Exercise 1

Set up an algorithm which converts a floating number in the decimal representation to a floating number in the binary representation. You may or may not use a scientific representation. Write thereafter a program which implements this algorithm following much of the same procedure outlined in program2.cpp or program2.f90 in chapter 2 of the lecture notes.

Exercise 2

2a) Make a program which sums

$$s_{\rm up} = \sum_{n=1}^{N} \frac{1}{n},$$

and

$$s_{\rm down} = \sum_{n=N}^{n=1} \frac{1}{n}.$$

The program should read N from screen and write the final output to screen.

2b) Compare $s_{\rm up}$ og $s_{\rm down}$ for different N using both single and double precision for N up to $N = 10^{10}$. Which of the above formula is the most realiable one? Try to give an explanation of possible differences. One possibility for guiding the eye is for example to make a log-log plot of the relative difference as a function of N in steps of 10^n with $n = 1, 2, \ldots, 10$. This means you need to compute $log_{10}(|(s_{\rm up}(N) - s_{\rm down}(N))/s_{\rm down}(N)|)$ as function of $log_{10}(N)$.

Exercise 3

We want you to compute the first derivative of

$$f(x) = tan^{-1}(x)$$

for $x = \sqrt{2}$ with step lengths h. The exact answer is $1/3$. We want you to code the derivative using the following two formulae

$$f'_{2c}(x) = \frac{f(x+h) - f(x)}{h} + O(h), \tag{1}$$

and

$$f'_{3c} = \frac{f_h - f_{-h}}{2h} + O(h^2), \tag{2}$$

with $f_{\pm h} = f(x \pm h)$.

(3a) Find mathematical expressions for the total error due to loss of precision and due to the numerical approximation made. Find the step length which gives the smallest value. Perform the analysis with both double and single precision.

(3b) Make thereafter a program (see programs under chapter 3 for examples) which computes the first derivative using Eqs. (1) and (2) as function of various step lengths h and let $h \rightarrow 0$. Compare with the exact answer.

Your program should contain the following elements:

- A vector (array) which contains the step lengths. Use dynamic memory allocation.
- Vectors for the computed derivatives of Eqs. (1) and (2) for both single and double precision.
- A function which computes the derivative and contains call by value and reference (for C++ users only).
- Eventually a function which writes the results to file.

(3c) Compute thereafter

$$\epsilon = log_{10}\left(\left|\frac{f'_{\mathrm{computed}} - f'_{\mathrm{exact}}}{f'_{\mathrm{exact}}}\right|\right),$$

as function of $log_{10}(h)$ for Eqs. (1) and (2) for both single and double precision. Plot the results and see if you can determine empirically the behavior of the total error as function of h.