

---

## Table of Contents

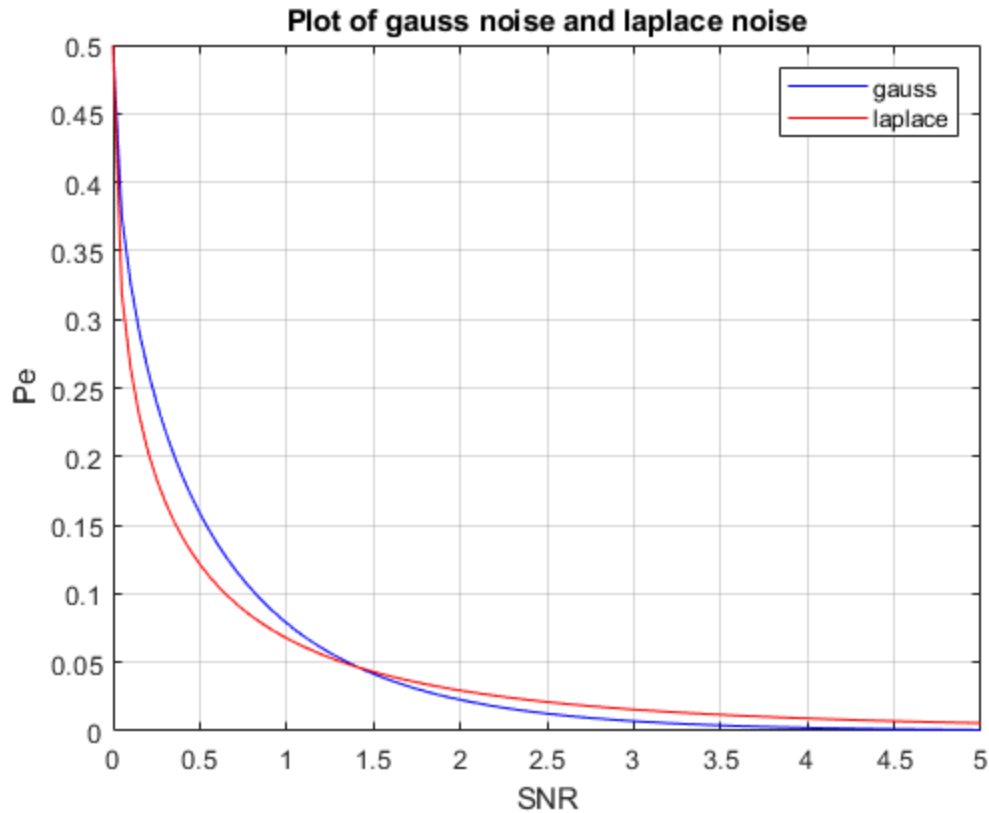
.....	1
Q1.1.5 .....	1
Q1.2.1-Q1.2.5 .....	2
Q1.2.6 .....	3
function laplace .....	3

```
disp('316389584_315871764');
```

```
316389584_315871764
```

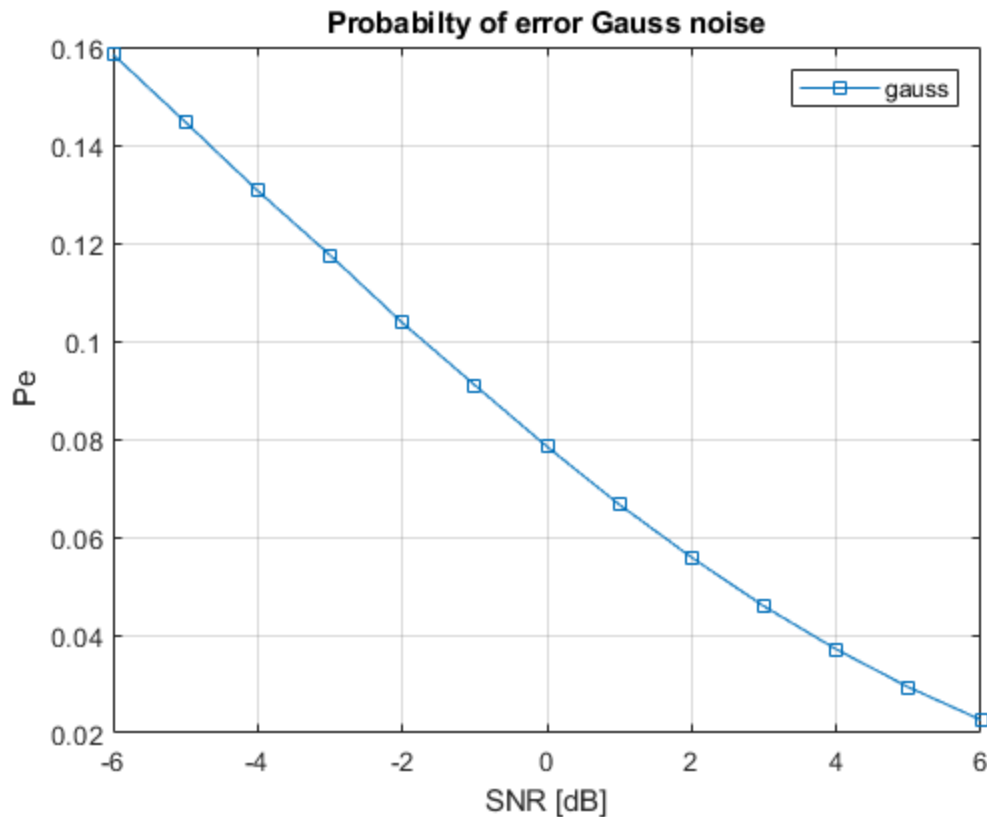
## Q1.1.5

```
SNR = linspace(0, 5, 100);  
gauss = qfunc(sqrt(2*SNR));  
laplace = 0.5*exp(-2*sqrt(SNR));  
plot(SNR, gauss, 'b');  
hold on;  
plot(SNR, laplace, 'r');  
xlabel('SNR');  
ylabel('Pe');  
legend('gauss', 'laplace');  
title('Plot of gauss noise and laplace noise');  
grid on;
```



