VINIT KUMAR SINGH: 16PH20036

Assignment – 2

Computer arithmetic 3: Consider the logistic map: $x_{n+1} = Ax_n\,(1 - x_n)$, where, x_n is the n^{th} iteration of *x* for a starting value of $0 \leq x \leq 1$. Here A is a constant.

(a) Write a code to generate the logistic map. Start by varying the value of *A* to observe the following:

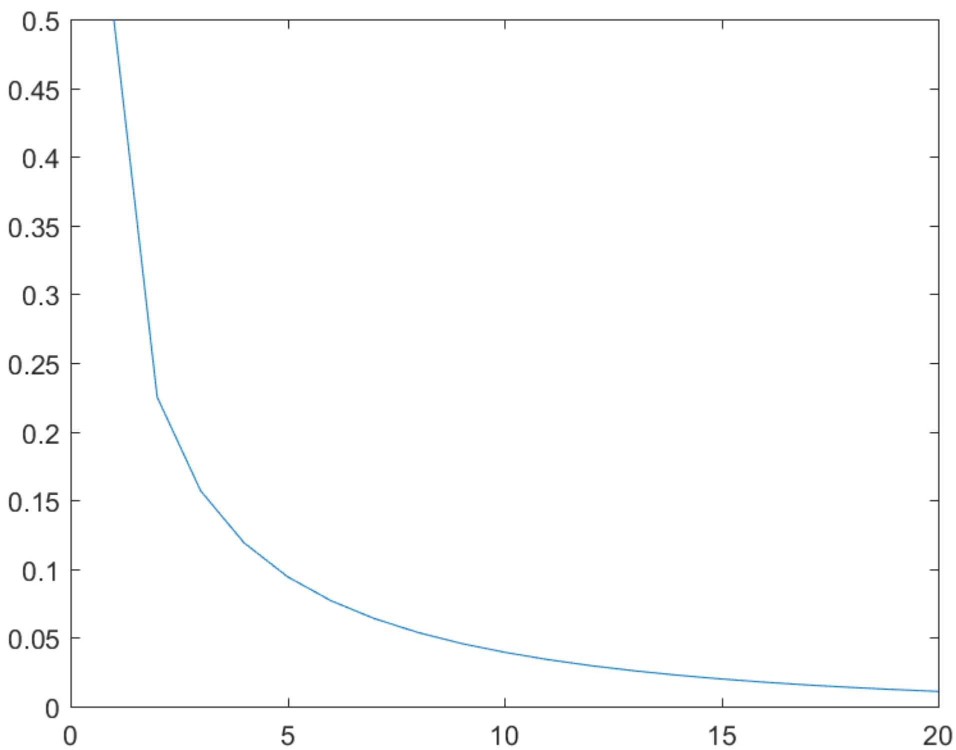
Solution:

```
function result = logi_1(A,x,n)
X=zeros(1,n);
X(1)=x;
for i=2:n
    X(i)=A*X(i-1)*(1-X(i-1));
end
plot(1:n,X);
result=X(length(X));
end
```

- With *A* between 0 and 1, the value of x_n will eventually go to zero, independent of the initial value of *x*.

