Problem 6

(a) To generate Brownian motion paths in $[0, 1]$ using the Karhunen-Loeve expansion, first define the following function.

Listing 1: BrownianMotionKL.m

```
function W_t = BrownianMotionKL(t, num_terms)
% Return a vector with the values of a BM path sampled
% at the elements of
% t, using the Karhunen-Loeve expansion
% t - Row vector of sample points
% num_terms - Number of terms in the Karhunen-Loeve expansion

W_t = zeros(1, length(t));
Z = randn(1, num_terms);

% It is more convenient to start the sum from k = 1.
for k = 1:num_terms
       coeff = sqrt(2) * Z(k) / ((k - 0.5) * pi);
       term = coeff * sin((k - 0.5) * pi * t);
       W_t = W_t + term;
end

end
```

and then execute the following script.

Listing 2: PlotBrownianMotionPaths.m

```
figure;
hold on;
title('Brownian motion paths in [0, 1], J = 5000');
xlabel('t')
ylabel('W(t)')

t = linspace(0, 1, 2000);

% plot 10 brownian motion paths
for k = 1:10
    y = BrownianMotionKL(t, 5000);
    plot(t, y);
end

plot([0 1], [0 0], 'color', 'black');

ylim([-2.5 2.5]);

hold off;
```

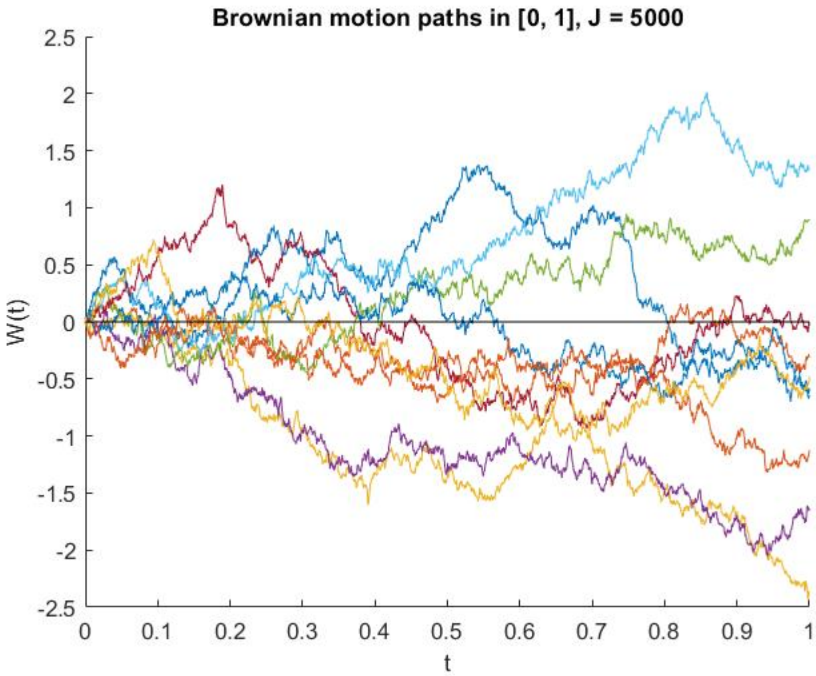

Figure 1: Sample script output.

(c) To plot $e_{M,J}(T)$, use the following code.

Listing 3: WJError.m

```
function [e_t] = WJError(t, M, J)

e_t = zeros(1, length(t));
wjs = zeros(M, length(t));

for k = 1:M
    wjs(k, :) = BrownianMotionKL(t, J);
end

wjs = wjs .^ 2;
column_sums = sum(wjs);

e_t = abs( (1 / M) * column_sums - t);

end
```

Listing 4: PlotWJError.m

```
J = floor(logspace(0, 3, 15)) + 1;
t = linspace(0, 1, 100);

figure;
hold on;
```

```
for ii = 1:length(J)
    y = WJError(t, 4000, J(ii));
    plot(t, y);
end

hold off;
```

This ouputs the following.