Numerical Algorithms (MU4IN910)

Practical 1 - Introduction to MATLAB and floating-point arithmetic

Exercise 1 (A computer counts wrong!)**.**

1. What should the following program normally display?

```
format longE;
function res = Higham(x)
   for i=1:52
      x=sqrt(x);
   end
   for i=1:52
      x = x.^2;
   end
   res = x;
end
Higham(4)
```

2. Run the program on a computer. What is the result? Explain.

3. Run the following program:

```
x = logspace(0, 1, 2013);
y = Higham(x);
plot(x, y, 'k.', x, x, '--')
```

   Explain the graph (identify points such as $y = x$).

Exercise 2 (Recursion)**.** We want to evaluate the definite integral I_n defined by

$$I_n = \int_0^1 x^n.e^{-x}dx.$$

For this we consider the following recurrence relation: $I_0 = 1 - e^{-1}$ et $I_n = -e^{-1} + n.I_{n-1}$.

1. Write a program using this recurrence relation for calculating this integral in double precision.

2. Give a simple enclosure of $x^n.e^{-x}$ on $[0,1]$ and then give an enclosure for I_n. Compare to the results of your calculations for different values of n.

3. Deduce from the previous formula a recurrence relation giving I_n as a function of I_{n+1}.

4. Write a program using this new recurrence relation to calculate I_n from $n + m$ by initializing arbitrarily with $I_{n+m} = 12$ and $m = 50$.

5. Comment, at n fixed, the results obtained for $m = 10$, $m = 20$, $m = 50$ and $m = 100$.

   For comparison here is the exact value of I_n for some values of n:

n	I_n
5	0.071302178109803159860
10	0.036461334624107272383

Exercise 3 (Using BLAS)**.**

1. Let A be a matrix of size $m \times n$. Write a MATLAB column-oriented program calculating

$$s_i = \sum_{j=1}^{n} |a_{ij}|$$

   for $i = 1, 2, \ldots, m$. Then use BLAS (with the `norm` command) to calculate the s_i. Compare the efficiency of these two algorithms.

2. Given 2 matrices A and B of size $n \times n$. Write a program which calculates AB. Compare the effectiveness of your program with the command `A*B`.

Exercise 4 (Implementation of the LU decomposition)**.**

1. Implement the LUP decomposition of a square matrix using the partial pivoting strategy.
2. Test your implementation on concrete examples.
3. Compare your implementation with the one of MATLAB (command `lu`).

Remark: for time measurements, you will use the commands `tic` and `toc` from MATLAB

```
tic;
program;
toc
```