A=0.9

X=0.5

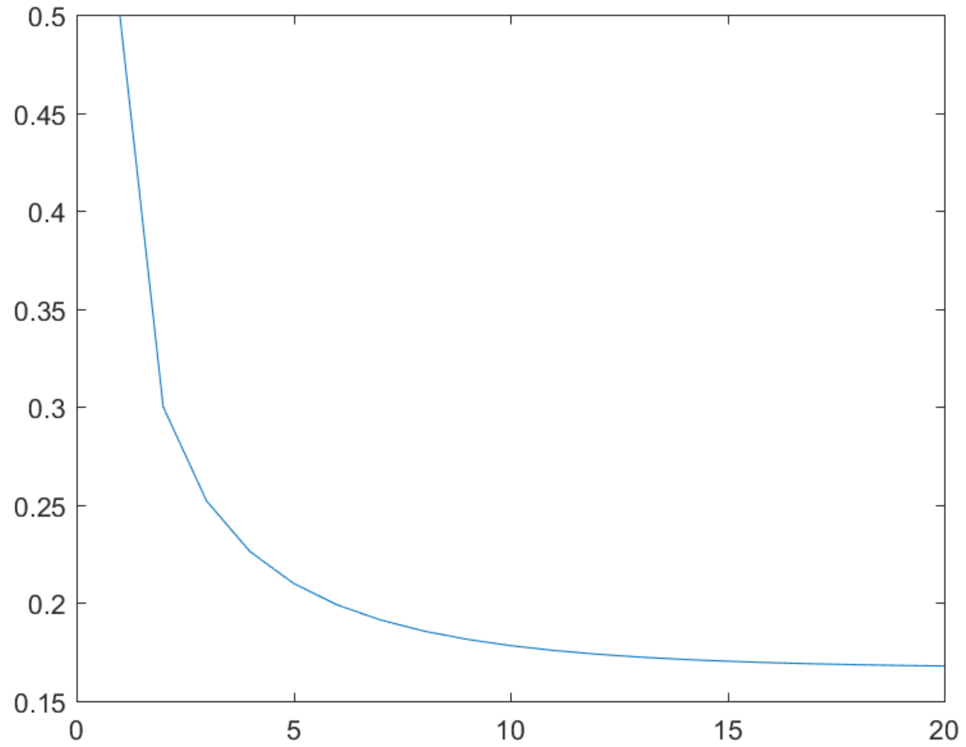


- With A between 1 and 2, the population will quickly approach the value.

