ELEN 160 Project

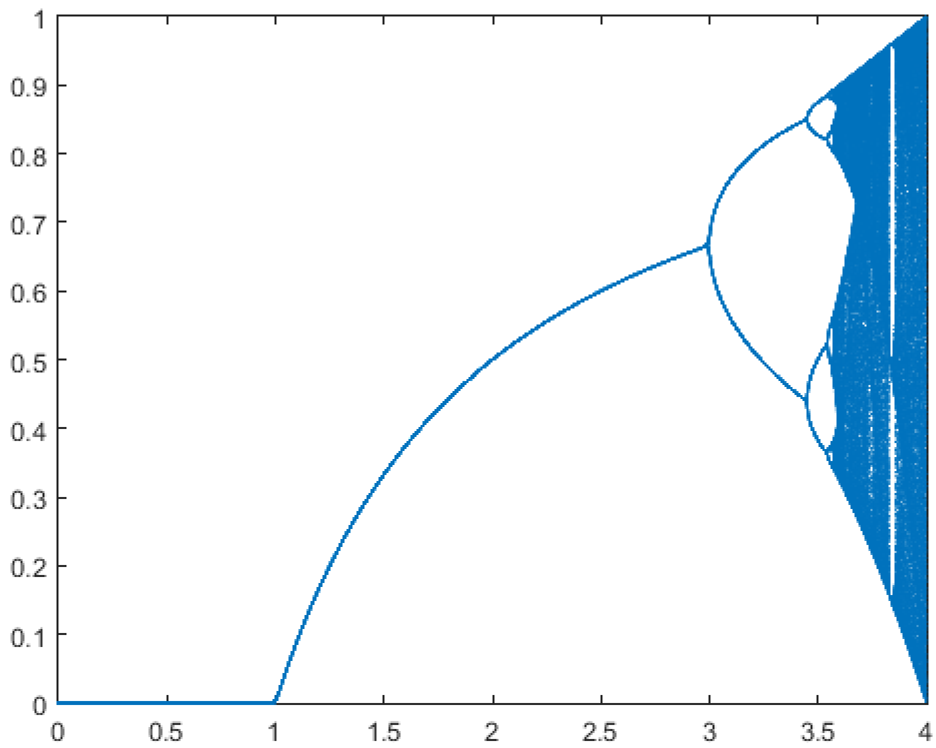
Table of Contents

By: Devon Quaternik & Matthew Salmanpour

Problem 1

```
%This is the code for the sequencer
%function [ xk ] = sequencer( x0,p,n )
% %Compute a sequence x(k) from x(0),p, and Kmax=n.
%
% xk=zeros(1,n);
% t=1;
% xk(1,t)=x0;
% while t<n
%     xk(1,t+1)=p*xk(t)*(1-xk(1,t));
%     t=t+1;
% end

pmin=0.001;  %matlab indices must start above 0
pmax=3.999;
x0=0.5;      %chosen x0
n=400;       %chosen length of sequence

p2=round(pmax*1000);
count=1;
px=zeros(1,n);

while count<p2
    pin=count/1000; %go back to decimal p
    for count1=1:n
        xk=sequencer(x0,pin,n); %gets the sequence for this p
        if count1>(n/2)         %removes first half to avoid
 transients
            count2=count1+(n-2)*count; %counts all the iterations
            iter(count2)=xk(count1);
            px(count2)=pin;
        end
    end
    count=count+1;
end

%plots the bifurcation diagram with small markers for visibility
plot(px,iter,'.','MarkerSize',2);
```

```
axis([floor(pmin),ceil(pmax),0,1])
```



