1 Application and Calculation Problems

1.1 Rat in a Markov Maze

a.

The MATLAB Code was attached as Appendix.

First and foremost, we should clear the assumption for the Markov chain model, which is about how Ratbert moves around the maze. I assumed the Ratbert will randomly choose its next place (obviously, Ratbert can also to stay at the same place.) in the maze with equal possibility and every time period for its decision are equal (I will call it as unit time in following text.) Now, come to the special points. For the electronic points, when the Ratbert arrive at the electronic points, it will leave that point with probability 100%. When the Ratbert arrive at the cheese, it will stay at that point with probability 100% to take its time to eat the cheese (after Ratbert leave the cheese point, the cheese will renew, so the Ratbert will have new cheese on its next stop). Come to the picture,

16 (Dog, come part e)	15	14 (Cheese),17 (Stay)	13
12	11	10	9 (Electronic)
8 (Electronic)	7	6	5
4	3	2	1

There are 16 points on the picture, we can give the Ratbert 17 states to stand for its status. (16 points standing for 16 states, and the No. 14 cheese point, the rat will stay at that point with probability 100%, I give the new state 17 to stand for this stay.)

We will conclude that when the Ratbert stand at one point, how many ways it can go for the next by the picture. For instance, when Ratbert start at point 1, it can go to point 5 or point 2 or stay at Point 1; When Ratbert stop at point 14, it will stay with probability 100% (state 17), at then it will go to point 15 or point 13 or stay at state 17. For point 8 and point 9, the Ratbert will leave the point with probability 100% (No stay!).

In conclude, the route the Ratbert can have as following.

(1, 1); (1, 2); (1, 5);

(2, 1); (2, 2); (2, 3); (2, 6);

(3, 2); (3, 3); (3, 4);

(4, 3); (4, 4); (4, 8);

(5, 1); (5, 5);

(6, 2); (6, 6); (6, 7);

(7, 6); (7, 7); (7, 11);

(8, 4);

(9, 10); (9, 13);

(10, 9); (10, 10); (10, 11);

(11, 7); (11, 10); (11, 11); (11, 12); (11, 15);

(12, 11); (12, 12); (12, 16);

(13, 9); (13, 13); (13, 14);

(14, 17);
(17, 13); (17, 17); (17, 15);
(15, 11); (15, 14); (15, 15);
(16, 12); (16, 16);

We can firstly give all these point the value 1 and then normalize them to meet the definition of the Markov probability transmit matrix. (Please checking the code for the 17*17 matrix)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0.333	0.333	0	0	0.333	0	0	0	0	0	0	0	0	0	0	0	0
2	0.25	0.25	0.25	0	0	0.25	0	0	0	0	0	0	0	0	0	0	0
3	0	0.333	0.333	0.333	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0.333	0.333	0	0	0	0.333	0	0	0	0	0	0	0	0	0
5	0.5	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0.333	0	0	0	0.333	0.333	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0.333	0.333	0	0	0	0.333	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0	0
10	0	0	0	0	0	0	0	0	0.333	0.333	0.333	0	0	0	0	0	0
11	0	0	0	0	0	0	0.2	0	0	0.2	0.2	0.2	0	0	0.2	0	0
12	0	0	0	0	0	0	0	0	0	0	0.333	0.333	0	0	0	0.333	0
13	0	0	0	0	0	0	0	0	0.333	0	0	0	0.333	0.333	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	0	0	0	0.333	0	0	0.333	0.333	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0.5	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0.333	0	0.333	0	0.333

15/10

b.

The MATLAB Code was attached as Appendix.

Set state 14 (The point 14) as an absorbing state, so the all other state are the transient state. Then we can come up with the substochastic matrix Q by deleting the state 14 corresponding row and column.

Let *i* be a transient state and consider Y_i is the total number of visits to *i*

$$\mathrm{Y_i} = \sum\nolimits_{n=0}^{\infty} I\{X_n = i\}$$

$$\mathbb{E}(\mathrm{Y_i}|\mathrm{X_0} = \mathrm{j}) = \mathbb{E}\left[\sum\nolimits_{n=0}^{\infty} I\{X_n = i\}|X_0 = j\right] = \sum\nolimits_{n=0}^{\infty} P\{X_n = i|X_0 = j\} = \sum\nolimits_{n=0}^{\infty} p_n(j, i)$$

We will see the $\mathbb{E}(\mathrm{Y_i}|\mathrm{X_0} = \mathrm{j})$ is the entry of *(j, i)* of the matrix:

$$I + P + P^2 + P^3 \ldots$$

10/10

Also the entry of *(j, i)* of the matrix:

$$I + Q + Q^2 + Q^3 \ldots = (I - Q)^{-1}$$

Based on this idea, we can get the matrix

$$M = (I - Q)^{-1}$$

And find the entry of (1, i) to know from state 1, the times Ratbert will pass state i. because the move of the Ratbert will take one unit time (under assumption), the expect time the Ratbert will take before it reach the chess is the $\sum_{i=1}^{16} M(1, i)$. Under my model, the Ratbert will take the time 112 unit time c.

The MATLAB Code was attached as Appendix.

Based on the M matrix on part b, the times the Ratbert will be shocked before it reach the cheese are equal to the times Ratbert arrived the shocked state before it reach the cheese point. On the Matrix M, it will be

$$M(1,8) + M(1,9) = 5.6667$$

5/5

d.

The MATLAB Code was attached as Appendix.

Let's set state 8, 9 and 14 as the absorbing states.

To calculate the probability the Ratbert ends up at the absorbing states 14.

Theoretically. We can order the states so that the recurrent states $r_1, \ldots r_k$ are recurrent states, and the $t_1, \ldots t_k$ are transient states.

$\alpha(t_i, r_j) = P\{X_n = r_j \text{ eventually} | X_0 = t_i\}$

$= \sum_{x \in S} P\{X_1 = x | X_0 = t_i\} P\{X_n = r_j \; eventually | X_1 = x$

$= \sum_{x \in S} p(t_i, x) \alpha(x, r_j)$

If we set A is the s*k matrix with entries $\alpha(t_i, r_j)$, we can have

$$A = S + QA = (I - Q)^{-1} S = MS$$

Note S is the substochastic matrix with deleting the row from state 8, 9 and 14 and deleting the column from states except state 8, 9 and 14

Then we got the matrix A, we will find the A(1.3) is the thing we need, the probability is 21.43% 10/10

e.

The MATLAB Code was attached as Appendix.

As the thing done above, let's clear the assumptions

1) The Cheese won't renew after the Ratbert ate it, so the state 17 disappear and point 14 became a normal point. On this the Markov Matrix is the size 16*16, only the two special points (state 8 and state 9). Other assumption holds like the part a.
2) Sam is a blind dog, so basically it won't know the road, so it got chance to meet the wall, at this situation, it will stay at the same state on the next epoch. (Even Sam was shocked, it also got chance to stay at that state since it is blind and unlearned.) When Sam made the decision, it got equal probability to stay and try to go to other 4 directions, although some of the direction are wall. (If there are 2 walls, 2 ways. Sam will stay at the point with probability 3/5; and 1/5 got to one of the way)
3) Sam and Ratbert moving in the same speed

To combine the two matrix together, I use the way $A = (I - 1) * 16 + J; B = (P - 1) * 16 + Q;$, the (I, P) stands for the Ratbert from state I to state P; and (J, Q) stands for the Sam from state J to Q. (I, J, P, Q. belonged to the {1,2,3,...,16}) We will find the A and B (both in the range of [1, 256]) can be present by unique pair parameter I, J and P, Q.

On this the combined matrix should be a 256*256 size matrix.

$$CMC_{AB} = MC_R(I, P) * MC_S(J, Q);$$

Now, we should the note when the Sam and Ratbert meet at the same point, the process stop. That is when the P=Q. we can got 16 absorbing states by $(P-1)*16+Q\ \{P=Q; P\&Q \text{ in } [1,16]\}$, then we can come up with the corresponding matrix Q and the find the matrix

$$M=(I-Q)^{-1}$$

15/15

Since, the Ratbert start at the state 14, and the Sam state at state 16, then the combined Matrix start at state 224. Follow the idea from part b, the expecting time for the Sam and Ratbert meet with each other is the "time = 17.3265"

f.

The MATLAB Code was attached as Appendix.

According to my model, after the Ratbert arrived at the point 13, it got probability 100% to go to state 17, then start at state 17 and then absorbed by point 13 again to enjoy next piece. We should use the M from part b.

Firstly, we should start from state 1 and expect 112 unit times to stop at point 13. Next, start from state 17 to expect 23 unit times to get next piece

Ok but could use concept of first return time

Total time=112 + 23*C; when the C is 5 and we got total time is 227

7/10

g.

The MATLAB Code was attached as Appendix.

Based on the part d, we should the probability the Ratbert get the cheese before shocked, the probability shocked by point 8 first; the probability shocked by point 9 first. On this we can know when the Ratbert firstly shocked by point 8, the expect shocked time "etime1"; and the Ratbert firstly shocked by point 8, the expect shocked time "etime2"; then use the probability in part d to get the expected shocked times. From my code, the expected shocked times is 1.6905

explain how combine results

~~10~~ ~~7~~/15

1.2 I Was a Snowball in Hell

For this question, we will use the idea about probability generating function. The property we will use is that

$$p_X(s)=\mathbb{E}s^X=\sum_{j=0}^{\infty}s^jP(X=j)=\sum_{j=0}^{\infty}s^jP_j$$

$$P_j=P(X=j)=\frac{1}{j!}\left(\frac{d}{ds}\right)^j p_X(s)|_{s=0}$$

$$\mathbb{E}X^m=\mathbb{E}\left(s\frac{d}{ds}\right)^m s^X|_{s=1}=\left(s\frac{d}{ds}\right)^m \mathbb{E}s^X|_{s=1}=\left(s\frac{d}{ds}\right)^m p_X(s)|_{s=1}$$

a.

Obviously, this is a branching process. Firstly, let's describe the process, $Z^0=m$ email send out and produce W^1 email back, and W^1 is a random variable under the probability $p_j^I=\{p_j^I\}_{j=0}^{\infty}$, under the W^1 backed email, each of them will produce email response $\{Y_1^1,Y_2^1,Y_3^1,Y_4^1\dots Y_{X^1}^1\}$, all Y_i^1 under the probability distribution $p_j^O=\{p_j^O\}_{j=0}^{\infty}$.

On this idea, the total response email $Z^1=\sum_{i=1}^{w^1}Y_i^1$. Next for the Z^1 , it will produce email back $\{X_1^1,X_2^1,X_3^1,X_4^1\dots X_{Z^1}^1\}$, $W^2=\sum_{i=1}^{Z^1}X_i^1$ (In the following text, $W^n=\sum_{i=1}^{Z^{n-1}}X_i^n$ stands for the number of backed emails on back round n, $Z^n=\sum_{i=1}^{W^n}Y_i^n$ stands for the response round n,)

The process should looked like $(Z^0,W^1,Z^1,W^2,Z^2,W^2,\dots Z^{n-1},W^n,Z^n)$

Firstly, we should note the Lemma from the notes on Nov.09. Picking the reference state i^*, we can form the matrix Q by deleting the row and column corresponding to the reference state i^*. Then we can form the vector β by define $\beta_j = P(T_{i_x}(1) = \infty | X_0 = j)$ with entries $\{\beta_j\}_{j \neq i_*}$, Then β is the maximal solution to the equation $Qx = x$, with $0 \leq x_j \leq 1$ $for\ j \in s \neq i_*$. When $\beta = 0$, the recurrence existed. Else, the transient existed.

