uc3m | Universidad **Carlos III** de Madrid

Departamento de Matemáticas

Numerical Calculus - 2022/2023 - Practice 1 Applied Mathematics and Computation 121

Name and Lastname: Group:

Hand in before 1pm on Monday the 3rd of October. All the solutions to the exercises must appear in a single `.m` file. Answers to questions must appear displayed on screen when running your code. You have plenty of time, but do not leave it for the last minute. Also, you may divide the task among the members of the team, since it contains several subtasks (writing the code, checking that the code does what expected, explaining the results,...), but all of you need to understand what others did since I will ask someone in the team to explain it. Name your file practice01teamXX.m where XX is the number of your team. On the first few lines of code write as comment the names and NIAs of all the members of the team.

The aim of this practice is to better understand truncation and roundoff errors while getting familiar with some MATLAB tools: functions and graphs.

Problem 1. Write a program that asks you for a (3 digit $+1$ sign) mantissa, and an exponent between -1 and 1 and translates numbers in base 10 with those mantissa and exponent into binary numbers with 8 digits $+1$ sign and exponents between -4 and 4, and then convert the number back from binary to base 10 with the original conditions, and provides you with: all 3 expressions of the number before, on base 2 and after, and the absolute and relative errors introduced in the process.

Now we are going to study some special sums: Suppose we take two natural numbers a and b such that $1 \leq b < a$. We are going to compute as precisely as possible the sum

$$S_0(a,b) = \sum_{n=1}^{\infty} \frac{1}{(an+b)(an+b+1)}.$$

Since the sum goes up to infinitely many terms, we need to *truncate* the process at some point. For this we are going to compute the sum of the first k terms, through a function that also depends on k, S_1:

$$S_1(k,a,b) = \sum_{n=1}^{k} \frac{1}{(an+b)(an+b+1)}.$$

Here k is another natural number $k \geq 2$. The sum $S_0(a,b)$ is approximately $S_1(k,a,b) + \frac{1}{a(ak+b+1)}$, and the truncation error is not exactly known, but it is at most

$$S_2(k,a,b) \leq \frac{1}{ak(k+1)}.$$

Each partial result is rounded and we want to determine the roundoff error that we introduce. For that you need to compute the number of floating-point operations according to the propagation of errors described in class. Remember every number may have a roundoff relative error of the machine epsilon (eps).

Problem 2. Write a program/rutine/function that asks for a value of k, a, and b, then

(1) Computes the sum $S_0(a,b)$ with the highest precission possible, summing first $S_1(k,a,b)$ by adding the first terms and then adding at each step the next one.

(2) Does the same but computing first the sum of the *last* terms of $S_1(k,a,b)$, then adds previous ones at each step.

(3) Computes the best possible bounds for roundoff and truncation errors with each method.

(4) Makes use of MATLAB *functions*.

Then combine it with another program/rutine that asks for a value of a and then plots on the same graph a family of curves, one for each value of b between 1 and $a-1$, representing how, as k takes the values $k = 2^1, 2^2, 2^3, ..., 2^{16}$, the total (roundoff plus truncation) error in the sum changes.

Interpret the results. Is there an optimal choice of k? (It could depend on the value of a and b) How large is then the contribution of the roundoff to the total error? How large, that of the truncation error? The explanation should appear as text on the screen when executing the program, together with the names of the functions you have created in the program.