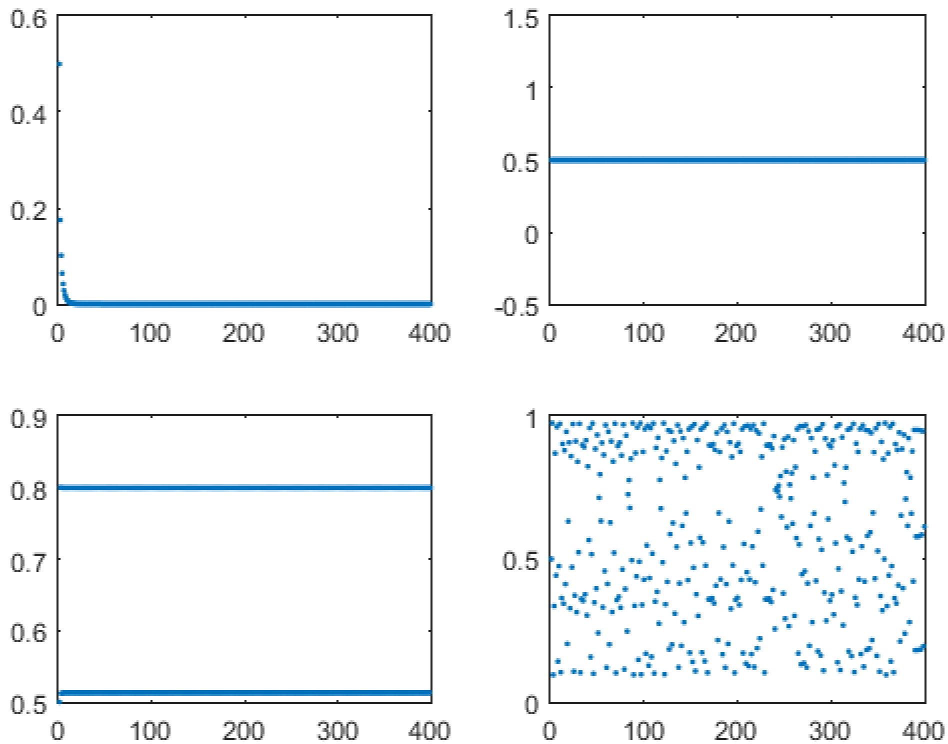
Problem 2

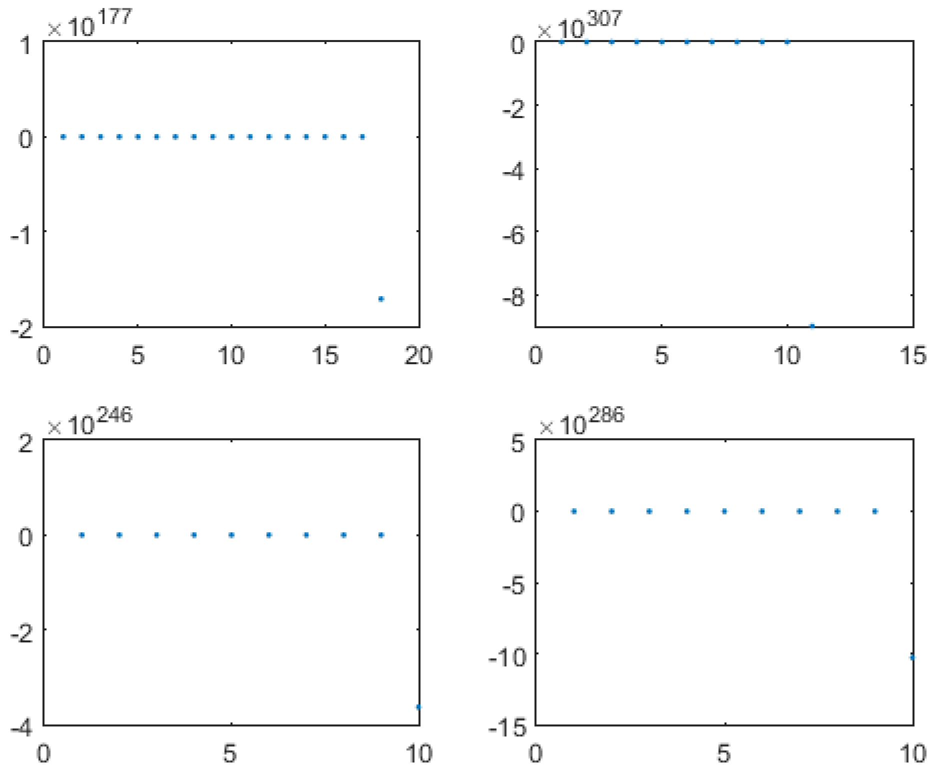
```
%Part A
%This does not remove transients at the beginning
x0=.5;
n=400;
p=.7;

for count1=1:n
        xk1=sequencer(x0,p,n); %gets the sequence for zero region
end


p2=2;

for count3=1:n
        xk2=sequencer(x0,p2,n); %gets the sequence for this p
end

p3=3.2

for count4=1:n
        xk3=sequencer(x0,p3,n); %gets the sequence for period doubling
end
```

```
p4=3.9;

for count5=1:n
        xk4=sequencer(x0,p4,n); %gets the sequence for the chaos
 region
end

%plots the four sequences together
figure(2);
subplot(2,2,1), plot(xk1,'.');
subplot(2,2,2), plot(xk2,'.');
subplot(2,2,3), plot(xk3,'.');
subplot(2,2,4), plot(xk4,'.');

%Part B.
%This part uses the same code for part a, just with x0 as 1.5 instead
 of .5
x0=1.5;
n=400;
p=.7;

for count1=1:n
        xk1=sequencer(x0,p,n); %gets the sequence for this p
end


p2=2;

for count3=1:n
        xk2=sequencer(x0,p2,n); %gets the sequence for this p
end

p3=3.2;

for count4=1:n
        xk3=sequencer(x0,p3,n); %gets the sequence for this p
end

p4=3.9;

for count5=1:n
        xk4=sequencer(x0,p4,n); %gets the sequence for this p
end

%plots the four sequences together
figure(3)
subplot(2,2,1), plot(xk1,'.');
subplot(2,2,2), plot(xk2,'.');
subplot(2,2,3), plot(xk3,'.');
subplot(2,2,4), plot(xk4,'.');

%As can be seen from the plots, using x0 at 1.5 instead of 0.5 does
 not
```

```
%work effectively. This is due to negative values that show up in the
%[1-x(k)] portion of the logistic map equation intially. This is
 supposed
%scaled between 0 and 1 and so x0 = 1.5 is problematic.
```

```
p3 =

    3.2000
```