Back to the question, first note f $\in \mathcal{A}$, then we can using the conclusion from part a and part b. Choosing k is the reference state. We can got the $Q = \{P_{ij}\}_{i \neq k, j \neq k}$, since we already know the $f^* \in \mathcal{A}$, and the $\sum_{j \in S} P_{ij} f^*(j) = f^*(i)$ $with\ i \neq k$; $f^*(i) = \sup_{f \in A} f(i)$, According to the lemma, we will find the f^* is the maximal solution to the equation $\sum_{j \in S} P_{ij} f^*(j) = f^*(i)$.

Furthermore, according to the $f^*(j) = \sup_{f \in A} f(j)$, and for the condition $f(j) > 0$ $for\ some\ j \in S$. Therefore, $f^*(j) > 0$ always hold (including the situation $\{f^*(j)\}_{j \neq k}$). Since $f^*(j) > 0$, according to the lemma, we got the matrix should be transient.

10/10

Appendix MATLAB code

1.1

a.b.c.d.f.g.

```
clear
clc
%% part a: To calculate the Markov Chain Probablity matrix
% difine the vector to discribe the route.
route = ...
    [1,1;1,2;1,5;
    2,1;2,2;2,3;2,6;
    3,2;3,3;3,4;
    4,3;4,4;4,8;
    5,1;5,5;
    6,2;6,6;6,7;
    7,6;7,7;7,11;
    8,4;
    9,10;9,13;
    10,9;10,10;10,11;
    11,7;11,10;11,11;11,12;11,15;
    12,11;12,12;12,16;
    13,9;13,13;13,14;
    14,17;
    17,13;17,17;17,15;
    15,11;15,14;15,15;
    16,12;16,16]';
```

```
MC = zeros(17);
for i=route
    MC(i(1,1),i(2,1))=1;
end
% To normolize the MC
for i = 1:1:17
    MC(i,:) = MC(i,:)/sum(MC(i,:));
end
%% part b: The average time (or step) the Ratbert reaches the cheese (state 14)
% To get the martix Q
Q = MC;
Q(14,:) = [];
Q(:,14) = [];
M = inv(eye(16)-Q);
time = sum(M(1,:));
%% part c: The times the Ratbert arrive the state 8 and state 9 before it arrive the cheese
stimes1 = M(1,8);
stimes2 = M(1,9);
stimes = stimes1 + stimes2;
%% part d: let's treat the state 14 and the state 8 and state 9 are all absorbing state.
Q1 = MC;
S1 = MC;
S = [];
N1 = [8,9,14];
N2 = [1,2,3];
Q1(N1,:) = [];
Q1(:,N1) = [];
S1(N1,:) = [];
S(:,N2) = S1(:,N1);
MS = inv(eye(14)-Q1) * S;
%% part f: The state 14 is the absorbing state
% at the first time, the state 1 is the start state. from second time, the
% state 17 will be the start state.
% for the first round
t1 = sum(M(1,:));
t2 = sum(M(16,:));
c = 5;% for instance, 5 pieces to make the Ratbert go to rest
t = t1+c*t2;
%% part g: The situation should be discussed by situation
% Fisrtly, the Ratbert start at state 1 and can be absorbed by state 7,8,13
% I use the MS in part b and get the three probability
% If the Ratbert absorbed by the state 7, we got a new matrix, state 7 can
% stay
route1 = ...
```

```
        [1,1;1,2;1,5;
        2,1;2,2;2,3;2,6;
        3,2;3,3;3,4;
        4,3;4,4;4,8;
        5,1;5,5;
        6,2;6,6;6,7;
        7,6;7,7;7,11;
        8,4;8,8;
        9,10;9,13;
        10,9;10,10;10,11;
        11,7;11,10;11,11;11,12;11,15;
        12,11;12,12;12,16;
        13,9;13,13;13,14;
        14,17;
        17,13;17,17;17,15;
        15,11;15,14;15,15;
        16,12;16,16]';
MC1 = zeros(17);
for i=route1
        MC1(i(1,1),i(2,1))=1;
end
for i = 1:1:17
        MC1(i,:) = MC1(i,:)/sum(MC1(i,:));
end
Q2 = MC1;
Q2(14,:) = [];
Q2(:,14) = [];
M2 = inv(eye(16)-Q2);
etimes1 = 1+M2(8,9);
%If the Ratbert absorbed by the state 8, we got a new matrix, state 8 can
%stay
route2 = ...
        [1,1;1,2;1,5;
        2,1;2,2;2,3;2,6;
        3,2;3,3;3,4;
        4,3;4,4;4,8;
        5,1;5,5;
        6,2;6,6;6,7;
        7,6;7,7;7,11;
        8,4;
        9,9;9,10;9,13;
        10,9;10,10;10,11;
        11,7;11,10;11,11;11,12;11,15;
        12,11;12,12;12,16;
```

```
    13,9;13,13;13,14;
    14,17;
    17,13;17,17;17,15;
    15,11;15,14;15,15;
    16,12;16,16]';
MC2 = zeros(17);
for i=route2
    MC2(i(1,1),i(2,1))=1;
end
for i = 1:1:17
    MC2(i,:) = MC2(i,:)/sum(MC2(i,:));
end
Q3 = MC2;
Q3(14,:) = [];
Q3(:,14) = [];
M3 = inv(eye(16)-Q3);
etimes2 = 1+M3(9,8);
Eetimes = MS(1,1) * etimes1 + MS(1,2)*etimes2 + MS(1,3)* 0;
```

e.

```
clear
clc
%% For the Ratbert, based on my assumption, The cheese won't renew after
% eaten by the Ratbert
route1 = ...
   [1,1;1,2;1,5;
    2,1;2,2;2,3;2,6;
    3,2;3,3;3,4;
    4,3;4,4;4,8;
    5,1;5,5;
    6,2;6,6;6,7;
    7,6;7,7;7,11;
    8,4;
    9,10;9,13;
    10,9;10,10;10,11;
    11,7;11,10;11,11;11,12;11,15;
    12,11;12,12;12,16;
    13,9;13,13;13,14;
    14,13;14,14;14,15;
    15,11;15,14;15,15;
    16,12;16,16]';
MC1 = zeros(16);
for i=route1
    MC1(i(1,1),i(2,1))=1;
```

```
end
% To normolize the MC
for i = 1:1:16
    MC1(i,:) = MC1(i,:)/sum(MC1(i,:));
end
%% For the dog, the dog is totally blind and unlearned
route2 = ...
    [1,2;1,5;
    2,1;2,3;2,6;
    3,2;3,4;
    4,3;4,8;
    5,1;
    6,2;6,7;
    7,6;7,11;
    8,4;
    9,10;9,13;
    10,9;10,11;
    11,7;11,10;11,12;11,15;
    12,11;12,16;
    13,9;13,14;
    14,13;14,15;
    15,11;15,14;
    16,12]';
MC2 = zeros(16);
for i = route2
    MC2(i(1,1),i(2,1))=1;
end
for i = 1:1:16
    MC2(i,i) = 5 - sum(MC2(i,:));
end
for i = 1:1:16
    MC2(i,:) = MC2(i,:)/sum(MC2(i,:));
end
%% To build up the conbined matrix
Cmc = [];
for i = 1:1:16
    for j = 1:1:16
        for p = 1:1:16
            for q = 1:1:16
                a = (i-1)*16 + j;
                b = (p-1)*16 + q;
                Cmc(a,b) = MC1(i,p)*MC2(j,q);
            end
        end
```

```
        end
end
Q = Cmc;
for i = 16:-1:1
        for j = 16:-1:1
                if i == j
                        a = (i-1)*16 + j;
                        Q(a,:) = [];
                        Q(:,a) = [];
                end
        end
end
M = inv(eye(240)-Q);
% we know the Ratbert start at point 14(after eating the chess), dog at
% point 16
i = 14;
j = 16;
state = (i-1)*16 + j;
time = sum(M(state,:));
```

1.2

c.

```
clear
clc
%% let's assume the n =10 m = 5
n = 10;
m = 5;
prob = [0.4,0.3,0.1,0.1,0.1];
syms a
fx = 0;
for i = 1:1:size(prob)
        fx = fx + prob(i) * a^(i-1);
end
fx = fx - a;
[a] = solve(fx);
a = vpa(a,3)
```

2.1

a. b. c.

```
clear
clc
ntime = 1000; % Montcarlo simulation
%% To clear the probability distribution
% for the type A agent, the jointed distribution can be
```

Perform the canonical decomposition of $\widetilde{M}$ to generate the Q matrix. Given that there are 16 absorbing states, ..ber of transient states is $256 - 16 = 240$. Hence $Q \in \mathbb{R}^{240 \times 240}$.

..p-3: Calculate $\mu_\tau = (I - Q)^{-1} f$ where $f \in \mathbb{R}^{240 \times 1}$ is a column vector with all entries as 1.

Step-4: In the column vector μ_τ, the initial state of the system $i = 240$, is associated with row number 226. Hence the required answer is $\mu_\tau(226)$.

The expected time to wait until Sam and Ratbert finds each other is equal to **54.4488**.

15/15

(f) In Part (b), we found the expected time for Ratbert to find the cheese given that it starts at State 16. The expected time is 85.1852. After eating the cheese for the 1^{st} time, Ratbert has to eat cheese $C - 1$ times before it goes to rest. The expected time to find the cheese for the last $C - 1$ times are equal. Let this time be τ. One trick to find τ is to observe that it is the *mean first return time* of state-3. Given that this markov chain in irreducible, it has a finite mean return time. τ is indeed given by a very simple formula

$$\tau = \frac{1}{\pi_3}$$

where π_3 is the probability corresponding to state 3 of the associated stationary distribution π. We can find π using the formula

$$\pi = \mathbf{1}^T (I - P + ONES)^{-1}$$

10/10

Using MATLAB we found that $\tau = 16$. Hence the expected time before Ratbert can take rest is $\mathbf{85.1852 + 16\,(C - 1)}$.

(g) The steps to solve this problem are as follows:

Step-1: Make a modified Probability Transition Matrix $\widetilde{P}$ in which states 3 (cheese state), 8 and 9 (shock states) are absorbing states.

Step-2: Calculate the probability of getting absorbed in states 3, 8 or 9 starting from state 16. This can be done by using the formula $U = (I - Q)^{-1} R$ where Q and R is obtained by canonical decomposition of $\widetilde{P}$. Let the probability of getting absorbed in states 3, 8 and 9 be p_3, p_8 and p_9. Observe that $p_3 + p_8 + p_9 = 1$.

Step-3: If the state of the modified markov chain gets absorbed in state 3 then the number of shock is 0. This happens with probability p_3. Hence the expected number of shock corresponding to this case is $p_3 \cdot 0$.

Step-4: Say that the state of the modified markov chain gets absorbed in state 8. This happens with probability p_8. To this end the number of shock received is 1 and the shock generator in state 8 will be turned off. Now we take the following sub steps:

- We make state 8 and 9 unabsorbing. This is done by setting the 8^{th} and the 9^{th} row of $\widetilde{P}$ equal to the 8^{th} and the 9^{th} row of P. We still keep state 3 absorbing.
- Now we have to find the number of shocks received in state 9 before reaching the cheese. This question can be abstracted as: *Starting from state 8, what expected number of epochs spent in state 9, before state 3 is visited?* To answer this question we will take steps similar to Part (c).
- Let the answer to the above question be N_8. Also it received 1 shock in the beginning. Hence the expected number of shock corresponding to this case is $p_8(N_8 + 1)$.

Step-5: Say that the state of the modified markov chain gets absorbed in state 9. This happens with probability p_9. Now we repeat the same steps as Step 4. The expected number of shock corresponding to this case is $p_9(N_9 + 1)$ where N_9 is the "*Expected number of epochs spent in state 8, before state 3 is visited, given that we start from state 9*".

Justify that pls?

Finally the expected number of shocks is $p_3 \cdot 0 + p_8(N_8 + 1) + p_9(N_9 + 1) = p_8(N_8 + 1) + p_9(N_9 + 1)$. We calculated p_8, p_9, N_8 and N_9 using MATLAB and finally the expected number of shock was calculated to be **1.791**. *As one may expect, the expected number of shocks in this case should be less than that in Part (c) because in this case one shock generator gets switched off. This intuition is found to be correct as* $1.791 < 6.2963$.

