Practical Work #1

The first session is devoted to the exploration of the software MATLAB. In practical works, we will study how to use MATLAB to compute approximate solutions of problem that we may not solve analytically.

1 Introduction to MATLAB

1.1 Standard functions

Open the file *TP1/part1/BasicOperations.m* and type the different operations in the command window of MATLAB. By Basic operations we means command that allow to create and use scalars, vectors, matrix and loops. Lines that begin by % are comments and are not executed by MATLAB. Some information about those basic commands are listed below.

Main tools for algorithms are loops

```
for k=1:10     while k<10     if k<10     else/elseif k>20
```

and must end with an `end`. Logical operations are

```
A==B (A equal to B)   ;  A~=B (A not equal to B)   ;  A<=B (A lower or equal to B)
```

```
A|B (A or B)   ;  A&B (A and B)
```

This is not mandatory to allocate memory for matrices but it is recommanded as it fastens the computation. To do so, you may use

```
A=zeros(m,n);
```

which sets A as a matrix with m rows and n columns. To refer to the entry at row i and column j, you may write `A(i,j)`. To fill a matrix by hand, use brackets:

```
A=[1 2 3;-5 7 1;0 1 2];
```

This yields

$$A = \begin{pmatrix} 1 & 2 & 3 \\ -5 & 7 & 1 \\ 0 & 1 & 2 \end{pmatrix}.$$

Standard operations on matrices, provided that A and B have compatible dimensions, read

```
A+B      A-B      A*B
```

Specific notations for the inverse are `A\B` and `A/B` which are equivalent to `inv(A)*B` and `A*inv(B)`. Notation `A'` corresponds to the transpose operation. The component-wise operations occur when a dot `.` precedes the operation symbol. For instance, if A and B are square matrices of order n, `A.*B` is a matrix or order n whose entries are $A_{ij}B_{ij}$. Likewise, if x is a vector of length n, `x.^ 2` is a vector whose ith-component is x_i^2.

Classical functions to create matrices are

```
zeros(m,n)      ones(m,n)      rand(m,n)      eye(m,n)
```

which correspond respectively to a matrix with m rows, n columns and 0-entries, 1-entries, random entries (with coefficients in $(0,1)$ and 1-entries on the diagonal / 0-entries elsewhere. Notice that if a single argument is specified, this yields a square matrix.

`diag(A)` is a vector containing the diagonal coefficients of matrix A while `diag(x)` is a diagonal matrix whose coefficients are the components of vector x. `tril(A)` is the lower triangular part of matrix A while `triu(A)` is the upper one (including the diagonal terms).

One must also mention the following functions: `det(A)` is the determinant of the square matrix A, `size(A)` returns the number of rows and columns of A, `length(x)` returns the number of rows of the column vector x, `eig(A)` is a vector made of the eigenvalues of A, `sum(x)` is the sum of all the components of x, `norm(x,p)` is the norm of x in ℓ_p (by default, $p = 2$), `max(p)` returns the largest entry of x, ...

The main function to plot figures is

```
plot(x,y,'options')
```

where x and y are two vectors with the same size. The options include the color of the lines, their thickness, their style (dashed, dotted, circle, cross, ...). To improve the quality of the figure, refer to functions such as `title, legend, xlabel, axis,` ...By default, any visualization function appears in `Figure(1)` and erases previous curves, to avoid this inconvenience, it is possible to show all curves on the same figure (`hold`) or on different figures by specifying `Figure(2)`, ...

To get information about a specific function `func`, type in the command window

```
>> help func
```

or open the product help.

1.2 Scripts and functions

There are two types of files when using MATLAB, namely *scripts* which are the location of the main algorithms and *functions* for parts of the algorithms which are called many times. Both types are implemented in the *editor*. Function file `func_name.m` look like

```
function [output variables] = func_name(input variables)
```

Inside the function, outputs must be computed thanks to the values of inputs.

Script files must contain the values of main parameters of the algorithm (such as the dimension of the problem, the bounds of the interval, ...) and the calls to functions. They must begin by

```
clear all; close all;
```

in order to clear all variables (which may remain from a previous computation) and close all figures.

To include a comment in a file (which is good for other readers!!!), use the % symbol. All characters occuring after will not be interpreted.
Each operating line must end with a `;`. Otherwise, the result of the line appears in the command window which may not be convenient.

