NOTE: All files all source files are in the .zip file turned in on learn

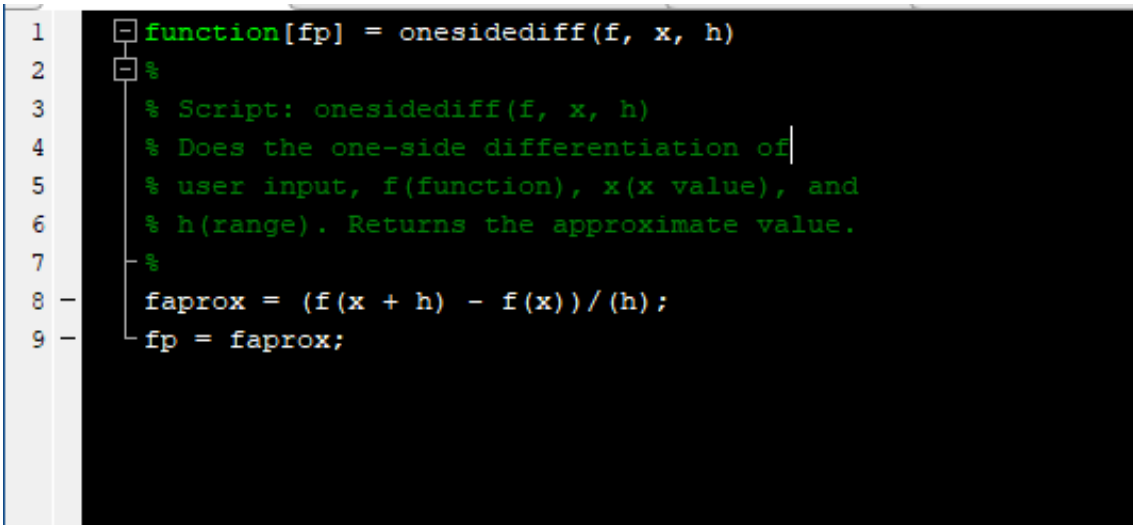
Damian Franco

dfranco24@unm.edu

CS 375 – 001

Homework 1

1) For question 1 on the homework we had to create a one-side differentiation function to approximate the derivative of a given function (f(x)). In this problem, the main goal is to see how the exact derivative of the function e^sin(x) differs from the one-side differentiation. Below, you will find the code of my function called onsidediff(f,x,h). You will also find how the exact derivative differs from the approximate derivative in the MATLAB generated graphs. I notice that the error is very noticeable on the one-side approximate.

