UNIVERSITY OF BOLOGNA - FACULTY OF ENGINEERING

INTERNATIONAL MASTER COURSE IN CIVIL ENGINEERING - A.Y. 2012/2013

INTRODUCTION TO NUMERICAL METHODS

LAB 1: MATLAB BASICS

a)

Working directly in the MATLAB Command Window, create the two matrices $A = \begin{bmatrix} 1 & -6 \\ 6 & 18 \end{bmatrix}$, $B = \begin{bmatrix} -1 & 2 \\ 10 & 6 \end{bmatrix}$, the column vector $v = [1 \;\; 2]^T$ and the scalar $k = 2$, check the existence (and the value, size, bytes and class) of the four created variables in the MATLAB Workspace, then compute:

(ADDITION)	A + k	A + v	A + B	k + A	v + A	B + A
(SUBTRACTION)	A - k	A - v	A – B	k – A	v – A	B - A
(MULTIPLICATION)	A * k	A * v	A * B	k * A	v * A	B * A
	A .* k	A .* v	A .* B	k .* A	v .* A	B .* A
(RIGHT and LEFT DIVISION)	A / k	A / v	A / B	A \ k	A \ v	A \ B
	A ./ k	A ./ v	A ./ B	A .\ k	A .\ v	A .\ B
(EXPONENTIATION)	v ^ k	A ^ k				
	v .^ k	A .^ k				
(CONCATENATION)	A || k	A || v	A || B	A = k	A = v	A = B
(TRANSPOSITION)	k^T	v^T	A^T			

PAY ATTENTION!!! The operations are not all computable: why?

b)

Working directly in the MATLAB Command Window, create the 4x4 matrix (square matrix of order 4):

$$A = \begin{bmatrix} 2 & 6 & -4 & 12 \\ -5 & -9 & 10 & 2 \\ -6 & 12 & 8 & 16 \\ 15 & -3 & 12 & 2 \end{bmatrix}$$

then:

a) Create a vector v formed with the elements (entries) of the second row of A;

b) Compute and store in the variable s the sum of the elements of v, after these have been divided (element by element) by the elements of the first column of A;

c) Create a matrix B formed with all the elements between the second and the fourth column of A;

d) Create a matrix C formed with all the elements of the first two rows and the last three columns of A;

e) Create A^t, the transpose of A;

f) Create a vector v formed with the minimums of the elements in each column of A^t ;

g) Create a vector v formed with the maximums of the elements in each row of A^t ;

h) Create a vector v formed with the sums of the elements in each row of A^t ;

i) Compute the sum of all the elements of A ;

c)

Create: 1) a MATLAB **script**; 2) a MATLAB **function**; that, given a natural number n , compute the sum S of the first n natural numbers by the formula $S = \sum_{k=1}^{n} k = \frac{n(n+1)}{2}$.

Use the script and the function to compute the sum S for different values of n … Which is the difference between **script** and **function**?

d)

An object is thrown vertically upwards with an initial velocity v_0 and reaches a height h in a time t, where $h = v_0 t - \frac{gt^2}{2}$. Write a MATLAB function that calculates (and provides as output) the time t required for reaching a given height h for a given value of v_0. The function takes as inputs h, v_0 and g.
Test the function in the cases: a) h = 170 m, v_0 = 60 m/sec e g = 9.81 m/s^2
b) h = 200 m, v_0 = 60 m/sec e g = 9.81 m/s^2 : What can you conclude?

[case a) 4.4580 - 7.7744 ; case b) complex result]

e)

After exploring (in the MATLAB Help) the following commands for the creation of special matrices:

▪ **zeros**	zero matrix
▪ **ones**	matrix of ones
▪ **eye**	identity matrix
▪ **diag**	diagonal matrix
▪ **tril**	lower triangular matrix
▪ **triu**	upper triangular matrix
▪ **hilb**	Hilbert matrix
▪ **vander**	Vandermonde matrix
▪ **rand**	matrix of random numbers
▪ **magic**	matrix with equal row, column and diagonal sums

then write a MATLAB function that takes as input a natural number n and provides as output a square n x n matrix $A = \{a_{i,j}\}_{i,j=1,\ldots,n}$ defined as follows:

$$a_{i,j} = \begin{cases} 2|i-j|+1 & \text{if } |i-j| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

What can you say about the matrix?

f)

f1) After exploring (in the MATLAB Help) the potentialities of the command **plot** (in particular, how to plot with given **color**, **style**, **marker**), write a MATLAB script that plots in the **same graph** the four functions:

$$y = \ln(x)\cos(x) \qquad y = e^{x}\log_{10}\left(\frac{x}{2}\right) \qquad y = \frac{\sin(2x)}{x} \qquad y = x^{3} - 4x \qquad \text{with } x \in [-2, 2]$$

using different types of line and a legend to distinguish the curves, axes labels, a title and the grid.

f2) Create a script that plots the functions in **four different graphs** in the same figure (command **subplot**).
f3) Create a script that plots the functions in **four different figures**.

PAY ATTENTION TO THE DOMAIN for the third function!!!

3D GRAPHICS

MATLAB provides several functions that allow creating different types of three-dimensional plots:

- Line plots
- Surface plots
- Contour plots

Line plots

Lines in three-dimensional space can be plotted by using the MATLAB function **plot3**; the syntax is:

plot3(x,y,z)

where *x, y, z* are vectors containing the *(x,y,z)* coordinates of a set of 3-D points defining the line.

