**CODE:**

```
clc;
clear all;
close all;
N = 10^6; % number of bits or symbols
rand('state',100); % initializing the rand() function
randn('state',200); % initializing the randn() function
% Transmitter
ip = rand(1,N)>0.5; % generating 0,1 with equal probability
s = 2*ip-1; % BPSK modulation 0 -> -1; 1 -> 1
n = 1/sqrt(2)*[randn(1,N) + 1i*randn(1,N)]; % white gaussian noise, 0dB variance
Eb_N0_dB = [-3:10]; % multiple Eb/N0 values
nErr = zeros(1,length(Eb_N0_dB)); % initialize error count
for ii = 1:length(Eb_N0_dB)
   % Noise addition
   y = s + 10^(-Eb_N0_dB(ii)/20)*n; % additive white gaussian noise
   % receiver - hard decision decoding
   ipHat = real(y)>0;
   % counting the errors
   nErr(ii) = sum(ip ~= ipHat); % count errors
end
simBer = nErr/N; % simulated ber
theoryBer = 0.5*erfc(sqrt(10.^(Eb_N0_dB/10))); % theoretical ber
% plot
close all
figure
semilogy(Eb_N0_dB,theoryBer,'b.-');
hold on
semilogy(Eb_N0_dB,simBer,'mx-');
axis([-3 10 10^-5 0.5])
grid on
```

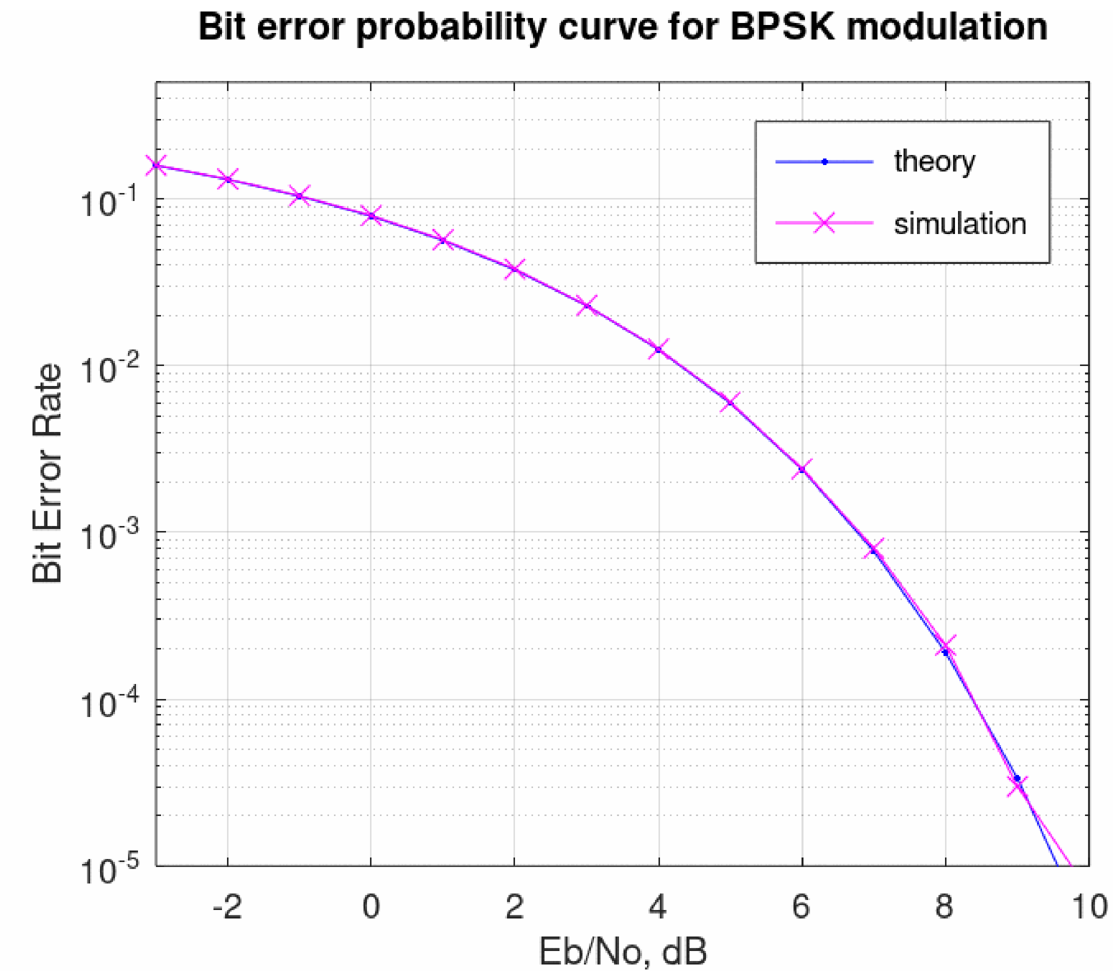
```
legend('theory', 'simulation');

xlabel('Eb/No, dB');

ylabel('Bit Error Rate');

title('Bit error probability curve for BPSK modulation');
```

**OUTPUT:**


Bit error probability curve for BPSK modulation
theory
simulation
Bit Error Rate
10^-1
10^-2
10^-3
10^-4
10^-5
Eb/No, dB