Homework 6 # Integration and Library Upkeep Due Oct 23, 2020

1: Update Your Library (20 pts)

By this point in time your library should contain the following routines: Your function declarations should be identical to the ones listed below. This includes the **function name**, the **number of parameters**, and the **order of parameters.**

- The Bisection method,
  `Bisect(double a, double b, double tol, double (*f)(double x))`
- Newton-Raphson method,
  `NewtonRaphson(double a, double tol, double maxit, double (*f)(double x), double (*df)(double x))`
- Golden-Section Search Method,
  `GoldenSectionSearch(double xu, double xl, double tol, double (*f)(double x))`
- Parabolic Interpolation,
  `ParabolicInterp(double x1, double x2, double x3, double tol, double (*f)(double x))`
- The Recursive Trapezoid Rule for functions,
  `CompositeTrapRule(double a, double b, double n, double avgdf2, double tol, double (*f)(double x))`
- Recursive Simpson's 1/3 Rule for functions,
  `CompositeSimps13(double a, double b, double n, double avgdf4, double tol, double (*f)(double x))`
- Simpson's Rules for data,
  `DataSimpsonsRule(double data[], double x[], double n)`
- Trapezoid Rule for data,
  `DataTrapezoidalRule(vector<double> x, vector<double> fx, int n)`
- Trapezoid Rule for function, non recursive,
  `TrapezoidRule(double a, double b, int n, double (*f)(double x))`
- 4th order Romberg Integration,
  `RombergIntegration(double a, double b, double tol, double (*f)(double x))`
- Adaptive Quadrature using Boole's Rule,
  `AdaptiveQuadrature(double a, double b, double tol, double (*f)(double x))`

Make sure that these routines work by testing them on the functions we covered in class,but do not submit any code for these tests. We will check them to make sure they are working using unit testing.

2: Integrating a Data Set (30 pts)

You've just been handed a set of experimental data to integrate, `xData` and `fxData`.

(a) Of the methods we've covered during lecture for integrating sets of data, which method would you choose use to integrate this data set? Why?

(b) Apply the method you chose to the set of data provided with this assignment. Report the integral value (sum).

3: Integrating a Function (50 pts)

The upward velocity of a rocket can be computed by the following formula:

$$v = u \ln\left(\frac{m_0}{m_0 - qt}\right) - gt$$

where v is the upward velocity, $u = 1850 m/s$ is the velocity at which the fuel is expelled relative to the rocket, $m_0 = 160,000 kg$ is the initial mass of the rocket, $q = 2500 kg/s$ is the fuel consumption rate, $g = 9.81 m/s^2$ is the acceleration of gravity, and t is the time.

(a) Report the height of the rocket after $30s$ using `RombergIntegration` using a `tol = 1e-4`.

(b) Report the height of the rocket after $30s$ using `AdaptiveQuadrature` using a `tol = 1e-4`.

4: Submission Details

Only the function used for Question 3 of this assignment should be defined within your Main.cpp.

Within the main function of your Main.cpp, you should be **only** be calling `DataSimpsonsRules` **OR** `dataTrapezoidalRule`, `RombergIntegration`, and `AdaptiveQuadrature`, with their appropriate parameters and reporting their returned values.

Submit your entire SciProgLib folder as a .zip file.