Problem set 1a: MATLAB Fundamentals and Graphics

Handed out: Thursday, September 15, 2016

Due: 23:55pm, Thursday, September 22, 2016

You must hand in two files:

- One file, named `your_name_E1a.pdf`, with the solutions to the exercises in this set, following the template available in the virtual campus. For every exercise, you must explain how you solved it, the necessary MATLAB code, the results obtained (run transcripts or plots), and some final comments (if any).
- The second file, named `your_name_E1a.zip`, must contain all the MATLAB code you used to solve the exercises, organized in one or more text (`.m`) files, to allow the teacher to check your solution.

Please, make sure that the code in the pdf file is **exactly the same** as in the zip file. Remember that all the MATLAB code is supposed to be **entirely yours**. Otherwise, you must clearly specify which parts are not, and properly attribute them to the source (names of collaborators, links to webpages, book references, ...). Read the Course Information document, also available in the virtual campus, for more details on this subject.

Exercise 1. Variables and operators.

(1) Compute the factorial of an integer number n without using any loop and without resorting to the **`factorial`** command (use function **`prod`**).

(2) The Pascal triangle[1] is a simple way to compute recursively the combinatorial numbers of increasing order exploiting that $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$. Starting with a vector of value $\begin{pmatrix}1 & 1\end{pmatrix}$ (that would correspond to $\binom{1}{k}$ for $k = 0, 1$), write the code that each time it is called it stores in the original vector the next row of the Pascal triangle, so the first time it is called it would produce $\begin{pmatrix}1 & 2 & 1\end{pmatrix}$, the second time $\begin{pmatrix}1 & 3 & 3 & 1\end{pmatrix}$ and so on.

(3) We want to test how many random integers that have been randomly generated in the range [4,100] satisfy simultaneously the following conditions: are multiples of 3 and are larger than 30. Given a length n (whose value is set at the beginning of your code), generate a set of n integers in the desired range (**`randi`**) and use the logical operators to test the desired conditions (operators **`>`**, **`==`** or **`~=`**, **`&`** and function **`rem`**) and find the amount of integers that satisfy them (**`find`**, **`length`**). Run your program for large values of n and verify that the amount of integers converges to the expected value.

[1] See https://en.wikipedia.org/wiki/Pascal's_triangle.

Exercise 2. Vector and matrix operations.

(1) Given a generic matrix $\mathbf{A}$ write the code that generates the matrix $\mathbf{R}$ that has the same dimensions but whose columns appear in the reversed order (**`size`**, operator "**`:`**"). Double check that your code works for matrices of different sizes and also for row vectors or column vectors (particular cases of matrices with one dimension equal to 1). Do not use any loop in your code.

(2) Generate the following matrices and vectors (functions **`zeros`**, **`ones`** and **`eye`**):

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{C} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

(3) Using concatenation, scalar by matrix multiplication, matrix transposition (`.'`) and the operator "**`:`**", build the following matrix employing only the matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ defined in the previous section:

$$\mathbf{M} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ -3 & 1 & 5 & 0 & 0 \\ -3 & 1 & 0 & 5 & 0 \\ -3 & 1 & 0 & 0 & 5 \end{pmatrix}$$

(4) Given a set of n integers $a_1, a_2, ...a_n$ generate a matrix whose (i, j)-th coefficient corresponds to the difference $a_i - a_j$ (**`meshgrid`**[2]).

(5) The Vandermonde matrix is a matrix with the terms of a geometric progression in each row, i.e. a $n \times m$ matrix

$$\mathbf{V} = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{m-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{m-1} \\ 1 & \alpha_3 & \alpha_3^2 & \cdots & \alpha_3^{m-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_n & \alpha_n^2 & \cdots & \alpha_n^{m-1} \end{pmatrix}$$

Write the code that given the values of m, n and $\alpha_1, ..., \alpha_n$ generates the corresponding Vandermonde matrix. Write this code in two lines using command **`meshgrid`** and operator operator "**`.^`**" or write it in one single line using command **`repmat`** and operator "**`.^`**" .