12/15

Again all the MATLAB code is included in Appendix A.

```
(row_delete,:)=[];
Q(:,col_delete)=[];

f=ones(240,1);

mu_tau=inv(eye(240)-Q)*f;
str=['Average Time before Ratbert and Sam meet each other = ',num2str(mu_tau(226))];
disp(str);

%%%%%%%%%%%%%%%%%%%%%%%%%% Code for 1.1.e. : END %%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%% Code for 1.1.f. : START %%%%%%%%%%%%%%%%%%%%%%

clearvars -except P temp

p_stnry=ones(1,16)*inv(eye(16)-P+ones(16,16));  % Resnick Formula
tau=1/p_stnry(3);

str=['Average Time before Ratbert can rest = ',num2str(temp),'+',num2str(tau),'(C-1)'];
disp(str);

%%%%%%%%%%%%%%%%%%%%%%%%%% Code for 1.1.f. : END %%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%% Code for 1.1.g. : START %%%%%%%%%%%%%%%%%%%%%%

clearvars -except P

P_tilde=P;
P_tilde(3,:)=0;
P_tilde(8,:)=0;
P_tilde(9,:)=0;
P_tilde(3,3)=1;
P_tilde(8,8)=1;
P_tilde(9,9)=1;

row_delete=[3 8 9];
col_delete=[3 8 9];
Q=P_tilde;
Q(row_delete,:)=[];
Q(:,col_delete)=[];

clear row_delete col_delete
row_delete=[3 8 9];
col_delete=[1 2 4 5 6 7 10 11 12 13 14 15 16];
R=P_tilde;
R(row_delete,:)=[];
R(:,col_delete)=[];

U=inv(eye(13)-Q)*R;
p8=U(13,2);
p9=U(13,3);
```

So we did some examples in MATLAB to ensure we were getting realistic results. Let's assume that $C = 5$. We use MATLAB to get the follwing results:

Average time to eat 2 pieces of cheese: 91.6667
Average time to eat 3 pieces of cheese: 106.6667
Average time to eat 4 pieces of cheese: 121.6667
Average time to eat 5 pieces of cheese: 136.6667

5/10

g) Suppose now that I have my own graduate student, Catbert, and his job is to watch Ratbert until he gets shocked for the first time. Once this happens, Catbert disables the shock generatr at the location where his first shock occurred. We now want to compute the average number of shocks before Ratbert can finally enjoy his well-deserved piece of cheese. The tool that we'll use to help us solve this problem is the law of total expectation which says that $\mathbb{E}(Y|C) = \sum_{k\in S}\mathbb{E}(Y|B_k, C)P(B_k|C)$. So, we look at

$$\mathbb{E}(\sum_{n=0}^{\tau-1} f(X_n)|X_0 = 1) =$$

$$\mathbb{E}(\sum_{n=0}^{\tau-1} f(X_n)|X_{s1} = 9, X_0 = 1)P(X_{s1} = 8|X_0 = 1) + \mathbb{E}(\sum_{n=0}^{\tau-1} g(X_n)|X_{s1} = 8, X_0 = 1)P(X_{s1} = 9|X_0 = 1)$$

where $X_{s1} = 8$ represents the first shock occuring in state 8, and $X_{s1} = 9$ represents the first shock occuring in state 9. Note we added the function $g(X_n)$. This is not meant to mean anything other than denote that the function we'll use in our computation is slightly different for state 9 than it is for state 8.

So, all we need to do is compute the probabilities that Ratbert hits state 8 first (before hitting the cheese or 9) and the probability that Ratbert hits state 9 first (before hitting the cheese or state 8). Fortunately, we've already computed these probabilities. In part d we computed the probability that Ratbert reaches the cheese without hitting any of the shock states. However, in that same matrix we pulled that probability from, we also have the probabilities that Ratbert reaches state 8 before cheese and state 9, as well as the probability that Ratbert reaches state 9 before cheese or state 8. So we have those pieces of information. Remember, here we made states 8 and 9 absorption states. Recall the matrix Q from part c. We are going to use the same formula that we used then to compute the expected number of time spent in the shock states. We will let f be a 15×1 vector where all the entries are 0 expect the 8^{th} entry is 1. Also let g be a 15×1 vector where all the entries are 0 except the 9^{th} entry is 1.

To stay consistent with out MATLAB code, we will the probability that Ratbert hits state 8 before the cheese or state 9 $U(1,1)$, and we will call the probability that Ratbert hits state 9 before the cheese or state 9 $U(1,2)$. Thus, our final formula for the expected number of shocks before Ratbert gets the cheese is

$$U(1,1)((I-Q)^{-1}g^{(T)}) + U(1,2)((I-Q)^{-1}f^{(T)})$$

where f,g and Q are defined as above.

We perform these computations easily in MATLAB and obtain the following results:

Number of times Ratbert gets shocked after Catbert removes the first one: 1.6905

Note that for most of the above parts we performed Monte Carlo simulations to corroborate our deterministic results. Fortunately we got approximately the same answers for the parts that were simulated above. The code is attached in the Appendix as well.

15/15

1.2 **I was a Snowball in Hell (75 points)** Suppose we are organizing an event and send out an email to m people to advertise and prepare. Each recipient will generate a random numer of responses which we will model as iid with pmf $p^I = \{p_j^I\}_{j=0}^{\infty}$. Each email that we receive will require us to send a random number of new outgoing emails which, again, will be modeled as iid with pmf $p^O = \{p_j^O\}_{j=0}^{\infty}$. This process will then repeat for every outgoing message we send. Suppose that the number of emails generated by each outgoing or incoming email are independent.

a) **(10 points)** First we will compute the mean and variance of the number of outgoing emails we must send out in the n^{th} round of the exchange. Let's start with the mean since it's more straightforward. We will let Y_j be the random variable denoting how many outgoing emails we send per incoming email, and Z_j be the random variable denoting how many incoming messages we receive per outgoing email. We want to use probability generating functions to help us compute the mean. We know that in general $\mathbb{E}(X) = (s\frac{d}{ds})\mathcal{P}_X(s)|_{s=1}$. So we will apply this formula to X_n were $X_N = \sum_{j=1}^{Z} Y_j$. Here X_n just represents the total number of outgoing emails at epoch n.

So $\mathbb{E}(X_n) = (s\frac{d}{ds})\mathcal{P}_{X_n}(s)|_{s=1}$ where $\mathcal{P}_{X_n}(s) = \mathcal{P}_{X_{n-1}}(\mathcal{P}_Z(\mathcal{P}_Y(s)))$ $\leftarrow$ why?
So

$$\begin{aligned}\mathbb{E}(X_n) &= (s\frac{d}{ds})\mathcal{P}_{X_{n-1}}(\mathcal{P}_Z(\mathcal{P}_Y(s)))|_{s=1}\\ &= s\mathcal{P}'_{X_{n-1}}(\mathcal{P}_Z(\mathcal{P}_Y(s)))\mathcal{P}'_Z(\mathcal{P}_Y(s))\mathcal{P}'_Y(s)|_{s=1}\\ &= 1\mathcal{P}'_{X_{n-1}}(\mathcal{P}_Z(\mathcal{P}_Y(1)))\mathcal{P}'_Z(\mathcal{P}_Y(1))\mathcal{P}'_Y(1)\\ &= 1\mathcal{P}'_{X_{n-1}}(\mathcal{P}_Z(1))\mathcal{P}'_Z(1)\mathcal{P}'_Y(1)\end{aligned}$$

as long as pdf is not defective

$$= \mathcal{P}'_{X_{n-1}}(1)\mathbb{E}(Z)\mathbb{E}(Y)$$

1.2 I was a Snowball in Hell

We are organizing an event and in the planning stage of it, we send out an email to a group of m people which then generates a series of emails back and forth between us and the original group of people henceforth referred to as incoming (from people to us) and outgoing (from us to people) emails.

We are given that incoming and outgoing emails are distributed according to

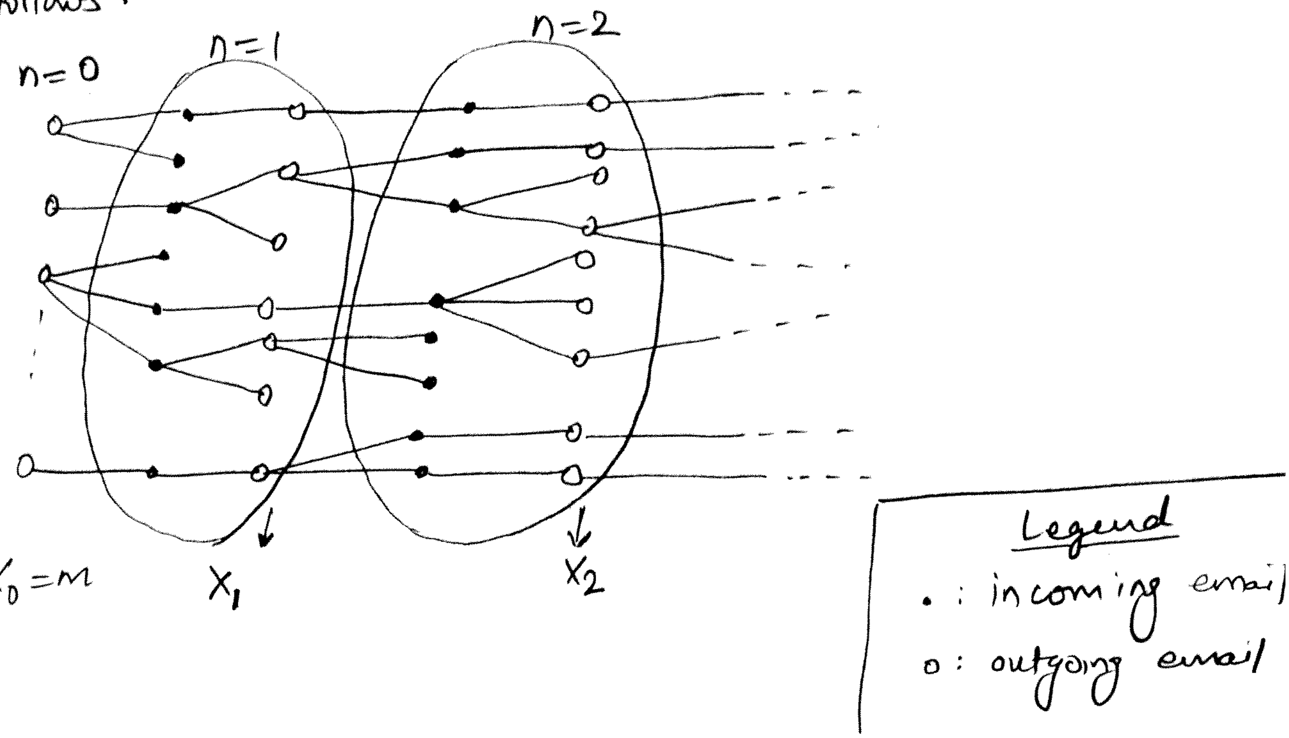
$$p^I = \{p_j^I\}_{j=0}^{\infty}$$

$$\& \; p^O = \{p_j^O\}_{j=0}^{\infty} \quad \text{respectively.}$$

Since in the next few parts, we are interested in # of outgoing emails in each exchange, we model the problem such that:

- → It's a countable state Markov chain with state space $S = \{0, 1, 2, ---\}$
- → At each epoch n, X_n represents the # of outgoing emails at the nth round of exchange.
- → Each round consists of receiving emails from the receipients and for each received email, sending out the new emails.
- → $n=0$ is also a round which corresponds to no received email and m outgoing emails that were sent out in the beginning.