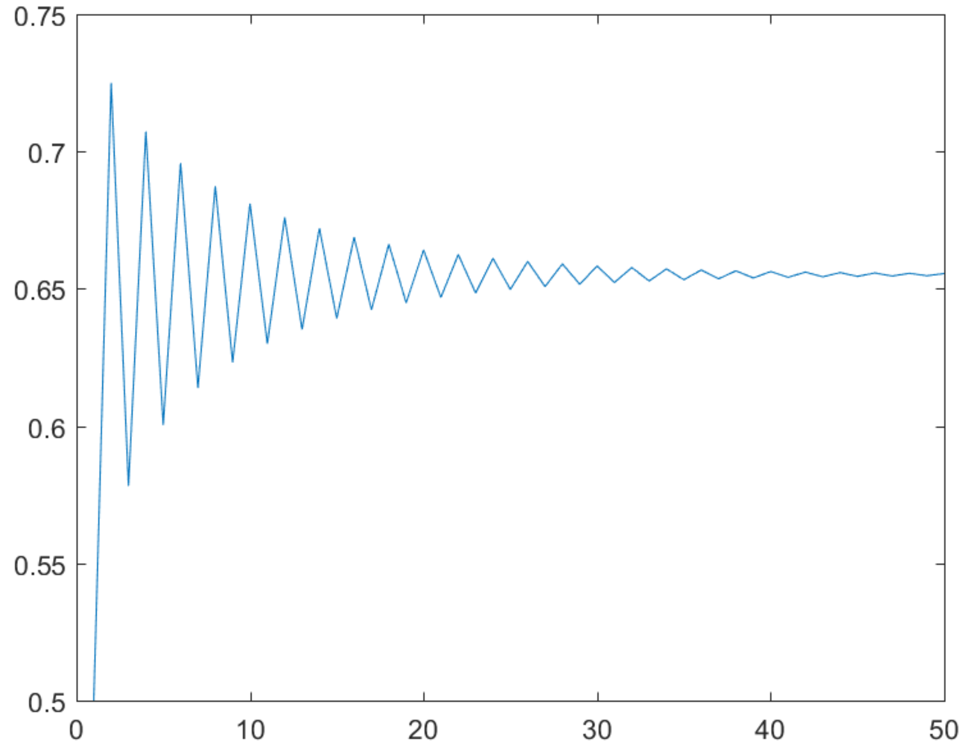
,

A=1.2

X=0.5



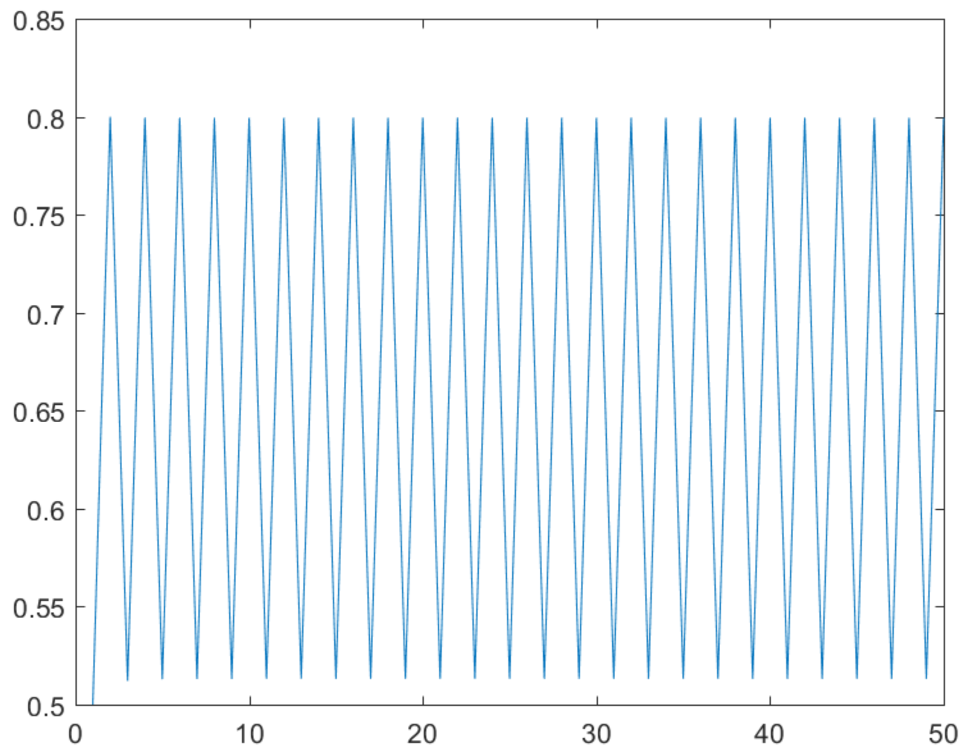
- With A between 2 and 3, xn will also eventually approach the value.



```
A=2.9
X=0.5
```

- With A between 3 and 1+6=3.449, from almost all initial conditions x_n will approach permanent oscillations between two values.

```
A=3.2
X=0.5
```



(b) For an initial value $x=0.3$, vary the value of A from 0.5 to 3.99 in total 250 steps. For each value of A, note the values of x_n for $n=150$, then make a plot

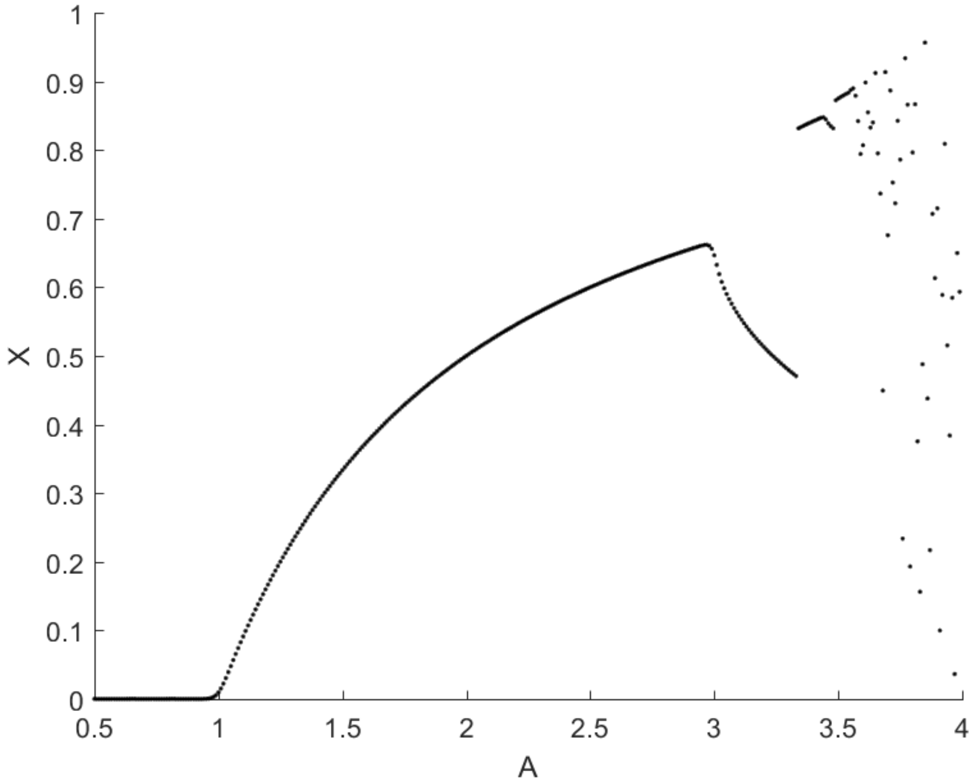
of A *vs.* x_n and see the bifurcation and chaos. Now change the initial value of x.

