

```
%Исходные данные
```

```
n=5;  
m=11;  
delta=0.5;  
Q=0.35;  
ro=8;
```

```
Mat=1.57;  
Sigm=0.5;
```

```
tmax=(m-1)*delta;  
for j=0:m-1  
    t(j+1)=j*delta;  
end
```

```
%реализация случайного процесса
```

```
P=normrnd(Mat, Sigm, [m 5]).*Q/ro
```

```
P = 11x5  
    0.0426    0.0523    0.0517    0.0578    0.0544  
    0.0828    0.0482    0.0756    0.0771    0.0791  
    0.0610    0.0409    0.0995    0.0777    0.0671  
    0.0697    0.0796    0.0775    0.0776    0.0482  
    0.0513    0.1297    0.0890    0.0607    0.0722  
    0.0348    0.0846    0.0336    0.0556    0.0628  
    0.0724    0.0518    0.0832    0.0558    0.0597  
    0.0673    0.0870    0.1155    0.0874    0.0531  
    0.0949    0.0440    0.0805    0.0282    0.0700  
    0.0862    0.0375    0.0350    0.0642    0.0283  
    ⋮
```

```
d=[];  
P=P';  
for i=1:m  
    d(i)=mean(P(:, i));  
end  
M=d
```

```
M = 1x11  
    0.0517    0.0726    0.0692    0.0705    0.0806    0.0543    0.0646    0.0821 ...
```

```
%График реализации случайного процесса
```

```
figure  
plot(t, P, '-r');  
title('График реализации случайного процесса');  
xlabel('t');  
ylabel({'P'; 'M'})
```

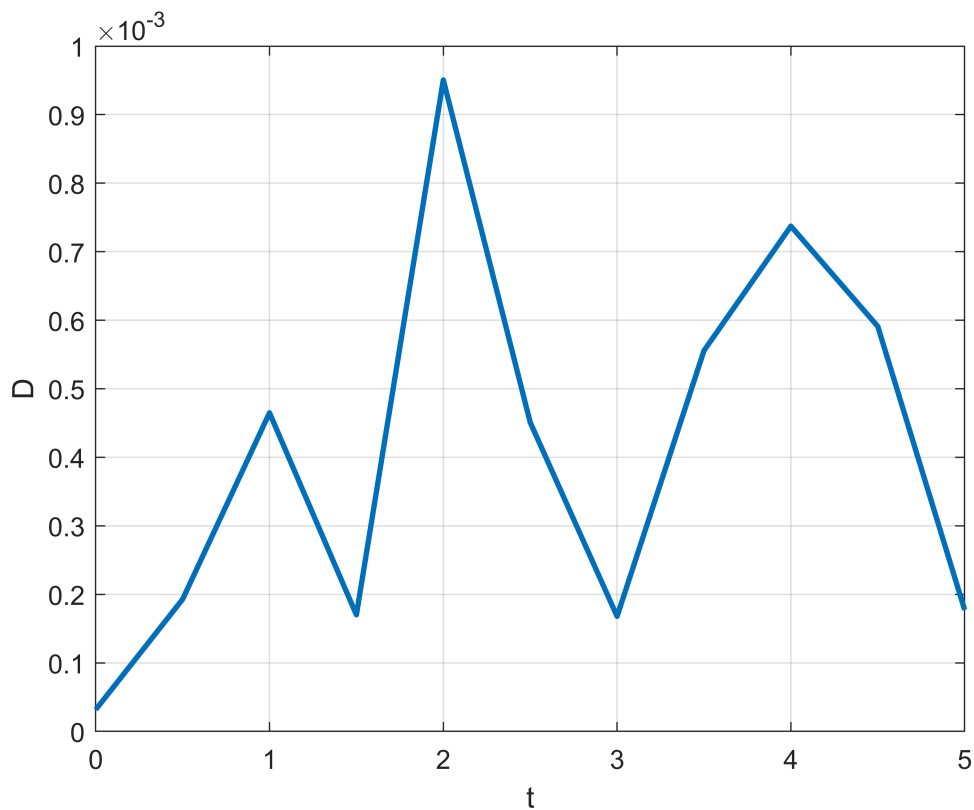
```
grid on  
hold on;
```

```
plot(t, M, '--b', 'LineWidth', 2);  
hold off;
```



%Вычисление дисперсии случайного процесса

```
d=[];
for i=1:m
    d(i)=var(P(:, i));
end
D=d';
plot(t, D, 'LineWidth', 2);
grid on
xlabel('t');
ylabel('D');
```



D'

```
ans = 1x11
10-3 x
0.0319    0.1929    0.4648    0.1701    0.9507    0.4510    0.1679    0.5556 ...
```

% Вычисление корреляционных моментов случайного процессов

```
n=0;
d=[];
for j=1:m
    for i=1:n+1
        d(i, j)=M(i)*M(j);
    end
    n=n+1;
end
```