(6) Use the code in the previous sections to generate a square Vandermonde matrix (choose $m = n$) and compute its determinant and its eigenvalues (**`det`**, **`eig`**) . Use the code in the previous sections to check that the determinant verifies:

$$det\left(\mathbf{V}\right) = \prod_{1 \le i < j \le n} \left(\alpha_i - \alpha_j\right)$$

$$det\left(\mathbf{V}\right) = \prod_{1 \le i \le n} \lambda_i$$

[2]The generation of a matrix that consists on the repetition of a column vector can also be obtained multiplying it by row vector $\begin{pmatrix} 1 & \ldots & 1 \end{pmatrix}$. The generation of a matrix that consist on the repetition of a row vector can also be obtained multiplying the column vector $\begin{pmatrix} 1 & \ldots & 1 \end{pmatrix}^T$ by the desired row vector. Furthermore, both can also be obtained using command **`repmat`**.

being λ_i the matrix eigenvalues. To verify the first identity you can use one **`for`** loop, to verify the second one you cannot use any loop (**`prod`**).

(7) Solve the following system of linear equations by using function **`inv`**. Then rewrite the code now using operator "****".

$$\begin{cases} 2x_1 + x_2 - 2x_3 = 4 \\ 4x_1 + 2x_2 - 3x_3 = 9 \\ -2x_1 + x_3 = -3 \end{cases}$$

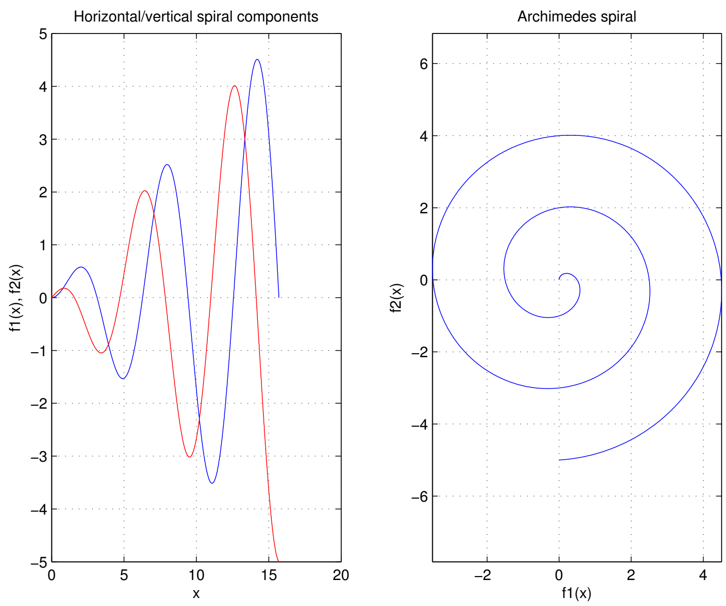
Exercise 3. Basic plots.

We want to generate a code to plot the Archimedes spiral.

(1) First, generate a vector **x** containing values between 0 and 5π and equally spaced by $\pi/200$ (operator "**`:`**" or function **`linspace`**). Next, plot the function $f_1(x) = \frac{x}{\pi}\sin x$ in that interval (functions **`sin`** and **`plot`**), and add labels to the plot with functions **`xlabel`**, **`ylabel`** and **`title`**.

(2) Add to the previous axes the plot of $f_2(x) = \frac{x}{\pi}\cos x$ in the same interval in red color (functions **`cos`**, **`plot`** and **`hold`**).

(3) Plot now in new axes (**`figure`**) $f_2(x)$ (vertical axis) as a function of $f_1(x)$ (horizontal axis) for the same range of values of x as before. Modify your software so that both plots are stacked horizontally in the same figure (**`subplot`**). Add a grid and scale the right hand plot so that both axis are depicted using the same scale (**`grid on`**,**`axis equal`**), so the final plot looks as shown in the figure.



Exercise 4. Computations with polynomials.

We want to plot the frequency response (magnitude and angle) of a FIR system whose zeros are located in $0.1 \pm 0.3j$, 0.5 and -1.

(1) Obtain the polynomial associated to the transfer function $H(z)$ that has the zeros in the desired locations[3] (**`poly`**):

[3]Remember that $\sqrt{-1}$ is written in Matlab as **`1i`**.

$$H(z) = \prod_i (z - z_i)$$

(2) Define a vector of equispaced frequencies in the range $[-\pi, \pi]$ and evaluate the magnitude of the frequency response of the system in the points of the unit circle corresponding tothese frequencies $z = e^{j\omega}$ (**`exp`**, **`polyval`**, **`abs`**).

(3) Plot the magnitude of the frequency response in dB's (**`plot`**, **`log10`**, **`axis`**, **`xlabel`**, **`ylabel`**).