Do you see any change in the plot?

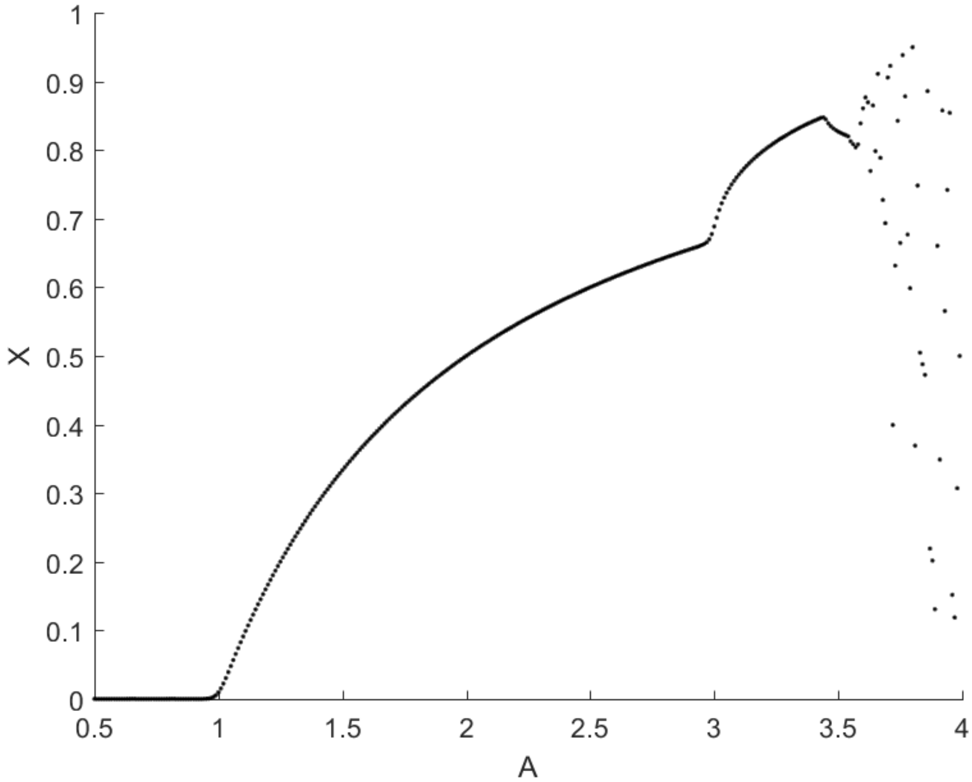
```
% To see the behaviour. I plotted the graph by varying
both x and A.

x=0.01:0.001:0.99;
hold on;
for j=1:length(x)
A=0.5:0.01:3.99;
L=zeros(1,length(A));
for i=1:length(A)
    L(i)=logi_1(A(i),x(j),100);
end
plot(A,L,'k.','MarkerSize',4)
```

```
end
xlabel('A')
ylabel('X')
hold off;
```

