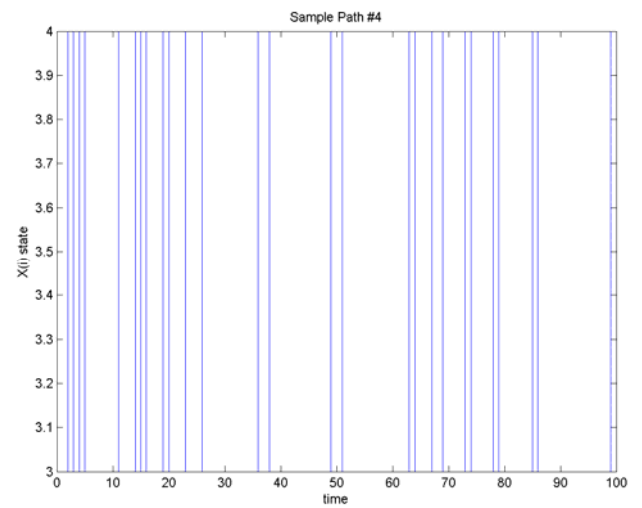
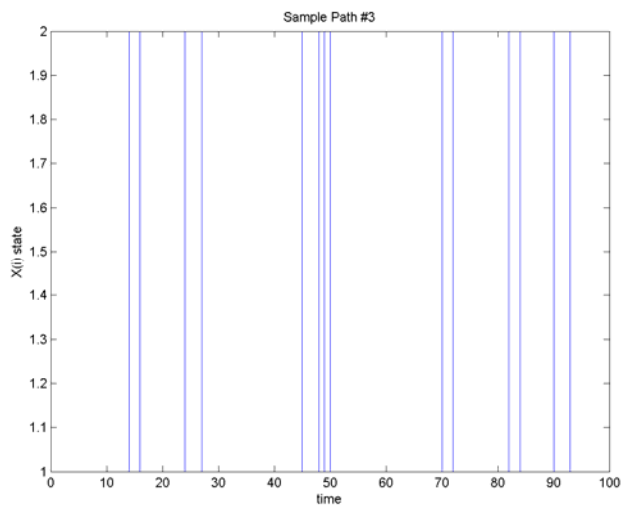
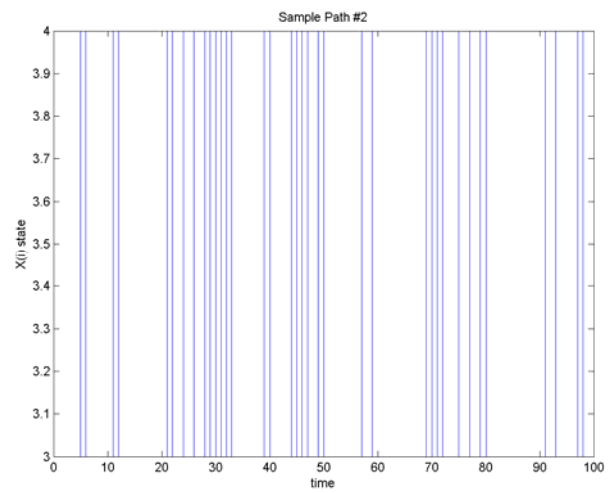
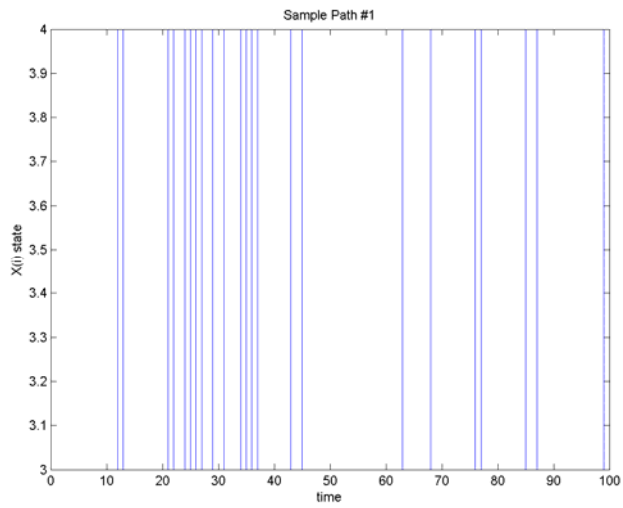
