CentraleSupelec **Numerical Methods, S8**

Introduction to Python

In this workshop, an introduction is given to Python programming. The three main elements that are of interest here are the use of array and control statements, and the use of plotting capacities in Python.

1 Definition and plotting of a function

1.1 First-dimensional case

We first write a code that evaluate the function $f(x) = x^3 + x^2 + x + 1$

1. Define a 1D array x containing N_x points between -1 and 1 (including first and end points).
2. Define the 1D array f which contains the evaluation of $f(x)$ at each point of array x.
3. Plot the function f versus x. Add label to x and y axes, and add the title '$f(x) = x^3 + x^2 + x + 1$'.
4. Change line style to red dashed line, and line width to 3

1.2 Two-dimensional case

Here the function now depends on two variables: $g(x, y) = x^2 + y^2 - 2xy$.

1. Construct the mesh point position x and y of the 2d space $[-1, 1]^2$ using the function mgrid.
2. Define the 2D array g which contains the evaluation of $g(x, y)$ at each point of array (x, y).
3. Plot the function g versus x and y using a surface plot. Add label to x and y axes, and add the title '$g(x) = x^2 + y^2 - 2xy$'.
4. Synthesize the 1D and 2D cases in a figure with two sub-plots.

2 Evaluation of the integral of a function

To evaluate the integral of a function the two most classical method are the rectangle method and the trapezoidal rule. Considering that the integral F of a function $f(x)$ over an interval $[-1, 1]$ can be decomposed into integrals over $N_x - 1$ sub-intervals $[x_{i-1}, x_i]$:

$$F = \int_{-1}^{1} f(x)\mathrm{d}x = \sum_{k=1}^{N_x-1} \int_{x_{i-1}}^{x_i} f(x)\mathrm{d}x \tag{1}$$

The integral over each interval can be evaluated by considering either a constant function (rectangle method) or a linear function (trapezoidal rule) between end points. The final formulas are then:

$$\text{Rectangle Method :} \quad F \approx \sum_{k=1}^{N_x-1} f(x_i)\Delta x \tag{2}$$

$$\text{Trapezoidal Rule :} \quad F \approx \sum_{k=1}^{N_x-1} \frac{f(x_i) + f(x_i + 1)}{2}\Delta x \tag{3}$$

1. Implement both formulas to evaluate the integral of the function $f(x)$.

2. To evaluate the order each method, evaluate the error for different number of points.

3. Plot the results in a log-log representation.

3 Root-finding algorithms

In this section, we now want to find the root x^* of the equation $f(x) = 0$. To do so, three method are considered : the fixed point iteration, the Newton method and the secant method. All methods end iterations as soon as a sufficient accuracy is achieved on the equation to be solved. The desired accuracy is determined by a user-defined threshold. The convergence of the algorithm can strongly depends on the initial guess x^0 used to start the iterations.

1. Fixed point iteration transforms the problem into $x = g(x)$ and performs the following iterations

$$x^{n+1} = g(x_n)$$

   Implement a fixed point iteration with $g(x) = f(x) + x$ starting from an initial guess x^0.

2. For a given guess for the solution x^n, the Newtons' method performs the following iterations:

$$x^{n+1} = x^n - f(x^n)/f'(x^n) \tag{4}$$

   Implement the Newton method for the function $f(x)$ starting from an initial guess x^0.

3. The secant method is given by

$$x^{n+1} = x^n - f(x^n)\frac{x^n - x^{n-1}}{f(x^n) - f(x^{n-1})}$$

   Implement the secant method for the function $f(x)$ starting from two initial guesses x^0 and x^1.

4. These methods are characterized by different rates of convergence α where $|e^{n+1}| \approx C|e^n|^\alpha$ and with $e^n = x^n - x^*$ the error at the n$^{\text{th}}$ iteration. Plot $\ln(e^{n+1}/e^n)$ and determine α for each method, knowing that $\ln(|e^{n+1}/e^n|) = \alpha \ln(|e^n/e^{n-1}|)$.