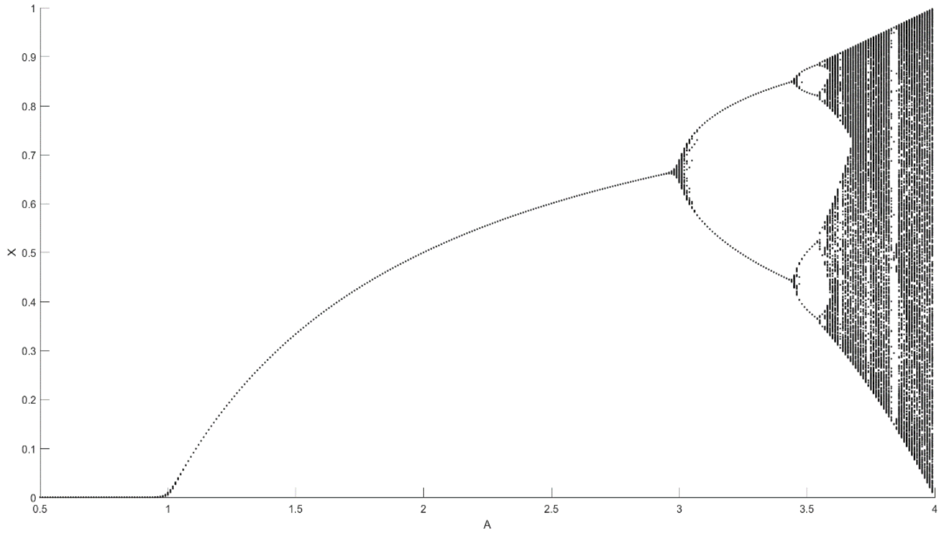
X=0.3



X=0.5

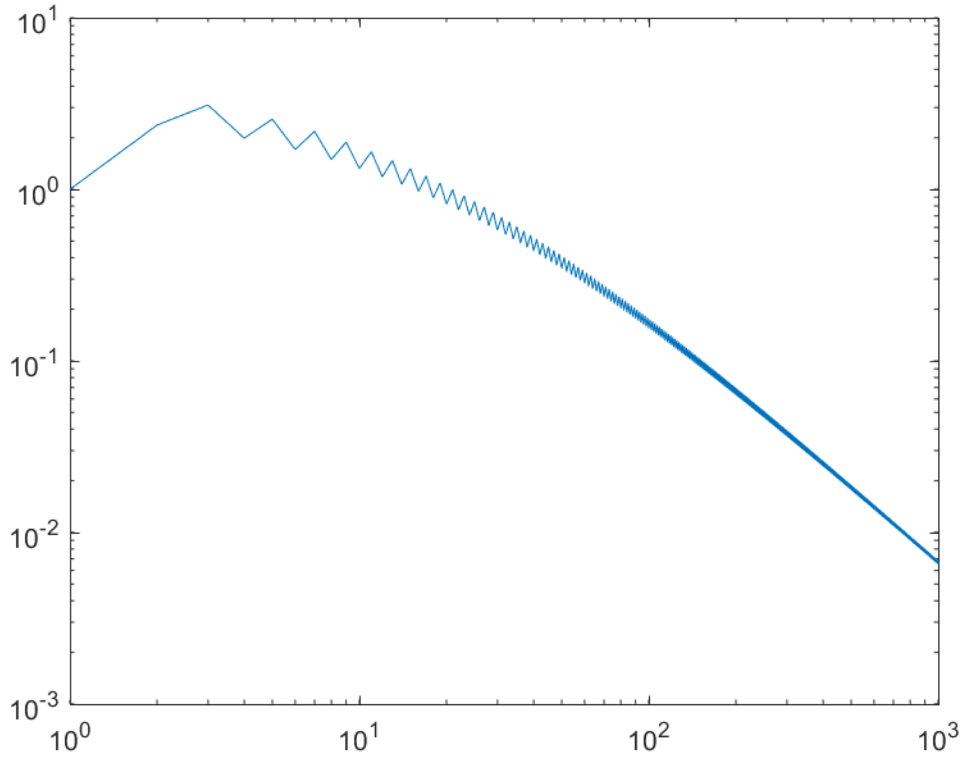


Variable X.