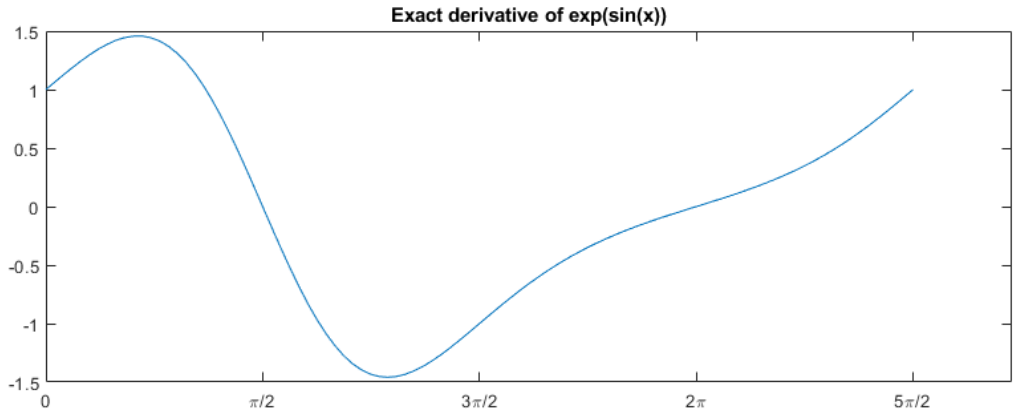


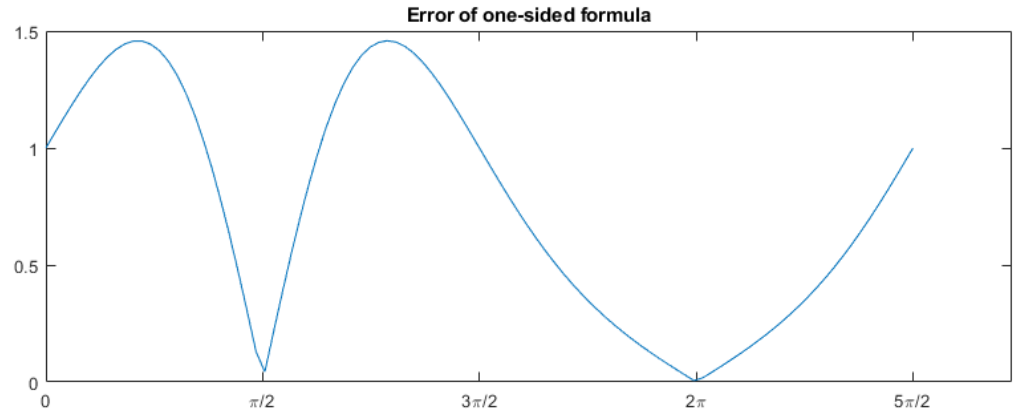
Source code (the onesidediff.m file is found in the zip file)

```
1    function[fp] = onesidediff(f, x, h)
2    %
3    % Script: onesidediff(f, x, h)
4    % Does the one-side differentiation of
5    % user input, f(function), x(x value), and
6    % h(range). Returns the approximate value.
7    %
8    faprox = (f(x + h) - f(x))/(h);
9    fp = faprox;
```

NOTE: All files all source files are in the .zip file turned in on learn

Graphs shown as directed in the problem (Prob1.png):



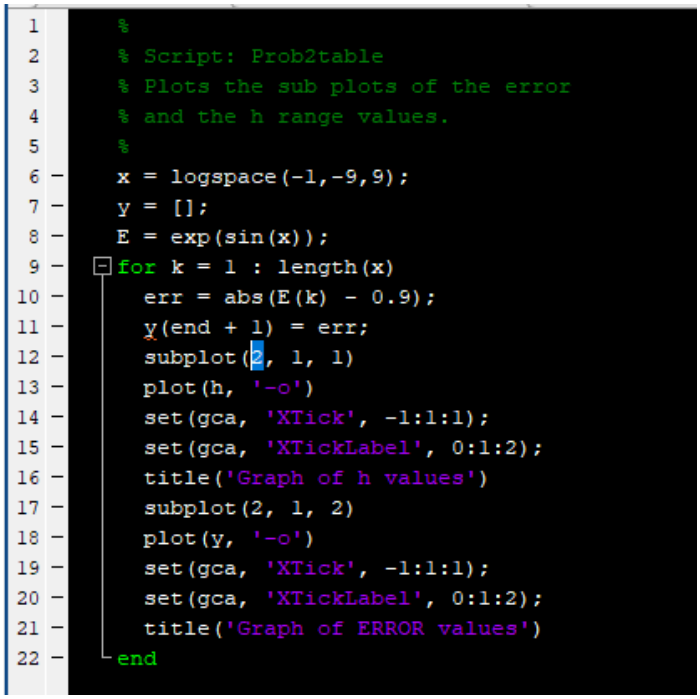


NOTE: All files all source files are in the .zip file turned in on learn

2) For problem 2, we had to use the onesidediff function to approximate the derivative of e^sin(pi). Then we had to make a table of the approximation value within the range 10^-1 through 10^-9. The approximation will be compared to the range (h) on two separate graphs. When looking at the graph, I observed that they both follow the same trend, but they vary differently when it comes to values. The errors seem to be 0.1 above what the values in the range are. Below is the source code, the table made, and the graphs generated in MATLAB.

