Programming Methods NA 2020: Assignment 2 - Mathematical Functions

prna@liacs.leidenuniv.nl

November 8, 2020

1 Introduction

In this assignment, you will learn some basic mathematical functions that will also be greeting you again later in your studies! Namely the factorial and exponential function.

2 Computing the factorial function

2.1 Background!

The factorial of an integer is: the number itself times all lower integers up to 1:

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 2 \cdot 1 \tag{1}$$

You will meet this function later in your physics and astronomy career often. Below we briefly present two examples, which you will encounter later in your courses. It is used in order to normalize the wave function of a harmonic oscillator in quantum mechanics:

$$|C_n|^2 = \frac{1}{n}|C_{n-1}|^2 = \frac{1}{n-1}\frac{1}{n-2}|C_{n-2}|^2 = \cdots = \frac{1}{n!}|C_0|^2. \tag{2}$$

We also use it to describe the state of a system of the microcanonical ensemble in statistical mechanics.

$$\Omega(E, N) = \frac{N!}{(N-m)!m!}. \tag{3}$$

2.2 The task

We now want you to compute this value yourself by writing a function called *vanilla_factorial*. This function must be implemented in pure python and may not use the *math.factorial* or the *np.math.factorial* function.
Use python's *time.time_ns()* to measure the number of nanoseconds each of the three implementations takes to calculate the factorials from 0 to 10000. Include the measurements in your report and reason about why which function performs the way it does, and which one you believe to be easiest and most convenient to use. Note that there are two ways of implementing this: As a for loop or as

a recursive function. We want you to elaborate on your choice in the report! Also, answer the following question in your report: Factorials get pretty big, very quickly. What is the largest factorial you can store as an `int64`, what is the largest value you can store as a `float64`?

3 Approximating the exponential function

3.1 Background

The exponential function is also prevalent in astronomy and physics, for example if we wish to solve a differential equation we get a solution that contains the exponential, for example a function whose derivative also depends on itself and time:

$$\frac{df(t)}{dt} = -atf(t) \implies f(x) = f(0)\exp\left(\frac{-at^2}{2}\right). \tag{4}$$

And when using statistics to describe a measurement in physics or observations in astronomy, we very often assume that they are taken from the Gaussian distribution with mean μ and standard deviation σ:

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right). \tag{5}$$

3.2 The exponential function as a series

The exponential can be written as an infinite series:

$$\exp(x) = \sum_{k=0}^{\infty}\frac{x^k}{k!} \tag{6}$$

3.3 Approximating

Write a function called *vanilla_exp* that approximates the exponential function from -1.5 to 1.5 in 0.1 increments, and compare it to *numpy.exp* function. What is the absolute error between your approximation, *numpy.exp* and *math.exp*? Also, as with the factorial function, make a performance comparison of the three different implementations calculating everything between -1.5 and 1.5 but in 0.001 increments. Include both error and performance in your report!

3.4 Report remarks

In the report, information from every task of the assignment should be presented. For every task, an explanation is given why the task is completed in the way it is. The report should include at least the following contents:

1. Introduction
2. Factorials
   (a) Performance comparison
   (b) Implementation choice

   (c) Answers to knowledge question

3. Exponential function
   (a) Absolute error comparison
   (b) Performance comparison
4. Conclusion
5. Discussion

3.5 Task remarks

- At the top of the python code, comments must specify the following information for *programmers*: what the program does, the date of the last edit, which version of the interpreter is used and the operating system that you worked on.
- Regarding the layout: Make use of empty lines, indentation, comments etc.

4 Submission

Deadline: 22 November 2020 before 23:59.
Only one person per pair of two submits.

- Submit the Python code (.py) and your report (.pdf) in BrightSpace.
- **Text and comments can be in either English or Dutch, as long as you stick to one of them for the entire assignment!**
- Your submitted files **must** be named according to the following scheme: XXXXXXX-YYYYYYY-opdr1.py and XXXXXXX-YYYYYYY-opdr2.pdf. XXXXXXX and YYYYYYY should be replaced with your student numbers e.g. 2484513-1528317-opdr2.py
- The submitted python file must contain the function vanilla factorial and vanilla exp. Both functions take one argument, namely a number, and as a result, return one.
- Specify the submission date and the names of the two creators everywhere, especially in the first line comments of the Python code as well as the header of your report.
- **Submissions can only be made digitally. Paper copies will be refused**
- Use Python 3.5 or above. Make sure your program is able to run on linux (by e.g. using the correct slash in file paths)

5 Grading key

- (consistent) Layout 15%
- Comments 15%
- Report 25%
- Information Block 5%
- Functionality 40%