



Principi di Bioingegneria

A.A. 2022/23

Lezione 6

Intro MATLAB 3

Prima Esercitazione

Vincenzo Catrambone, PhD

vincenzo.catrambone@unipi.it



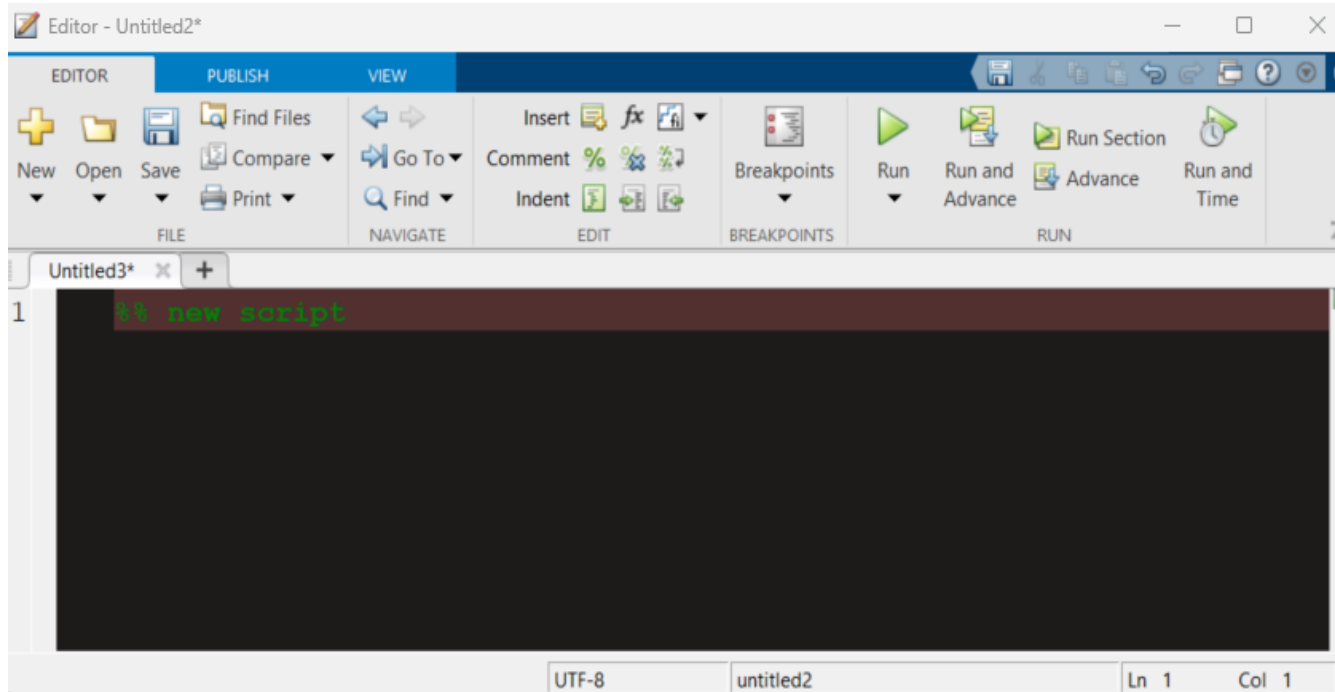
Script

Funzioni

Esercizi

Esercitazione in aula

Script



MATLAB is a programming language, as well as an interactive computational environment. Files that include code in the MATLAB language are called **M-files**.

M-files are ordinary ASCII text data written in MATLAB's language; they must have the filename extension ".m" at the end of their name (like **myfunction.m**). This extension is needed for this data to be interpreted by MATLAB.

There are two types of M-Files:

- **M-File Scripts:** Scripts do not accept input arguments or return output arguments. They operate on data in the workspace.
- **M-File Functions:** Functions can accept input arguments and return output arguments. Internal variables are local to the function.

