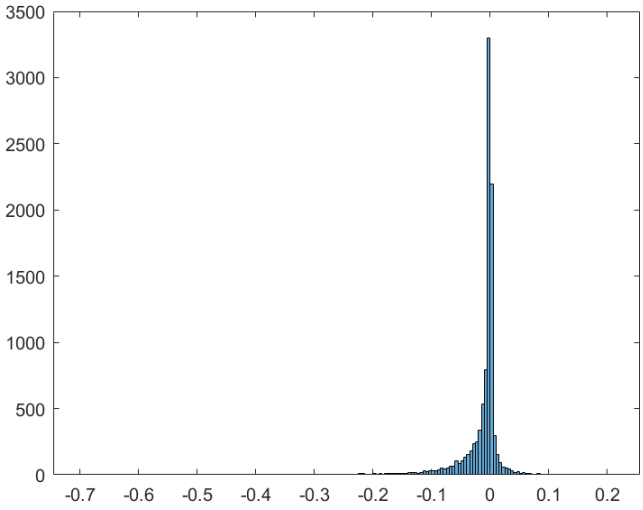
Question 1

Consider the following model similar to the good/bad crops example done in class:
$N(t+1) = (1+r^3)N(t)$
Here N is the population size, t is time and r is a random variable that is normally distributed with mean $\mu = -0.1$ and standard deviation $\delta = 0.2$. This means $r \approx N(-0.1, 0.2)$. Also assume that $N(0) = 100$. The goal is now to investigate what happens to the population over time.
a) Classify this model
b) Find the fixed points
c) Produce a histogram of 10000 samples of the random parameter r^3. Compute the mean and standard deviation of the generated histogram (This is a computer question)
d) What do you expect to happen to the population over time? Support your answer using c).
e) Run a single trial/simulation of the model up to time $t = 200$. Graph N as a function of t. (This is a computer question).
f) Now run 1000 trials up to t=50. Produce a histogram and again find the mean and standard deviation. (This is a computer question).
g) Verify your numerical results by comparing them to your conclusion from d).

16 marks question 1 a) 1 mark b) 1 mark c) 4 marks d) 2 marks e) 2 marks f) 4 marks g) 2 marks

a) LUDS (Linear Univariate Discrete Stochastic)
1 mark

b) Fixed point is 0
1 mark
c) Note this question is stochastic so they could have different answers however the answers have to be around:
newmean = -0.0131
newstand = 0.0397

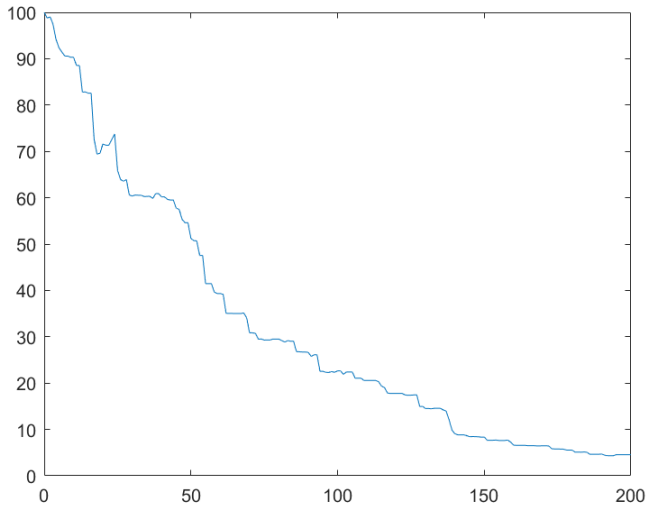



```
mu=-0.1;
stand=0.2;
simulations=10000; %pick number of simulations
r(1:simulations)=0; %This now sets all simulation run initial conditions
rng(1,'twister'); %This generates a seed value for a random number generator.
% What this means is if I pick the seed of 1, and ask for 10
% random numbers. I can then come back later, again have this
% line of code asking for 10 random numbers will give me
% the exact same random numbers. This basically helps and saves reproducibility
for j=1:simulations
    r(j)=normrnd(mu,stand)^3; %normal random variable mean mu standard deviatio
end
newmean=mean(r)
newstand=std(r)
histogram(r)
```