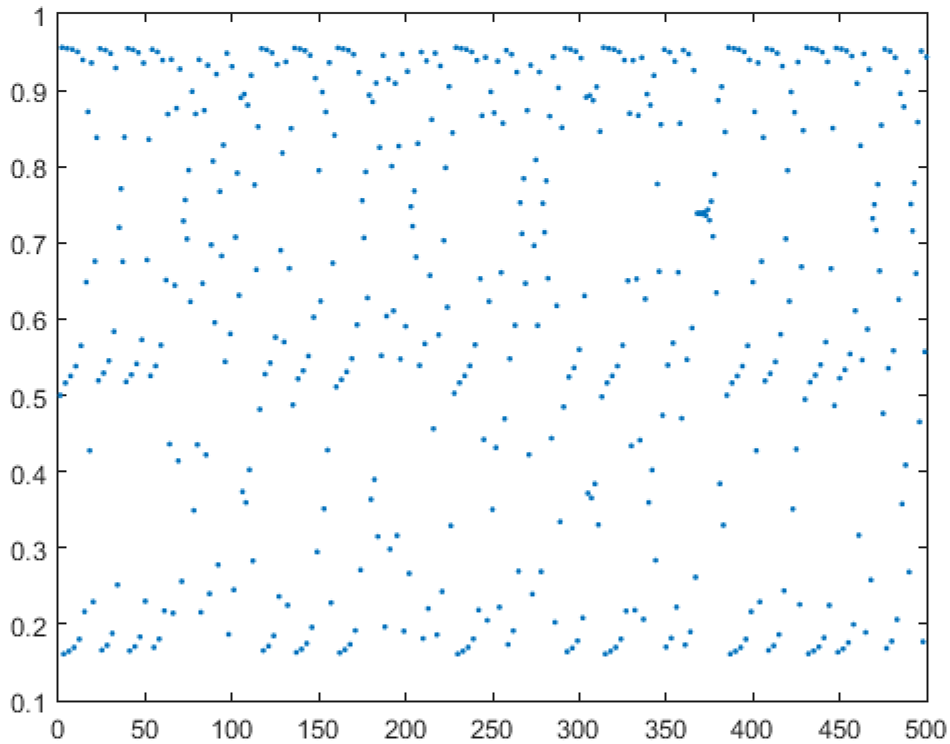


Problem 3

```
%The Answer to Problem 3 can be found in the pages following Problem 5
```

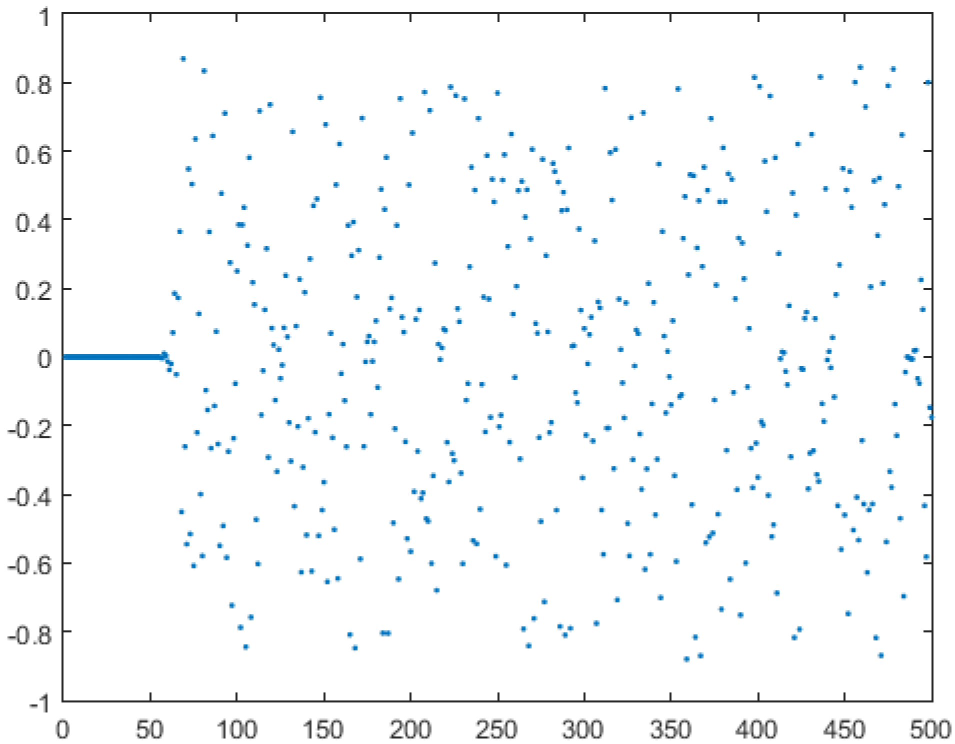
Problem 4

```
x0=.5;
n=500;
p=3.82394; %this is the value that gives one disturbance around k=375

for count1=1:n
        xk1=sequencer(x0,p,n); %gets the sequence for this p
end

figure(4);
plot(xk1,'.');
```



Problem 5

```
x0=.5;
n=500;
p=3.9;

for count1=1:n
        xk1=sequencer(x0,p,n); %gets the sequence for this p
end

x01=.50000001;
for count1=1:n
        xk2=sequencer(x01,p,n); %gets the sequence x0 10^-8 away
 from .5
end
xk=xk1-xk2; %find difference of two sequences

figure(5);
plot(xk,'.');

%It can be visibly seen from the plot that it takes until x(k) = 50
 for the
%solutions to become visibly distinct. Until x(k) = 50 it is uniform
 and
%after that the chaos can be seen.
```



Published with MATLAB® R2015a