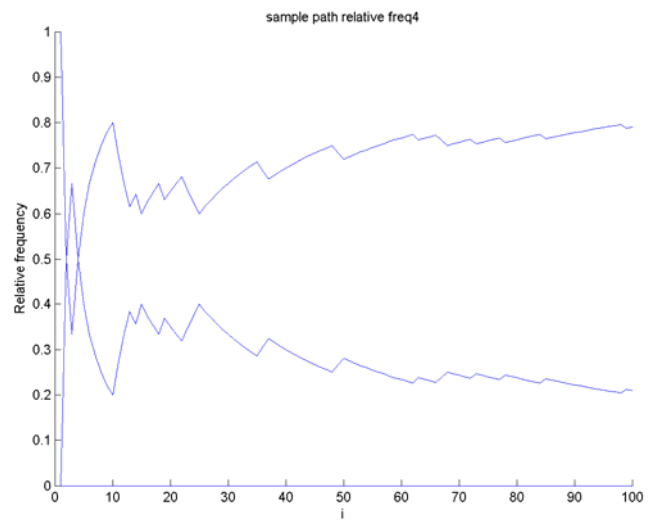
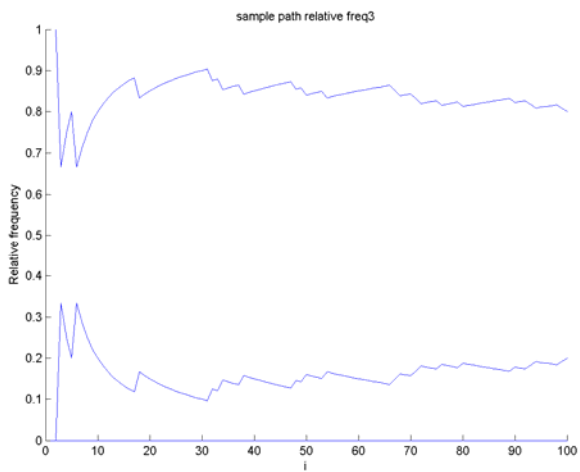
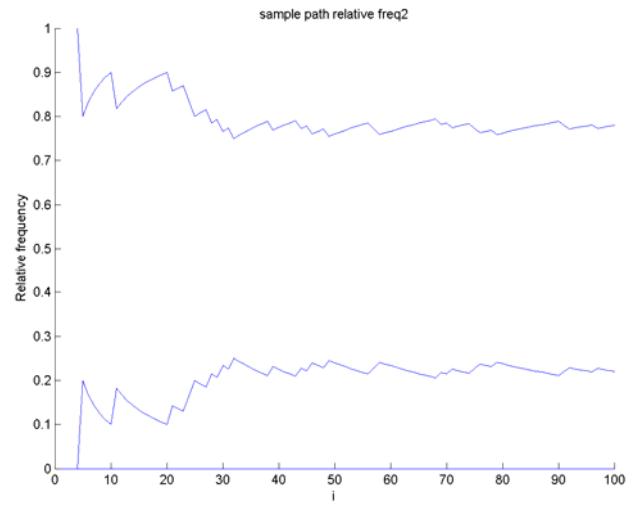
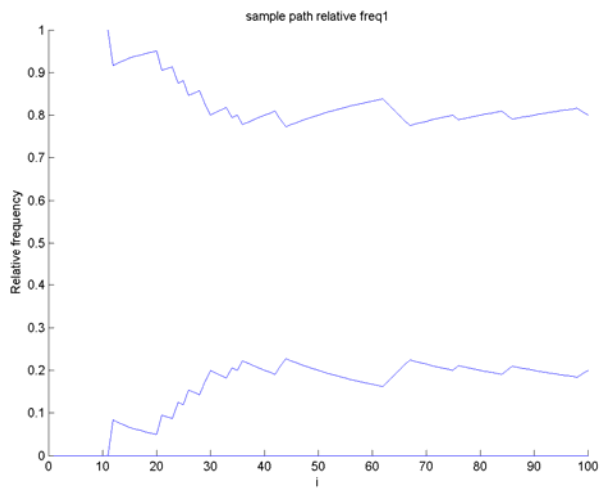