Mx=d

```
Mx = 11x11
0.0027    0.0038    0.0036    0.0036    0.0042    0.0028    0.0033    0.0042 ...
0    0.0053    0.0050    0.0051    0.0058    0.0039    0.0047    0.0060
0    0    0.0048    0.0049    0.0056    0.0038    0.0045    0.0057
0    0    0    0.0050    0.0057    0.0038    0.0046    0.0058
0    0    0    0    0.0065    0.0044    0.0052    0.0066
0    0    0    0    0    0.0029    0.0035    0.0045
0    0    0    0    0    0    0.0042    0.0053
0    0    0    0    0    0    0    0.0067
0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0
```

⋮

```
n=5;
d=zeros(m, m);
for s=1:n
    d=d+KS(s, P, m);
end
ks=d;
Kx=(ks./n-Mx).*(n/(n-1))
```

```
Kx = 11×11
10-3 ×
    0.0319   -0.0159    0.0257   -0.0013    0.0383    0.0570   -0.0366    0.0225 ...
         0    0.1929    0.1801   -0.0808   -0.4039   -0.2350    0.0944   -0.0840
         0         0    0.4648    0.0167   -0.2609   -0.3325    0.1987    0.2745
         0         0         0    0.1701    0.1437   -0.0087    0.0133    0.2432
         0         0         0         0    0.9507    0.4469   -0.1217    0.2729
         0         0         0         0         0    0.4510   -0.2449   -0.1239
         0         0         0         0         0         0    0.1679    0.1339
         0         0         0         0         0         0         0    0.5556
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         ⋮
```

%Среднеквадратическое отклонение случайного процесса
Sigm=sqrt(D)

```
Sigm = 11×1
    0.0056
    0.0139
    0.0216
    0.0130
    0.0308
    0.0212
    0.0130
    0.0236
    0.0272
    0.0243
         ⋮
```

%Вычисление нормированной корреляционной функции

```
n=0;
d=[];
for j=1:m
    for i=1:n+1
        d(i, j)=Sigm(i)*Sigm(j);
    end
    n=n+1;
end
S=d;

n=0;
d=[];
```

```

for j=1:m
    for i=1:n+1
        d(i, j)=Kx(i,j)/S(i,j);
    end
    n=n+1;
end
Px=d

```

```

Px = 11x11
    1.0000   -0.2026    0.2116   -0.0178    0.2202    0.4752   -0.5001    0.1692 ...
         0    1.0000    0.6015   -0.4459   -0.9432   -0.7969    0.5245   -0.2566
         0         0    1.0000    0.0594   -0.3924   -0.7262    0.7110    0.5401
         0         0         0    1.0000    0.3574   -0.0316    0.0789    0.7911
         0         0         0         0    1.0000    0.6825   -0.3045    0.3755
         0         0         0         0         0    1.0000   -0.8897   -0.2475
         0         0         0         0         0         0    1.0000    0.4385
         0         0         0         0         0         0         0    1.0000
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         ⋮

```

%Осреднение математического ожидания, дисперсии и среднеквадратического отклонения

```
MN=mean(M)
```

```
MN = 0.0668
```

```
DN=mean(D)
```

```
DN = 4.0824e-04
```

```
SigmN=sqrt(sum(D))/m
```

```
SigmN = 0.0061
```

%Осреднение нормированной корреляционной функции по ансамблю реализаций

```

n=0;
d=[];
for j=1:m
    d(j)=0;
    s=1;
    for i=n+1:m
        d(j)=d(j)+Px(s,i);
        s=s+1;
    end
    Sr(j)=d(j)/s;
    n=n+1;
end
PxN=Sr

```

```

PxN = 1x11
    0.9167    0.1670   -0.0554   -0.2098   -0.0020    0.0878   -0.0452    0.2035 ...

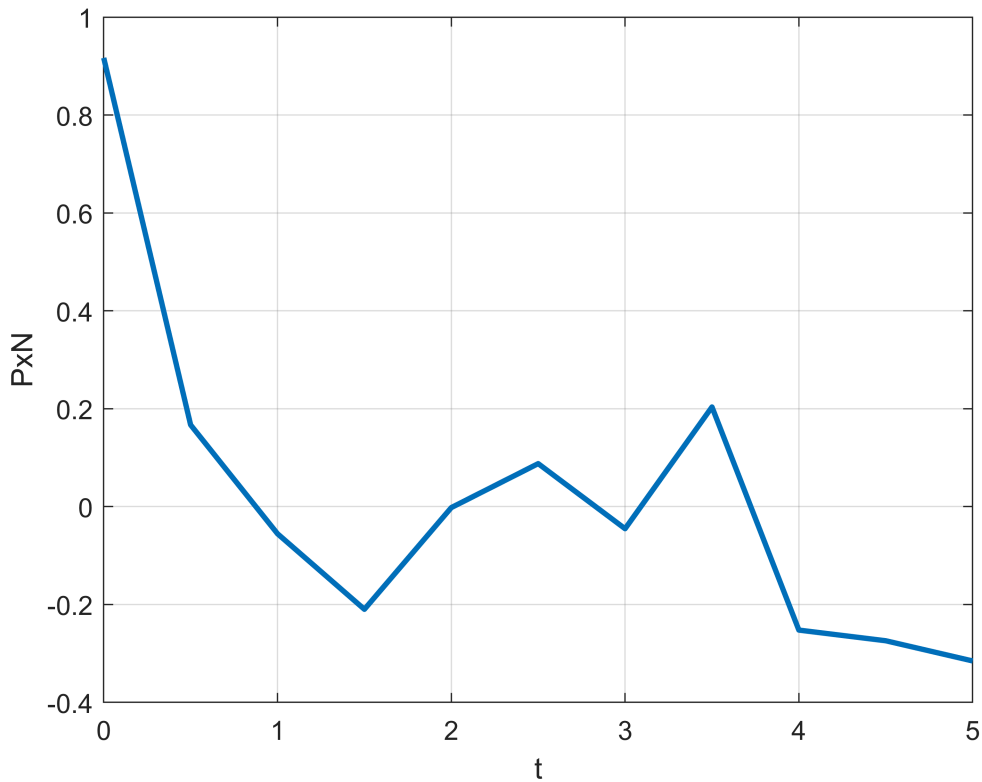
```

```

for i=0:m-1
    tau(i+1)=i*delta;
end

