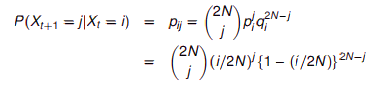
**Question 3**

The Wright Fisher model is explained as follows. Assume a simple haploid model; it consists of a population of 2N genes (or alternatively – N diploid organisms) of random reproduction, with each haploid possessing either allele A1 or allele A2. Initially, we may disregard mutation as well as selective forces. At each time-step, a gene (allele) reproduces some number of offspring (which are the exact copies of itself) and dies immediately after that; thus has a life-span of only one generation. The process, modeled thus, describes how the genes get transmitted from one generation to the next

Process of birth and death in the population remains hidden. The only observable is the frequency of alleles changing from generation to generation. The allele frequency of the next generation is governed only by a genetic drift. The Genetic drift is defined as a force that reduces heterozygosity by the random loss of alleles. We should just focus on the frequency of allele A1 in the population of 2N haploids. This is a process of changing from one generation to the next in terms of a Markov Chain, where the state X of the chain corresponds to the number of haploids (genes) of type A1.

In any generation X takes one of the values 0, 1, . . . , 2N, which constitutes a state space. Denote the value taken by X in generation t as Xt . The model assumes that genes for the generation t + 1 are derived by sampling with replacement from the genes of generation t. Thus, the make-up of the next generation is determined by 2N independent Bernoulli trials so that Xt is a binomial random variable. Let the initial generation consist of i genes of type A1 and 2N − i ,genes of type A2. Then we define a probability of success (resulting in allele A1) pi and a probability of failure qi (resulting in allele A2) for each Bernoulli trial as pi = i/2N, qi = 1 – i/2N. The process generates a Markov Chain {Xn}, where Xn is the number of A1 genes in the nth generation, among a constant population size of 2N individuals. Basically, Xt+1 is a binomial random variable with index 2N and parameter (probability of success) Xt/2N.

The transition probabilities from Xt = i to Xt+1 = j for this Markov Chain are computed according to the binomial distribution as



A matrix for which all the column vectors are probability vectors is called transition or stochastic matrix. A Markov chain is a process that consists of a finite number of states and some known probabilities pij, where pij is the probability of moving from state j to state i. Of particular interest is a probability vector **p** such that **Ap**=**p**, that is, an eigenvector of **A** associated to the eigenvalue 1. Such vector is called a steady state vector. The Perron–Frobenius theorem, asserts that a [real square matrix](https://en.wikipedia.org/wiki/Real_square_matrix) with positive entries has a unique largest real [eigenvalue](https://en.wikipedia.org/wiki/Eigenvalue) and that the corresponding [eigenvector](https://en.wikipedia.org/wiki/Eigenvector) can be chosen to have strictly positive components, and also asserts a similar statement for certain classes of nonnegative matrices. A Markov chain is ergodic if there is a number N such that any state can be reached from any other state in at most N steps (in other words, the number of steps taken are bounded by a finite positive integer N). In case of a fully connected transition matrix, where all transitions have a non-zero probability, this condition is fulfilled with N=1. A Markov chain with more than one state and just one out-going transition per state is either not irreducible or not aperiodic, hence cannot be ergodic.

**Code Description**

The code starts off with prompting the user to enter the value of the location of the A1 Allele in the gene. Initialize an array of zeros as in the code and the change the value of the location entered by the user to 1. Consider N to be 100. We then initialize a transition or stochastic matrix of 201x201 to zeros. We then populate each element of the matrix using the formula described above. Nchoosek function gives us nCr value. We choose n=1000 as the time indice. We then calculate the output as in the code below by multiplying with the transition matrix a number of times. The output is plotted and the results can be seen to be in a steady state.

**Code**

clc;

clear all;

close all;

user=input('Enter the location of A1 allele');%Accepting user input

input=zeros(1,201); % initial distribution

input(user)=1;%Passing user input

N = 100; % number of individuals

% transition matrix

P=zeros(2\*N+1,2\*N+1);

for i = 1:2\*N+1

for j = 1:2\*N+1

P(i,j) = nchoosek(2\*N,j-1)\*((i-1)/(2\*N))^(j-1)\*(1-(i-1)/(2\*N))^(2\*N-j+1);

end

end

n=1000; % number of time steps to take

output=zeros(n+1,2\*N+1); % clear out any old values

t=0:n; % time indices

output(1,:)=input; % generate first output value

for i=1:n,

output(i+1,:) = output(i,:)\*P;

%a tolerance check to automatically stop the simulation when the density is close to its steady-state

LIT =(output(i+1,:)- output(i,:)); %calculate the threshold

if all(LIT == 1)

break;

end

end

plot(t,output);

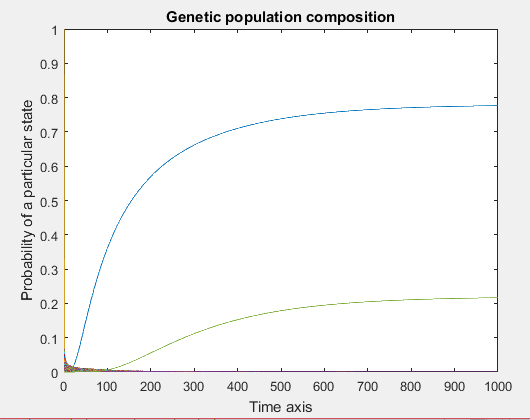
xlabel('Time axis');

ylabel('Probability of a particular state')

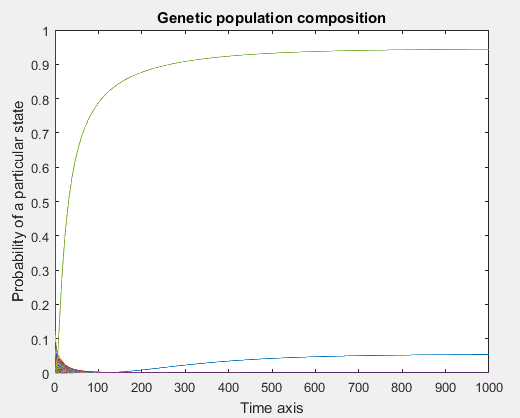
title('Genetic population composition')

Output

For user input =45



For user input=190



**Analysis**

As can be seen from the graphs above with different allele initial distribution, the steady state reaches more quickly in the first case when compared to the second graph. Hence, this defies the Perron-Frobenius theorem and the Markov chain ergodic theorem as can be clearly seen from above.