1.
OUTPUT





Sample path relative Frequency at i=100

When state= 1

v2_Rel_Freq =

0 0 0.2000 0.8000

When state= 2

v2_Rel_Freq =

0 0 0.2200 0.7800

When state= 3

v2_Rel_Freq =

0.1600 0.8400 0 0

When state= 4

v2_Rel_Freq =

0.1600 0.8300 0 0

>> vectoreig=v(1:4,2)/sum(v(1:4,2))

vectoreig1 =

0.1667

0.8333

0

0

>> vectoreig=v(1:4,4)/sum(v(1:4,4))

vectoreig2 =

0

0

0.2222

0.7778

Sample path relative Frequency at $i=10000$, to display convergence.

When state= 1

v2_Rel_Freq =

0 0 0.2234 0.7766

When state= 2

v2_Rel_Freq =

0 0 0.2264 0.7736

When state= 3

v2_Rel_Freq =

0.1627 0.8373 0 0

When state= 4

v2_Rel_Freq =

0.1647 0.8353 0 0

Comment

From the setup, one can notice that not all states communicate with each other. The transition matrices is irreducible when separated into two disjoint matrices that within the matrices do communicate with each other. Therefore if the state enters into one of these two matrices, it will continue and remain in the state, as seen in relative frequencies graphs.

CODE

```
%prob1
clear all
clc
trans = [0.5 0.5 0.0 0.0;
         0.1 0.9 0.0 0.0;
         0.0 0.0 0.3 0.7;
         0.0 0.0 0.2 0.8];

n=100;
state = zeros(4,n);
path=zeros(n);
for i=1:4;
    rand1 = rand/2;
    rand2 = rand/2;
    state(:,1)=[rand1 0.5-rand1 rand2 0.5-rand2];
    for j=1:n;
        x = rand;
        y = find(x<cumsum(state(:,j))));
        path(j,i) = min(y);
        state(:,j+1)=trans(path(j,i),:);
        for k=1:4;
            count(j,i,k)=length(find(path(1:j,i)==k))/j;
        end;
    end;
end;

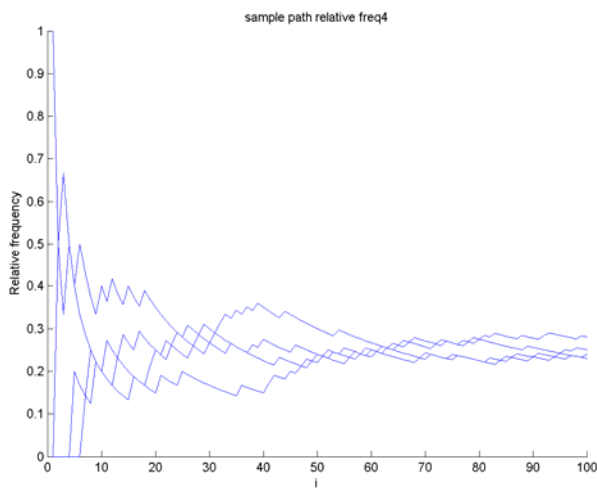
for i=1:4;
    figure(i);
    stairs(path(:,i));
    xlabel('time')
    ylabel('X(i) state')
    title(['Sample Path #' int2str(i)]);
    saveas(figure(i),['Prob1 Sample path ' int2str(i) '.png'])
end;
x1 = [1:1:100];
for i=1:4;
    figure(i+4);
    hold on
    plot(x1,count(:,i,1));
    plot(x1,count(:,i,2));
    plot(x1,count(:,i,3));
    plot(x1,count(:,i,4));
    xlabel('i');ylabel('Relative frequency');
    title(['sample path relative freq' int2str(i)]);
    saveas(figure(i+4),['Prob1 sample path rel freq ' int2str(i) '.png'])
    hold off
end;
for i=1:4;
    [v,d] = eig(trans');
    v1_eigen = v(1:4,1)/sum(v(1:4,1))
    v2_Rel_Freq = [count(n,i,1), count(n,i,2), count(n,i,3), count(n,i,4)]
end;

vectoreig1=v(1:4,2)/sum(v(1:4,2))
vectoreig2=v(1:4,4)/sum(v(1:4,4))
```

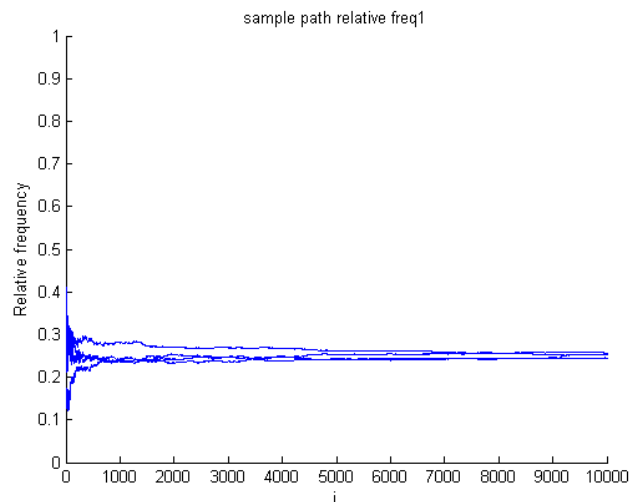
2.