Source code for making the table: (Prob2Table.m)

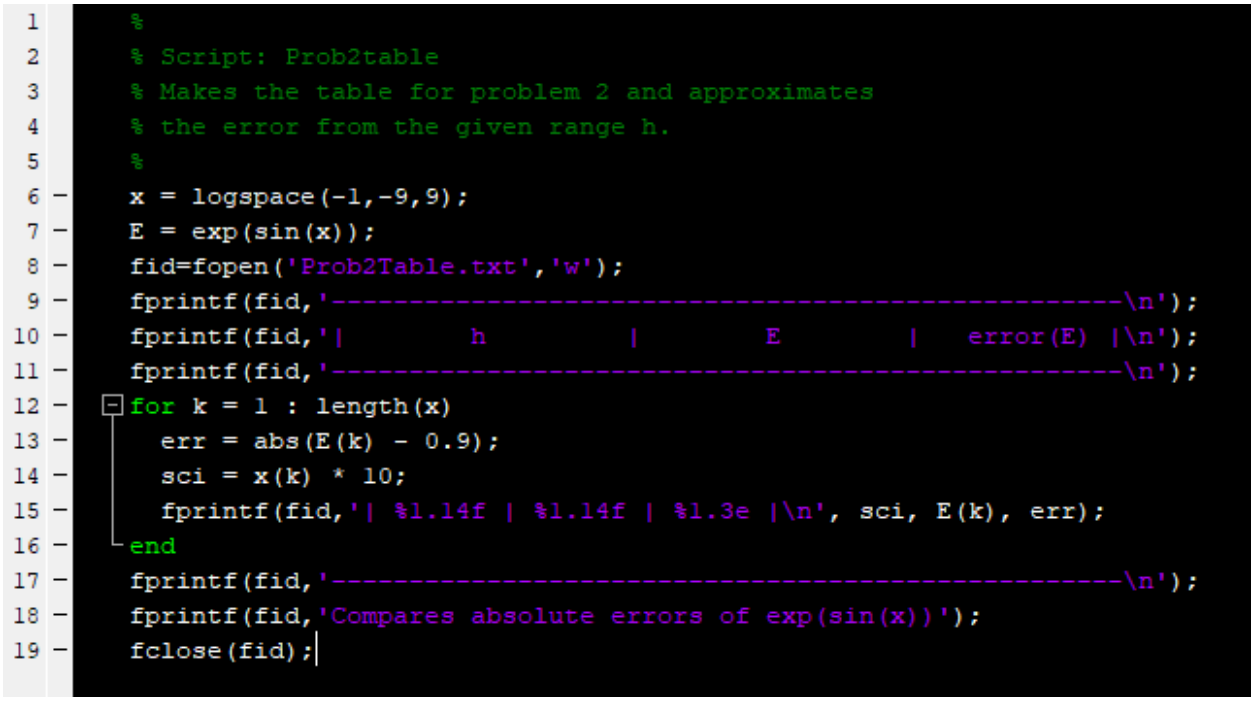
```
1     %
2     % Script: Prob2table
3     % Plots the sub plots of the error
4     % and the h range values.
5     %
6     x = logspace(-1,-9,9);
7     y = [];
8     E = exp(sin(x));
9     for k = 1 : length(x)
10      err = abs(E(k) - 0.9);
11      y(end + 1) = err;
12      subplot(2, 1, 1)
13      plot(h, '-o')
14      set(gca, 'XTick', -1:1:1);
15      set(gca, 'XTickLabel', 0:1:2);
16      title('Graph of h values')
17      subplot(2, 1, 2)
18      plot(y, '-o')
19      set(gca, 'XTick', -1:1:1);
20      set(gca, 'XTickLabel', 0:1:2);
21      title('Graph of ERROR values')
22    end
```



NOTE: All files all source files are in the .zip file turned in on learn

Source code for making the table: (Prob2Plot.m)

```
%
% Script: Prob2table
% Makes the table for problem 2 and approximates
% the error from the given range h.
%
x = logspace(-1,-9,9);
E = exp(sin(x));
fid=fopen('Prob2Table.txt','w');
fprintf(fid,'-----------------------------------------------------\n');
fprintf(fid,'|        h         |        E        |   error(E) |\n');
fprintf(fid,'-----------------------------------------------------\n');
for k = 1 : length(x)
  err = abs(E(k) - 0.9);
  sci = x(k) * 10;
  fprintf(fid,'| %1.14f | %1.14f | %1.3e |\n', sci, E(k), err);
end
fprintf(fid,'-----------------------------------------------------\n');
fprintf(fid,'Compares absolute errors of exp(sin(x))');
fclose(fid);
```