Script

Editor - C:\Users\asus\OneDrive - University of Pisa\studio symbolicTE microS\MicroStateAnalysis_CP.m

EDITOR PUBLISH VIEW

New Open Save Find Files Compare Print FILE

Go To Find NAVIGATE

Insert Comment Indent EDIT

Breakpoints BREAKPOINTS

Run Run and Advance RUN

Run Section Advance RUN

Run and Time RUN

MicroStateAnalysis_CP.m

```
31 EEGdir = 'E:\EEG rest+CP+rest';
32
33 % retrieve a list of all EEG files in EEGdir
34 EEGFiles = dir(EEGdir);
35 load('E:\Events');
36
37 %% load datasets in EEGLAB
38 sub = 0;
39 for ss = 1:length(Events)
40     if isempty(Events{ss,4})
41         continue
42     end
43     ss2 = 0;
44     for ssl = 3:length(EEGFiles)
45         if contains(Events{ss,4}, EEGFiles(ssl).name(1:end-4))
46             ss2 = ssl;
47             break
48         end
49     end
50     if ~ss2
51         continue
52     end
53     sub = sub + 1;
54
55     load(fullfile(EEGdir, EEGFiles(ss2).name));
56     EEG.setname = EEGFiles(ss2).name;
57
58     [ALLEEG, EEG, CURRENTSET] = eeg_store( ALLEEG, EEG, 0 );
59     eeglab redraw
60 end
61 num = size(ALLEEG, 2);
62
63 % select data for microstate analysis
64 [EEG, ALLEEG] = pop_micro_selectdata( EEG, ALLEEG, 'datatype', 'spontaneous', 'avgref', 1, ...
```

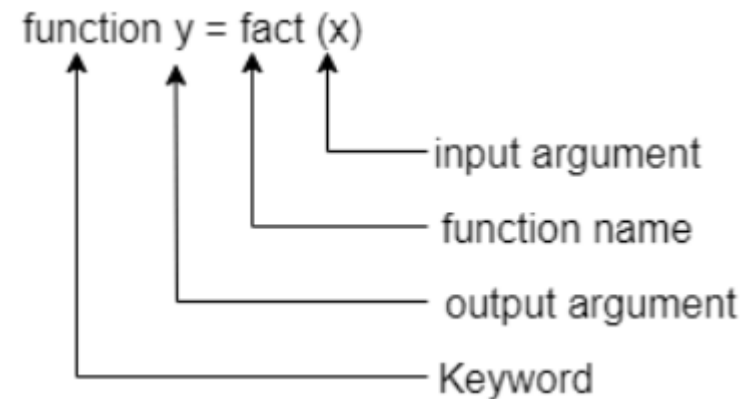
UTF-8 script Ln 34 Col 24

Functions

Functions are M-files that can obtain input arguments and return output arguments. The names of the M-file and the function should be the same. Functions perform on variables within their own workspace, which is also called the local workspace, separate from the workspace you access at the MATLAB command prompt, which is known as the base workspace.

Function files are like programs or subroutine in FORTRAN, operations in PASCAL, and functions in C.

A function file starts with a function definition line, which has a well-defined record of inputs and outputs. Without this line, the file develops into a script file.



Functions

```
function f = fact(n)      Function definition line
% Compute a factorial value.    H1 line
% FACT(N) returns the factorial of N,    Help text
% generally indicated by N!
% put simply, FACT(N) is PROD(1:N).    Comment
f = prod(1:n);            Function body
```

! Scrivere il nome di una funzione
! qualsiasi nella command window, right
! click, 'Open selection'

Function definition line (functions only)	It defines the function name, input and output argument.
H1 line	A one-line summary definition of the program, displayed when you request help on an entire directory, or when you use lookfor.
Help text	A more detailed definition of the program, displayed together with the H1 line when you request help on a specific function
Function or script body	Program code that executes the actual evaluation and assigns values to any output argument.
Comments	Text in the body of the programs that describe the internal working of the program.

Functions

```
factorial.m x +
1 function n = factorial(n)
2 %FACTORIAL Factorial function.
3 %   FACTORIAL(N) for scalar N, is the product of all the integers from 1 to N,
4 %   i.e. prod(1:N). When N is an N-D matrix, FACTORIAL(N) is the factorial for
5 %   each element of N. Since double precision numbers only have about
6 %   15 digits, the answer is only accurate for N <= 21. For larger N,
7 %   the answer will have the correct order of magnitude, and is accurate for
8 %   the first 15 digits.
9 %
10 %   Class support for input N:
11 %       float: double, single
12 %       integer: uint8, int8, uint16, int16, uint32, int32, uint64, int64
13 %
14 %   See also PROD.
15
16 %   Copyright 1998-2012 The MathWorks, Inc.
17
18 N = n(:);
19 if ~isreal(n) || any(fix(N) ~= N) || any(N < 0)
20     error(message('MATLAB:factorial:NNegativeInt'))
21 end
22 if isa(n, 'double')
23     thres = 171;
24 elseif isa(n, 'single')
25     thres = 35;
26 elseif isinteger(n)
27     thres = 21;
28 else
29     error(message('MATLAB:factorial:unsupportedType'));
30 end
31 N = min(N, thres);
32 m = max(1, max(N));
33 Fa = cumprod([1 1 2:m]);
34 n(:) = Fa(N+1);
```

