Math 4610 Tasksheet 4: Derivative Approximations and Sources of Errors

- Due Date: Friday, Oct. 7, 2022

The following tasks comprise the second homework assignment for the class. In the class, we have talked about a few basic content items.

Task 1. Approximation of the Second Derivative of a Function.

Using Taylor series prove that the approximation

$$f''(x_0) \approx \frac{f(x_0 - h) - 2f(x_0) + f(x_0 - h)}{h^2}$$

is of order h^2. Write the expression in the form of

$$\left| f''(x_0) - \frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2} \right| \leq C\, h^2$$

This is similar to the work we did on the first derivative approximation. You can write out the details using pencil and paper to do the derivation. If your work is handwritten, scan the work into a pdf file and include the pdf in your submission.

Task 2. Computer Code For Second Derivative Approximation.

Create a code to perform the approximation of the derivative given in the previous task. The code should accept as input

- the function, $f(x)$,
- the point, x_0, where the approximation is to be computed, and
- the initial increment, h parameter.

Output the result of the approximation. Test your code on the function

$$f(x) = \frac{(x - \frac{\pi}{2})\, tan^2(x)}{x^2 + 65}$$

for $x_0 = \frac{\pi}{4}$. Print out a few values of the derivative approximation of the given function.

Task 3. Computational Convergence Study

Write a routine that will fit data sets to a linear polynomial as described in class. The code should accept as input

- a data array, x, containing the input values, and
- a data array, y, of the measured output.

The output should include two real numbers, a and b, the approximate coefficients for the linear fit. Test your code on the data provided by the code in Task 2 of this assignment. Use a point like $x_0 = 1.0$ for the function defined in the task.

Task 4. Starting a Shared Library for Computing Errors.

Create 2 routines, one for computing the absolute error and one for computing the relative error between a number, say u, and some approximation, v, of the original exact value. Also include routines that output the number of digits of accuracy for both single and double precision. Create a shared library for collecting routine for computing errors that is separate from the root finding archive created earlier in the course.

Task 5. Implementing the Explicit Euler Method.

In class we considered the approximate solution of initial value problems of the form

$$\frac{dx}{dt} = f(t, x(t))$$

with initial condition $x_t)0) = x_0$ via the Explicit Euler method. Implement this method for application to the Logistic equation

$$\frac{dP}{dt} = \alpha\, P - \beta\, P^2$$

with $P(0) = P_0$. Your code should accept as input

- the growth constant, $\alpha > 0$,

- the limiting constant, $\beta > 0$, and
- the initial value, P_0, to start the simulation.

Test your codes for some given values of the parameters:

- test 1: $\alpha = 0.2$, \beta=0.0005, and $P_0 = 10.0$
- test 2: $\alpha = 0.01$, \beta=0.0005, and $P_0 = 10.0$
- test 3: $\alpha = 2.0$, \beta=0.0005, and $P_0 = 10.0$