An introduction to Matlab
AIV M1 bootcamp

Antoine Frénoy
antoine.frenoy@inserm.fr

September 2014

The course material is available on `http://perso.crans.org/frenoy/matlab2014.html`.

Attendance

Attendance to the class is mandatory for all students. If you already know Matlab and find these exercises too easy, ask us and we will happy to find more complicated exercises for you. If you have to skip a class we expect you to catch up before the following class. Please arrive on time.

Evaluation

The goal of this class is to give you tools to survive this semester. We will not grade you, and the only goal of tests is to evaluate the efficiency of our teaching. However you should not underestimate the importance of this class: not being comfortable with Matlab would seriously compromise your chances of succeeding your M1.

Prerequisite

Typing. Nothing else.
You do not need any prior programming nor biological knowledge. A part of the exercises relies on real published dataset, but we choose simple enough problems so biological knowledge is never the bottleneck.

Documentation

Because we are lazy, the first thing we are going to teach you is to use Matlab documentation. For all the following exercises, we thus provide the name of the Matlab functions you should use so you can efficiently use this documentation. This will allow you to go at your own speed.

Chapter 1

Exercises

A. The very beginning

1. Start Matlab.
2. Calculate $32.4/23.145 * 8.3$, store the result in the variable `x`.
3. Type `doc sin` in the command window.
4. Calculate $sin(\frac{4\times\Pi}{5})$, store the result in the variable `y`.
5. Write a Matlab script to swap the content of the variables `x` and `y`.

B. Functions and very simple algorithms

1. Write a matlab function `sq` that takes one number and returns its square.
2. Write a matlab function `m2sum` that takes two numbers and returns their sum. *Documentation:* **function**.
3. Write a matlab function that takes one integer n and plots n times "hello". *Documentation:* **for, disp**.
4. Write a matlab script that plots the squares of all the integers between 0 and 100.
5. Write a matlab function `sequence0` that takes an integer `n` and returns the n^{th} element of the sequence (u_n) defined by:
   - $u_0 = 1$
   - $u_{n+1} = sin(u_n) + 1$
6. Write a matlab function `sequence1` that takes an integer `n` and returns the n^{th} element of the sequence (u_n) defined by:
   - $u_0 = 1$
   - $u_{n+1} = u_n + 3 * n - 4$
7. Write a matlab function `sequence2` that does the same thing for the sequence defined by:
   - $u_0 = 17$
   - $u_{n+1} = u_n/2$ if u_n is even, $3 * u_n + 1$ otherwise.

Documentation: **if, mod**.

8. Write a matlab function `pra0` that takes seven numbers and returns their sum if one of them is zero, their product otherwise.
9. Write a matlab function `pra1` that takes three numbers and returns the "middle" one (the one that is nor the highest nor the lowest).
10. Write a matlab function `pra2` that takes three numbers and returns the product of the two highest numbers.
11. Write a matlab function `pra3` that takes four numbers and returns `1` if at least two of the numbers are equals, `0` otherwise.

C. More advanced algorithms

1. Write a matlab function `mgcd` that takes two integers and returns their greater common divider. *Documentation:* **while**.
2. Write a matlab function `misprime` that takes a positive integer and tells whether it is a prime number. *Documentation:* **sqrt, floor, break**.
3. Calculate the sum of the first 500 prime integers.

D. Algorithmic on vectors

1. Type in Matlab the following commands and understand what they do.

```
x=[2,7,-3,1,2,5]
x(2)
x(2)*2
x(2)=3
x(2)=[]
x*4
x>3
x==2
```

2. Write a matlab function `mini` that takes a vector and returns its lowest element. *Documentation:* **for, Inf**.
3. Write a matlab function `mini2` that takes a vector and returns its lowest element and the index of this element.
4. Type in Matlab the following commands and understand what they do.

```
x=[2,7,-3,1,5,2,9]
x([3 1])
x(2:5)
x(7:-1:1)
```

5. Write a matlab function `reverse` that takes a vector and reverses it (the first element becomes the last one and so on). *Documentation:* **size, length, zeros**.
6. Write a matlab function `incsort` that takes a vector and sorts it in ascending order.

E. Loading and writing data

1. Load the variable `m` contained in the file `load1.mat`. Sum the elements of `m` and put the result in a variable `s` that you will store in the file `save1.mat`. *Documentation:* **load, sum, save**.
2. Load the file `data1.xls` and sum the third line of numerical values. *Documentation:* **xlsread**.