## Q1.2.1-Q1.2.5

```
SNRDB = -6:6;
SNR = db2mag(SNRDB);
Es = 1;
symbols = randsrc(10^5,1,[-Es,Es]);
white_gauss = sqrt((Es./SNR)./2).*randn([10^5,1]);
r_gauss = symbols + white_gauss;
r_gauss(r_gauss>=0) = Es;
r_gauss(r_gauss<0) = -Es;
Pe_gauss = sum(r_gauss+symbols == 0)/10^5;
figure;
plot(SNRDB,Pe_gauss,'-s');
grid on;
legend('gauss')
xlabel("SNR [dB]"); ylabel("Pe");
title("Probabilty of error Gauss noise");
```



## Q1.2.6

```

symbols_laplace = randsrc(10^5,1,[-Es,Es]);
white_laplace = laprnd(10^5, 1, 0, sqrt((Es./SNR)./2));
r_laplace = symbols_laplace + white_laplace;
r_laplace(r_laplace>=0) = Es;
r_laplace(r_laplace<0) = -Es;
Pe_laplace = sum(r_laplace+symbols_laplace == 0)/10^5;
figure;
plot(SNRDB,Pe_laplace,'-o');
grid on;
xlabel("SNR [dB]"); ylabel("Pe");
title("Probabilty of error laplace noise");
legend('laplace');
figure;
plot(SNRDB,Pe_laplace,'-o');
grid on;
hold on;
plot(SNRDB,Pe_gauss,'-s');
title("laplace vs gauss");
legend('laplace', 'gauss');

```

## function laplace

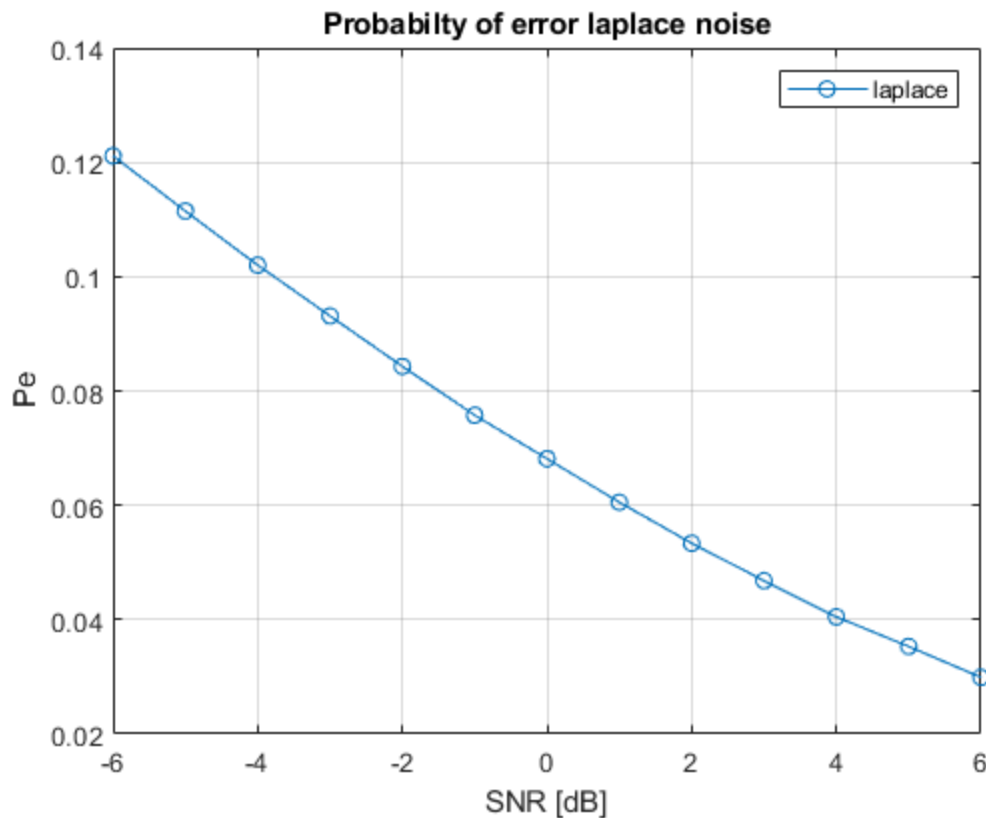
```
function y = laprnd(m, n, mu, sigma)
```

