# Errors in numerical analysis

Numerical analysis - University of Luxembourg - Session 1

Exercises

## Python documentation

* numpy: <https://numpy.org/doc/>
* matplotlib: <https://matplotlib.org/stable/gallery/subplots_axes_and_figures/index.html>
* sympy <https://www.sympy.org/en/index.html>

## Exercises on numerical errors

Exercise 1. **Absolute and relative errors**

Step 1: Print the relative and absolute errors of the Stirling approximation

for different values of n. Explain your observations for large values.

* Implement a function that computes the Stirling approximation "def Stirling(n): return ...", and another one that computes the errors "def Errors(x, xref): ... return abs err, rel err"
* For factorials, you can use “from scipy.special import factorial”
* For simple plotting, you can use the following concise form
  + plt.figure(figsize=(7,4))
  + plt.plot(N, err0, 'm-o', label = 'Stirling') or plt.loglog(N, err0, 'm-o', label = 'Stirling')

Step 2: Do again the comparison with more terms of the asymptotic development

* You can implement more functions def Stirling2(n) and def Stirling3(n):

Comment on the accuracy of the approximation.

Exercise 2. **Round-off vs discretization error**

Let and

1. Write a Taylor expansion of around with a step size , and derive a formula to approximate of first order (see the lecture slides).

2. Plot the absolute error in a log-log scale as a function of the step size , where is in the range . What is going on when is too small?

3. Add another plot by changing the numpy floating point precision using np.float32, for both and . Please comment on your observations

* use np.logspace to generate the h values

## Exercises on floating point systems

Exercise 3. **Decimal approximations**

Compute decimal approximations of the real number 0.1 for 32-bit floating number precision in base 2. Then do the same thing for 0.25. Explain the difference that you observe. What about 0.35?

* You can use f-strings for printing the values print(f'\nSingle Precision (32-bit): {single\_precision:.20f}')

Exercise 4. **Decimal Conversion of a Single-Precision Floating Point Number**

Find the decimal equivalent of the following 32-bit precision machine number (sign, exponent, significand):

0 10000000 10010010000111111011011

* Mantissa 1’s are in position 1,4,7,12-17,19-20, 22-23

This decimal number approximates a well-known number. How many significant digits does this 32-bit representation achieve?

## Exercises on stability and conditioning

## Exercise 5**.** **Polynomials**

Evaluate and plot the polynomial function written in its current factored and developed forms. Look in particular at the behavior close to the root . You can use the interval equally spaced by 100 points. Explain your observations.

* One can use sympy to find expression for the developed form

## Exercise 6**.** **Catastrophic cancellation**

Recall the quadratic formula to find the root of . Test the usual formula for the coefficients. . Explain what is going on, and propose a method to compute both roots accurately.