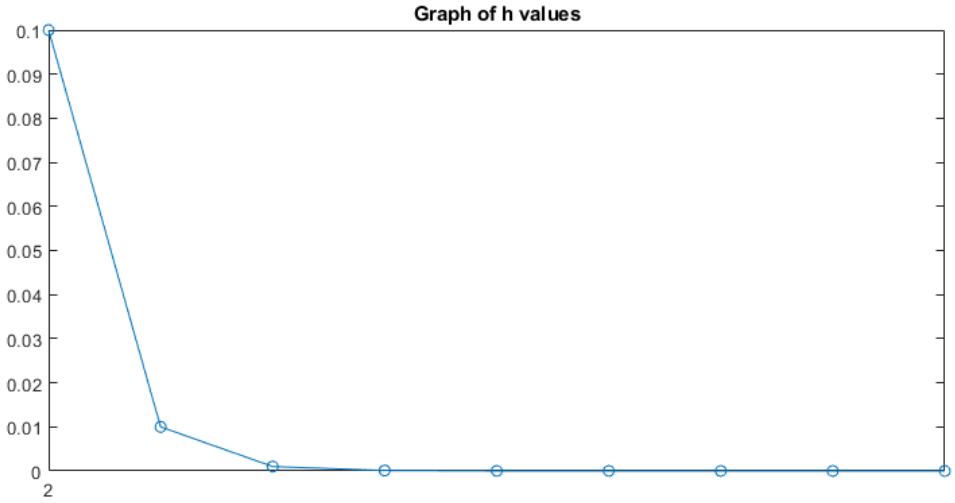
NOTE: All files all source files are in the .zip file turned in on learn

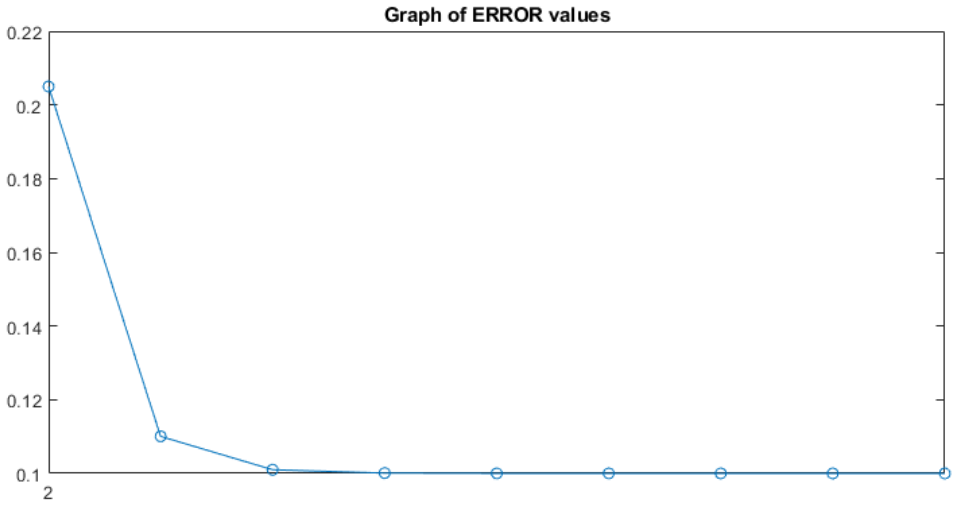
Table showing the errors compared to h. (Prob2Table.txt)

h	E	error(E)
1 * 10^-1	1.10498683033169	2.050e-01
1 * 10^-2	1.01004999874333	1.100e-01
1 * 10^-3	1.00100049999987	1.010e-01
1 * 10^-4	1.00010000500000	1.001e-01
1 * 10^-5	1.00001000005000	1.000e-01
1 * 10^-6	1.00000100000050	1.000e-01
1 * 10^-7	1.00000010000000	1.000e-01
1 * 10^-8	1.00000001000000	1.000e-01
1 * 10^-9	1.00000000100000	1.000e-01

