ME2 Computing- Tutorial 10: Root finding

Learning outcomes:

- Being able to find roots of a function through bisection and Newton-Raphson methods
- Being able to solve numerically systems of non-linear equations

Before you start

In your H drive create a folder *H:\ME2CPT\Tutorial10* and work within it.

Task A: Bisection method

Write a Python function *mybisection*, to determine the root of the equation $f(x) = 0$, within a given interval $[a, b]$ and a specified accuracy ε, using the bisection method. $f(x)$ is a known function and might be implemented as a separate Python function.

Test your code by finding the root of the equation:

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

Find and compare the root with accuracies ε = 0.1, ε = 0.01 and ε = 0.001

Task B: Bisection method with a discrete function $f(x_n)$

Often the function $f(x)$ is available only in a discrete form, as a set of values $f(x_n)$, for a finite number of points. In such cases the evaluation of the function at any x, as requested by the bisection method, can be obtained through interpolation. Use a copy of the code in Task A and amend it to find roots from a discrete set of values.

Test your code with the discrete set of values stored in files *x.txt, fx.txt*, containing values of the independent variable x and the function $f(x)$, respectively.

Task C: Newton-Raphson method

Write a Python function *myNewton*, to determine the root of the equation $f(x) = 0$, given an initial guess x_0 and a specified accuracy ε, using the Newton-Raphson method. $f(x)$ is a known function and might be implemented as a separate Python function.

Task D: Newton-Raphson method for systems of non-linear equations

Write a Python function *mySystem*, to determine the solutions of system of non-linear equations, using the Newton-Raphson method.