```

```
figure
plot(tau, PxN, 'LineWidth', 2);
grid on
xlabel('t');
ylabel('PxN');
```



```
%Вычисление нормированной спектральной плотности
%Добавление к исходной матрице нормированной корреляционной функции времени
%нулей
```

```
Q1=[0 0 0 0 0];
```

```
Q=[PxN Q1];
```

```
T=[t Q1];
```

```
%Число элементов новой матрицы
```

```
N=m+length(Q1);
```

```
%Выполнение быстрого преобразования Фурье
```

```
W=fft(Q);
```

```
W1=abs(W);
```

```
%Вычисление частот
```

```
i=0:length(W)-1;
```

```
Omega=(i+1).*(delta/m);
```

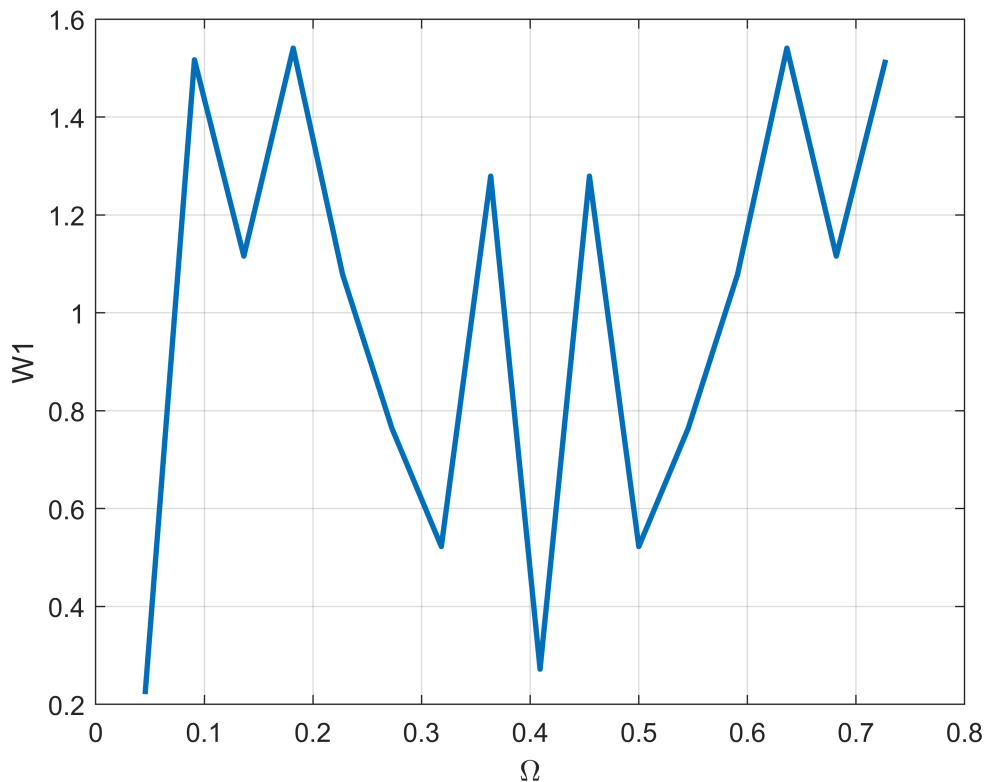
```
Omega0=delta/m
```

$\Omega_0 = 0.0455$

$\Omega_N = (N/2+1) \cdot \Delta / m$

$\Omega_N = 0.4091$

```
figure
plot(Ω, W1, 'LineWidth', 2);
grid on
xlabel('\Omega');
ylabel('W1');
```



```
function [d]=KS(s, P, m)
%Вспомогательная функция расчета KS при вычислении корреляционных мометнов случайного
%процесса
d=[];
P=P';
for k=1:s
    n=0;
    for j=1:m
        for i=1:n+1
            d(i,j)=P(i,k)*P(j,k);
        end
        n=n+1;
    end
end
```

end