Recursive functions (examples)

```
factorial.m  my_factorial.m  +
1  function F = my_factorial(n)
2      % my version of a factorial
3
4      if n <= 1
5          F = 1;
6      else
7          F = n*my_factorial(n-1);
8      end
9
10 end
11
```

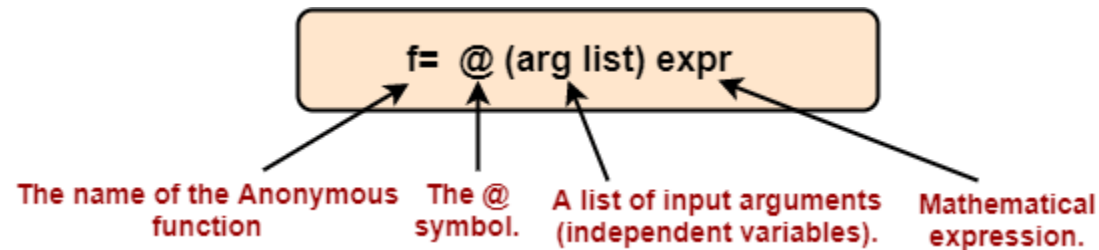
Try to write a recursive and a not-recursive function to calculate factorial, see the time needed for computation at increasing inputs

```
factorial.m  my_factorial.m  +
1  function F = my_factorial(n)
2      % my version of a factorial
3
4      if isnan(n) || isinf(n)
5          disp('Error: input is NaN or infinite')
6          return
7      elseif ~isreal(n) || (n < 0)
8          disp('Error: input is complex or negative')
9          return
10     elseif numel(n) ~= 1
11         disp('Error: input is an array')
12     end
13     n = round(n);
14
15     if n <= 1
16         F = 1;
17     else
18         F = n*my_factorial(n-1);
19     end
20
21 end
22
```


Anonymous function

An anonymous function is a simple (one-line) user-defined function that is defined without creating a separate function file (M-file). Anonymous functions can be defined in the Command Window, within a script file, or inside a user-defined function.

An anonymous function is generated by typing the following command:



```
>> c = 2;  
>> f = @(x) c*x  
f =  
    @(x)c*x  
>> f(3)  
ans =  
     6
```

Where `f` is the function handle. The input list can contain a single variable or several variables separated by commas. After creating the function, we can use it with its handle to evaluate the function or pass it as an argument to other functions.

For example, create an anonymous function that evaluates and return the area of a circle:

