```
%%This programm compares uniform, trapez and Simpson-Sceme for %%
%%approximation of solutions to the Fredholm integral equation.%%
%%written by Tim Jaschek as a part of his bachelor thesis%%

%%Used to generate data for Tabular 6.1 and 6.2 %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Import the class Kernels which contains some Kernels and
%integration scemes.
Kernels;

%Parameter for the Number of approximation steps
N = 45;

%Generation of different Kernels
BrownianMotion = Kernels.KMat(1,N);
BrownianBridge = Kernels.KMat(2,N);
ExponentialKer = Kernels.KMat(3,N);

%BROWNIAN MOTION
%Solve Fredhol integralequality with different Scemes
[lambda1,Phi1] = Kernels.uniform_Sceme(BrownianMotion);
[lambda2,Phi2] = Kernels.trapez_Sceme(BrownianMotion);
[lambda3,Phi3] = Kernels.simpson_Sceme(BrownianMotion);
%Compute analytic solutions for first Eigenvalues
lambda = [lambda1(1) lambda2(1) lambda3(1)];
Phi = [Phi1(:,1) Phi2(:,1) Phi3(:,1)];
la = (2/pi)^2;
ph = zeros(N+2,1);
for i=1:N+2
   ph(i) = sqrt(2)* sin(0.5*pi*((i-1)/(N+2)));
end
%plot(linspace(0,1,N+2),ph,linspace(0,1,N+2),Phi(:,1))
%Compute the error terms
absolute_error_lambda = abs(la(1)-lambda)
relative_error_lambda = abs(la(1)-lambda)/la(1)*100
absolute_error_phi = zeros(1,3);
relative_error_phi = zeros(1,3);
for i=1:3
     absolute_error_phi(i) = max(abs(ph-Phi(:,i)));
     relative_error_phi(i) = absolute_error_phi(i)/max(abs(Phi(:,↙
i)));
end
absolute_error_phi
```

```
44 relative_error_phi*100
45
46 %BROWNIAN BRIDGE
47 %Solve Fredhol integralequality with different Scemes
48 [lambda1,Phi1] = Kernels.uniform_Sceme(BrownianBridge);
49 [lambda2,Phi2] = Kernels.trapez_Sceme(BrownianBridge);
50 [lambda3,Phi3] = Kernels.simpson_Sceme(BrownianBridge);
51 %Compute analytic solutions for first Eigenvalues
52 lambda = [lambda1(1) lambda2(1) lambda3(1)];
53 Phi = [Phi1(:,1) Phi2(:,1) Phi3(:,1)];
54 la = (1/pi)^2;
55 ph = zeros(N+2,1);
56 for i=1:N+2
57    ph(i) = sqrt(2)* sin(pi*((i-1)/(N+2)));
58 end
59 plot(linspace(0,1,N+2),ph,linspace(0,1,N+2),Phi(:,1))
60 %Compute the error terms
61 absolute_error_lambda = abs(la(1)-lambda)
62 relative_error_lambda = abs(la(1)-lambda)/la(1)*100
63 absolute_error_phi = zeros(1,3);
64 relative_error_phi = zeros(1,3);
65 for i=1:3
66      absolute_error_phi(i) = max(abs(ph-Phi(:,i)));
67      relative_error_phi(i) = absolute_error_phi(i)/max(abs(Phi(:,↙
i)));
68 end
69 absolute_error_phi
70 relative_error_phi*100
71
72
73
74
```