

```

1 %%This class is a collection of functions to expand a Brownian✓
motion wrt.
2 %%to different orthonormal bases and compute the Total Mean✓
Squared Error
3 %%for the approximation with the n-th partial sum.
4 %%written by Tim Jaschek as a part of his bachelor thesis%%
5
6 %%Used to generate FIGURE 3 and the data for TABULAR 5.1 in this✓
thesis %%
7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8
9 classdef MSE
10     properties (Constant)
11         %Parameter for the partial sum
12         n = 20;
13
14         %Parameter for the accuracy of the Kernels
15         N = 1000;
16     end
17     methods (Static)
18         function approximation()
19             %load the classes BrMo and Kernels
20             BrMo;
21             Kernels;
22             %%%generate a Brownian Motion with N steps%%
23             p = 10000; %parameter for the accuracy of the BM
24             X = BrMo.BrownianMotion(p,MSE.N);
25             %%%get ORTHONORMAL BASES
26             %%%the analytic KLT Eigenfunctions
27             %%%the HaarWavelets
28             %%%Eigenfunctions of Brownian Bridge
29             %%%Eigenfunctions of Exponential Kernel
30             [Z,lambda,psi] = BrMo.compute_KLT_components(MSE.N);
31             psi = psi.';
32             Haar = BrMo.Haar(1000);
33             [lambda,Bridge] = Kernels.trapez_Sceme(Kernels.KMat(2,✓
MSE.N));
34             [lambda,Exp] = Kernels.trapez_Sceme(Kernels.KMat(3,MSE.✓
N));
35             %%%Compute the Approximations%%
36             X_hat = MSE.Approx(X,psi(:,1:MSE.n));
37             X_tilde = MSE.Approx(X,Haar(:,1:MSE.n));
38             X_bridge = MSE.Approx(X,Bridge(:,1:MSE.n));
39             X_exp = MSE.Approx(X,Exp(:,1:MSE.n));

```

```

40         %%%plot both%%%
41         figure
42         hold on;
43         plot(linspace(1/MSE.N,1,MSE.N),X)
44         plot(linspace(1/MSE.N,1,MSE.N),X_hat)
45         plot(linspace(1/MSE.N,1,MSE.N),X_tilde)
46         plot(linspace(1/MSE.N,1,MSE.N),X_bridge)
47         %plot(linspace(1/MSE.N,1,MSE.N),X_exp)
48         hold off;
49     end
50     function meansquare()
51         %load the classes BrMo and Kernels
52         BrMo;
53         Kernels;
54
55         disp('Compute TMSE of Karhunen-Loeve-Base...')
56         [Z,lambda,psi] = BrMo.compute_KLT_components(MSE.N);
57         psi = psi.';
58         Mse = zeros(1,MSE.N);
59         for i=1:30
60             X = BrMo.BrownianMotion(5000,MSE.N);
61             X_hat = MSE.Approx(X,psi(:,1:MSE.n));
62             for j=1:MSE.N
63                 Mse(j) = Mse(j) + (X(j)-X_hat(j))^2;
64             end
65         end
66         Mse = Mse/30;
67         TMse = 0;
68         h=1/MSE.N;
69         for j=1:MSE.N-1
70             TMse = TMse + h/2 * (Mse(j) + Mse(j+1));
71         end
72         TMse
73         %%%%%%%%%Haar%%%%%%%%
74         disp('Compute TMSE of Haar-Base...')
75         Haar = BrMo.Haar(1000);
76         Mse = zeros(1,MSE.N);
77         for i=1:30
78             X = BrMo.BrownianMotion(5000,MSE.N);
79             X_hat = MSE.Approx(X,Haar(:,1:MSE.n));
80             for j=1:MSE.N
81                 Mse(j) = Mse(j) + (X(j)-X_hat(j))^2;
82             end
83         end

```

```

84         Mse = Mse/30;
85         TMse = 0;
86         h=1/MSE.N;
87         for j=1:MSE.N-1
88             TMse = TMse + h/2 * (Mse(j) + Mse(j+1));
89         end
90         TMse
91         %%%%%%%%%Bridge%%%%%%%%
92         disp('Compute TMSE of Brownian-Bridge-Base...')
93         [lambda,Bridge] = Kernels.trapez_Sceme(Kernels.KMat(2,MSE.N));
94         Mse = zeros(1,MSE.N);
95         for i=1:30
96             X = BrMo.BrownianMotion(5000,MSE.N);
97             X_hat = MSE.Approx(X,Bridge(:,1:MSE.n));
98             for j=1:MSE.N
99                 Mse(j) = Mse(j) + (X(j)-X_hat(j))^2;
100            end
101        end
102        Mse = Mse/30;
103        TMse = 0;
104        h=1/MSE.N;
105        for j=1:MSE.N-1
106            TMse = TMse + h/2 * (Mse(j) + Mse(j+1));
107        end
108        TMse
109        %%%%%%%%%Exponential%%%%%%%%
110        disp('Compute TMSE of Exponential-Base...')
111        [lambda,Exp] = Kernels.trapez_Sceme(Kernels.KMat(3,MSE.N));
112        Mse = zeros(1,MSE.N);
113        for i=1:30
114            X = BrMo.BrownianMotion(5000,MSE.N);
115            X_hat = MSE.Approx(X,Exp(:,1:MSE.n));
116            for j=1:MSE.N
117                Mse(j) = Mse(j) + (X(j)-X_hat(j))^2;
118            end
119        end
120        Mse = Mse/30;
121        TMse = 0;
122        h=1/MSE.N;
123        for j=1:MSE.N-1
124            TMse = TMse + h/2 * (Mse(j) + Mse(j+1));
125        end

```

```
126         TMse
127     end
128     function X_hat = Approx(X,Phi);
129         %%assume X has N steps on [0,1] and we have at least n✓
130         Eigenvectors with N steps each%%
131         X_hat = zeros(1,MSE.N);
132         A = zeros(1,MSE.n);
133         h = 1/MSE.N;
134         for i=1:MSE.n
135             %%compute the integrals via trapez sceme%%
136             for j=1:MSE.N-1
137                 A(i) = A(i) + h/2 * (X(j)*Phi(j,i) + X(j+1)✓
138                 *Phi(j+1,i));
139             end
140         end
141         X_hat(i) = dot(A,Phi(i,1:MSE.n));
142     end
143 end
144 end
145
146
147
148
149
```