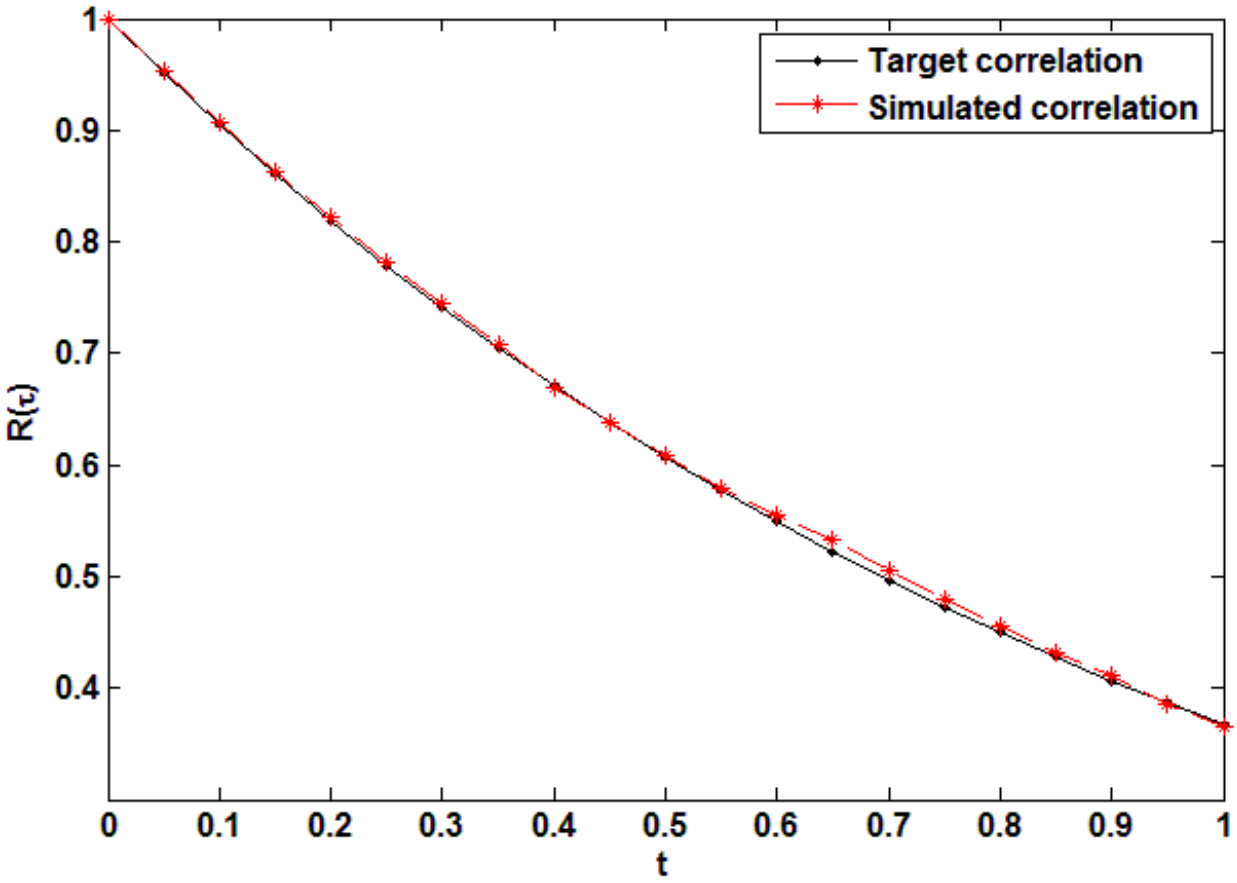
```
%% THIS PROGRAM WILL SIMULATE CORRELATED STANDARD NORMAL FIELD USING K-L
%% EXPANSION METHOD (HOME ASSIGNMENT 1)
% SUBMITTED BY: Rudraprasad Bhattacharyya
%-----------------------------------------
clc; clear all;close all;
% Discretization of "t" domain
start_point=0;
end_point=1;
N_interval=20;          % No. of intervals
interval_size=(end_point-start_point)/N_interval;
N_points=N_interval+1;
t=linspace(start_point,end_point,N_points);
N_simulations=10000;                % Total number of experiments
%% Construction of target Auto-correlation matrix R
for i=1:1:N_points
    for j=1:1:N_points
        tau=(j-i)*interval_size;
        R(i,j)=exp(-abs(tau));
    end
end
R1=R(:,1);
%% Simulation of correlated random field
% EVALUATION OF EIGENVALUE AND EIGEN FUNCTIONS
[f,lambda]=eig(R);
for simulation=1:N_simulations

        for i=1:N_points
               g(:,i)=sqrt(lambda(i,i)).*f(:,i)*randn;
        end

    G_storage(:,simulation)=sum(g,2);
end
%% Check:Estimation of simulated data
% Calculation of correlation coefficient
G_storage=G_storage';
for i=1:N_points
    g_1=G_storage(:,i);
    for j=1:N_points
        g_2=G_storage(:,j);
          Expectation_g_1g_2=(g_1'*g_2)/N_simulations;
          Expectation_g1=mean(g_1);
          Expectation_g2=mean(g_2);
          std_dev_g1=std(g_1);
          std_dev_g2=std(g_2);
          computed_correlation=abs((Expectation_g_1g_2-(Expectation_g1)*(Expectation_g2))/
(std_dev_g1*std_dev_g2));
          computed_correlation_matrix(i,j)=computed_correlation;
    end
end
C1=computed_correlation_matrix(:,1);
% Plot the correlation
figure;
plot(t,R1,'k.-',t,C1,'r*-');xlabel('t');ylabel('R(\tau)');
legend('Target correlation','Simulated correlation');
```


Target correlation
Simulated correlation
R(τ)
t
1
0.9
0.8
0.7
0.6
0.5
0.4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1