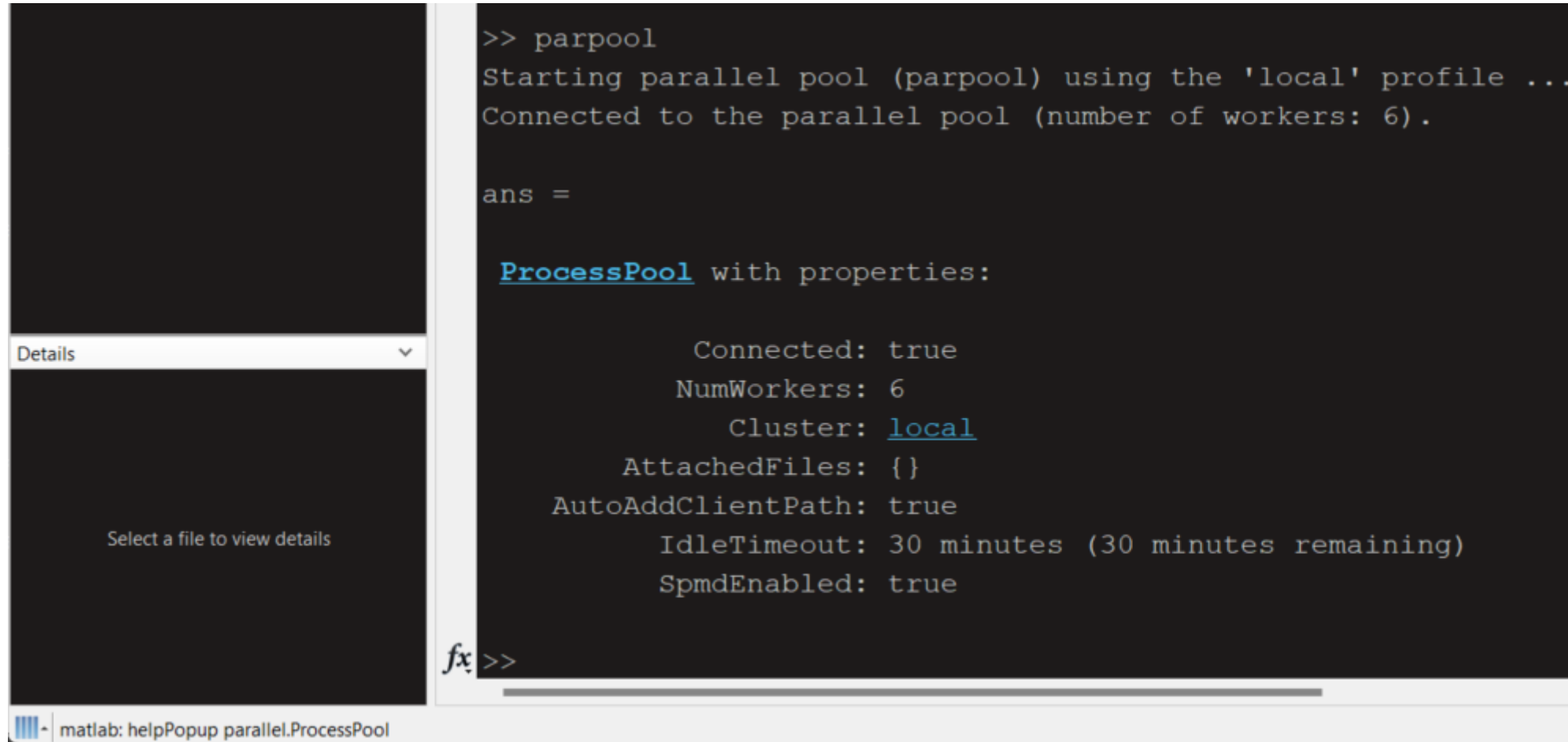
```
>> cirarea = @(radius) pi * radius.^ 2;
```

```
>> class(cirarea)  
ans =  
    function_handle
```

```
>> cirarea(4)  
ans =  
    50.2655  
>> cirarea(1:4)  
ans =  
    3.1416    12.5664    28.2743    50.2655
```

Parallel computing

Ormai tutti i PC hanno un certo numero di core (Dual Core, Quad Core), con i comandi `parpool` e `parfor` è possibile far girare più iterazioni in parallelo



```
>> parpool
Starting parallel pool (parpool) using the 'local' profile ...
Connected to the parallel pool (number of workers: 6).

ans =

    ProcessPool with properties:

        Connected: true
        NumWorkers: 6
        Cluster: local
        AttachedFiles: {}
        AutoAddClientPath: true
        IdleTimeout: 30 minutes (30 minutes remaining)
        SpmdEnabled: true

fx >>
```

Details ▾

Select a file to view details

matlab: helpPopup parallel.ProcessPool

Parallel computing

```
162 - parfor ch = 1:Nch
163 -     [EEG_Power(ch,:)] = ExtractPower(EEG(ch,:));
164 - end
```

Esempio ipotetico di applicazione per estrarre la potenza dai canali EEG in parallelo

```
146 % model running for each EEG channel
147
148 if Nch > 1
149     parfor ch = 1:Nch
150         [HeartToBrain(ch,:), BrainToLF(ch,:), BrainToHF(ch,:), HeartToBrain_sigma(ch,:), HeartToBrain_mc(ch,:), time_bhi(ch)] = ...
151         BHI_InsideModel(TFR_EEG(ch,:), TFR_HRV, CPr, CSr, wind, time_tfr);
152     end
153     time_bhi = time_bhi(1);
154 else
155     [HeartToBrain, BrainToLF, BrainToHF, HeartToBrain_sigma, HeartToBrain_mc, time_bhi] = ...
156     BHI_InsideModel(TFR_EEG, TFR_HRV, CPr, CSr, wind, time_tfr);
157 end
158
```

Saving or loading

Save all variables from the workspace in a binary MAT-file, `test.mat`. If `filename` is a variable, use function syntax.

```
filename = 'test.mat';  
save(filename)
```

