Exercise 01

Numerical Approximations

Deadline: Please hand in your protocol in **pdf format** by **Thursday, 11th May, 10.15 a.m.** to **saleksic@zedat.fu-berlin.de**. The proctol should contain Python code, plots, and comments if necessary.

1.1 Series limit (30 Points)

In this exercise, you can practice the use of for-loops and while-loops. Consider the convergent series

$$log(x+1) = \sum_{n=0}^{\infty}(-1^n)\frac{x^{n+1}}{n+1} \quad x \in (-1, 1] \tag{1}$$

(a) Write a program that calculates the series using a for-loop and then vectorize the script. Pick a real number x and test the convergence of the series. (Hint: to implement the series, you have to replace infinity by a number N).

(b) Write a program which numerically estimates the limit of the series by evaluating the first N terms. Plot the estimate as a function of N up to $N = 20$.

(c) Write a program which evaluates the series until the increment is less than a pre-defined threshold. How many terms are needed until the increment is less than 10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}?

(d) Often one does not know in advance how long it will take until a certain threshold is met. To prevent a program from running too long, one stops the calculation if either the threshold or a maximum number of iterations is reached. Modify the program from (c) such that it stops if one of the following conditions is met:

- the increment is less than a predefined threshold,
- a maximum number of terms have been evaluated.

1.2 Monte Carlo 1 (25 Points)

(a) Generate two sequences of 1000 random real numbers drawn from the interval $[-5, 5]$.

(b) Interpret the two sequences as x and y coordinates. Produce a scatter plot.

(c) Add a circle with radius $r = 5$ and center $(0, 0)$ to the scatter plot

(d) Numerically estimate the area of the circle as

$$A_{circle} \approx \frac{N}{N_{tot}} \cdot A_{square}$$

where N is the number of data points within the square and N_{tot} is the total number of random data points.

(e) Increase the number of random data points. Does the result improve?

1.3 Monte Carlo 2 (30 Points)

Consider the function

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

(a) Evaluate the integral $\int_2^6 f(x)\mathrm{d}x$ using a Monte Carlo method. Consider a rectangle defined by the points $\{[2,0],[6,0],[6,f(6)],[2,f(6)]\}$, then generate $N_{tot} = 100$ random points within the rectangle and count how many points N fall under the curve.
(Hint: draw N_{tot} random numbers from the interval $[2,6]$ for the x-values, and N_{tot} random numbers from the interval $[0,f(6)]$ for the y-values).
Finally, the value of the integral is

$$\int_2^6 f(x)\mathrm{d}x \approx \frac{N}{N_{tot}} \cdot A_{rectangle}$$

(b) Increase the number of points N_{tot} and plot the integral result as a function of N_{tot}. Compare the estimate with the exact value of the integral (hint: use scipy.integrate library to compute the real value of the integral).

1.4 Neighbor lists (15 Points)

(a) Generate two sequences of $N = 100$ random real numbers drawn from the interval $[-5,5]$.

(b) For each datapoint, write out the list of neighbors within a radius $r = 0.5$.

(c) Write a matrix M of size $N \times N$ where each element M_{ij} is 0 if the distance between the point i and j is larger than r, otherwise is 1. Save matrix to a file by using savetxt comand of Numpy.

(d) What is the average number of neighbors for each datapoint?