CMPSC/Mathematics 451
MATLAB Program Two
Due 1 April 2020
Spring 2020

You are to write MATLAB (or Octave) functions to implement the composite Trapezoid rule and the composite Simpson's Rule for approximate computation of the integral

$$I = \int_a^b f(x)dx.$$

Both integration methods will have $(n{+}1)$ evenly space mesh points $x_0, x_1, \ldots, x_n$ where

$$x_k = a + kh, \quad h = (b-a)/n.$$

Given n you can generate the mesh points with the two MATLAB statements

```
h=(b-a)/n;
x=a:h:b;
```

Your function for the Trapezoid rule should have the first line

function int=trap(f,a,b,n)

where f is a function handle for the function to be integrated. I recommend reviewing the MATLAB intrinsic functions **feval** for function evaluation and **sum** for summing the entries of a vector. Your code should be very short and should avoid loops. The output *int* is $T_0(h)$ (but don't use this notation in your code). I suggest treating $n = 1$ as a special case.

As part of your Simpson's rule routine, you will need a function called **update_trap** for updating the trapezoid rule. It has the first line

function new_int=update_trap(f,a,b,n,old_int)

where a,b, and n have the same meaning as for **trap**. If we let $h = (b-a)/n$, then $old\_int = T_0(h)$. The output argument $new\_int = T_0(h/2)$.

This uses the formula in the notes that produces $T_0(h/2)$ from $T_0(h)$ and the function f evaluated at the midpoints

$$x_{k+1/2}, = a + (k + 1/2)h, \quad k = 0\colon n - 1.$$

Again, this code for this function should be short.

The last function implements the composite Simpson's rule using the functions **trap** and **update_trap**. It's first line is

function [int,err]=simpson(f,a,b,n).

where

$$int = T_1(h) = (4 * T_0(h/2) - T_0(h))/3$$

and err is

$$err = (T_1(2h) - T_1(h))/15$$

is the error estimate from the notes. You may assume that n is divisible by 2. Thus **simpson** will make one call to **trap** (for $n/2$) to compute $T_0(2h)$ and two calles to **update_trap** to compute $T_0(h)$ and $T_0(h/2)$. This function should also be very short.

Test your code on three functions that I will put in the folder **Program Two Examples** in the MATLAB Assignments Folder on Canvas. They are **pifunc.m**, **logderiv.m**, and **erfderiv.m** and they are all one line functions. For these three functions, use the Trapezoid rule and Simpson's Rule with $n = 32, 64, 128$. Also produce compute $T_0(h) - I$ and $T_1(h) - I$ for the "exact" integrals. For Simpson's Rule, also produce the estimated error.

- **pifunc.m**. Compute the integral over the interval $[0, 1]$. This function is

  $$pifunc(x) = 1/(1 + x^2).$$

  You should get an approximation of π.

- **logderiv.m**. Compute the integral over the interval $[1, e]$ where $e =$ **exp**(1) is the natural logarithm base. This function is

  $$logderiv(x) = 1/x.$$

  The integral should be approximately 1.

- **erfderiv.m**. Compute the integral over the intervals $[0, 1]$ and $[0, 3]$. This function is

$$erfderiv(x) = \frac{2}{\sqrt{\pi}} e^{-x^2}.$$

  You should get good approximations of **erf**(1) and **erf**(3) from the **erf** function in MATLAB.

Please turn in all three functions, your calling script, and your results. I recommned use of the function **diary** in MATLAB to produce a diary file for your output.

Quiz Four will involve using parts of this project to construct a function for Romberg integration.