=zeros(t,1);
    vec(1) = getNext(R,N0);
    for i=2:t
        vec(i) = getNext(R,vec(i-1));
    end
end
```