Pictorially speaking, the model can be represented as follows:



Modelled this way, we see that it's a branching process and is a simple modification of Galton-Watson model with the two offspring distributions given as p_0^I & p_0^O.

→ We first proceed to write the stochastic update rule for this Markov chain in the form of a recurrence relation involving a random sum.

If we have:

$Y_{n,k}$ = # of outgoing emails from kth agent in nth epoch.

& $Z_{n,k}$ = # of incoming emails from kth agent in nth epoch.

Then, the recurrence formula for state variable X_n can be written as follows:

$$\boxed{X_{n+1} = \sum_{k=1}^{X_n} \sum_{t=1}^{Z_{n,k}} Y_{n,t}}$$

The expression above is self explanatory. Since we want total # of outgoing emails at each epoch, we first sum over all the outgone emails in the last round (the outer sum) the number of incoming emails. But since each incoming email generates a random # of outgoing emails, it gives rise to inner sum for each incoming email.

→ Now, before moving onto all the parts, we compute the recurrence relation for the pgf (probability generating function) of X_n since we know we're going to need it.

$$P_{X_{n+1}}(s) = \mathbb{E}\, s^{X_{n+1}} \qquad \left[\begin{array}{l}\text{by def}^{n} \\ P_X: \text{pgf of } X\end{array}\right.$$

$$= \mathbb{E}\, s^{\sum_{k=1}^{X_n} \sum_{t=1}^{Z_{n,k}} Y_{n,t}}$$

$$= \mathbb{E}\, s^{\sum_{k=1}^{X_n} H_{n,k}} \qquad \left[\begin{array}{l}\text{where} \\ H_{n,k} = \sum_{t=1}^{Z_{n,k}} Y_{n,t}\end{array}\right.$$

$$= \mathbb{E} \prod_{k=1}^{X_n} s^{H_{n,k}}$$

$$= \sum_{j=1}^{\infty} \mathbb{E}\left[\prod_{k=1}^{X_n} s^{H_{n,k}} \,\middle|\, X_n = j\right] P(X_n = j)$$

[Law of total expectation

$$= \sum_{j=1}^{\infty} \mathbb{E}\left[\prod_{k=1}^{j} s^{H_{n,k}} \,\middle|\, X_n = j\right] P(X_n = j)$$

[plugged $X_n = j$ in sum limit]

$$= \sum_{j=1}^{\infty} \mathbb{E}\left[\prod_{k=1}^{j} s^{H_{n,k}}\right] P(X_n = j)$$

[as $H_{n,k}$ are independent of X_n

$$= \sum_{j=1}^{\infty} \prod_{k=1}^{j} \mathbb{E}\left[s^{H_{n,k}}\right] P(X_n = j)$$

[Since $Z_{n,k}$ & $Y_{n,k}$ & therefore $H_{n,k}$ are independent and we know $E[XY] = E[X]E[Y]$ if X & Y are independent

$$= \sum_{j=1}^{\infty} \prod_{k=1}^{j} P_{H_{n,k}}(s) \quad P(X_n = j)$$

[from defn of pgf for $H_{n,k}$

$$= \sum_{j=1}^{\infty} \left(P_{H_n}(s)\right)^j P(X_n = j)$$

[Since all agents have same distribution $H_{n,k} = H_n$.

$$= \mathbb{E}\left(P_{H_n}(s)\right)^{X_n}$$

$$= P_{X_n}\left(P_{H_n}(s)\right)$$

[from defn of pgf again].

So our next task is to find the pgf $P_{H_{n,k}}(s)$

but notice that $H_n = \sum_{t=1}^{Z_{n,k}} Y_{n,t} = \sum_{t=1}^{Z_n} Y_{n,t}$

has essentially the same form as

$$X_{n+1} = \sum_{k=1}^{X_n} H_{n,k}$$

for which we just found the pgf.

∴ Therefore, to find the pgf $P_{H_{n,k}}(s)$, all we need to do is replace the <u>subscript</u> & <u>argument</u> in the expression we just obtained.

i.e. $$P_{H_n}(s) = P_{Z_n}\left(P_{Y_n}(s)\right)$$

[where we've used the fact that all agents have same outgoing distribution i.e. $Y_{n,t} = Y_n$

Now, we also know that distribution of incoming & outgoing emails are independent of the epoch & are identical.

Therefore, $H_n = H$
because $Z_n = Z$
& $Y_n = Y$ $\forall n$.

$$\Rightarrow P_{H_n}(s) = P_H(s) = P_Z\left(P_Y(s)\right)$$

We plug this into previously obtained result for $P_{X_{n+1}}(s)$ to get:

$$\boxed{P_{X_{n+1}}(s) = P_{X_n}\left(P_Z\left(P_Y(s)\right)\right)}$$

where $P(Z=j) = p_j^I$ $j = 0, 1, 2, \ldots$ (incoming)

$P(Y=j) = p_j^O$ $j = 0, 1, 2, \ldots$ (outgoing)

This provides the recurrence relation for pgf of X_n.

We can now proceed to individual parts:

a) We first find the mean of X_{n+1}

using the pgf, we know

$$E X_{n+1} = s \frac{d}{ds} P_{X_{n+1}}(s) \Big|_{s=1}$$

$$= s \frac{d}{ds} P_{X_n}(P_Z(P_Y(s))) \Big|_{s=1}$$

$$= s\, P'_{X_n}(P_Z(P_Y(s)))\, P'_Z(P_Y(s))\, P'_Y(s) \Big|_{s=1}$$

(using chain rule)

$$= 1 .\, P'_{X_n}(P_Z(P_Y(1)))\, P'_Z(P_Y(1))\, P'_Y(1)$$

Now, $P_Y(1) = \sum_{j=0}^{\infty} p_j^{o} 1^j = 1$

$$\Rightarrow P_Z(P_Y(1)) = P_Z(1) = \sum_{j=0}^{\infty} p_j^{I} 1^j = 1$$

$$\Rightarrow E X_{n+1} = 1 .\, P'_{X_n}(1) .\, 1 .\, P'_Z(1) . 1 .\, P'_Y(1)$$

$$1 .\, P'_{X_n}(1) = s \frac{d}{ds} P_{X_n}(s) \Big|_{s=1} = E X_n$$ (property of pgf)

similarly $1 .\, P'_Z(1) = s \frac{d}{ds} P_Z(s) \Big|_{s=1} = EZ$

& $1 .\, P'_Y(1) = EY$

$$\Rightarrow E X_{n+1} = E X_n\, E(Z)\, E(Y)$$

by recursion, we get

$$\boxed{E X_n = [E(Z) E(Y)]^n\, E X_0}$$

. Here $EX_0 = m$ for us.

To compute the variance, we first find the second-order moment for X_{n+1}

$$\therefore \quad E X_{n+1}^2 = \left(s\frac{d}{ds}\right)^2 P_{X_{n+1}}(s)\Big|_{s=1}$$

$$= s\frac{d}{ds}\left[s P'_{X_n}(P_Z(P_Y(s)))\, P'_Z(P_Y(s))\, P'_Y(s)\right]\Big|_{s=1}$$

(as done on previous page for EX_{n+1}

$$= P'_{X_n}(P_Z(P_Y(s)))\, P'_Z(P_Y(s))\, P'_Y(s)$$
$$+ s P'_{X_n}(P_Z(P_Y(s)))\, P'_Z(P_Y(s))\, P''_Y(s)$$
$$+ s P'_{X_n}(P_Z(P_Y(s)))\, P''_Z(P_Y(s))\left[P'_Y(s)\right]^2$$
$$+ s P''_{X_n}(P_Z(P_Y(s)))\left[P'_Z(P_Y(s))\right]^2\left[P'_Y(s)\right]^2\Big|_{s=1}$$

$$= P'_{X_n}(P_Z(P_Y(1)))\, P'_Z(P_Y(1))\, P'_Y(1)$$
$$+ 1.\, P'_{X_n}(P_Z(P_Y(1)))\, P'_Z(P_Y(1))\, P''_Y(1)$$
$$+ 1.\, P'_{X_n}(P_Z(P_Y(1)))\, P''_Z(P_Y(1))\left[P'_Y(1)\right]^2$$
$$+ 1.\, P''_{X_n}(P_Z(P_Y(1)))\left[P'_Z(P_Y(1))\right]^2\left[P'_Y(1)\right]^2$$

Now, just like before $P_Y(1) = 1 \Rightarrow P_Z(P_Y(1)) = P_Z(1) = 1$

$P'_Y(1) = E(Y) \quad P'_Z(P_Y(1)) = P'_Z(1) = E(Z)$

& $P'_{X_n}(P_Z(P_Y(1))) = P'_{X_n}(1) = E(X_n)$

$$\Rightarrow E X_{n+1}^2 = EX_n\, EZ\, EY + EX_n\, EZ\, P''_Y(1)$$
$$+ EX_n\, P''_Z(1)\left[EY\right]^2 + P''_{X_n}(1)\left[EZ\right]^2\left[EY\right]^2$$

Now, $P''_Y(1) = s\frac{d}{ds}\left(P'_Y(s)\right)\Big|_{s=1} = \left(s\frac{d}{ds}\right)\left(s\frac{d}{ds}\right)(P_Y(s))\Big|_{s=1} - s\frac{d}{ds}P_Y(s)\Big|_{s=1}$

$$= \left(s\frac{d}{ds}\right)^2 P_Y(s)\Big|_{s=1} - \left(s\frac{d}{ds}\right)P_Y(s)\Big|_{s=1}$$

$$= EY^2 - EY$$

$\therefore$ Similarly $\quad P''_{Z}(1) = EZ^2 - EZ$

& $\quad P''_{X_n}(1) = EX_n^2 - EX_n$.

Therefore,
$$EX_{n+1}^2 = \cancel{EX_n E(Z) E(Y)} + EX_n E(Z)\left[EY^2 - \cancel{EY}\right] + EX_n[EY]^2\left[EZ^2 - EZ\right] + [EZ]^2[EY]^2\left[EX_n^2 - EX_n\right]$$
$$= EX_n EZ\left[EY^2 - (EY)^2\right] + EX_n (EY)^2\left[EZ^2 - (EZ)^2\right] + (EZ)^2(EY)^2 EX_n^2$$
$$= EX_n EZ\, V(Y) + EX_n (EY)^2 V(Z) + [E(Z)]^2[E(Y)]^2 EX_n^2$$
[where V : variance]

$$Var(X_{n+1}) = EX_{n+1}^2 - [EX_n]^2$$
$$= EX_n EZ\, V(Y) + EX_n (EY)^2 V(Z) + (E(Z))^2(E(Y))^2 EX_n^2 - (EX_n)^2 (E(Z))^2 (E(Y))^2$$
$$= EX_n EZ\, V(Y) + EX_n (EY)^2 V(Z) + (E(Z))^2 (E(Y))^2 V(X_n)$$

$$\Rightarrow V(X_{n+1}) = \left[EZ\, V(Y) + (E(Y))^2 V(Z)\right] EX_n + (EZ)^2(EY)^2 V(X_n)$$
$$= \underbrace{\left[EZ\,V(Y) + (EY)^2 V(Z)\right]\left[EZ\,EY\right]^n EX_0}_{A} + (EZ)^2(EY)^2 V(X_n)$$
$$= A + (EZ)^2(EY)^2 V(X_n)$$
$$= A + (EZ)^2(EY)^2\left[A + (EZ)^2(EY)^2 V(X_{n-1})\right]$$
$$= \left[1 + (EZ)^2(EY)^2\right] A + (EZ)^4(EY)^4 V(X_{n-1})$$
$$- - - -$$
$$= \left[1 + (E(ZY))^2 + \cdots + [E(ZY)]^{2n}\right] A + \cancel{[E(ZY)]^{2n+2} V(X_0)} \to 0$$

15/10

$\left\{\begin{array}{l} \text{as } E(Z)E(Y) = E(ZY) \quad \because Z \text{ \& } Y \text{ are independent} \\ \text{\& } V(X_0) = 0 \text{ since } X_0 = m \text{ as given.} \end{array}\right.$

Therefore, we get
$$\boxed{V(X_n) = \left(\frac{[E(ZY)]^n - 1}{E(ZY) - 1}\right)\left[EZ\,V(Y) + (EY)^2 V(Z)\right]\left[E(ZY)\right]^{n-1} EX_0}$$

where $EX_0 = m$

5/10

$Var(X_n) = E(X_n^2) - E(X_n)^2$ Replacing all the $E(X_{n-1})$ terms we get:
$Var(X_n) = m(E(I)E(\theta))^n + E(X_{n-1}^2)(E(I)E(\theta))^2 + m(E(I)E(\theta))^{n+1}$
$+ mE(\theta)(E(I)E(\theta))^n - m^2(E(I)E(\theta))^{2n}$

b) Find the probability the email stops at the n^{th} exchange.
Assuming this means at exactly epoch n. Even if its not, I will calculate the extinction probability by n as an intermediate calculation.

