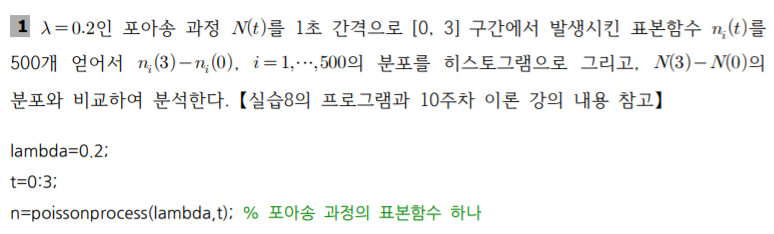
**Poisson Process, Brownian Motion Process,**

**and Stationary Gaussian Process**

201620350 김지영



- MATLAB Code

clear; clf;

lambda=0.2;

t=0:3;

M=500;

r=zeros(1,M);

R=zeros(1,M);

for i=1:M

n=poissonprocess(lambda,t);

plot(t,n), hold on;

axis([0 3 0 10]), grid on;

r(i)=n(4)-n(1);

N=poissonprocess(lambda,t)

R(i)=N(4)-N(1);

end

s=poissonarrivals(lambda,max(t));

N=fn\_count(s,t);

function s=poissonarrivals(lambda,T)

n=ceil(1.1\*lambda\*T);

s=cumsum(exponentialrv(lambda,n));

while (s(length(s))< T)

s\_new=s(length(s))+ ...

cumsum(exponentialrv(lambda,n));

s=[s; s\_new];

end

s=s(s<=T);

end

function x=exponentialrv(lambda,m)

x=-(1/lambda)\*log(1-rand(m,1));

end

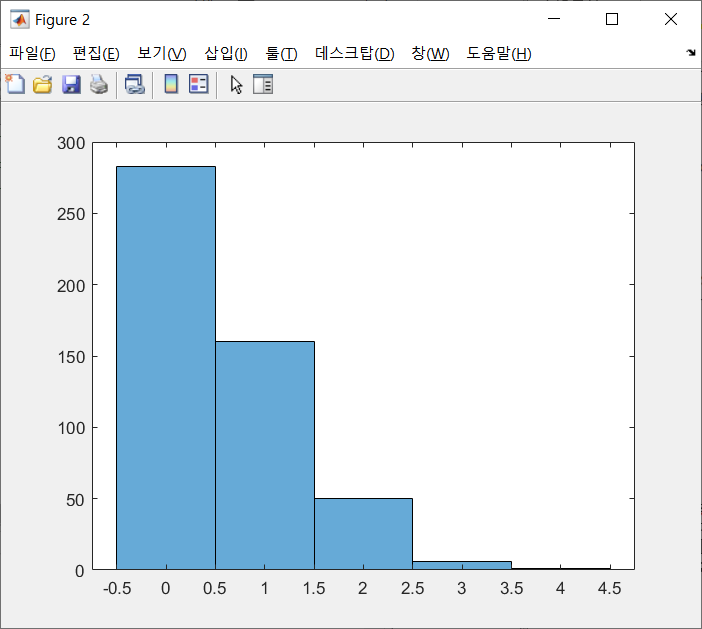
function n=fn\_count(x,y)

[MX,MY]=ndgrid(x,y);