Example:
```
t = [0:0.1:10*pi] ;
x = exp(-t/20) .* cos(t);   y = exp(-t/20) .* sin(t);   z = t;
plot3(x,y,z);
xlabel('x');  ylabel('y');  zlabel('z');
```

Surface plots

A generic function of two variables *z = f(x,y)* represents a surface when plotted in three-dimensional space. The MATLAB functions **mesh** and **surf** allow to generate, respectively, a *wireframe* and a *solid* surface plot. Before being used, both functions require a grid of points in the *xy* plane (function *f* domain) to be created by means of the MATLAB function **meshgrid**, so that the *z = f(x,y)* function is then calculated in correspondence of the grid points. The syntax of the function **meshgrid** is:

[X,Y] = meshgrid(x,y)

After the grid has been generated, the surface plot can be created by the **mesh** (wireframe surface) or the **surf** (solid surface) function. The syntax is:

mesh(X,Y,Z) **surf(X,Y,Z)**

Examples:

a) using the MATLAB function **mesh** to visualize the two variables function $z = f(x,y) = sin(x)cos(y)$:

```
x = linspace(0,2*pi,50);
y = linspace(0,pi,50);
[X,Y] = meshgrid(x,y);
Z = sin(X).*cos(Y);
mesh(X,Y,Z);
xlabel('x'); ylabel('y'); zlabel('z');
grid on;
```

b) using the Matlab function **surf** to visualize the two variables function $z = f(x,y) = exp(-(x^2+y^2) / 3)$:

```
[X,Y] = meshgrid(-3:.2:3 , -2:.2:2);
Z = exp( -(X.^2+Y.^2)/3 );
surf(X,Y,Z);
xlabel('x'); ylabel('y'); zlabel('z');
grid on;
```

Contour plots

In topography, three-dimensional earth surface is generally represented by two-dimensional maps, called *contour maps*. These maps contain *contour lines* (or *isolines*), that are lines joining points of equal elevation above a given level, such as the mean sea level. More generally, the shape of two variables functions can be better understood by means of contour plots. In MATLAB, these plots can be created by using the function **contour**. The syntax is:

contour(X,Y,Z)

The usage of this function is similar to that of **mesh** and **surf**, in the sense that a grid of points has to be preliminarily created by means of the MATLAB function **meshgrid**.

Example:
```
[X,Y] = meshgrid(-3:.2:3 , -2:.2:2);
Z = exp( -(X.^2+Y.^2)/3 );
contour(X,Y,Z);
xlabel('x'); ylabel('y'); zlabel('z');
grid on;
```

Refer to the MATLAB Help to gather further information on functions that generate three-dimensional graphs (*doc graph3d*)

- — - — - — - — - — - — - — - — -

g)

After exploring (in the MATLAB Help) the potentialities of the command **plot3**, write a MATLAB script that plots the 3-dimensional curve *f(t)* defined by the following parametric equations:

$$f(t): \begin{cases} x = a \cdot t \cdot \cos(t) \\ y = a \cdot t \cdot sen(t) \\ z = b \cdot t \end{cases}$$

where a and b are real constants.
Create a plot of the curve *f(t)* for each of the following four cases:

a) $b = 0.1$ set $a = 1$ and $-10\,\pi \le t \le 10\,\pi$

b) $b = 0.5$
c) $b = -0.1$
d) $b = -5$

Compare the four obtained plots by means of the command **subplot**.

h)

Consider the following function of two variables:

$$z = f(x,y) = \frac{x + y}{1 + x^2 + y^2}$$

Represent the function, that is the surface **(**hyperbolic paraboloid**),** in the domain $-1 \le x \le 1$, $-1 \le y \le 1$, using the Matlab commands seen before (*mesh, surf, contour, meshc, surfc* …) and trying different graphics options (*colormap, shading, axis*…).

i)

Create the contour and surface plots for the following two variables function:

$$z = f(x,y) = -x^2 + 2xy + 3y^2$$

This function has a critical point at *(x=0 , y=0)*, but the point corresponds neither to a minimum nor to a maximum (inflectional point): which isolines correspond to this point?

CREATION OF MOVIES

MATLAB offers the possibility to store a series of plots and then to show them one after the other so as to obtain a real animation. The basic idea is to store tidily the plots within the columns of a proper matricial data structure. The MATLAB function **movie**, then, allows to show the animation.
Firstly, the plots to be shown have to be generated (usually through a *for* loop) and stored, by means of the MATLAB command **getframe**, in the columns of the matrix. It might be useful to set the axes limits through the command **axis([x_{min} x_{max} y_{min} y_{max}])**.

Example:
```
x=linspace(0,pi);
for k=1:5
   plot(x,sin(k*x));
   grid;
   title('Sin(kx), k=1,2,3,4,5');
   axis([0,pi,0,1]);
   xlabel('x');
   ylabel('y');
   M(:,k)=getframe; % the current k^th plot is stored in the k^th column
end
```

Then, the movie is simply shown by the command:

movie(M)

It is worth noticing (in the *Workspace* panel) the large amount of memory required to store the movie (matrix *M*)!

Refer to the Matlab Help to gather further information on functions that generate movies

j)

Create a movie for the one variable function:

$$y = f(x; a) = ax^2 - ax$$

with $a = 0,1,2,3,4,5,6,7,8,9$ in the interval $x \in [-0.5, 1.5]$

k)

Create a movie for the two variables function:

$$z = f(x, y; k) = \sin(k\pi x)\cos(k\pi y)$$

with $k = 0,1,2,3,4,5$ in the interval $(x, y) \in [0, 1] \times [0, 1]$