$P(\bigcup_{i=1}^{n}\{X_i = 0\} \mid X_0 = m) \equiv$ Probability of extinction by n.
$= P_{X_n}(0) = P_{X_{n-1}}(P_I(P_\theta(0))) = P_{X_{n-1}}(P_I(p_0^\theta))$
We know $P_{X_0}(s) = s^m \Rightarrow P_{X_1}(s) = P_{X_0}(P_I(P_\theta(s))) = (P_I(P_\theta(s)))^m$
$P_{X_2}(s) = P_{X_1}(P_I(P_\theta(s))) = P_{X_0}(P_I(P_\theta(P_I(P_\theta(s))))) = (P_I(P_\theta(P_I(P_\theta(s)))))^m$
Therefore $P_{X_n}(s)$ is n compositions of $P_I \circ P_\theta$ raised to the m.
$P_{X_n}(0)$ can then be calculated numerically.
Now extinction at exactly epoch n can be found easily by
$P(\text{Extinct AT } n) = P(\text{Extinct BY } n) - P(\text{Extinct BY } n-1)$
$= P_{X_n}(0) - P_{X_{n-1}}(0) = \hat{P}_{X_n}(0)$
See Matlab for example calculations using these formulas
Ex: $\{p^I\} = \{\frac{3}{8}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}\}$ (0 1 2 3 4) $\{p^\theta\} = \{\frac{3}{7}, \frac{1}{7}, \frac{2}{7}, \frac{1}{7}\}$ (0 1 2 3) $m = 5$

10/10 $P_{X_2}(0) = 0.0866$, $P_{X_3}(0) = 0.1174$, $P_{X_4}(0) = 0.1357$, $P_{X_5}(0) = 0.1465$, $P_{X_{10}}(0) = 0.1611$
$\hat{P}_{X_2}(0) = 0.0446$, $\hat{P}_{X_3}(0) = 0.0272$, $\hat{P}_{X_4}(0) = 0.0158$, $\hat{P}_{X_5}(0) = 0.0092$, $\hat{P}_{X_{10}}(0) = 0.0007$
The eventual extinction probability converges to 0.1623 and the extinction probability at exactly n decays to 0.

c) Find the probability the emailing never stops. Using what was calculated in b), the probability to never stop is 1 minus the probability it goes extinct by n, as $n \to \infty$. This

10/10 is $1 - \lim_{n\to\infty} P_{X_n}(0)$. Using the same example distributions as I did in b), I stated $P_{X_n}(0)$ converged to 0.1623 for large n. This gives a probability of $1 - 0.1623 = 0.8377$ for the email chain to explode.

d) Now with each batch of outgoing email we add an additio random number of emails N, iid with pmf. $p^N = \{p_j^N\}_{j=0}^{\infty}$

1.2 b)

```
%Find the extinction probability at exactly epoch n for the email branching
%process with m original recipients.
clear

pI=[3,2,1,1,1];      %iid dist. for incoming emails
pO=[3,1,2,1];        %iid dist. for outgoing email
pI=pI/sum(pI);  %normalize pI
pO=pO/sum(pO);  %normalize pO

m=5;            %Number of initial emails
n=100;           %Extinction probabilities for all integers <=n

Q=zeros(1,n);   %Vector to store extinction probabilities by n
for i=1:n
    s=0;
    for j=1:i
        s=genP(genP(s,pO),pI);
    end
    Q(i)=s^m;
end

G=zeros(1,n);  %Vector for extinction at epoch n
G(1)=Q(1);
for i=2:length(Q)
    G(i)=(1-Q(i))*(Q(i)-Q(i-1));
end
```

Calculates the generating function at s of given distribution.

```
>> Q(1:15)

ans =

  Columns 1 through 5

    0.0377    0.0866    0.1174    0.1357    0.1465

  Columns 6 through 10

    0.1529    0.1567    0.1590    0.1603    0.1611

  Columns 11 through 15

    0.1616    0.1619    0.1621    0.1622    0.1623
```

Extinction by n

```
>> G(1:15)

ans =

  Columns 1 through 5

    0.0377    0.0446    0.0272    0.0158    0.0092

  Columns 6 through 10

    0.0054    0.0032    0.0019    0.0011    0.0007

  Columns 11 through 15

    0.0004    0.0002    0.0001    0.0001    0.0001
```

Extinction at n

```
>> Q(100)
ans =
  0.1623
```

Eventual extinction prob

c) Now we want to find the probability that we are dealing with the emails forever. i.e. (20)

P (process never dies out)

= 1 - P(process dies out at some epoch n)

= 1 - a(m)

where a(K) is defined similar to lecture notes as:

$a(K) = P\{X_n = 0 \text{ for some } n > 0 \mid X_0 = K\}$.

{since we've $X_0 = m$, we use the value $a(m)$}

$= (a(1))^K$ {since all agents are identical

$= (a)^K$ where $a = a(1)$

In class, we saw that for pgf recurrence relation given by $P_{X_{n+1}}(s) = P_{X_n}(P_Y(s))$,

solving for a yields:

$$a = P_Y(a)$$

In our case, since $X_0 = m$ the above eqn changes slightly when we use pgf of X_{n+1} $P_{X_{n+1}}(s)$ in the proof.

where $P_{X_{n+1}}(s) = P_{X_n}(P_Z(P_Y(s)))$.

The modified result is a nonlinear eqn in a as:

$$\boxed{a = \sum_{j=0}^{\infty} a^j \sum_{k=1}^{\infty} P(Y_1 + Y_2 + \dots Y_k = j)\, P_k^I} \quad — (*)$$

(with exactly the same steps as followed in class ~~proof~~, proof is on next page)

$$a = P\left(\bigcup_{n=1}^{\infty}\{X_n = 0\} \,\middle|\, X_0 = 1\right)$$

$$= \sum_{j=0}^{\infty} P\left(\bigcup_{n=1}^{\infty}\{X_n = 0\} \,\middle|\, X_1 = j, X_0 = 1\right) P(X_1 = j \mid X_0 = 1)$$

[Law of Total Probability

$$= P\left(\bigcup_{n=1}^{\infty}\{X_n = 0\} \,\middle|\, X_1 = 0, X_0 = 1\right) P(X_1 = 0 \mid X_0 = 1)$$
$$+ \sum_{j=1}^{\infty} P\left(\bigcup_{n=1}^{\infty}\{X_n = 0\} \,\middle|\, X_1 = j, X_0 = 1\right) P(X_1 = j \mid X_0 = 1)$$

$$= 1.\, P(X_1 = 0 \mid X_0 = 1) + \sum_{j=1}^{\infty} P\left(\bigcup_{n=2}^{\infty}\{X_n = 0\} \,\middle|\, X_1 = j, X_0 = 1\right) P(X_1 = j \mid X_0 = 1)$$

(as X_1 can not be 0)

$$= 1.\, P(X_1 = 0 \mid X_0 = 1) + \sum_{j=1}^{\infty} P\left(\bigcup_{n=2}^{\infty}\{X_n = 0\} \,\middle|\, X_1 = j\right) P(X_1 = j \mid X_0 = 1)$$

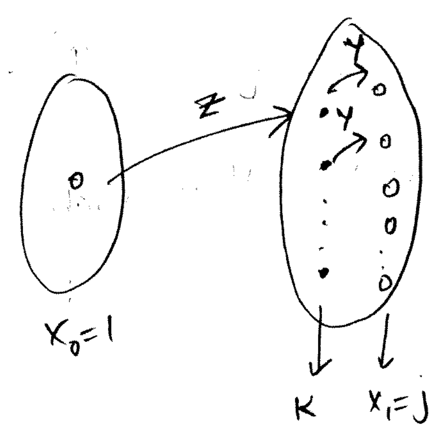
(by Markov Property)

$$= P(X_1 = 0 \mid X_0 = 1) + \sum_{j=1}^{\infty} a(j)\, P(X_1 = j \mid X_0 = 1)$$

(by time shift $n \to n+1$)

$$\Rightarrow \boxed{a = \sum_{j=0}^{\infty} a^j\, P(X_1 = j \mid X_0 = 1)}$$

Now, since we've two offspring distribution at each epoch, we'll need to see how the $P(X_1 = j \mid X_0 = 1)$ comes out depending on different possible cases.



$$a = \sum_{j=0}^{\infty} a^j \sum_{k=1}^{\infty} P(X_1 = j \mid X_0 = 1, Z = k) \quad P(Z = k \mid X_0 = 1)$$

[By law of total probability and noting that we need at least one incoming mail to have $X_1 = j$ eventually].

$$= \sum_{j=0}^{\infty} a^j \sum_{k=1}^{\infty} P(X_1 = j \mid Z = k) \; p_k^I$$

[Since with Z fixed, X_1 would be independent of X_0).

Now $Z = k$ implies there are k incoming emails. For those incoming emails, we want sum of all the outgoing emails to be j. Therefore:

$$a = \sum_{j=0}^{\infty} a^j \sum_{k=1}^{\infty} P(Y_1 + \dots + Y_k = j) \; p_k^I$$

where $\{Y_i\}$ are iid with distribution $\{p_l^O\}_{l=0}^{\infty}$ each

This proves the result written on page 20.

Complete Q1

Now we look at the expression obtained for EX_n in part (b)

$$E X_n = [E(ZY)]^n m$$

clearly if $E(ZY) < 1$ (subcritical) $\Rightarrow \lim_{n\to\infty} EX_n = 0$

or $E(ZY) = 1$ (critical) $\Rightarrow \lim_{n\to\infty} EX_n = m$

and first of these situations will lead to extinction for certain since the values obtained for the limits above show that X_n can never exhibit explosive growth.

[In other words, the only sol^n to eq^n ⊛ on p (20) is $a=1$ i.e process will certainly die out].

[$E(ZY)=1$ is the critical case]

Now, for the case $E(ZY) > 1$

we solve $a = \sum_{j=0}^{\infty} a^j \sum_{k=1}^{\infty} P(Y_1 + \dots + Y_k = j)\, P_k^I$ — c1)

The solution of above equation will give the extinction probability in the case of $X_0 = 1$

but here, since we start with $X_0 = m$, the extinction probability will be :

$= a(m)$ [where a is soln to c1) above]

Therefore, combining everything:

$$P(\text{process never dies out}) = \begin{cases} 0 & E(ZY) < 1 \\ 1 - a^m & E(ZY) \geq 1 \text{ (where } a(m) \text{ is as described above)} \end{cases}$$

10/10

1.2 I Was a Snowball in Hell (computational part)

For the computational part of this question, we write a MATLAB program. We are given that we can pick a reasonable probability distributions for p^I, p^O and p^N.

a,b,c) For parts a, b and c, we assume that p^I and p^O are Poisson distributions with different means. We now compute the answers to these parts using the formulas derived on paper. We pick the value of m to be 100 assuming that many emails were sent in the beginning.