To launch a programme, it suffices to type the name of the script file in the command window

```
>> script_name
```

1.3 An example of script

To execute a sequence of commands without the need to type them line by line in the command window of MATLAB, we may create scripts. Open the file *TP1/part1/scriptexample*. This script use functions defined in other files, the function *functionexample1* for instance is defined in *TP1/part1/functionexample1.m* (only one function may be defined per file).

To execute the scipt type *scriptexample* (the scipt name) in the command window of MATLAB, your current directory need to contain your script (and the functions that are called by your script). In MATLAB we manipulate vector of finite size, the x-axis is represented by a finite number of value defined by nx here. Change the definition of nx : $nx = 5$, $nx = 100$ or $nx = 1000$, launch the script and observe the influence of the choice of nx on figures.

2 A first program to compute approximations of e

The goal of this exercise is to compute approximations of e. We consider the two following sequences

$$sequence\ 1\ :\ u_n = \left(1+\frac{1}{n}\right)^n,\ n \geq 1$$

$$sequence\ 2\ :\ v_n = exp\left(\frac{1}{2n}\right)\left(1+\frac{1}{n}\right)^n,\ n \geq 1$$

Those two sequences converge toward e when n goes to infinity, we will use this fact to compute approximation of e.

1. Plot the N first terms of those sequences and compare them to e.

   To do so modify files *TP1/part2/sequence1.m* and *TP1/part2/sequence2.m* according to the definitions above. Then execute the script TP1/part2/mainsciptpart2 with different value of N what do you observe ?

2. Compute approximations of e with those sequences.

   We approximate e by u_{n1} (or v_{n2}). The question is how to choose $n1$ (or $n2$). To have an accurate approximation they should be chosen large. Nonetheless most methods that we will use in practical works are based on recursive sequences, so that large value of $n1$ leads to a long time of computation. We need a criterion to assess that we are close enough to the exact solution e up to an error ϵ (number of significant digits).

   criterion 1 : we choose $n1$ such that $|u_{n1} - e| \leq \epsilon$

   criterion 2 : we choose $n1$ such that $|u_{n1+1} - u_{n1}| \leq \epsilon$

   Use those two criteria and different values of ϵ, compare approximations obtained with sequence 1 and sequence 2 what do you observe ?

   Remark : the second criterion do not require to know the exact solution e.

   Remark 2 : In MATLAB scalars have only a finite number of digits, the smallest value of ϵ that we may chose is the smallest scalar represented in *Matlab* and called *machine epsilon* of order 10^{-16} for *Matlab*.

   Remark 3 : type *format long* in the command window to show scalar with 16 digits. (By default *format short* show scalar with only 5 digits)

3. report these data in a table for $\epsilon \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$

3 Taylor series expansions

The goal of this exercise is to investigate the accuracy of Taylor series expansions. Let us set

$$f(x) = \exp x$$

and

$$\varphi_1(x) = 1 + x, \qquad \varphi_2(x) = 1 + x + \frac{x^2}{2}, \qquad \varphi_3(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6}, \qquad \varphi_4(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}.$$

For a given $M > 0$, $E_k(M)$ is the global error between f and φ_k, *i.e.*

$$E_k(M) = \max_{x \in (-M,M)} |f(x) - \varphi_k(x)|.$$

1. Plot the functions on the interval $(-5,5)$.
2. Plot the errors for $M \in (0,5)$ on a second figure.
3. Given a threshold $\varepsilon > 0$, determine M_k such that $E_k(M_k) = \varepsilon$. Which function φ_k is the most accurate approximation of f?

4 The Syracusa conjecture

The Syracusa sequence is defined by

$$u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{if } u_n \text{ is even,} \\ 3u_n + 1 & \text{if } u_n \text{ is odd.} \end{cases}$$

This sequence is parametrized by its first term u_0. The conjecture is that for any $u_0 \in \mathbb{R}$, there exists $N_0 \in \mathbb{Z}_+$ such that $u_{N_0} = 1$.

1. Write down a function which computes the 50 first terms of the sequence corresponding to a given u_0.
2. Plot the 50 first terms of the sequences characterized by $u_0 \in \{1,2,\ldots,15\}$. What do you observe?
3. Same question for $u_0 \in \{27,97,703\}$.
4. Modify your function in order to yield the maximal value of the sequence and the smallest index for which $u_n = 1$.
5. Report these data in a table for $u_0 \in \{1,2,\ldots,30\}$.