F. Working with matrices

1. Type in Matlab the following commands and understand what they do.

```
m=[1,7,2; 8,9,2; 5,6,1; 2,1,4]
m(3,1)
m(3,:)
m(1:2,3)
m*3
m>3
```

2. Load the matrix `A` from the file `a.mat`.
3. Multiply element by element the third and fifth column of `A`, *ie* create a colum vector `z` such that $\forall i, z(i) = A(i,3) * A(i,5)$. *Documentation:* **arith**.
4. Create a column vector that contains the sum of each line of `A`.
5. Sum all the elements of `A` that belong to an even column.
6. Count the number of elements of `A` that are greater than 2. *Documentation:* **find, numel**.
7. Sum the elements of `A` that are greater than 2 and smaller than 50.

G. Algorithmic again

Load the file `practice.mat`, and read in the help the section **MATLAB/Language Fundamentals/Matrices and Arrays/Indexing/Matrix Indexing**.

1. How many numbers appear several times in `V1`? Only count once each non unique element. *Documentation:* **unique**.
2. Sum the numbers that appear exactly once on each column of `B`.
3. Find the smallest of the elements of `C` that appear only on one line (they can appear several times but not on different lines).
4. Find the elements of `D` that are such that the sum of the column they belong to is equal to the sum of the line they belong to.
5. Calculate the product of the elements of `E` that are equal to the number of their column.
6. In matrix E, what is the number of the column that contains the highest number of even elements? *Documentation:* **max**.
7. How many elements of E that are strictly greater than 26?
8. Calculate the sum of the elements of E that are strictly greater than 26.
9. What is the greatest element of E that is a multiple of 3 but not a multiple of 5?

H. And again

1. Reorganize the columns of E to get a matrix F with the same columns in a different order such that the first row of F is in increasing order (two elements that belonged to the same column of E must still belong to the same column of F, you just move the columns without "breaking" them). *Documentation:* **sort**.

2. Write a function that takes a vector and tells you whether there are in this vector two consecutive elements (next to each other in the vector) that would still be consecutive after sorting the vector in increasing order. If that is simpler you can consider that the input vector contains only once each value. For example if `X=[1, 7, 3, 5, 2, 6]`, when sorting this vector `3` and `5` will still be following each other (in the same order) so the answer if yes. If `X=[3, 5, 2, 4, 7, 6]`, the answer is no.

I. Text printing and parsing

1. Write a matlab function `nicetext` that takes a matrix and prints the highest element of each column indicating the number of the column:
   ```
   The highest element of column 1 is 15
   The highest element of column 2 is 7
   ...
   ```
   Documentation: **fprintf**.

2. The file `samples.txt` contains a measure for each sample identified by a letter and a digit (for example `A1`). We consider that samples identified with the same letter (but different digits) belong to the same group. Print the average value of the measure for each group. *Documentation:* **fopen, fscanf, mean**.

J. Distances

1. The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. Write a matlab function `hamming` that takes two vectors and returns their hamming distance. *Documentation:* **assert**.

2. **(difficult)** The Levenshtein distance between two strings is defined as the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character. Write a matlab function `levenshtein` that takes two vectors and returns their Levenshtein distance. *Documentation:* **return**.

K. Plotting 2D curves

1. Plot the graph of the function $x \to 3 * x^2 - 6 * x + 1$ over [0,10]. *Documentation:* **plot**.

2. Plot the graphs of the functions sin and cos^2 over [0,5] with different colors. Add a legend to identify each function. *Documentation:* **legend**.

L. Randomness and plotting histograms

1. Create a 10000 by 5 matrix `A` filled with uniformly distributed random numbers between 0 and 10. Create a vector `V` containing the sums of each line of `A`. Plot the distribution of `V`. *Documentation:* **rand, hist**.

2. Create a 100 by 1 matrix `B` filled with normally distributed random numbers with mean 2 and standard deviation 1.5. Plot the distribution of `B`. *Documentation:* **randn**.

M. Plotting points

1. Load the file `points.mat`. It contains a list `l` of 2D points. Plot these points using "x" as marker on a 2D graph. *Documentation:* **scatter**.
2. Zoom to plot only $x \in [0, 5]$ and $y \in [-1, 1]$. *Documentation:* **xlim, ylim**.
3. Name the x-axis "time" and the y-axis "sales". *Documentation:* **xlabel, ylabel**.
4. The file `points.mat` also contains standard error for each point in variable `err`. Plot the same graph as in q. 1 with error bars. *Documentation:* **errorbar**.