1 mark for mean
1 mark for standard deviation
2 marks for histogram.
d) I expect the population to die out overtime as there will be an overall decline of the population since the mean of r and hence r^3 is negative.
1 mark for conclusion.
1 mark since from c mean is negative or decline is around .9

e) Again this question is stochastic so it should look "something like

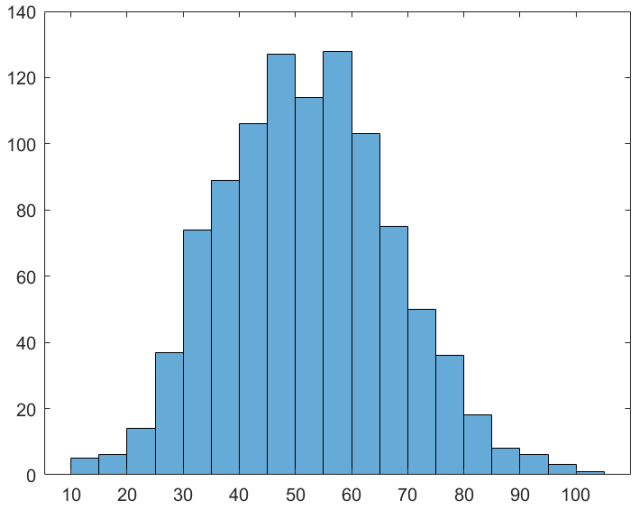
this"




```
mu=-0.1;
stand=0.2;
simulations=200; %pick number of simulations
r(1:simulations)=200; %This now sets all simulation run initial conditions
n(1:simulations+1)=100;
count(1:simulations+1)=1;
rng(1,'twister'); %This generates a seed value for a random number generator.
% What this means is if I pick the seed of 1, and ask for 10
% random numbers. I can then come back later, again have this
% line of code asking for 10 random numbers will give me
% the exact same random numbers. This basically helps and saves reproducibility
count(1)=1;
for j=1:simulations
    r(j)=normrnd(mu,stand)^3; %This sets r1 to be a random value between 0 and
    n(j+1)=(1+r(j))*n(j);
    count(j+1)=j+1;
end
plot(count-1,n); %recall this makes sure the initial condition is at the
                 %correct time
```

2 marks for a plot looking something like this. Should be some stochastic jumps and should be going to 0.

f)Again stochastic but should look something like this:



newmean = 52.8058
newstand = 15.1433
1 mark for mean
1 mark for standard deviation
2 marks for histogram.

```
mu=-0.1;
stand=0.2;
simulations=50; %pick number of simulations
trials=1000;
r(1:trials,1:simulations)=200; %This now sets all simulation run initial condit
n(1:trials,1:simulations+1)=100;
count(1:trials:simulations+1)=1;
rng(1,'twister'); %This generates a seed value for a random number generator.
% What this means is if I pick the seed of 1, and ask for 10
% random numbers. I can then come back later, again have this
% line of code the asking for 10 random numbers will give me
% the exact same random numbers. This basically helps and saves reproducibility
count(1)=1;
for i=1:trials
    for j=1:simulations
        r(i,j)=normrnd(mu,stand)^3; %This sets r to be a random normally
                                    %distributed value with our desired
                                    %mean and standard deviation
        n(i,j+1)=(1+r(i,j))*n(i,j);
```

```
    end
end
newmean=mean(n(1:trials,simulations))
newstand=std(n(1:trials,simulations))
histogram(n(1:trials,simulations))
```

g) Yes this agrees with part d where the populations were expected to die out since most of the histogram is in/near the 0 bin. 1 mark conclusion
1 mark noting that the largest bin is the 0/near 0 bin

