AERO-222: Introduction to Aerospace Computation, Spring 2023
Homework #1, Due Date: Thursday, February 9, 2023

Show all work and justify your answers!

Instructions

- *This homework contains both handwritten and coding problems and shall be submitted according to the following guidelines.*
- *Hardcopy:*
  - *Due on CANVAS at 11:59 PM on the day of the deadline.*
  - *Shall include screenshots of any hand-written work.*
  - *For coding problems, the hardcopy shall include any relevant derivations and emphasize the final results (i.e. boxed, highlighted, etc.).*
  - *Shall be submitted as a single file according to the provided template with the following naming scheme:* "LastnameHW#.pdf"
  - *If preferable, you can put all of your work into a single Jupyter notebook (.ipynb) with photos of your hand-written work as well. Markdown allows for images.*
- *Coding Submission:*
  - *Due on CANVAS at 11:59 PM on the day of the deadline.*
  - *Shall be submitted as a single file according to the provided template with the following naming scheme:* "LastnameHW#.py" or "LastNameHW#.ipynb".
  - *The script shall print out all outputs asked for in the problem.*
- *Late submissions will be accepted with a 10 point deduction per day late.*

---

1. **Root-finding Algorithms (Coding Problem) (15 pts)** Calculate the roots (to an absolute error $\varepsilon_x < 1e-08$) of

$$f(x) = 3x^2 \sin x - x \cos x + 4 = 0$$

Use the following 3 methods:

- Bisection method
- Secant method
- Regula-Falsi method

These methods require a set of starting points/bounds. Use $x_1 = -2$ and $x_2 = +2$, which will meet the initial requirements for all 3 methods. For each method, calculate and plot $f(x_n)$ as a function of the iteration number n. Which method converges fastest?

2. **Round-off Error (Coding Problem) (10 pts)** Write a script to:

   (a) Evaluate the cubic polynomial, $f(x)$, at $x = 1.32$, using default machine precision and again using only three significant digits at each arithmetic operation. Calculate the absolute and relative error of the final result:

   $$f(x) = 3.15\,x^3 - 2.11\,x^2 - 4.01\,x + 10.33$$

   (b) Repeat part (a) but do it with *nested multiplication*. Compare errors. Which takes fewer operations? The nested form is:

   $$f(x) = [(3.15\,x - 2.11)\,x - 4.01]\,x + 10.33$$

3. **Truncation Error (Coding Problem) (20 pts)** Evaluate $f(x) = e^{-5}$ to five digits of precision using the following two approaches:

   (a) $e^{-5} = \sum_{k=0}^{10} \frac{(-5)^k}{k!} = \sum_{k=0}^{10} \frac{(-1)^k 5^k}{k!}$

   (b) $e^{-5} = \frac{1}{e^5} \approx \frac{1}{\sum_{k=0}^{10} \frac{(5)^k}{k!}}$

   Note that the true value to five digits of precision is 1.8316×10^{-2}. Which formula gives more accurate results and why? Plot the error of each approach as a function of iteration number.

4. **Taylor Series (15 pts)** Expand the function, $f(x) = x^4 + \sin x$, by Taylor series up to degree 3 for the following cases:

   (a) centered at $x_0 = 4$ and evaluated at $x_1 = x_0 + 0.2$;

   (b) centered at $x_0 = 3$ and evaluated at $x_1 = x_0 - 0.7$.

5. **Variables and Computer Precision (10 pts)** Answer the following questions:

   (a) Which can store a larger number, a signed int or unsigned int?

   (b) Which uses more memory, a float or double? Which is more precise?

   (c) If I'm trying to establish if two integers are equal, what's the easiest way to compare their values in code? Write the statement that would achieve this?

(d) If I'm trying to establish if two real numbers (double or float) are equal, what's the "correct" way to compare their values in code? Write the statement that would achieve this.

(e) What is Python default machine precision?

6. Base Conversion (10 pts) Show all steps:

(a) Convert 1482 from decimal to binary.

(b) Convert 0.1 from decimal to binary using 1 byte (8 bits).

(c) Does adding zeros to the back (ones side) of a base 10 integer change the value? What about a base 2 (binary) integer?

(d) Now consider the value to be a decimal. Does anything change?