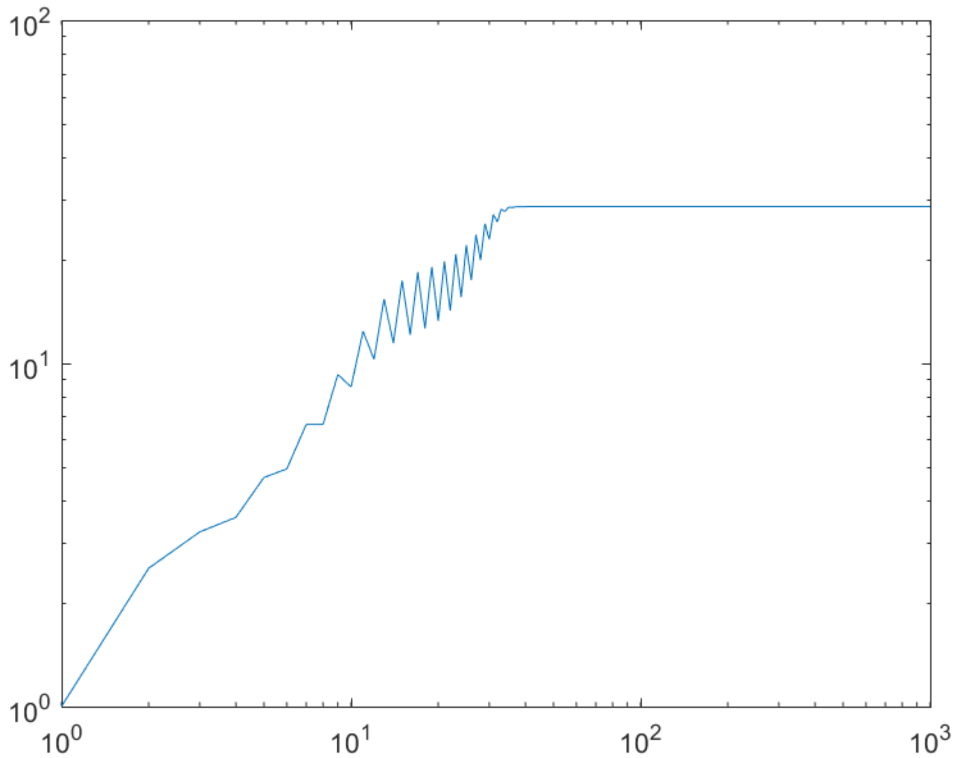


(c) For $A = 3.0$, choose two points x and x' close to 1 where, $x' = x + 0.01$ and iterate. Plot $log(|x_n - x'_n| / 0.01)$ as a function of n in log-log representation. See if it is approaching a straight line for high n. Check for other values around 3, and you will see error dropping to 0 much sooner. This happens because at A=3, near the bifurcation memory, the systems approaches equilibrium in a dramatically slow manner.

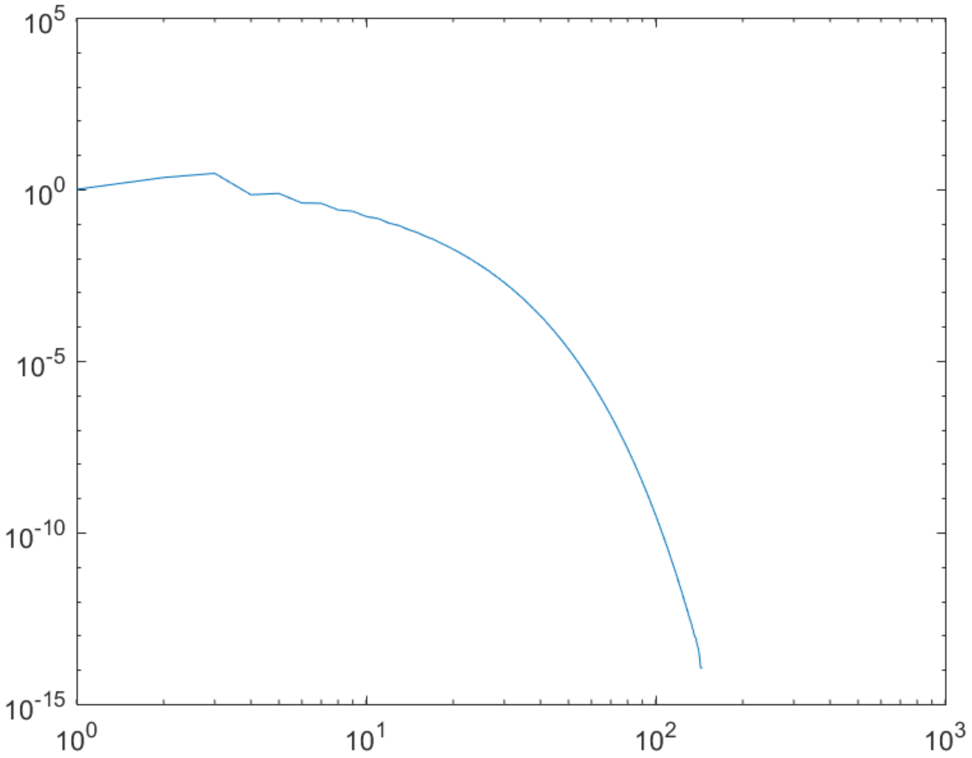
```
function result = logi_3(A,x,n)
X=zeros(1,n);
X_=zeros(1,n);
X(1)=x;
X_(1)=x-0.01;
for i=2:n
    X(i)=A*X(i-1)*(1-X(i-1));
    X_(i)=A*X_(i-1)*(1-X_(i-1));
end

loglog(1:n,(abs(X-X_)/0.01));
result=0;
end
```



A=3.0



A=3.2



A=2.8

Root finding 1: How many real roots does the polynomial $f(x) = 2x^3 - 5$ has? Find the roots using the method of bisection.