N. Advanced plotting

1. Let's consider the function $f(x, y) = sin(6 \times x) + y$. Plot f over $[0, 1] \times [0, 1]$ as a 3D graph, *ie* $z = f(x, y)$. *Documentation:* **surf**.
2. Plot the same function as a color 2D graph, *ie* (x, y) is colored according to the value of $f(x, y)$. Add a color scale. *Documentation:* **pcolor, colorbar**.
3. Put the last two graphs on the same figure. *Documentation:* **subplot**.
4. Export this figure as a pdf file named `f1.pdf`. *Documentation:* **print**.

O. Basic statistics

1. We randomly picked 20 men and 15 women among the world population, and recorded their size in the file `humansizes.mat`. Plot the histogram of sizes using different colors for men and women. Perform a statistical test that will estimate whether men are statistically taller than women. *Documentation:* **kstest, ttest2**.
2. We did the same for Northern Gannet, with 30 females and 28 males, and saved the results in the file `birdsizes.mat`. Perform a statistical test that will estimate whether there is a size difference between males and females.

P. More practice in parsing, analyzing and plotting data

1. The file `plants.txt` contains a list of plants identified by their location on a field (two coordinates between 0 and 100) and their types (A, B or C). More precisely, each line contains in the order x coordinate (decimal), y coordinate (decimal), and type (1, 2 or 3 for A, B or C).
   - Load the data. *Documentation:* **importdata**.
   - Make a 2D representation of the field, plotting each plant with different markers and colors. *Documentation:* **scatter, plot**.
   - Divide the space in a 50×50 grid and make three color plots, each one indicating the density of one type of plant in each grid location. *Documentation:* **pcolor, colormap**.
   - Find a way to represent both densities of plant A, plant B and plant C on the same color plot. *Documentation:* **reshape, image**.

- Plot the distribution of the distances between two random plants of type A.
- Same question for plant B and plant C.
- Compare the 2 previous distributions with what you would get for $10,000$ plants randomly positioned on the field (each coordinate coming from a uniform distribution between 0 and 100).
- Do you think there is an interference between plant A and plant B? Make graphs to support your hypothesis.

2. The file `trees.txt` contains a list of trees with their height, a number quantifying the amount of water they get, a number quantifying their exposure to sun, and a number quantifying their exposure to wind.

   - Load the data. *Documentation:* **fopen, fscanf**.
   - Plot the distribution of the heights of the trees.
   - Plot with two different colors on the same graph the distribution of the heights of the 20% trees that get the most water and of the 20% trees that get the less water. *Documentation:* **bar, hist**.
   - How does water seem to affect tree height? Make an other graph that supports your hypothesis.
   - How does sun exposure seem to affect tree height? Make a graph that supports your hypothesis.
   - Do the "input" variables (water, sun exposure, wind exposure) seem independents? Make graphs to support your hypothesis.
   - What variable seems to have the smallest effect on tree's height? Make graphs that support your hypothesis.
   - Make a color plot of tree's height as a function of the two other variables. You will have to "discretize" the variable space.

Q. First order differential equations

Use Matlab to get a numerical solution (*ie* a 2D curve) of the following differential equations, and use your math skills to get a formal solution when you can. We will always solve over [0,1]. *Documentation:* **ode45**.

1. $y' = 2$ with initial condition $y(0) = -1$
2. $y' - 3 \times y = 2$ with initial condition $y(0) = 1$
3. $y'(x) + 2y(x) = e^{2x}$ with initial condition $y(0) = 0$
4. $\frac{dy}{dx} - 5y = e^{5x}$ with initial condition $y(0) = 1$
5. $\dot{x} - x = sin(t)$ with initial condition $x(0) = 1$
6. $(2 + t)\frac{dy}{dt}(t) = 2 - y(t)$ with initial condition $y(0) = 1$

R. Second order differential equations

Use matlab to get a numerical solution (*ie* a 2D curve) of the following differential equations, and use your math skills to get a formal solution when you can. We will always solve over [0,1].