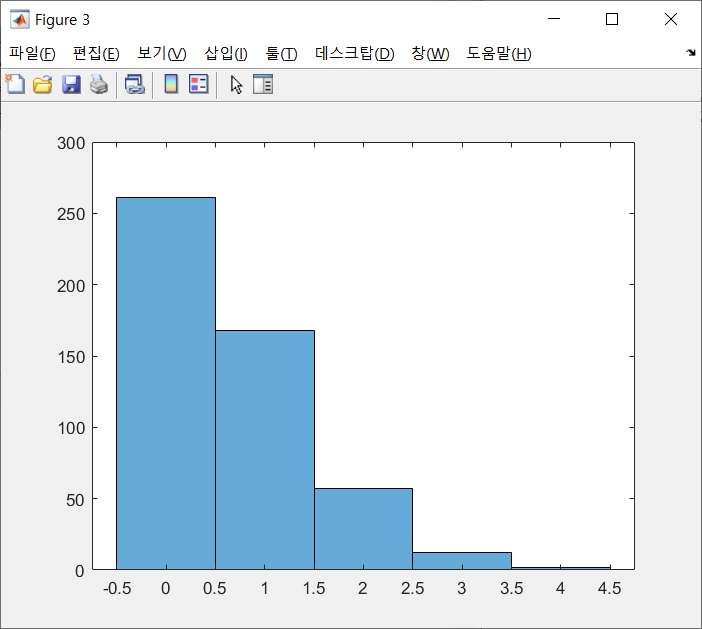
n=(sum((MX<=MY),1))';

end

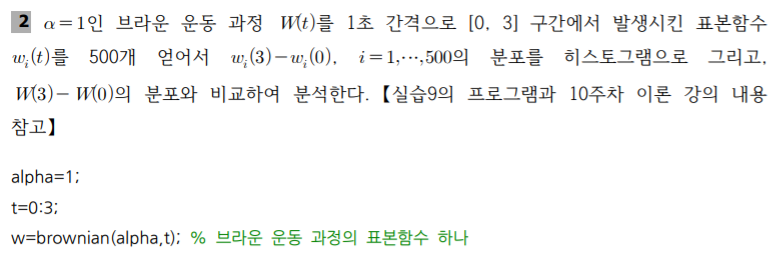
>> figure(2), histogram(r)



>> figure(3), histogram(R)



=> figure 2는 표본함수 중에 n\_i(3)-n\_i(0) 의 분포를 히스토그램으로 나타낸 것이고, figure 3은 N(3)-N(0)의 분포를 히스토그램으로 나타낸 것입니다. 구간 [0,3]에서의 도착 수를 나타내는 N(3)-N(0)은 기댓값이 lambda(3-0)인 포아송 확률변수를 의미합니다. 표본함수의 분포 히스토그램과 비교했을 때, 거의 비슷한 분포를 가지는 것을 확인할 수 있습니다.



- MATLAB Code

clear; clf;

t=0:3;

alpha=1;

N=500;

result=zeros(1,N);

for i=1:N

w=brownian(alpha,t);

plot(t,brownian(alpha(1),t)), hold on;

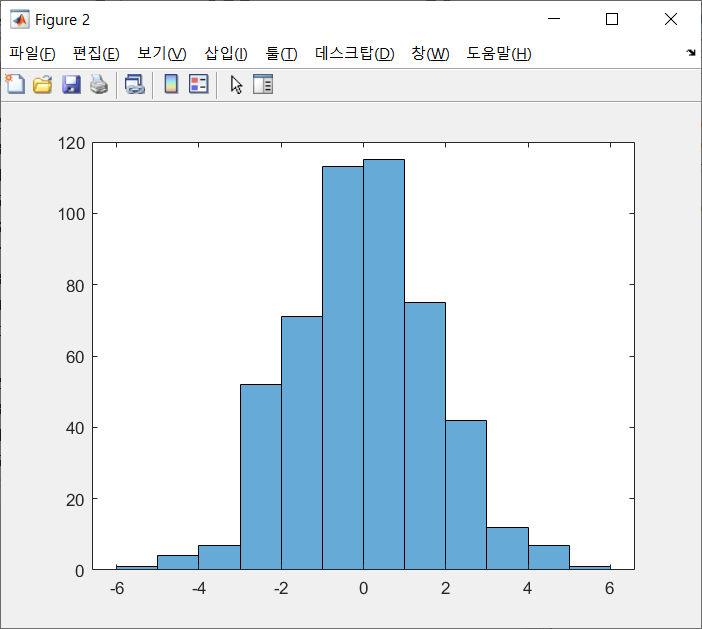
axis([0 3 -5 5]), grid on;

legend(['\alpha=', num2str(alpha)]);