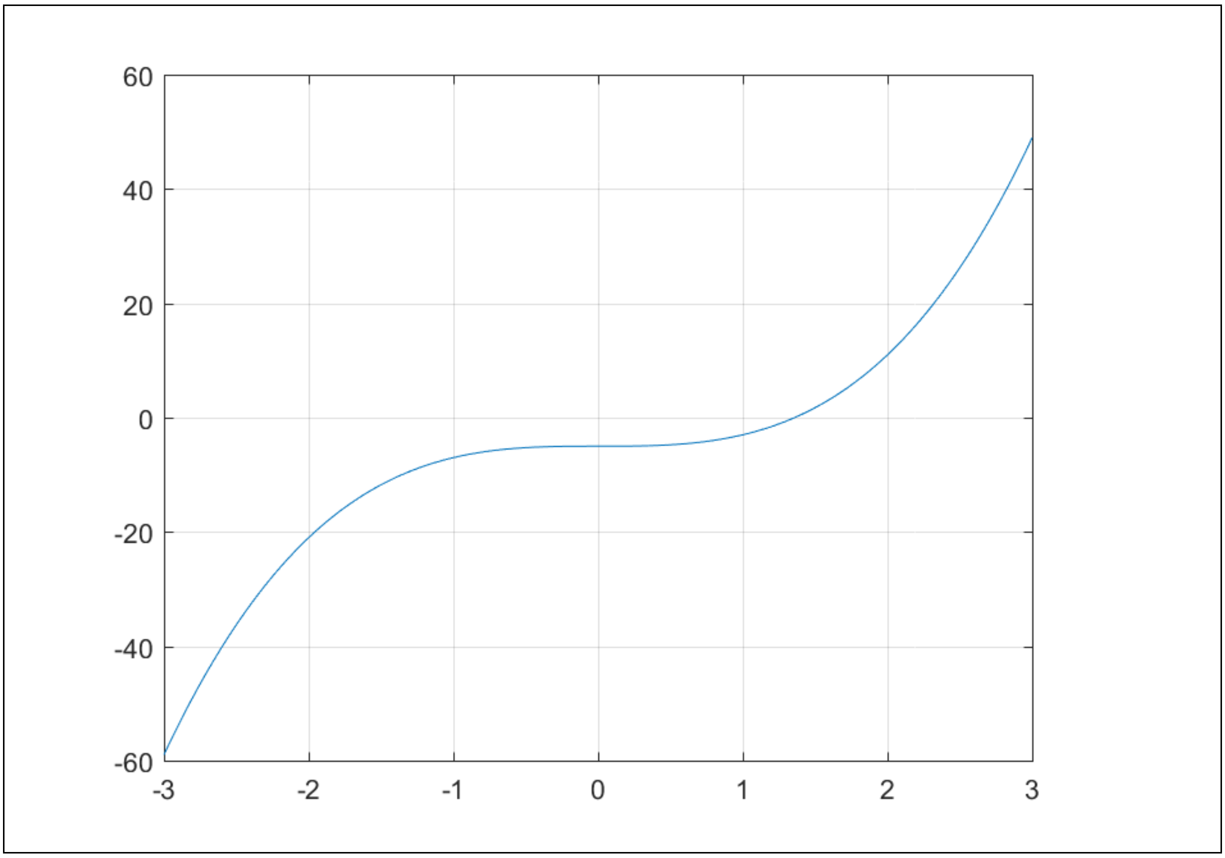
```
a=1;
b=2;
while abs(b-a)>0.0001 && f1(a)*f1(b)<0
    mid=(a+b)/2;
    if f1(a)*f1(mid)<0
        b=mid;
    else
        a=mid;
    end
end
disp(a);

X=-3:0.01:3;
plot(X,f1(X))
grid on;
```

Function:

```
function res = f1 (x)
    res=2*(x.^3)-5;
end
```

Result:

X = 1.3572



Root finding 2: Solve the equation $f(x) = x^3 - 0.165x^2 + 3.99310^{-4}$ using Newton-Raphson method.

