I. Roko

Assignment #3 *The answers should be submitted until Wednesday October 26, 16:00*

Exercise 1

To calculate numerically one of the roots of the equation $x^2 - x - 1 = 0$, we can use the following iterative procedure

$$x_{k+1} = 1 + \frac{1}{x_k}, \qquad k = 1, 2, \ldots$$

where x_1 is the initial value (first guess). During the iterations, the solution x_k produced by the algorithm will get closer and closer to the true root.

In the following questions, different alternatives are proposed to program this approximation with R.

1. For $x_1 = 1$, use the `for` loop to iterate 7 times the procedure (i.e. compute the value of x_8). Here we do not need to save the successive values of x_k in a vector.

Now we will no longer iterate a predefined number of times, but will continue the iterations as long as the (absolute) difference between two successive approximations is greater than a given tolerance δ. Thus the algorithm for solving the equation is [1]:

```
x_old = 0
x_new = 1
while CONDITION do
    x_old = x_new
    x_new = 1 + 1/x_old
end while
```

where *CONDITION* should be replaced by the appropriate expression.

2. Program this procedure and execute it for $\delta = 10^{-10}$.

3. Add *control instructions*, in order to force the iterations to stop if a maximum number `maxit = 30` of iterations is reached.

[1] You can also use e.g. `x0` and `x1` instead of `x_old` and `x_new`. Note that here we are using the language of "algorithms" to define the procedure, whereas R's syntax for "while" loops is different (see course notes).

4. What is the solution you get? Replace the solution in the expression $x^2 - x - 1$ and verify that it is indeed (numerically) equal to 0.

Exercise 2

The approximation of function $\exp(x)$, for a given x, can be obtained by the following sum:

$$\hat{e}^x = \left(\sum_{k=0}^{m} \frac{1}{k!} \right)^x .$$

The precision of the approximation increase by increasing m. In fact, for $m \to \infty$, the approximation $\hat{e}^x$ to the true value of $\exp(x)$.

1. Calculate the approximation of $\exp(x)$ following the equation above, with the help of a `for` loop, for $m = 100$ and $x = 3$ (the R command `factorial` can be used). Use the name `exp1` for $\hat{e}^x$.

2. Compute the approximation above without using a `for` loop. Test your approach for the same m and x. Use the name `exp2` this approximation.

3. Display your approximation of $\exp(3)$ for $m = 5$ and compare it to the true value (calculate the *absolute error*[2]).

4. By using a `for` loop, calculate the *absolute error* for $m = 3, 4, \ldots, 20$, and compare the results.

5. To find a good approximation of $\exp(3)$ use the `while` loop with the condition: `absolute error` $>$ `10^(-10)`. Give the value of the optimal m.
   Hint: initialise m at 5, and increase it by 1 at each iteration.

6. Add *control instructions*, in order to force the iterations to stop if a maximum number $maxit = 100$ of iterations is reached.

[2] Reminder: $Absolute\,error\ = |Approximation - True\,value|$