Since we are assuming Poisson distribution for offspring distribution Y and Z in the developed expressions, we first need to write the PGF of a Poisson distribution with mean λ. From the definition of PGF:

$$\mathcal{P}_X(s) = \sum_{x \geq 0} p_X(x)\, s^x$$

From the definition of the Poisson distribution:

$$\forall k \in \mathbb{N}, k \geq 0 : p_X(k) = \frac{e^{-\lambda}\lambda^k}{k!}$$

So:

$$\begin{aligned}
\mathcal{P}_X(s) &= \sum_{k \geq 0} \frac{e^{-\lambda}\lambda^k}{k!} s^k \\
&= e^{-\lambda} \sum_{k \geq 0} \frac{(\lambda s)^k}{k!} \\
&= e^{-\lambda} e^{\lambda s} \qquad \text{(by Taylor Series Expansion for Exponential Function)} \\
&= e^{\lambda(s-1)}
\end{aligned}$$

Therefore, in all our obtained expressions, we plug the above PGF wherever needed corresponding to $\mathcal{P}_Z$ and $\mathcal{P}_Y$ and find the values of required quantities.

Results obtained for particular values of $E(Z)$ and $E(Y)$ are as follows:

EZ/EY	n	$Mean$	$Variance$	Probability that email stops at n^{th} exchange	Probability that email stops eventually
EZ = 0.2 EY = 0.5	5	0.001000	0.001667	0.999262	1.000000
EZ = 1.1 EY = 0.6	5 15 35	12.52 0.19 0.000048	51.55 0.92 0.000227	0.008703 0.935185 0.999984	1.000000
EZ = 1.0 EY = 1.1	10	259.37	8680.89	0.000001	0.000085
EZ = 1.3 EY = 1.8	5	7015.83	1013857.83	0.000000	0.000000

Code for these first 3 parts is as follows:

Listing 2: Matlab code for HW3 Question 2abc

```
function HW3Q2abc(n)
%without noise case
EZ = 1.0; VZ = EZ;
EY = 1.0; VY = EY;
m = 100;
n = 10; %if no value is given in input while running
EX0 = m;

%% part a)
EXn = (EZ*EY)^n*EX0;
VXn = (((EZ*EY))^n-1)/(EZ*EY-1)*(EZ*VY+EY^2*VZ)*(EZ*EY)^(n-1)*EX0;
fprintf('WITHOUT␣noise␣case:␣EZ␣=␣%f,␣EY␣=␣%f,␣n␣=␣%d␣\n',EZ,EY,n);
fprintf('Mean␣␣␣␣:␣%f␣\n', EXn);
fprintf('Variance:␣%f␣\n', VXn);

%% part b) we compute P(Xn=0) by G_Xn(0)
temp = 0; %corresponds to s=0
for i=1:n %recursion from this point on
   temp = compute_G(temp,EY);
   temp = compute_G(temp,EZ);
end
GXn0 = temp^m; %G_Xn(0)
st = ['st';'nd';'rd';'th']; %for display purpose
fprintf('Probability␣that␣email␣stops␣at␣%d%s␣exchange:␣%f␣\n',n,st(min(4,n),:),GXn0);

%% part c) solving the nonlinear equation is hard so will find out the G_Xn(0) as n -> infinity
GXinf0 = compute_G(compute_G(GXn0,EY),EZ);
diff = abs(GXn0-GXinf0);
while(diff>1e-10)%computing the pgf GXn at 0 until it converges (for n->infinity)
   temp = compute_G(compute_G(GXinf0,EY),EZ);
   diff = abs(GXinf0-temp);
   GXinf0 = temp;
end
GXinf0 = GXinf0^m; %G_Xinf(0)
fprintf('Probability␣that␣email␣stops␣eventually:␣%f␣\n',GXinf0);
return
end

function G = compute_G(s,lambda) %function to compute PGF of poisson distribution with mean lambda
   G = exp(lambda*(s-1));
   return
end
```

d) For part d, we assume p^N is also a Poisson distribution with some mean, and use the formulas obtained in the noise case to compute the required quantities. Results obtained in this case for particular values of $E(Z)$, $E(Y)$ and $E(T)$ are as follows:

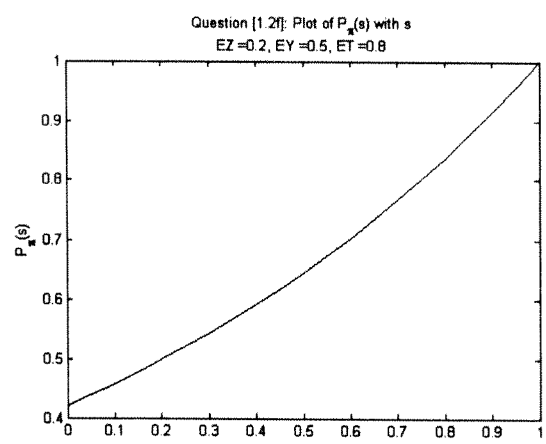
EZ/EY	n	$Mean$	$Variance$	Probability that email stops at n^{th} exchange	Probability that email stops eventually
EZ = 0.2 EY = 0.5 ET = 0.8	5	0.889880	0.801667	0.419854	0.420161
EZ = 0.2 EY = 0.5 ET = 3.8	5	4.22	3.80	0.016252	0.016264

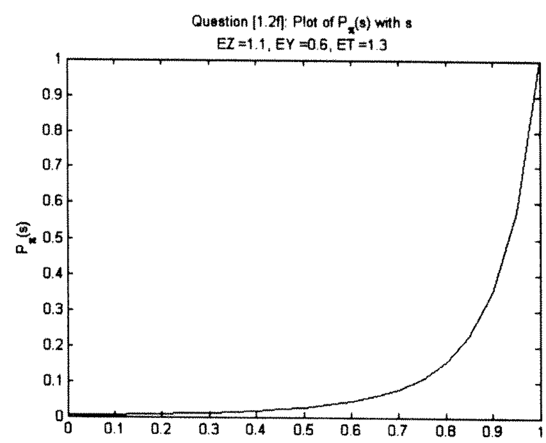
EZ/EY	n	$Mean$	$Variance$	Probability that email stops at n^{th} exchange	Probability that email stops eventually
EZ = 1.1 EY = 0.6 ET = 1.3	5 15 35	15.86 4.012 3.82	52.85 2.222 1.3	0.000849 0.077071 0.082201	0.082202
EZ = 1.1 EY = 0.6 ET = 3.8	5 15 35	22.30 11.35 11.17	55.35 4.72 3.8	0.000010 0.000634 0.000673	0.000673
EZ = 1.3 EY = 1.8, ET = 0.5	5	7041.64	1013858.33	0.000000	0.000000

Observations: We notice that results obtained are very well expected. In the without noise case, when $EZ * EY < 1$, we get eventual extinction with probability 1. We also see that there is very small room when $EZ * EY > 1$ which leads to a probability of extinction not equal to 0. In the noise case as well, the results follow similar pattern, but we see lot more variations in values because of noise factor. This time we see that even when $EZ * EY < 1$, the probability of eventual extinction is non-zero. With higher EZ or EY or higher ET, we see that the drop in extinction probability which is expected.

e) On paper only.

f) In part e, we had obtained the expression that was written earlier. We observed that it wasn't really solvable by hand because it consisted infinite compositions. We solve it here numerically using our own distributions. The graphs of limiting distributions $\mathcal{P}_\pi(s)$ for different choices of $E(Z)$, $E(Y)$ and $E(T)$ look like this:





$$p_{X_2}(0) = p_0^N \times p_{N_n}(\mathbb{P}_{IO}(0)) \times \mathbb{P}_{IO}^{(2)}(0)^m$$

....

$$p_{X_n}(0) = p_0^N \left(\prod_{i=1}^{n-1} p_{N_n}\left(\mathbb{P}_{IO}^{(i)}(0)\right) \right) \times \mathbb{P}_{IO}^{(n)}(0)^m \qquad (*)$$

Just to make my notation clear:

$$\mathbb{P}_{IO}(s) = p_{I_{n,k}}\left(p_{O_{n,l}}(s)\right), \mathbb{P}_{IO}^{(n)}(s) = \underbrace{\mathbb{P}_{IO}\left(\mathbb{P}_{IO}\left(\ldots\left(\mathbb{P}_{IO}(\mathbb{P}_{IO}(s))\right)\right)\right)}_{n\ fold}$$

Then I can use (*) to compute the probability $\Pr(X_n = 0|X_0 = m)$. This makes perfect sense since the three terms in this formula can be interpreted as:

- $p_0^N = \Pr(N_{n-1} = 0)$, represents the probability that in the last epoch there is no additional emails added;
- $\prod_{i=1}^{n-1} p_{N_n}\left(\mathbb{P}_{IO}^{(i)}(0)\right)$, represents the probability that the ones and their offspring generated by additional emails in first n-1 epochs die out;
- $\mathbb{P}_{IO}^{(n)}(0)^m$, as we have already seen, represents the probability that the original m emails and their offspring die out.

Hence, the required probability can be computed as:

$$\Pr(X_n = 0|X_{n-1} \neq 0) = p_{X_n}(0) - p_{X_{n-1}}(0)$$

The code **StopRolling2.m** implemented this approach, which taken three Poisson parameters $(\lambda_I, \lambda_O, \lambda_N)$ together with m and n and output the termination probability at epoch n. forever ? 8/15

e. If such stationary distribution exist we must have:

$$X_n \sim \pi, \qquad X_{n+1} \sim \pi$$

$$\mathbb{P}_\pi(s) = \mathbb{P}_\pi(p_{I_{n,k}}\left(p_{O_{n,l}}(s)\right) \times p_{N_n}$$

5/5

f. w

2. Numerical Computations

2.1 Branching Out in Two Ways (60 plus 25 bonus points)

a. Here I implemented the Monte Carlo simulations of this branching process with the Matlab function **twoWayBranching.m**, which takes the following input parameters:

jointProbA: the joint probability distribution $p^{(A)}$, as a matrix
jointProbB: the joint probability distribution $p^{(B)}$, as a matrix
N: the total number of epochs being simulated in this simulation
runNum: The total number of simulations conducted
Astart: The starting number of agent A, $X_0^{(A)}$
Bstart: The starting number of agent B, $X_0^{(B)}$

$$\begin{aligned}\mathcal{P}_\pi(s) &= \mathcal{P}_\pi(\mathcal{P}_Y(s))\,\mathcal{P}_N(s)\\ \mathcal{P}_\pi(s) &= \mathcal{P}_\pi(\mathcal{P}_I(\mathcal{P}_O(s)))\,\mathcal{P}_N(s)\end{aligned} \tag{29}$$

3/5

Equation 29 is the required relationship.

(f) It may not be possible to directly solve equation 29 to get $\mathcal{P}_\pi(s)$. We instead use the idea that if $\mathcal{P}_\pi(s)$ exists, then we can express it as a limit of $\mathcal{P}_{X_n}(s)$. Using equation 27 we can already express $\mathcal{P}_{X_n}(s)$ in terms of repeated composition of $\mathcal{P}_N(s)$, $\mathcal{P}_O(s)$ and $\mathcal{P}_I(s)$. Hence

$$\begin{aligned}\mathcal{P}_\pi(s) &= \lim_{n\to\infty} \mathcal{P}_{X_n}(s)\\ &= \lim_{n\to\infty} \left(\mathcal{P}_Y^n(s)\right)^m \prod_{k=0}^{n-1} \mathcal{P}_N\left(\mathcal{P}_Y^k(s)\right)\end{aligned} \tag{30}$$

10/10

where $\mathcal{P}_Y(s) = \mathcal{P}_I(\mathcal{P}_O(s))$.