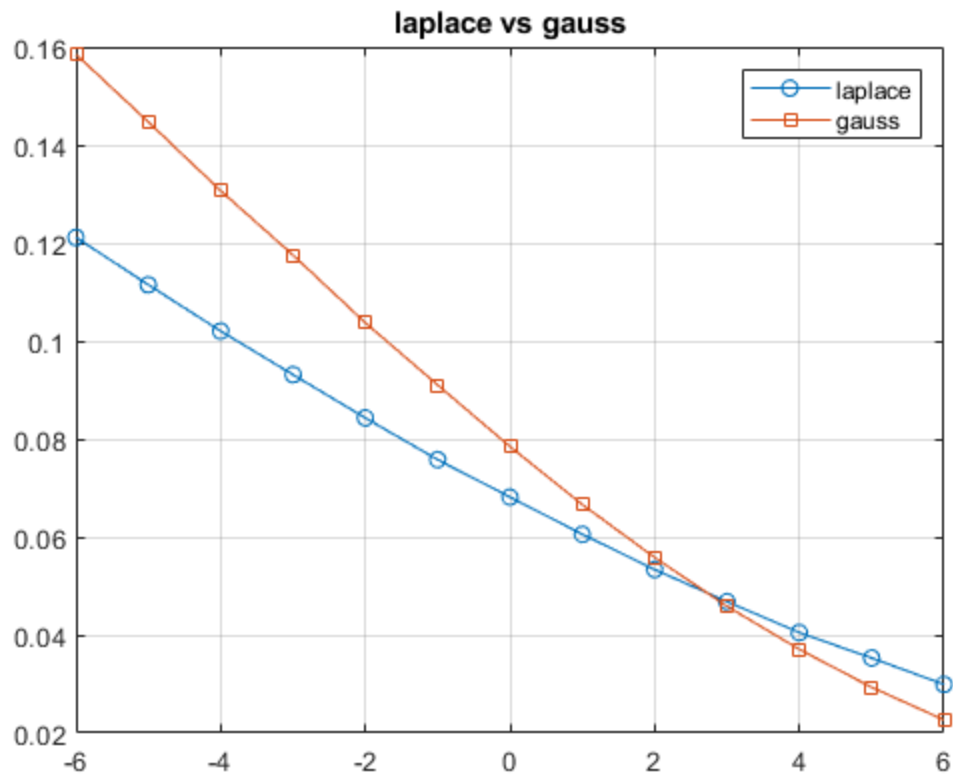
---

```

%LAPRND generate i.i.d. laplacian random number drawn from laplacian
distribution
% with mean mu and standard deviation sigma.
% mu : mean
% sigma : standard deviation
% [m, n] : the dimension of y.
% Default mu = 0, sigma = 1.
% For more information, refer to
% http://en.wikipedia.org/wiki/Laplace\_distribution
% Author : Elvis Chen (bee33@sjtu.edu.cn)
% Date : 01/19/07
%Check inputs
if nargin < 2
    error('At least two inputs are required');
end
if nargin == 2
    mu = 0; sigma = 1;
end
if nargin == 3
    sigma = 1;
end
% Generate Laplacian noise
u = rand(m, n)-0.5;
b = sigma ./ sqrt(2);
y = mu - b .* sign(u).* log(1- 2* abs(u));
end

```





*Published with MATLAB® R2023a*