Tutors: C. Czaban, A. Dromard, J. Glesaaen, M. Neuman, D. Palao, C. Schäfer, A. Sciarra

Tutorial IV

November 14

Along this tutorial we will exercise several control flow statements. In programming, some tasks can be carried out equivalently, at least from the operational point of view, following different approaches. In particular, in this tutorial we will practice some basic procedures to make the computer repeating tasks. We will focus our attention on the four options:

(A) `while` loops;

(B) `do-while` loops;

(C) `for` loops;

(D) `goto` and `if`/`if-else`.

But remember two important generic remarks. First, not all of the above options fit naturally always. Although normally all of them are technically possible in a particular case, usually some choice is much better/worse than the others. Second, please do not use the `goto` in your code beyond this tutorial, if you can avoid it. In general code without `goto`s is easier to understand and mantain than code with `goto`s. See the discussion about that issue in the relevant section of chapter 3 in the book by Kernighan and Ritchie for more details and examples.

Excercise 1 [*Factorial*] Write four programs that compute the factorial of positive integer numbers, using in each of them only one of the four options above mentioned: (A), (B), (C) and (D).

Excercise 2 [*Secant method*] Write a program that computes the zeros of a one-dimensional real-valued function, $f(x)$, using the so-called *secant method*[1]. Realize the control flow in four different ways, using in each case only one of the above options: (A), (B), (C) and (D).

To structure your program in a clean way, write a `C` function, `double f(double x)`, that will correspond to the function $f(x)$ of which the zeros are computed. Also write the following functions:

- `double secant_zero_A(double x_min, double x_max, double epsilon)`,
- `double secant_zero_B(double x_min, double x_max, double epsilon)`,
- `double secant_zero_C(double x_min, double x_max, double epsilon)`, and
- `double secant_zero_D(double x_min, double x_max, double epsilon)`,

each of them being a different implementation of the *secant method* where the control flow is realized, as already mentioned, using only one of the four options above ((A), (B), (C) and (D)), and where `x_min` and `x_max` correspond to the lower and upper bounds of an interval in which a zero is guaranteed, i.e. $f(x_{\min})f(x_{\max}) < 0$. The parameter `epsilon` is the relative numerical accuracy. The functions should return when the relative difference between the value corresponding to the actual iteration and the value corresponding to the previous one is smaller than `epsilon`.

Test your program. In particular, check that the four implementations of the secant method are consistent among them. Also check that the results are correct in a couple of particularly simple cases, e.g. finding the zeros of $\cos x$.

[1] See, e.g. the Wikipedia for more details.

Excercise 3 [*Transcendental equation in quantum mechanics*] Consider a particle of mass m in the presence of the following one-dimensional finite potential well in quantum mechanics,

$$V(x) = \begin{cases} -V_0 & \text{for } |x| < L/2 \\ 0 & \text{otherwise} \end{cases} .$$

Solve the Schrödinger equation analytically (you can get help from your favourite text book for this task). At some point, for computing the energy levels, you have to find the zeros of a function which is transcendental. That means that you cannot calculate the zeros of this function hence the eigenvalues of the Hamiltonian analytically.

Therefore, use your *secant method* program, tuning it adequately, to determine the two energy eigenvalues that exist for

$$\begin{aligned} m &= 50 \text{ MeV}/c^2 \\ V_0 &= 500 \text{ MeV} \\ L &= 4 \text{ fm} \end{aligned}$$

up to six significant digits; in other words, the tolerance, `epsilon`, must be 10^{-6}.