EXAMPLE

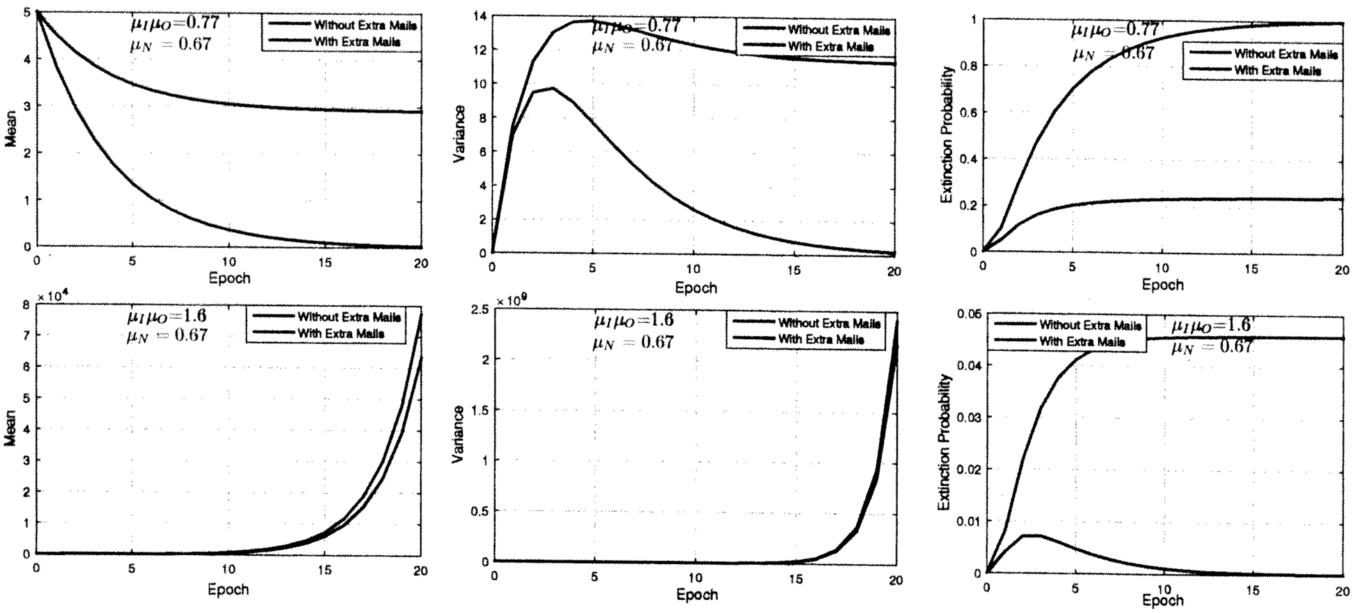


Figure 2: Figure to study the effect of $\mu_O\mu_I$ and extra mails on the mean, variance and extinction probability of the number of outgoing mails.

what does that mean? Should say what they are

We randomly generated two sets of p_O and p_I, one with $\mu_I\mu_O \approx 0.8 < 1$ and the other with $\mu_I\mu_O \approx 1.5 > 1$. We also generated a common p_N for both these sets with $\mu_N \approx 0.7$. We now do two kind of comparative study: **I)** The effect of $\mu_O\mu_I$. **II)** The effect of extra mails.

−5

The effect of $\mu_O\mu_I$: When $\mu_I\mu_O < 1$, the mean and variance of the number of outgoing mails converges to a finite value as time increases. If $\mu_I\mu_O > 1$, both mean and variance of the number of outgoing mails increase exponentially with time. The extinction probability increases with decrease in $\mu_O\mu_I$, Indeed, if there are no extra mails and $\mu_O\mu_I < 1$ then extinction probability converges to 1 while if $\mu_O\mu_I > 1$ then the extinction probability converges to a finite value.

The effect of extra mails: With extra mails mean and variance of the number of outgoing mails increases. With no extra mails, the mean and the variance were converging to 0 if $\mu_I\mu_O < 1$. However with extra mails it is converging to a finite value. If $\mu_O\mu_I < 1$ then with no extra mails the extinction probability converges to 1 but with extra mails it converges to a finite value. If $\mu_O\mu_I > 1$ and there are extra mails, then the extinction probability converges to 0.

---

want prob to ~~@~~ stop at epoch n

Table 2: Probability of Extinction

	Epoch							
	0	1	2	3	5	10	15	20
Extinction Probability	0	0.0224	0.0250	0.0252	0.0252	0.0252	0.0252	0.0252

5/10

C. Question 1.2.c

10/10

From the above table, we can find from the third epoch, the extinction probability remains almost same. We can judge the extinction probability converges to the value of 0.0252, as $n \to \infty$. So if $X_n \neq 0$, it means you need to deal with the email regarding the event. The probability is 1-0.0252=0.9748.

The extinction probabilities with varying n can be found in the attached document.

III. Question 2

A. Question 2.1.a

Let X_n^A denote the number of agents of type A at epoch n, and X_n^B denote the number of agents of type B at epoch n. Construct a matrix for $P^{(A)}$, where $P_{ij}^{(A)}$ denotes the probability that $y^{(AA)} = i - 1$ and $y^{(AB)} = j - 1$. Note that the sum of all entries in P^A is 1. Construct a similar matrix for $p^{(B)}$, where $P_{ij}^{(B)}$ denotes the probability that $y^{(BA)} = i - 1$ and $y^{(BB)} = j - 1$. Then the update rule is

$$X_{n+1}^A = \sum_{k=1}^{X_n^A} y_{k,n}^{(AA)} + \sum_{k=1}^{X_n^B} y_{k,n}^{(BA)}, \quad X_{n+1}^B = \sum_{k=1}^{X_n^A} y_{k,n}^{(AB)} + \sum_{k=1}^{X_n^B} y_{k,n}^{(BB)}, \tag{16}$$

where $y_{k,n}^{(AA)}$ is the number of offspring of type A produced from the kth agent of type A at epoch n.

Based on the probability defined in matrix $P^{(A)}$, we divide the range from 0 to 1 into $m \times n$ intervals, where m and n are the numbers of rows and columns of $P^{(A)}$. The process is same to matrix $P^{(B)}$. To simulate the process, at epoch n, we sample $X_n^A + X_n^B$ numbers that are uniformly distributed in [0,1]. From the intervals where the sampled numbers lie in, we can determine the number of offspring of type A and B. Sum up the offspring of type A and B of each agent respectively. Then we get X_{n+1}^A, X_{n+1}^B. Repeat the process.

15/15

B. Question 2.1.b

First we set the matrices of $P^{(A)}$ and $P^{(B)}$ as

$$P^{(A)} = P^{(B)} = \begin{pmatrix} 0.05 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.25 \end{pmatrix}$$

The result is as follows:

Table 3: Numbers of agents of type A and type B

Agent Type	Epoch					
	0	1	2	5	8	10
A	100	235	585	7338	96340	539380
B	100	231	550	7333	96384	540197

We can see in this case, both types coexist and flourish.
The case that both types go to extinct:

$$P^{(A)} = P^{(B)} = \begin{pmatrix} 0.5 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix}$$

The result is shown as follows:
The case that one agents takes over:

$$P^{(A)} = P^{(B)} = \begin{pmatrix} 0.25 & 0.05 \\ 0.3 & 0.05 \\ 0.3 & 0.05 \end{pmatrix}$$

The result is as follows:

Table 4: Numbers of agents of type A and type B

Agent Type	Epoch					
	0	1	5	10	20	50
A	100	58	53	34	12	0
B	100	101	80	55	20	0

Table 5: Numbers of agents of type A and type B

Agent Type	Epoch					
	0	1	5	10	15	20
A	100	198	409	1023	2636	6729
B	100	32	68	140	376	954

Suppose the number of agents of type A and B are $X_n^A \in \mathbb{Z}_+$ and $X_n^B \in \mathbb{Z}_+$, and $X_n^A + X_n^B > 0$, we can find the expected number of offspring in the next epoch:
1) case 1:

$$\begin{aligned}
X_{n+1}^A &= (X_n^A + X_n^B) \cdot (0 \times (0.05 + 0.1 + 0.1) + 1 \times (0.1 + 0.2 + 0.1) + 2 \times (0.1 + 0.1 + 0.25) \\
&= 1.3(X_n^A + X_n^B) > X_n^A \\
X_{n+1}^B &= (X_n^A + X_n^B) \cdot (0 \times (0.05 + 0.1 + 0.1) + 1 \times (0.1 + 0.2 + 0.1) + 2 \times (0.1 + 0.1 + 0.25) \\
&= 1.3(X_n^A + X_n^B) > X_n^B
\end{aligned}$$

In this case, both types flourish. The formula shows the number of type A is equal to the number of type B, which can be verified in the simulation of case 1.
2) case 2:

$$\begin{aligned}
X_{n+1}^A &= (X_n^A + X_n^B) \cdot (0 \times (0.5 + 0.1 + 0.1) + 1 \times (0.1 + 0.1 + 0.1) \\
&= 0.3(X_n^A + X_n^B) \\
X_{n+1}^B &= (X_n^A + X_n^B) \cdot (0 \times (0.05 + 0.1) + 1 \times (0.1 + 0.1) + 2 \times (0.1 + 0.1) \\
&= 0.6(X_n^A + X_n^B) \\
X_{n+1}^A + X_{n+1}^B &= 0.9(X_n^A + X_n^B) < X_n^A + X_n^B
\end{aligned}$$

Therefore, in this case, the expected total amount decreases epoch by epoch, and both types go to extinction.
3) case 3:

$$\begin{aligned}
X_{n+1}^A &= (X_n^A + X_n^B) \cdot (0 \times (0.25 + 0.05) + 1 \times (0.3 + 0.05) + 2 \times (0.3 + 0.05) \\
&= 1.05(X_n^A + X_n^B) \\
X_{n+1}^B &= (X_n^A + X_n^B) \cdot (0 \times (0.25 + 0.3 + 0.3) + 1 \times (0.05 + 0.05 + 0.05) \\
&= 0.15(X_n^A + X_n^B) \\
X_{n+1}^A + X_{n+1}^B &= 1.2(X_n^A + X_n^B) > X_n^A + X_n^B
\end{aligned}$$

10/10

In this case, the expected total amount increases epoch by epoch, and type A takes over, the ratio of type A and type B is 7, which can be verified the result of the corresponding simulation.

C. *Question 2.1.c*

Here we consider the scenario of cell division and mutation. Type A denotes normal cells, and Type B denotes cancer cells. Each epoch one cell can remain the same type, mutate into another type, divide into 2 cells of the same type, or die.
1) case 1: for the normal cells

$$P^{(A)} = \begin{pmatrix} 0.3(\text{death}) & 0.1(\text{become cancer cell}) \\ 0.2(\text{no change}) & 0 \\ 0.4(\text{divide into 2 cells}) & 0 \end{pmatrix}$$

For the cancer cells

$$P^{(B)} = \begin{pmatrix} 0.25(\text{death}) & 0.2(\text{no change}) & 0.5(\text{divide into 2 cells}) \\ 0.05(\text{become normal cells}) & 0 & 0 \end{pmatrix}$$

Table 6: Numbers of normal cells and cancer cells

Agent Type	Epoch					
	0	1	5	10	15	20
Normal cells	100	107	198	362	774	1854
Cancer cells	100	133	339	1013	2717	7226

Then we can compute the number of cells at each epoch in the simulation: We can find in this case the cancer cells dominates.
2) case 2: Now we employ chemotherapy to deal with cancer. This method can kill both normal cells and cancer cells. The probability distribution matrices become

$$P^{(A)} = \begin{pmatrix} 0.4(\text{death}) & 0.1(\text{become cancer cell}) \\ 0.2(\text{no change}) & 0 \\ 0.3(\text{divide into 2 cells}) & 0 \end{pmatrix}$$

$$P^{(B)} = \begin{pmatrix} 0.5(\text{death}) & 0.2(\text{no change}) & 0.25(\text{divide into 2 cells}) \\ 0.05(\text{become normal cells}) & 0 & 0 \end{pmatrix}$$

The number of cells at each epoch in the simulation now is:

Table 7: Numbers of normal cells and cancer cells

Agent Type	Epoch					
	0	1	5	10	15	20
Normal cells	100	72	25	16	6	4
Cancer cells	100	78	19	4	2	0

+16

We can find though this is an effective method to handle cancer cells, it does harm to normal cells as well.

