Math 104A - Homework 4

Due 7/14

1 Write a function that takes as input an interval $[a, b]$, a number subintervals N, and a function f, and outputs the approximate derivative of f at the nodes $x_i = a + i \cdot h$, for $i = 0, \dots, N$, where $h = \frac{b-a}{N}$. Use the three point formulas:

$$f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_1) - f(x_2)], \tag{1}$$

$$f'(x_i) = \frac{1}{2h}[-f(x_{i-1}) + f(x_{i+1})], \quad \text{for} \quad i = 1, \dots, N-1, \tag{2}$$

$$f'(x_N) = \frac{1}{2h}[f(x_{N-2}) - 4f(x_{N-1}) + 3f(x_N)]. \tag{3}$$

An example code template in MatLab is:

```
function yp = approx_deriv(a,b,N,fstring)

    f = inline(fstring);
    x = linspace(a,b,N+1);
    y = f(x);

    % code for approximating the derivative goes here

end
```

2 Write another function that plots the results from problem 1. It should take as input the interval $[a, b]$, the number of subintervals N, the function f, and the derivative f'. The function should plot both the exact derivative and the approximate derivative at the nodes $x_i = a + i \cdot h$. An example code template is:

```
function [] = plot_approx_deriv(a,b,N,fstring,fpstring)

    fp = inline(fpstring);
    x = linspace(a,b,N+1);
    yp = fp(x);
    yp_approx = approx_deriv(a,b,N,fstring);

    % code for plotting results goes here

end
```

Test your code for the interval $[0.2, 2]$ on the function $f(x) = \sin(1/x)$, for $N = 10, 20, 40, 80, 160$. **Label your plots clearly** (i.e. include a legend, label the axes, give a title) and include them in your homework.

3 Write a code that implements composite Simpson's rule (algorithm 4.1 in the book). Test your code for the interval $[0, 1]$ on the function $f(x) = \frac{e^x + e^{-x}}{2}$ with $N = 10, 20, 40, 80, 160$. An example code template is:

```
function I = comp_Simpsons(a,b,N,fstring)

    f = inline(fstring);

    % code for composite Simpson's rule goes here.

end
```

4 Write a code that has as input an interval $[a, b]$, a function f, the antiderivative F, and a *vector* of N's, and plots the error for each of the N's versus the step sizes h. An example code template is:

```
function [] = plot_comp_Simpsons_error(a,b,fstring,Fstring,Nvec)

    f = inline(fstring);
    F = inline(Fstring);

    n = length(Nvec);
    hvec = zeros(n,1);
    errorvec = zeros(n,1)

    % code for calculating the error and h for each N goes here.

    % code for plotting the error goes here.

end
```

Now modify your code to plot the *logarithm* of the error versus the *logarithm* of h. Use this modified code to plot the errors for $N = 10, 20, 40, 80, 160, 320, 640$ using the same interval and function as in problem 3. As before, make sure to **label your plots clearly**. How does the slope of your log-log plot compare to the error term given for composite Simpson's rule?