Assignment 1

In this exercise sheet, we practice on the concepts of machine precision, floating point arithmetic and condition of a problem.

Graded exercise

Exercise 6 (Working with machine precision)
Consider the function

$$f(x) = \frac{1 - \cos(x)}{x^2}, \tag{1}$$

for $x \in [10^{-9}, 10^{-6}] =: I$. A graph of the function should look as in Figure 1.

(a) In PYTHON, write the code to plot the function (1) on the interval I using the logarithmic scale on the x-axis. Executing the code, which plot do you obtain? Most probably, you will obtain something different from Figure 1.

Hint: For plotting, the PYTHON library `matplotlib` can be of great help. The function `semilogx` can help you with setting the logarithmic scale on the x-axis.

Hint: To plot, use many evaluation points, of the order of 50000.

(b) Provide an explanation for the plot obtained in the previous task. Explain in particular the reason for the oscillatory behavior and for the plateau preceeding it.

(c) Modify the expression in (1) so as to produce the plot of the function as in Figure 1. Code the obtained expression and produce the plot.

Other exercises

Exercise 7 (Testing with machine precision)

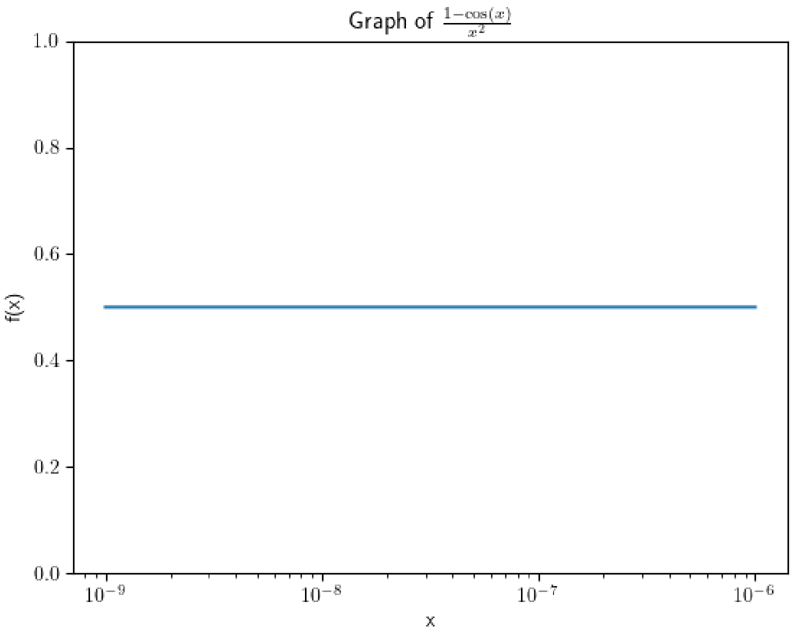

Figure 1: Plot of the function in (1).

(a) Print the value of the machine epsilon in PYTHON. Note that, in PYTHON, `eps` is defined as the distance between 1 and the smallest float > 1, so its value is twice the one defined in the lecture, see discussion at the end of Section 1.2 of the lecture notes.

(b) In PYTHON, print the results of the operations $10^{16} + 1 - 10^{16}$ and $10^{16} - 10^{16} + 1$, respectively, *using floats*. Why do we obtain two different results?

Hint: Remember that PYTHON saves `1` as an integer and `1.` as a float.

Exercise 8 (Floating point representation)

(a) Represent the following numbers in double precision (64 bits): 46.875 and 0.1.

Example: For 8.5 we would have

$$(8.5)_{10} = (1000.1)_2 = (1.0001)_2 \cdot 2^3 = \underbrace{\boxed{0}}_{\text{Sign}}\ \underbrace{\boxed{0}\boxed{1}\boxed{1}}_{\text{Exponent}}\ \underbrace{\boxed{0}\boxed{0}\boxed{0}\boxed{1}}_{\text{Mantissa}},$$

where we remind that, since we always have $d_1 = 1$ when using the normalization, we do not need to store it.

(b) In double precision, can we compute `0.0000000000000000003 - 0.0000000000000000002` without rounding errors? Why?

Exercise 9 (Condition of the division)

Consider the problem to compute the quotient $y = x_1/x_2$ of two nonzero reals x_1, x_2.

(a) What are the componentwise (aka elementwise) condition numbers of y with respect to x_1 and x_2?

(b) Is the problem well-conditioned?