QUESTION 2

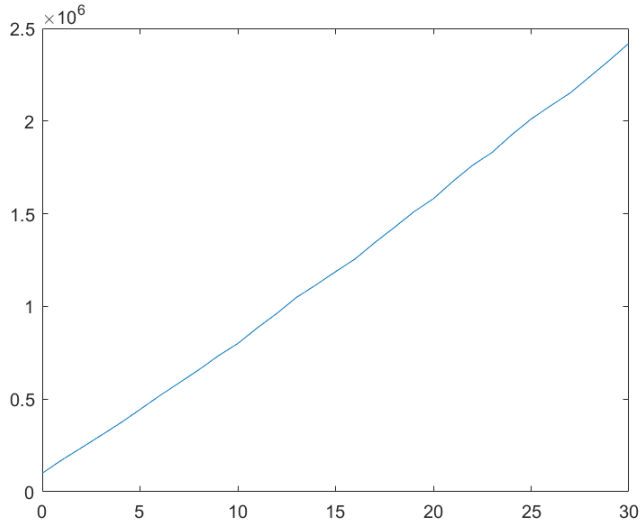
An employee at a large corporation enjoys a yearly wage of 120000 dollars. She spends half of this money and deposits the other half in her savings account at the bank. Due to fluctuations in the world economy, the banks yearly interest rate varies from year to year. In a given year, chances are 1 in 4 that the interest rate is 0.5%, 1 in 2 it is 1%, and 1 in 4 it is 1.5%. Additionally, the employee gets a yearly bonus that she deposits directly into her savings account. The bonus is an amount that is normally distributed with mean 8000 dollars and standard deviation 1600 dollars.

You may assume that the employees entire wage is paid out on January 1st, while the bonus she receives is paid out on December 31st. Let M(t) be the amount of money in the savings account and t is time (in years). Assume M(0) = 100000 so that the employee starts off with 100000 dollars. We are interested in what happens to M over time.

a) Create a model for M by writing down a recursive formula for M(t).
b) Classify your model.
Use computer software to answer the following questions.
c) Run a single trial of the model up to t = 30. Show a plot of M as a function of t.
d) Now run 10000 trials of the model up to t = 30. Display a histogram of M at t = 30. Compute the mean and standard deviation of this distribution.
e) Run 10000 more trials of the model and calculate how many years it will most likely take for the employee to cross the million dollar mark on her savings account (i.e., balance exceeding 1000000). Hint: add an if-statement in your time-stepping loop to calculate when one million dollar is.

question 2 14 marks
a) 4 marks
b) 1 mark
c) 2 marks
d) 4 marks
e) 3 marks

a) M(t+1)=(60000+M(t))(1+r)+b
4 marks (total) and if this is off, punish the rest accordingly since everything will be different
1 for the 60000
1 for the +b (or whatever letter they choose at the end)
1 for the (1+r) multiplying the 60000+M(t)
1 for correctly having everything setup
b) Linear Univariate Discrete Stochastic
1 mark
c) Again stochastic so something like this:



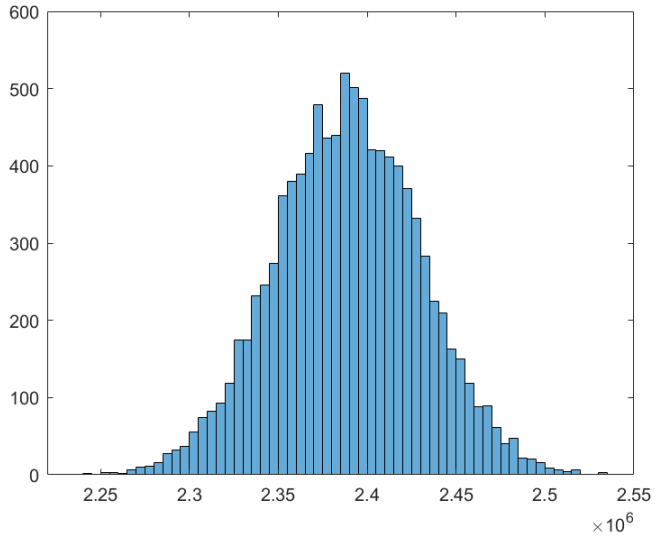
2 marks should be increasing like a line.

```
mu=8000;
stand=1600;
simulations=30; %pick number of simulations
trials=1;
b(1:trials,1:simulations)=200; %This now sets all simulation run initial condit
```

```
m(1:trials,1:simulations+1)=100000;
count(1:trials:simulations+1)=1;
rng(1,'twister'); %This generates a seed value for a random number generator.
% What this means is if I pick the seed of 1, and ask for 10
% random numbers. I can then come back later, again have this
% line of code asking for 10 random numbers will give me
% the exact same random numbers. This basically helps and saves reproducibility
count(1)=1;
for i=1:trials
    for j=1:simulations
        r1=rand;
        if r1<.25 % if else ladder to take care of the 3 different interest rat
            r=0.001;
        elseif r1<.75
            r=0.01;
        else
            r=0.015;
        end
        b(i,j)=normrnd(mu,stand); %This sets bonus to be a random normally dist
        m(i,j+1)=(1+r)*(60000+m(i,j))+b(i,j);
        count(j+1)=j+1;
    end
end
plot(count-1,m); %recall this makes sure the initial condition is at the correc
```

d) Again stochastic so numbers somewhat around these. newmean = 2.5207e+06
newstand = 3.1567e+04

8



1 mark mean

1 mark standard deviation

2 marks for a graph looking something like mine (should be bell shaped).

```
mu=8000;
stand=1600;
simulations=30; %pick number of simulations
trials=10000;
b(1:trials,1:simulations)=200; %This now sets all simulation run initial condit
m(1:trials,1:simulations+1)=100000;
count(1:trials:simulations+1)=1;
rng(1,'twister'); %This generates a seed value for a random number generator.
% What this means is if I pick the seed of 1, and ask for 10
% random numbers. I can then come back later, again have this
% line of code asking for 10 random numbers will give me
% the exact same random numbers. This basically helps and saves reproducibility
count(1)=1;
for i=1:trials
    for j=1:simulations
        r1=rand;
        if r1<.25
            r=0.005;
        elseif r1<.75
            r=0.01;
        else
            r=0.015;
```

```
        end
        b(i,j)=normrnd(mu,stand); %This sets r to be a random normally distribu
        m(i,j+1)=(1+r)*(60000+m(i,j))+b(i,j);
        count(j+1)=j+1;
    end
end
newmean=mean(m(:,end))
newstand=std(m(:,end))
histogram(m(:,end))
```

e) Average Millionaire Years = 12.9623

```
mu=8000;
stand=1600;
simulations=30; %pick number of simulations
trials=10000;
b(1:trials,1:simulations)=200; %This now sets all simulation run initial condit
m(1:trials,1:simulations+1)=100000;
count(1:trials:simulations+1)=1;
rng(1,'twister'); %This generates a seed value for a random number generator.
% What this means is if I pick the seed of 1, and ask for 10
% random numbers. I can then come back later, again have this
% line of code asking for 10 random numbers will give me
% the exact same random numbers. This basically helps and saves reproducibility
count(1)=1;
millionaire(1:trials)=0;
for i=1:trials
    for j=1:simulations
        r1=rand;
        if r1<.25
            r=0.005;
        elseif r1<.75
            r=0.01;
        else
            r=0.015;
        end
        b(i,j)=normrnd(mu,stand); %This sets r to be a random normally distribu
        m(i,j+1)=(1+r)*(60000+m(i,j))+b(i,j);
        count(j+1)=j+1;
        if m(i,j+1)>1000000 && m(i,j)<1000000
            millionaire(i)=j;
        end
    end
```

```
end
newmean=mean(millionaire)
```

2 marks for an answer around 13 years, since stochastic