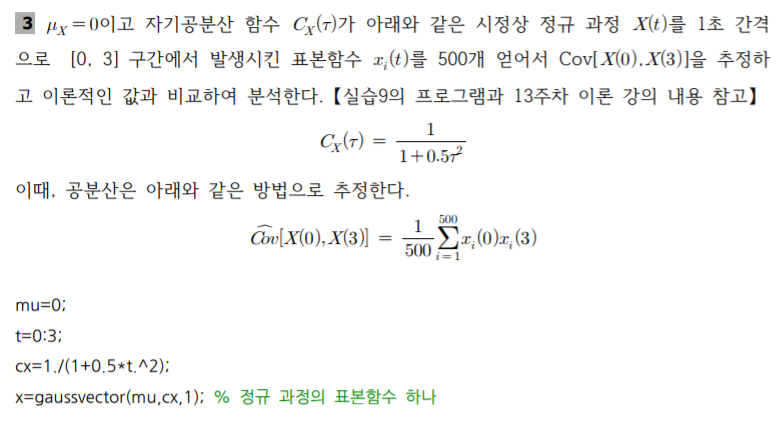
result(i)=w(4)-w(1);

end

>> figure(2), histogram(result)



=> 브라운 운동 과정 W(t)는 0보다 큰 시간구간의 길이 tau에서 W(t+tau)-W(t)는 가우시안 (0, sqrt(alpha\*tau)) 확률변수를 만족한다는 성질을 가지고 있습니다. 위의 주어진 브라운 운동과정은 alpha=1 이고 [0, 3] 구간에서 움직이므로 가우시안 (0, sqrt(3)) 확률변수를 나타낸다고 할 수 있습니다. 가우시안 (0, sqrt(3))은 평균이 0이고 표준편차가 sqrt(3) 인 분포를 나타내며, 위의 figure 2 에서 구한 히스토그램을 보면 중심이 0이고 0을 기준으로 편차가 거의 sqrt(3)이므로 이론 값과 실제 구한 히스토그램의 분포가 거의 비슷함을 알 수 있습니다.



- MATLAB Code

clear; clf;

t=0:3;

n=500;

mu=0;

tau=0:3;

result=zeros(1,n);

sum=zeros(1,n);

cx=1./(1+0.5\*t.^2);

x=gaussvector(mu,cx,n);

plot(t,x), hold on;

axis([0 tau(end) -4 4]), grid on;

for i=1:n %% 공분산 추정을 위한 for문

result(i) = x(1,i)\*x(4,i);

sum(i+1) = sum(i)+result(i);

end

%% function

function x=gaussvector(mu,C,m)

if (min(size(C))==1)

C=toeplitz(C);

end

n=size(C,2);

if (length(mu)==1)

mu=mu\*ones(n,1);

end

[U,D,V]=svd(C);

x=U\*(D^(0.5))\*randn(n,m)...

+(mu(:)\*ones(1,m));

end

=> 표본함수 x를 500개 발생시키고, 공분산을 추정하기 위해 작성한 코드입니다. x(0)과 x(3)에 해당하는 실제 배열 값인 x(1)과 x(4)의 곱을 n이 1부터 500까지 증가할 때마다 반복해서 더해준 결과, sum(501) = 94.3277 이었고 이 결과를 500으로 나눈 값이 공분산의 추정 값이 되는 것입니다. 결과는 0.1886이며, 시정상 정규과정의 공분산을 구하는 이론적인 방식은 cov[X(t1),X(t2)] = C\_x(t1-t2) 이고, 주어진 구간 [0,3] 내에서 공분산을 계산한 결과 약 0.1818 입니다. 추정값(0.1886)과 이론값(0.1818)을 비교해보면, 오차율이 3.74%으로 거의 비슷합니다.