1. $y'' - 2y' + 2y = xe^x$ with initial conditions $y(0) = 1$ and $y'(0) = -1$

2. $\ddot{y} - 4\dot{y} + 4y = 2(t-2)e^t$ with initial conditions $y(0) = 1$ and $\dot{y}(0) = -1$

3. $\frac{d^2x}{dt^2} - 4\frac{dx}{dt} + 13x = 10\cos(2t) + 25\sin(2t)$ with initial conditions $x(0) = 1$ and $\frac{dx}{dt}(0) = -1$

S. System of differential equations

1. Use matlab to solve the following system over [0,100] with parameters $r = 0.3$ and $m = 0.1$.

   - a' = r*sin(t)*a + m*a*b - m*a*c
   - b' = r*cos(t)*b + m*b*c - m*b*a
   - c' = r*sin(t/2)*c + m*c*a - m*c*b

Chapter 2

Project

A. Monty Hall

From Wikipedia:

> The Monty Hall problem is a famous problem in probability. The problem is based on a television game show from the United States, Let's Make a Deal. It is named for the host of this show, Monty Hall. In the problem, there are three doors. A car (prize of high value) is behind one door and goats (booby prizes of low value) behind the other two doors. First, the player chooses a door but does not open it. Then the host, who has knowledge of what is behind every door, opens a different door which he is certain has a goat behind it (opening either door with equal chances if the car is behind the player's door). Last, the host lets the player choose whether to keep what is behind the first door or to change choices to the third door (the one the host did not open). The rules of the problem are that the host has to open a door with a goat behind and has to let the player switch. The question is whether changing choices increases the chances of getting the car.

Questions

1. Do you think that changing choices increases the chances of getting the car?
2. Write Matlab code to simulate the problem. Was your guess right?
3. Can you explain the result you get with a mathematical proof?

B. Applied epidemiology

A class of 50 (N_{tot}) students is working hard to get their master degree. However they need some distractions, and each friday night they drink a lot, then randomly hook up (formation of 20 random couples, 10 random students being too drunk for this kind of activities). One student has mononucleosis, a highly infectious disease: an individual hooking up with one infected individual has $P_{inf} = 1/2$ chances of becoming infected. Luckily this disease is curable, and each week an infected individual has $P_{cur} = 1/10$ chances of recovery, after which we consider he is immunized and can not be infected again.

Modeling choices

- We take time $t = 1$ at initial situation (one infected student, nobody is immunized).
- At each time point we apply first the "recovery" process then the "infection" process.

Questions

1. Run 1000 simulations of the spread of the disease.
2. Draw the expected number of infected individuals as a function of (discrete) time, with standard error.
3. At time t1 = 6 weeks, draw the histogram of the number of infected individuals.
4. Draw the histogram of the time t_f at which the class becomes entirely disease free.
5. Same question for the histogram of time t_{half} at which half of the class is immunized. Exclude from this histogram the simulations in which the number of immunized students never reaches half of the class.
6. We now vary the initial number of infected students Ni. Draw the variance of the time t_f as a function of Ni.
7. Taking $Ni = 10$ and $N_{tot} = 500$, make a color plot of the total number of infected students as a function of P_{inf} and P_{cur}.

C. Social interaction network

In this exercise, we will use some data that come from Marcel Salathé's group. They measured contact networks in a high school to be able to predict the spread of infectious diseases and find the best vaccination strategy (you can see the publication: `http://www.pnas.org/content/early/2010/12/08/1009094108.full.pdf`). We will perform simple statistical analyses on these data.

Description of the raw data

Each node represents one person, identified by a unique number between 1 and 789. The folder `nodes` contains one file per node, with one line per interaction with another node. The lines are composed of the number of the other node and the time at which the interaction occurred. For example if the file `node-1` contains one line `2 1500` it means that persons identified by numbers 1 and 2 were interacting at time 1500 (arbitrary unit). All persons have one of the following role: student, teacher, staff, other. The file `roles` indicates the role of each person: 1 for student, 2 for teacher, 3 for staff, 4 for other.

Loading the data

We suggest that you read first the questions so you can choose the best way to store the data. Be careful that if A interacts with B at time t, then B interacts with A, however for some reason this reciprocal connection is not always reported.

Questions

1. Who interacted with the biggest number of other (unique) people during the day? What is his role?
2. Is there a statistical difference in total number of interactions (with unique people, meaning that we do not count several times several interactions between the same pair of persons) during the day between students and teachers?
3. Do students interact statistically more with other students than with teachers (still counting only interactions with unique people)?

D. Mortality from influenza

All the data we will use in this exercise come from `http://epidemiology.mit.edu/`. This is a real dataset used by epidemiology researchers, working mainly on cancer but having also other data. We will focus on mortality by influenza and try to produce simple figures from raw data.