1) With initial guess of *x(0) = 0.05*.
2) With initial guess of *x(0)=0.11*,

Why does the 2nd case do not offer any solution? You may start of by plotting the function.

3) Can you find another initial guess which will lead to no solution? Explain why.

```
x=0.05;
while abs(f2(x))>1e-7
    x=x-f2(x)/f2_(x);
end
disp(x);

x=0.16;
while abs(f2(x))>1e-7
    x=x-f2(x)/f2_(x);
end
disp(x);

X=0:0.001:0.2;
plot(X,f2(X))
grid on;

x=20; %%0.05
while abs(f2(x))>1e-7
    x=x-f2(x)/f2_(x);
    if abs(f2(x))>1000
        disp('Diverged');
        break;
    end
end
disp(x);
```

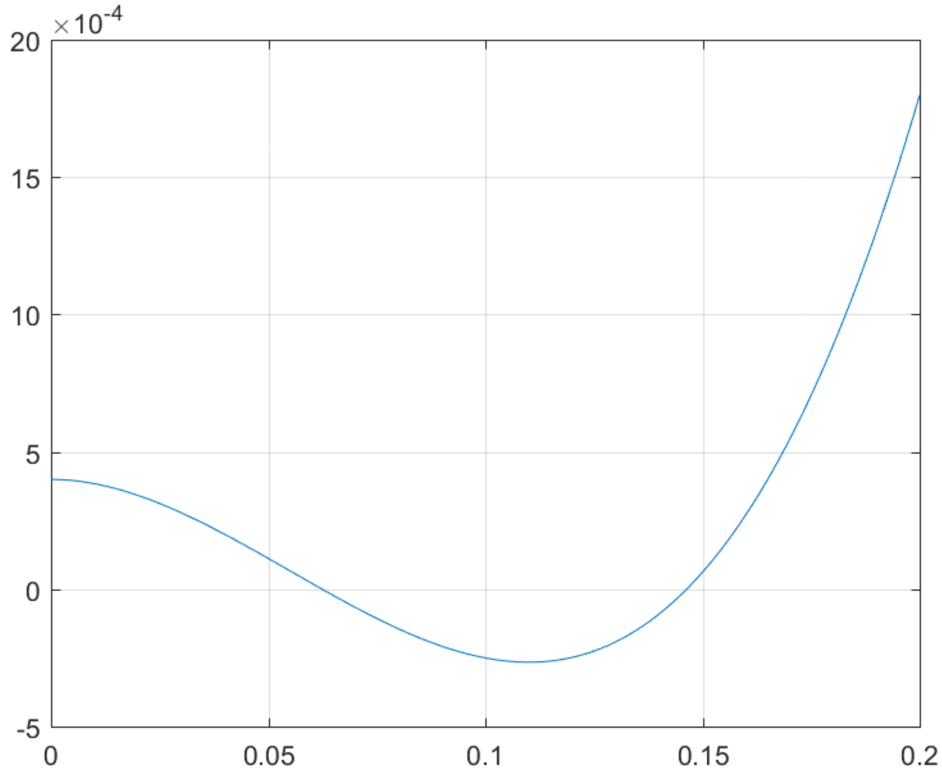
Functions:

```
function res = f2 (x)
    res=x.^3 - 0.165*x.^2 + 3.993e-4;
end

function res = f2_ (x)
    res=3*x^2 - 0.33;
end
```

Results:

1. X = 0.0624
2. Diverges. Since, the initial slope goes to zero.

3. X(0) = 20 will also lead to no solution. The iterator diverges in this case.



Root finding 3: Now find the roots for the polynomial of last problem by method of bisection. See if choosing *x(0)=0.11* as one of the initial bound work. Compare the number of iterations it takes to converge to a root. Also check *x*=

1.8 as the starting point

```
a=0;
b=0.11;
c=0; %Stores the number of iterations.
disp(f2(a)*f2(b))
while abs(b-a)>1e-7 && f2(a)*f2(b)<0
    c=c+1;
    mid=(a+b)/2;
    if f2(a)*f2(mid)<0
        b=mid;
    else
        a=mid;
    end
end
disp((a+b)/2);
disp(c);
```

Result:

a=0, b=0.11.
Root: 0.0624
Steps: 21

a=0.11, b=2
Root: 0.1464
Steps: 20

It works well for x=0.18.