Computer Practise 4

Computer Practise with MATLAB: more series and sequences and errors

4.1 Exercises.

1. The exercises can be done individually or in groups of at most two people. Exercises “shared” or copied from your companions or form previous years will be graded with 0 points.

2. Each group has to email to **virginia.muto@ehu.eus** a zip file with the name *group.zip* that has to contain the script files that solve the questions stated at the end of the page from (a) to (f). The files have to contain the explication of the used commands. Each group has one week to solve the exercises and late deliver will be penalized with half the mark.

3. The name(s) and surname(s) of the author(s) have to appear also in the first line of every Matlab file. Unnamed files will not be graded.

4. Run always the codes before submitting them.

5. Besides the functions used in the previous classes, the used MATLAB functions for the next exercises are:

   - The instruction that allows to chose different possibilities:
     $if \quad condicion \quad \ldots \quad else \quad \ldots \quad end.$
   - The instructions that allows to perform loops:
     $for\, ni = ni_{ini}, ni_{step}, ni_{fin} \quad ... \quad end \quad \text{or} \quad while \quad condicion \quad .... \quad end.$
   - The instruction that allows to print out data with format *fprintf*:

     ```
     fprintf(formatSpec,A1,...,An)
     fprintf(fileID,formatSpec,A1,...,An).
     ```

     For instance

```
% To print out integer numbers in the screen
fprintf('%10d \n' ,ii)
% To print out real numbers with 12 decimals in a file
fprintf(fileID,'%18.12f %18.12f \n' ,xx, yy).
```

The file will be created using the instruction *fopen* and it will be closed using *fclose* (e.g. `CubicValues.txt` of the second problem).

```
fileID = fopen(filename,permission)
fclose(fileID)
```

As example

```
fileID = fopen('CubicValues.txt','w')
⋮
fclose(fileID)
```

Note: Open the text file containing data using the WordPad software and not the "Bloc de notas".

- The *double(n)* command that change an integer into a real double precision number.
- Intrinsic functions as *exp(x)*, *sqrt(x)*, etc.

(a) Write a MATLAB program/script that computes the partial sum of N terms of the series

$$\sum_{n=0}^{\infty} \frac{2^n}{n!}.$$

for $N = 10, 20, 30$. Do you identify the obtained value?

(b) Build a MATLAB program/script in order to find all integers numbers between 0 y 9999 such that the sum of the cube of its digits is equal to the number itself. For instance, $1^3 + 5^3 + 3^3 = 153$. Print out the values which satisfies the given condition in an output file, called `CubicValues.txt`.

(c) Write a MATLAB program/script that generates a graphic showing the definition of limit of the sequence

$$b_n = 1 + \frac{(-1)^n}{n^2},$$

for $\varepsilon = 0.1, 0.001, 0.00001$. Use the program to discover from which term b_n of the sequence the condition $|b_n - 1| < \varepsilon$ is satisfied.

(d) Even if the functions

$$f_1(x) = \sqrt{x}\,\left(\sqrt{x+3} - \sqrt{x}\right) \qquad \text{y} \qquad f_2(x) = \frac{3\,\sqrt{x}}{\sqrt{x+3} + \sqrt{x}}$$

are equal in a mathematical sense (say why in your M-file), they do have different behaviours from a numerical point of view.

Write a MATLAB program/script that uses the *fprintf* command and generates a table with the results of evaluating f_1 y f_2 at the points $x = 1, 10, 10^2, \ldots, 10^{16}$.

Use also the *semilogx* command that allows to generate a plot in a logarithmic scale (to the base 10) for the x axis and a lineal scale for the y axis.

If you compare the results, which one looks better? Which is the reason? Write your answers in your M-file.

(e) Write a MATLAB program/script for the main program which asks for the argument of a function called ''nt'', that will return as output the number of terms needed to approximate π up to n decimals significant digits, using the following series:

$$\pi = 4 \sum_{i=0}^{\infty} \frac{(-1)^i}{2\,i+1}.$$

Use the following definition: a number p^* is said to approximate p with n significant digits, if n is the biggest no negative integer such that

$$\left| \frac{p^* - p}{p} \right| \le 5.0 \times 10^{-n}.$$

Use the *input* command to ask for the value n.

Store the different approximations of π in a file called `PiApprox.txt` using the *fprintf* command.

(f) Build a MATLAB program/script for each of the two algorithms seen in theoretical classes in order to evaluate the function

$$f(x) = \frac{exp(x) - 1}{x} = \sum_{i=0}^{\infty} \frac{x^i}{(i+1)!}.$$

Algorithm 1	Algorithm 2
	$y = e^x$
if $x = 0$	if $y = 1$
$f = 1$	$f = 1$
Else	Else
$f = (e^x - 1)/x$	$f = (y - 1)/\log y$
end	end

Check the results shown in the following table. Moreover, compare these results with the ones obtained computing the partial sum of the series given by the first 30 terms.

x	Algorithm 1	Algorithm 2
10^{-5}	1.000005000006965	1.000005000016667
10^{-6}	1.000000499962184	1.000000500000167
10^{-7}	1.000000049433680	1.000000050000002
10^{-8}	$9.999999939225290 \times 10^{-1}$	1.000000005000000
10^{-9}	1.000000082740371	1.000000000500000
10^{-10}	1.000000082740371	1.000000000050000
10^{-11}	1.000000082740371	1.000000000005000
10^{-12}	1.000088900582341	1.000000000000500
10^{-13}	$9.992007221626408 \times 10^{-1}$	1.000000000000050
10^{-14}	$9.992007221626408 \times 10^{-1}$	1.000000000000005
10^{-15}	1.110223024625156	1.000000000000000
10^{-16}	0	1