- The process is irreducible because there exists positive probability which is associated with each transition from any state, say i , to any other state, j
- The process is aperiodic. This can be shown by noting that there exists positive probability that is associated with traveling from state 1 and returning to state 1 in one step. Therefore, the greatest common factor is 1 and thus by definition the chain is aperiodic.

OUTPUT



When $i=100$



When $i=10000$

The stationary probability is
 $v1_eigen =$

0.2500
0.2500
0.2500
0.2500

Sample path relative Frequency at $i=10000$, to display convergence.

When state= 1

v2_Rel_Freq =

0.2540 0.2420 0.2530 0.2510

When state= 2

v2_Rel_Freq =

0.2440 0.2500 0.2570 0.2490

When state= 3

v2_Rel_Freq =

0.2490 0.2640 0.2430 0.2440

When state= 4

v2_Rel_Freq =

0.2430 0.2470 0.2640 0.2460

total2 =

1.1250

avg1 =

1.1244

Comment

For each state, the averages along that path converge to the stationary probability. The above two values are approximately close to one another. Obtained 1.125 by the formula and through simulation, obtained 1.124

CODE

```
%prob1
clear all
clc

trans = [0.1 0.3 0.4 0.2;
         0.2 0.1 0.3 0.4;
         0.4 0.2 0.1 0.3;
         0.3 0.4 0.2 0.1];

n=10000;
state = zeros(4,n);
path=zeros(n);

for i=1:4;
    rand1 = rand/2;
    rand2 = rand/2;
    state(:,1)=[rand1 0.5-rand1 rand2 0.5-rand2];
    for j=1:n;
        x = rand;
        y = find(x<cumsum(state(:,j))));
        path(j,i) = min(y);
        state(:,j+1)=trans(path(j,i),:);

        for k=1:4;
            count(j,i,k)=length(find(path(1:j,i)==k))/j;
        end;
    end;
end;

% for i=1:4;
%     figure(i);
%     stairs(path(:,i));
%     xlabel('time')
%     ylabel('X(i) state')
%     title(['Sample Path #' int2str(i)]);
%     saveas(figure(i),['Prob 2 -Sample path ' int2str(i) '.png'])
% end;
%
x1 = [1:1:100];
for i=1:4;
    figure(i+4);
    hold on
    plot(x1,count(:,i,1));
    plot(x1,count(:,i,2));
    plot(x1,count(:,i,3));
    plot(x1,count(:,i,4));
    xlabel('i');
    ylabel('Relative frequency');
    title(['sample path relative freq' int2str(i)]);
    saveas(figure(i+4),['Prob 2 - sample path relative freq ' int2str(i) '.png'])
    hold off
end;
```

```
for i=1:4;
[v,d] = eig(trans');
v1_eigen = v/sum(v)
v2_Rel_Freq = [count(n,i,1), count(n,i,2), count(n,i,3), count(n,i,4)]
end;

f=[2 1 2.5 -1];
total2 = dot(f,v1_eigen)

n1=10000;

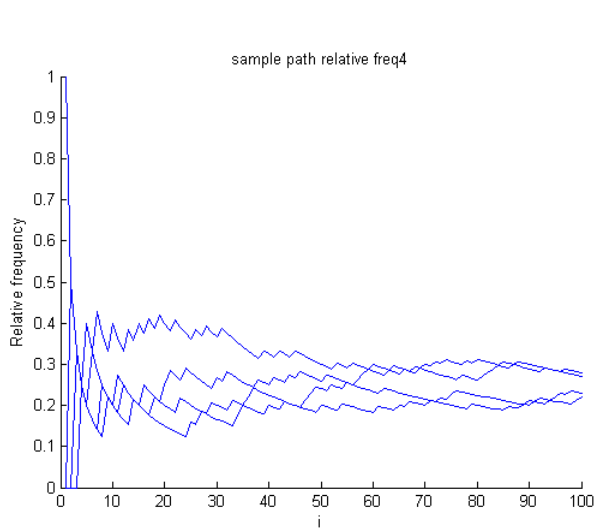
x(1) = ceil(4*rand);

for i=1:n1
[m,n]=sort(trans(x(i),:));
exitloop=0;
j=1;
u1=rand;
while exitloop==0;
    if u1<sum(m(1:j));
        x(i+1)=n(j);
        exitloop=1;
    else
        j=j+1;
    end
end
end

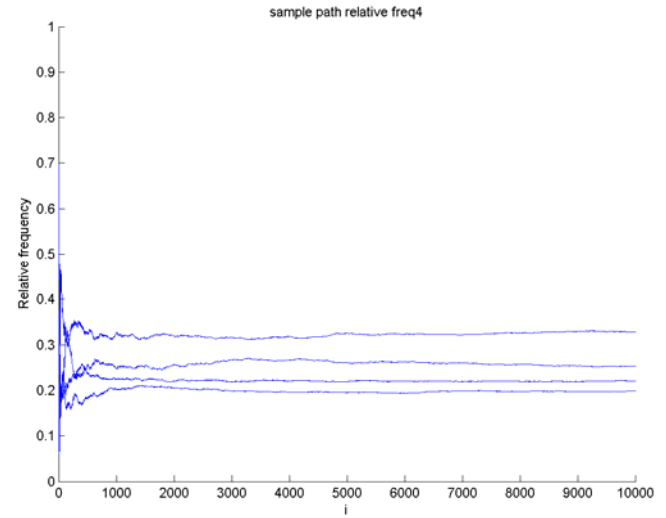
for i=1:n1;
    total1(i)=sum(f(x(i)));
end;

avg1=sum(total1)/n1
```

