****

**MATH 3305**

**PROBABILITY AND RANDOM PROCESSES**

**PROJECT: Part I**

Fethiye Gül DİKAYAK - 17070007008

Sinem VATANSEVER - 16070003004

Instructor

Prof. Volkan RODOPLU

Fall 2020

**PART I-A:**

function [] = Simulation()%This is the main function

L = 3;

N = 14;

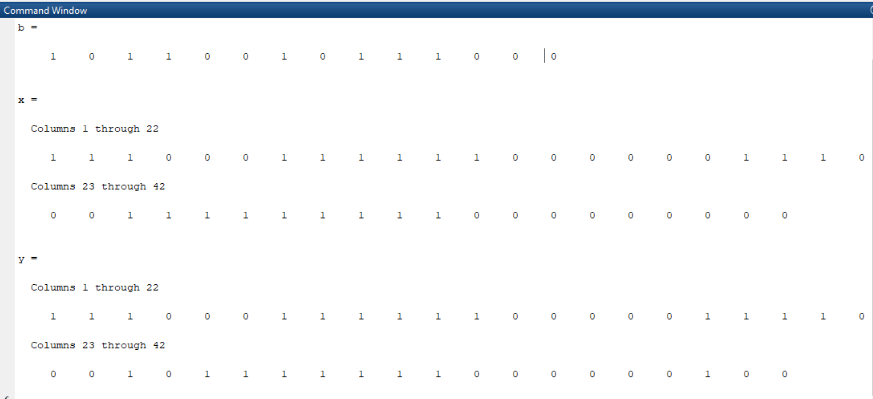
b = GenerateBits(N) %Generates N independent bits {0,1} with equal probability

x = ChannelEncoder(b) %Copies the bits 3 times

y = Channel(x) %wrapper function

end

*Results of the Simulation Code:*



*Other functions in Simulation Code:*

function b = GenerateBits(N) %Generate bits

b = randi([0 1],1,N);

end

function x = ChannelEncoder(b)

L = 3; %The parameter for the repetition code. This parameter can be changed.

x = RepetitionEncoder(b, L);

end

function x = RepetitionEncoder(c,L)

“x = reshape(repmat(c,L,1),1,[]);” %Retrieved from: https://in.mathworks.com/matlabcentral/answers/57388-is-there-a-function-in-matlab-that-creates-a-binary-repetition-coder

end

function y = Channel(x)

y = BinarySymmetricChannel(x);

end

function [output] = BinarySymmetricChannel(x)

E = 10^-1;

“output = bsc(x,E);” %"Retrieved from: https://ch.mathworks.com/help/comm/ref/bsc.html#:~:text=ndata%20%3D%20bsc(data%2Cprobability)%20passes%20the%20binary%20input,array%20in%20GF(2)."

end

**PART I-B:**

*Codes and results for a-b-c:*

%% a)

N = 100;

x = 1;

b = GenerateBits(N);

Matrix = RunningFrequencyComputation(b,x);

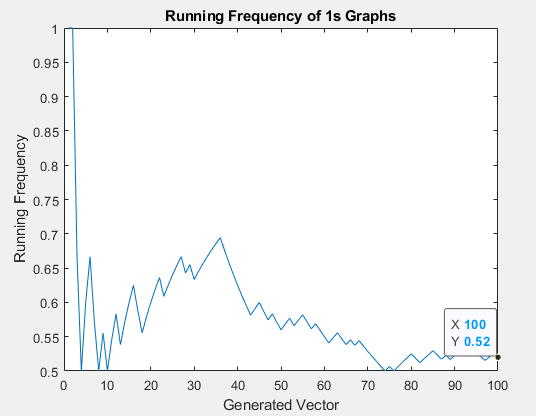
figure(1),

plot(Matrix)

title('Running Frequency of 1s Graphs')

ylabel('Running Frequency')

xlabel('Generated Vector')



%% b)

N = 100;

x = 0;

b = GenerateBits(N);

Matrix = RunningFrequencyComputation(b,x);

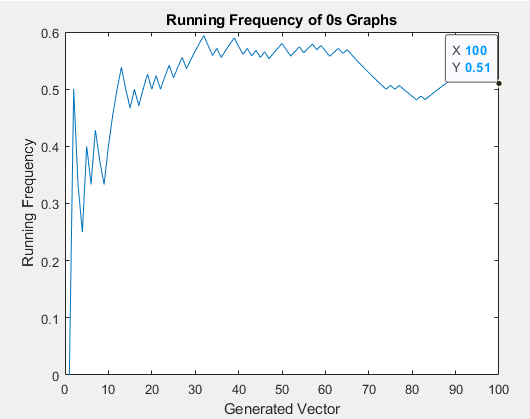
figure(2),

plot(Matrix)

title('Running Frequency of 0s Graphs')

ylabel('Running Frequency')

xlabel('Generated Vector')



* Graph of a part the running frequency of 1's is converging to 0.52 when N = 100. and Graph of b part the running frequency of 0's is converging to 0.51 when N = 100. If we say that N=1000 Graph of a part the running frequency of 1's is converging to x ' and Graph of b part the running frequency of 0's is converging to 0.5. So, we can clearly see frequency becomes more equally to for both cases.

%% c)

c = [0 1 0 1];

L\_equal\_1 = RepetitionEncoder(c,1)

L\_equal\_2 = RepetitionEncoder(c,2)

L\_equal\_3 = RepetitionEncoder(c,3)

L\_equal\_4 = RepetitionEncoder(c,4)

