

Practical Work #1

The first session is devoted to the exploration of the software MATLAB. It is dedicated to numerical analysis and in particular to linear algebra.

1 Introduction to MATLAB

1.1 Files

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.

1.2 Standard functions

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 recommended 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 $A \setminus B$ and A / B which are equivalent to $\text{inv}(A) * B$ and $A * \text{inv}(B)$. 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 of order n whose entries are $A_{ij}B_{ij}$. Likewise, if x is a vector of length n , $x.^2$ is a vector whose i th-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 , ...

1.3 Figures

The main function 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), ...

1.4 Use

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

```
>> script_name
```

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

```
>> help func
```

or open the product help.

To include a comment in a file (which is good for other readers!!!), use the `%` symbol. All characters occurring 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.

2 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

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

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 ?

3 The Syracuse conjecture

The Syracuse 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 $u_0 \in \{1, 2, \dots, 15\}$. 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 $u_0 \in \{1, 2, \dots, 30\}$.

4 The Cholesky decomposition (exercise 4)

We recall that any symmetric positive-definite matrix can be written as $B^T B$ where B is upper triangular with nonnegative diagonal entries.

1. Let C be an invertible matrix. What can you say about the matrix $C^T C$?
2. Implement the algorithm taking care of the hypotheses of the statement.
3. Work out your programme for relevant matrices.
4. Implement the resolution of upper and lower triangular linear systems.
5. Write out a script solving $Ax = b$ for some given $b \in \mathbb{R}^d$ and $A \in \mathcal{M}_d(\mathbb{R})$ symmetric positive-definite.