3. OUTPUT



When i=100



When i=10000

The stationary probability is

v1_eigen =

**0.1975
0.3333
0.2469
0.2222**

Sample path relative Frequency at $i=10000$, to display convergence.

When state= 1

v2_Rel_Freq =

0.1991 0.3245 0.2541 0.2223

When state= 2

v2_Rel_Freq =

0.1986 0.3199 0.2591 0.2224

When state= 3

v2_Rel_Freq =

0.2002 0.3332 0.2453 0.2213

When state= 4

v2_Rel_Freq =

0.1980 0.3281 0.2536 0.2203

total2 =

1.1235

avg1 =

1.1200

Comment

For each state, the averages along that path converge to the stationary probability. The above two values are approximately close to one another. Obtained 1.1235 by the formula and through simulation, obtained 1.1200

CODE

```
%prob3

clear all
clc

trans = [0.1 0.3 0.4 0.2;
         0.2 0.4 0.0 0.4;
         0.0 0.3 0.5 0.2;
         0.5 0.3 0.2 0.0];

n=10000;
state = zeros(4,n);
path=zeros(n);

for i=1:4;
    rand1 = rand/2;
    rand2 = rand/2;
    state(:,1)=[rand1 0.5-rand1 rand2 0.5-rand2];
    for j=1:n;
        x = rand;
        y = find(x<cumsum(state(:,j)));
        path(j,i) = min(y);
        state(:,j+1)=trans(path(j,i),:);

        for k=1:4;
            count(j,i,k)=length(find(path(1:j,i)==k))/j;
        end;
    end;
end;

% for i=1:4;
%     figure(i);
%     stairs(path(:,i));
%     xlabel('time')
%     ylabel('X(i) state')
%     title(['Sample Path #' int2str(i)]);
%     saveas(figure(i),['Prob3-Sample path prob3 ' int2str(i) '.png'])
% end;

x1 = [1:1:10000];
for i=1:4;
    figure(i+4);
    hold on
    plot(x1,count(:,i,1));
    plot(x1,count(:,i,2));
    plot(x1,count(:,i,3));
    plot(x1,count(:,i,4));
    xlabel('i');
    ylabel('Relative frequency');
    title(['sample path relative freq' int2str(i)]);
    saveas(figure(i+4),['Prob3-sample path relative freq ' int2str(i) '.png'])
    hold off
```

```
end;

for i=1:4;
    [v,d] = eig(trans');
    v1_eigen = v/sum(v)
    v2_Rel_Freq = [count(n,i,1), count(n,i,2), count(n,i,3), count(n,i,4)]
end;

f=[2 1 2.5 -1];
total2 = dot(f,v1_eigen)

n1=1000;

x(1) = ceil(4*rand);

for i=1:n1
    [m,n]=sort(trans(x(i),:));
    exitloop=0;
    j=1;
    u1=rand;
    while exitloop==0;
        if u1<sum(m(1:j));
            x(i+1)=n(j);
            exitloop=1;
        else
            j=j+1;
        end
    end
end
end

for i=1:n1;
    total1(i)=sum(f(x(i)));
end;

% for i=1:4
%     total2(i)=sum(f(i)*v1_eigen(i));
% end

avg1=sum(total1)/n1
```