Description of the raw data

The file `influenza.xls` contains data about mortality by influenza in United States for European American population between 1900 and 2006. "EAM" refers to European American Males, and "EAF" refers to European American Females. "Raw Data" is the number of deaths from influenza at each time point and in each age category, "Population" is the number of people alive at each time point and in each age category, "1 minus TOT" is 1 minus the mortality by any cause at each time point and in each age category.

Loading the data

We suggest that you first read the whole exercise before choosing which data you will store and how you will store them. Be careful about possibly missing data. You will probably have to make approximations, interpolations, ...

Output

1. A first kind of graph we want to extract from these data would show us, for a given time point (parameter), what is the mortality by influenza (y-axis) for each age category (x-axis).
2. A second kind of graph we want to extract from these data would show us, for a given age category (parameter), what is the mortality by influenza (y-axis) at each time point (x-axis).
3. A third kind of graph we want to extract from these data would show us, for people born in a given year (parameter), what is the mortality by influenza (y-axis) at each age (x-axis).

Plot one graph of each category. Do not forget to label the axes of your plot, add a legend, ...

E. Seam Carving

The following links explain an automated image resizing algorithm called seam carving. Watch the Youtube video, read the Wikipedia page. The third link is the paper by the inventors of this technique which provides more details if you need.

- `http://www.youtube.com/watch?v=6NcIJXTlugc`
- `http://en.wikipedia.org/wiki/Seam_carving`
- `http://perso.crans.org/frenoy/matlab2012/seamcarving.pdf`

1. Try to summarize with your words what is seam carving and what are the interests and weaknesses compared to other resizing techniques.
2. We want you to implement seam carving with Matlab. Here are the steps we suggest:
   - Identify the arguments of the Matlab function.
   - Load the image (*Documentation:* **imread**.).

- Compute the energy matrix.
- Choose and implement an algorithm to find the best seam to remove.
- Perform the removal and update the energy matrix.
- Loop the last two steps.
- Save the image (*Documentation:* **imwrite**.).

3. Test on several images (`http://images.google.com/`).
4. Every improvement or innovation you can do is welcome! For example you can try to use other energy functions, to improve the performance of your program, to add a user interface ...

F. Iterated prisoner dilemma

The prisoner dilemma is a classical game-theory problem in which two players are going to decide at the same time to cooperate or not with the other player. When both choose to cooperate, they get a payoff of 1. When none cooperate, they get -1. When only one cooperate, he gets -2 and the other (that does not cooperate) gets 2.

Briefly said, when both player cooperate it is a win-win situation. However a player that does not cooperate get even more because he "exploits" the other. But if both do not cooperate it becomes a lose-lose situation. What makes the game interesting is that "players" will encounter each other a high number of times, and will remember previous interactions, so they can choose to act according to their "history" with a particular player.

The goal of this exercise is to write the best possible strategy, to win first against a determined number of "bots", then against the strategy of other students. A strategy is a Matlab function that takes the arguments `myID, opponentID, GameHistory, IdFight, ResultMat` (described below) and returns 1 if you wish to cooperate, 0 if you wish not to cooperate.

- `myID` is your ID (a number that identifies you in the history of interactions). It is assigned randomly at the beginning of the tournament and stays the same.
- `opponentID` is the ID of your opponent (that identifies him in the history of interactions)
- `GameHistory` represents the history of interactions you had, with this opponent and with other players. Each line represent a past interaction, formed of the number of the "fight", the ID of opponent A, the ID of opponent B, the behavior of opponent A (1 for cooperate, 0 for not cooperate), the behavior of opponent B. `GameHistory(j,:)`=[IdFight, ID opponent A, ID opponent B, decision A, decision B] for jth fight. Note that for each line of this history you can be either opponent A or opponent B (this can be different each time).
- `IdFight` is the ID of the current fight.
- `ResultMat` is the `2x2` matrix of payoffs (`[D vs D, D vs C; C vs D, C vs C]`).

1. Make sure you have folder `PD` (part of `http://perso.crans.org/frenoy/matlab2014.tar.gz`) in your Matlab folder. Have a look on the already defined strategies (`Bot*.m`).
2. Write a strategy that respects the specifications described above, and modify `TournamentScript.m` so it adds your strategy to the competition with predefined bots.

3. When you are satisfied with your strategy, send it to me by email, and we will make a competition with other students strategies, past year students strategies, and my strategy, in addition to existing bots.