Compares absolute errors of exp(sin(x)) to h

Graphs shown as directed in the problem (Prob2.png):



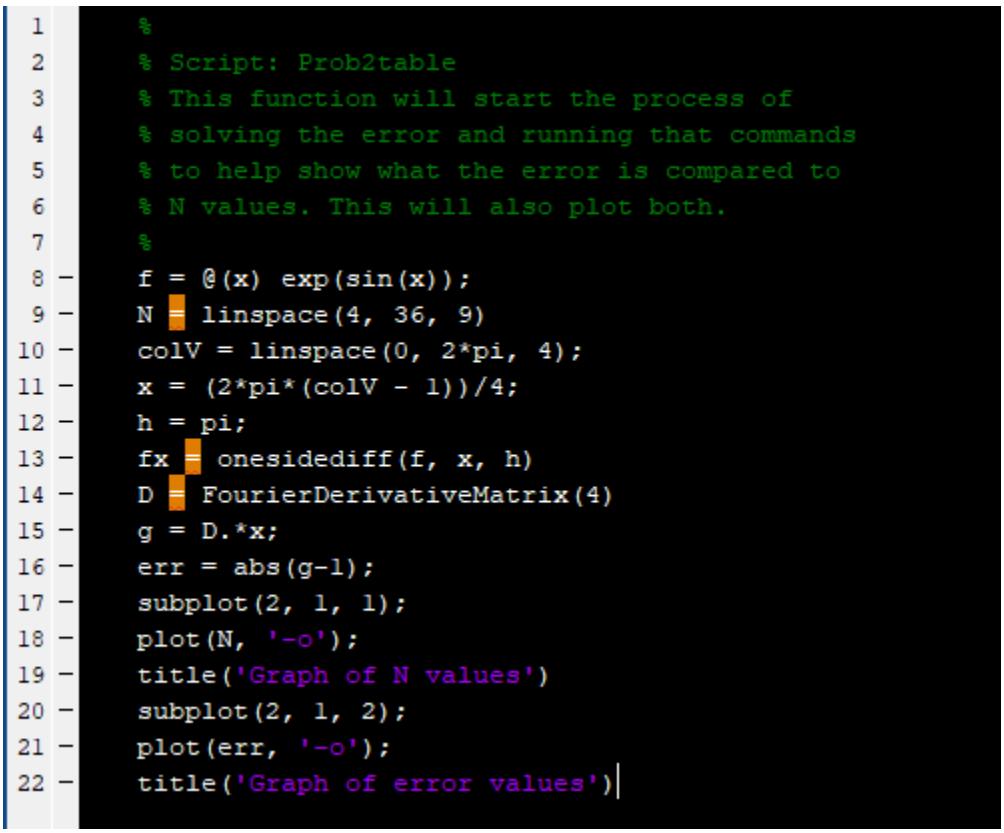


NOTE: All files all source files are in the .zip file turned in on learn

3) Problem 3 was the hardest on the homework and for this problem we must approximate the errors in brand new way of calculating a derivative. The function is called the FourierDerivativeMatrix.m and we were assigned to use that to calculate exp(sin(x)). I observed when running a test for all N's that the errors will range from high to low and as the N increase so will the error size.

Source code (Prob3Script.m):

```
%
% Script: Prob2table
% This function will start the process of
% solving the error and running that commands
% to help show what the error is compared to
% N values. This will also plot both.
%
f = @(x) exp(sin(x));
N = linspace(4, 36, 9)
colV = linspace(0, 2*pi, 4);
x = (2*pi*(colV - 1))/4;
h = pi;
fx = onesidediff(f, x, h)
D = FourierDerivativeMatrix(4)
g = D.*x;
err = abs(g-1);
subplot(2, 1, 1);
plot(N, '-o');
title('Graph of N values')
subplot(2, 1, 2);
plot(err, '-o');
title('Graph of error values')
```



NOTE: All files all source files are in the .zip file turned in on learn

Graphs shown as directed in the problem (Prob3.png):