IV. QUESTION 3

A. *Question 3.1*

Proof: since $f^*(j) = \sup_{f \in \mathcal{A}} f(j)$,
- from $f(k) = 0$, we have $f^*(k) = 0$,
- from $0 \le f(j) \le 1$ fro all $j \in S$, we have $0 \le f^*(j) \le 1$ for all $j \in S$,
- from f is subharmonic at all $j \ne k$ with respect to P, we have $\sum_{j \in S} P_{ij} f(j) \ge f(i)$ for all $j \ne k$, hence

$$\sup\left(\sum_{j \in S} P_{ij} f(j) \ge f(i)\right) \ge \sup(f(i)) = f^*(i).$$

Since $P_{ij} \ge 0$ for all $i, j \in S$, we have

$$\sup\left(\sum_{j \in S} P_{ij} f(j)\right) \le \sum_{j \in S} P_{ij} \cdot \sup(f(j)) = \sum_{j \in S} P_{ij} f^*(j).$$

Hence the inequality

$$\sum_{j \in S} P_{ij} f^*(j) \ge f^*(i)$$

5/5

holds for all $i \ne k$.
All the three conditions get satisfied, hence $f^* \in \mathcal{A}$.

B. *Question 3.1.b*

Suppose the equality $\sum_{j \in S} P_{ij} f^*(j) = f^*(i)$ does not hold for some $i \ne k, i \in S$, we choose one inequality to analyze, for example let $i = m$. From the conclusion in Question 3.1.a, since $\sum_{j \in S} P_{ij} f^*(j) \ne f^*(m)$, then we can have

$$\sum_{j \in S} P_{mj} f^*(j) > f^*(m).$$

```
% Problem 2.1.a
clear;clc;
prompt='Please input an m*n matrix for p_A (note that m<=5, n<=5):\n';
p_A=input(prompt);
% the probability transition matrix for type A
[m_A, n_A]=size(p_A);
p_A=p_A/sum(sum(p_A));
% conduct normalization to get the probability

% record the probability interval and correspomding row index and column
% index
index_A=zeros(m_A*n_A,3);
for j=1:n_A
    for i=1:m_A
        index_A(i+(j-1)*m_A,:)=[sum(p_A(1:i+(j-1)*m_A)) i j];
        % recoord the probability sum, corresponding row index, column
        % index
    end
end

prompt='Please input an m*n matrix for p_B (note that m<=5, n<=5):\n';
p_B=input(prompt);
% the probability transition matrix for type B
[m_B, n_B]=size(p_B);
p_B=p_B/sum(sum(p_B));
% conduct normalization to get the probability

% record the probability interval and correspomding row index and column
% index
index_B=zeros(m_B*n_B,3);
for j=1:n_B
    for i=1:m_B
        index_B(i+(j-1)*m_B,:)=[sum(p_B(1:i+(j-1)*m_B)) i j];
        % recoord the probability sum, corresponding row index, column
        % index
    end
end

% the initial number of agents of type A and B
A=100;
B=100;

% the simulation epochs
length=20;
n_agents=zeros(length,2);
for i=1:length
    sample=rand(A+B,1);
    offspring=[0 0];
    for j=1:A
        k=find(sample(j)<=index_A(:,1),1,'first');
        % find the corresponding probability interval
        offspring=offspring+index_A(k,2:3)-1;
        % get the corresponding number of offsprings of type A and B
    end
    for j=A+1:A+B
        k=find(sample(j)<=index_B(:,1),1,'first');
```

```
        % find the corresponding probability interval
        offspring=offspring+index_B(k,2:3)-1;
        % get the corresponding number of offsprings of type A and B
    end
    % update the number of agents of type A and B
    A=offspring(1);
    B=offspring(2);
    n_agents(i,:)=offspring;
end
```

$$\mathbf{E}(X_{n+1}^A) = s\frac{d}{ds}p_{X_{n+1}^A}(s)\,|_{s=1} = s\frac{d}{ds}p_{X_n^A}\left(p_{Y^{AA}}(s)\right)p_{X_n^B}\left(p_{Y^{BA}}(s)\right)|_{s=1}$$

$$= s\left(p'_{X_n^A}\left(p_{Y^{AA}}(s)\right)p'_{Y^{AA}}(s)p_{X_n^B}\left(p_{Y^{BA}}(s)\right)\right.$$

$$\left.+\, p_{X_n^A}\left(p_{Y^{AA}}(s)\right)p'_{X_n^B}\left(p_{Y^{BA}}(s)\right)p'_{Y^{BA}}(s)\right)|_{s=1}$$

$$= p'_{X_n^A}\left(p_{Y^{AA}}(1)\right)p'_{Y^{AA}}(1)p_{X_n^B}\left(p_{Y^{BA}}(1)\right) + p_{X_n^A}\left(p_{Y^{AA}}(1)\right)p'_{X_n^B}\left(p_{Y^{BA}}(1)\right)p'_{Y^{BA}}(1)$$

$$= p'_{X_n^A}(1)p'_{Y^{AA}}(1) + p'_{X_n^B}(1)p'_{Y^{BA}}(1) = \mathbf{E}(X_n^A)\mathbf{E}(Y^{AA}) + \mathbf{E}(X_n^B)\mathbf{E}(Y^{BA})$$

Similarly,

$$\mathbf{E}(X_{n+1}^B) = \mathbf{E}(X_n^A)\mathbf{E}(Y^{AB}) + \mathbf{E}(X_n^B)\mathbf{E}(Y^{BB})$$

Write in matrix form,

$$\begin{pmatrix}\mathbf{E}(X_{n+1}^A) & \mathbf{E}(X_{n+1}^B)\end{pmatrix} = \begin{pmatrix}\mathbf{E}(X_n^A) & \mathbf{E}(X_n^B)\end{pmatrix}\begin{pmatrix}\mathbf{E}(Y^{AA}) & \mathbf{E}(Y^{BA})\\ \mathbf{E}(Y^{AB}) & \mathbf{E}(Y^{BB})\end{pmatrix}$$

By iteration,

$$\begin{pmatrix}\mathbf{E}(X_n^A) & \mathbf{E}(X_n^B)\end{pmatrix} = \begin{pmatrix}\mathbf{E}(X_0^A) & \mathbf{E}(X_0^B)\end{pmatrix}\begin{pmatrix}\mathbf{E}(Y^{AA}) & \mathbf{E}(Y^{BA})\\ \mathbf{E}(Y^{AB}) & \mathbf{E}(Y^{BB})\end{pmatrix}^n$$

If eigenvalue of matrix $\begin{pmatrix}\mathbf{E}(Y^{AA}) & \mathbf{E}(Y^{BA})\\ \mathbf{E}(Y^{AB}) & \mathbf{E}(Y^{BB})\end{pmatrix}$ is no greater than one, elements of $\begin{pmatrix}\mathbf{E}(Y^{AA}) & \mathbf{E}(Y^{BA})\\ \mathbf{E}(Y^{AB}) & \mathbf{E}(Y^{BB})\end{pmatrix}^n$ are all bounded. Thus $\begin{pmatrix}\mathbf{E}(X_n^A) & \mathbf{E}(X_n^B)\end{pmatrix}$ is bounded. This implies there is no probability that $\begin{pmatrix}X_n^A & X_n^B\end{pmatrix} \to \infty$, because this will give a infinity weight to large n in $\begin{pmatrix}\mathbf{E}(X_n^A) & \mathbf{E}(X_n^B)\end{pmatrix}$. Hence extinction is certain. + 3 more detail...

3.1 *Another Criterion for Transience...*

According to the definition of $\mathcal{A}$, $f(k) = 0$ for all $f \in \mathcal{A}$. Hence ✓

$$f^*(k) = \sup_{f\in\mathcal{A}} f(k) = 0$$

Since $f^*(j)$ is the supremum of $f(j)$, and as mentioned in the definition, 1 is an upper bound of $f(j)$, we have

$$f^*(j) \le 1$$ ✓

Also, for a specific f,

$$f^*(j) \ge f(j) \ge 0$$

Now consider state $i \ne k$, by definition, for all $f \in \mathcal{A}$, ✓

$$f(i) \le \sum_{j\in S} P_{ij}f(j) \le \sum_{j\in S} P_{ij}f^*(j)$$

This equation implies that $\sum_{j\in S} P_{ij}f^*(j)$ is an upper bound of $f(i)$. Thus

$$f^*(i) \le \sum_{j\in S} P_{ij}f^*(j)$$ ✓

All of the above shows that f^* meets the requirements of $\mathcal{A}$, hence $f^* \in \mathcal{A}$.

5/5

Suppose for state $l \neq k$,

$$\sum_{j \in S} P_{lj} f^*(j) = a > f^*(l)$$

Define a new function g

$$g(j) = \begin{cases} f^*(j), & j \neq l \\ a, & j = l \end{cases}$$

It is obvious that $g(j) \geq f^*(j)$ for all $j \in S$.
Since $l \neq k$

$$g(k) = f^*(k) = 0$$

As $P_{lj} \geq 0$ and $f^*(j) \geq 0$,

$$a = \sum_{j \in S} P_{lj} f^*(j) \geq 0$$

Also, $f^*(j) \leq 1$,

$$a = \sum_{j \in S} P_{lj} f^*(j) \leq \sum_{j \in S} P_{lj} = 1$$

Hence $0 \leq g(j) \leq 1$ for all $j \in S$.
For all $i \neq l$ and $i \neq k$,

$$\sum_{j \in S} P_{ij} g(j) \geq \sum_{j \in S} P_{ij} f^*(j) \geq f^*(i) = g(i)$$

For $i = l$,

$$\sum_{j \in S} P_{ij} g(j) = \sum_{j \in S} P_{lj} g(j) \geq \sum_{j \in S} P_{lj} f^*(j) = a = g(l) = g(i)$$

Hence,

$$\sum_{j \in S} P_{ij} g(j) \geq g(i)$$

Thus $g \in \mathcal{A}$. However, $g(l) = a > f^*(l)$, this is contradictory to the assumption that $f^*(j) = \sup_{f \in \mathcal{A}} f(j)$ for all $j \neq k$. Therefore,

$$\sum_{j \in S} P_{lj} f^*(j) \leq f^*(l)$$

holds for all $l \neq k$. Together with the conclusion above, we have

$$\sum_{j \in S} P_{ij} f^*(j) = f^*(i)$$

$-$ 10/10

Now looking at an irreducible Markov chain with transition probability matrix P which has a function f that $f \in \mathcal{A}$. According to the conclusion above, there must be a function $f^* \in \mathcal{A}$ satisfy

$$\sum_{j\in S} P_{ij} f^*(j) = f^*(i)$$

for all $i \neq k$. As $f^*(k) = 0$, the equation can be re-write to

$$\sum_{j\in S} P_{ij} f^*(j) = f^*(i), i, j \neq k$$

Define a matrix Q which delete the row and column corresponding to state k from P, we have

$$\sum_{j\in S^*} Q_{ij} f^*(j) = f^*(i)$$

where S^* is formed with elements of S except k. In other words, $f^*(j), j \in S^*$ is a solution of the equation $Qx = x$. Since $0 \le f^*(j) \le 1$ for all $j \in S$ and $f^*(j) > 0$ for some $j \in S$, it is a nonzero, nonnegative, bounded solution. Hence the Markov chain must be transient.

10/10