Otherwise, you also can use command syntax.

```
save test.mat
```

Remove the variables from the workspace, and then retrieve the data with the `load` function.

```
clear  
load('test.mat')
```

```
load(filename)  
load(filename,variables)  
load(filename,'-ascii')  
load(filename,'-mat')  
load(filename,'-mat',variables)
```

```
S = load(__)
```

```
load filename
```

```
p = rand(1,10);  
q = ones(10);  
save('pqfile.mat','p','q')  
save('pqfile.txt','p','q','-ascii')
```

Create a bar chart and save it as a PNG file.

```
x = [2 4 7 2 4 5 2 5 1 4];  
bar(x);  
saveas(gcf,'Barchart.png')
```

Esercizi

1. Assume that we want to create a vector of increasing integer values from *mymin* to *mymax*. We will write a script *createvec* that receives two input arguments, *mymin* and *mymax*, and returns a vector with values from *mymin* to *mymax* in steps of one. First, we would make sure that the value of *mymin* is less than the value of *mymax*. If not, we would need to exchange their values before creating the vector. How would we accomplish this?
2. Write a script *printsindegorrad* that will:
 - Prompt the user for an angle.
 - Prompt the user for (r)adians or (d)egrees, with radians as the default.
 - If the user enters 'd', the *sind* function will be used to get the sine of the angle in degrees; otherwise, the *sin* function will be used. Which sine function to use will be based solely on whether the user enters a 'd' or not. A 'd' means degrees, so *sind* is used; otherwise, for any other character the default of radians is assumed so *sin* is used.
 - Print the result.
3. How could you write a function to determine whether an input argument is a scalar, vector, or matrix?
4. The systolic and diastolic blood pressure readings are found when the heart is pumping and the heart is at rest, respectively. A biomedical experiment is being conducted only on subjects whose blood pressure is optimal. This is defined as a systolic blood pressure less than 120 and a diastolic blood pressure less than 80. Write a script that will prompt for the systolic and diastolic blood pressures of a person, and will print whether or not that person is a candidate for this experiment, or not.
5. The pH of an aqueous solution is a measure of its acidity. The pH scale ranges from 0 to 14, inclusive. A solution with a pH of 7 is said to be neutral, a solution with a pH greater than 7 is basic, and a solution with a pH less than 7 is acidic. Write a script that will prompt the user for the pH of a solution, and will print whether it is neutral, basic, or acidic. If the user enters an invalid pH, an error message will be printed.

Esercizi

5. Write a script that will prompt the user for a temperature in degrees Celsius, and then an 'F' for Fahrenheit or 'K' for Kelvin. The script will print the corresponding temperature in the scale specified by the user. For example, the output might look like this:

```
Enter the temp in degrees C: 29.3
```

```
Do you want K or F? F
```

```
The temp in degrees F is 84.7
```

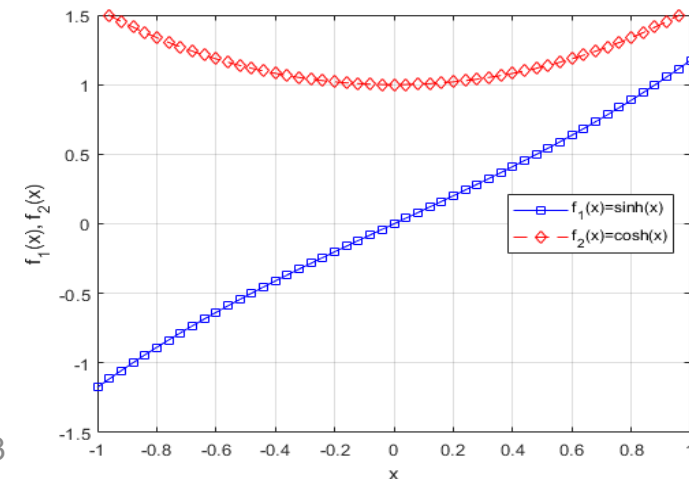
The format of the output should be exactly as specified. The conversions follow:

$$F = 9/5 C + 32$$

$$K = C + 273.15$$

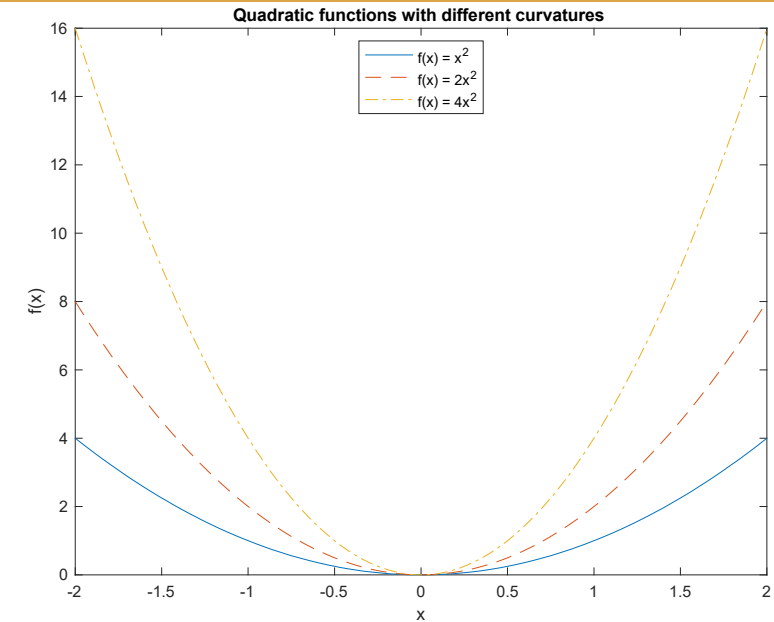
6. Write a function `isdivby4` that will receive an integer input argument, and will return logical 1 for true if the input argument is divisible by 4, or logical false if it is not.
7. A Pythagorean triple is a set of positive integers (a, b, c) such that the square of c is equal to the sum of the squares of a and b . Write a function `ispythag` that will receive three positive integers (a, b, c) in that order and will return logical 1 for true if they form a Pythagorean triple, or 0 for false if not.
8. Evaluate following two functions in the interval $x \in \langle -1, 1 \rangle$ for 51 values: $f_1(x) = \sinh(x)$, $f_2(x) = \cosh(x)$.

- Use the function `plotto` to depict both f_1 and f_2 so that:
- both functions are plotted in the same axis,
- the first function is plotted in blue with \square marker as solid line,
- the other function is plotted in red with \diamond marker and dashed line,
- limit the interval of the y-axis to $[-1.5, 1.5]$,
- label the axes (x-axis: x , y-axis: $f_1(x), f_2(x)$),
- apply grid to the graph,
- add a legend associated to both functions.



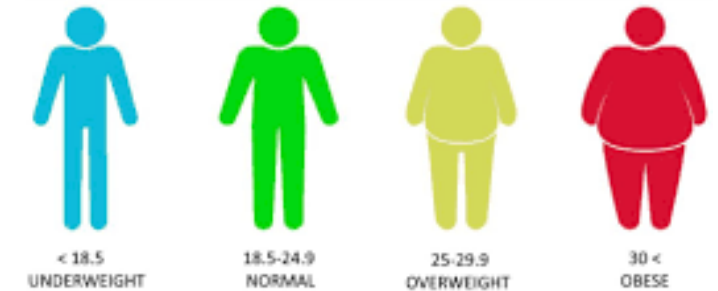
Esercizi

9. Generate the following graph: $y_1 = x.^2$; $y_2 = 2*x.^2$; $y_3 = 4*x.^2$;
Title is: Quadratic functions with different curvatures
ALTERNATIVE: save y_1, y_2, y_3 in a cell array or structure and make the plot retrieving the data from the cell array /structure
10. Biomedical engineers are developing an insulin pump for diabetics. To do this, it is important to understand how insulin is cleared from the body after a meal. The concentration of insulin at any time t is described by the equation $C = C_0 e^{-30t/m}$ where C_0 is the initial concentration of insulin, t is the time in minutes, and m is the mass of the person in kilograms. Write a script that will graphically show how the weight of the person influences the time for insulin to be cleared from the body. It will show in a 2×1 subplot the concentration of insulin for two subjects: one who weighs *120 lb*, and one who weighs *300 lb*. For both, the time should increment from 0 to 4 minutes in steps of 0.1 minute, and the initial concentration should be 85. The concentration over time will be shown in each subplot, and the weight of the person should be in the title. The conversion factor is $1 \text{ lb} = 0.4536 \text{ kg}$. To better compare, use consistent axes for both plots.



Esercizi

11. Write a program to calculate and print the area and circumference of a circle. There should be one script and three functions to accomplish this (one that prompts for the radius, one that calculates the area and circumference, and one that prints).
12. Write a function that calculates the Body Mass Index (BMI) of a person, taking as input their mass and height, and gives the associated label.
13. Write a function that converts a point from cartesian (x,y,z) to polar coordinates (r,p)



Esercitazione

- 1) Si crei una funzione Matlab che data una stringa 's' di DNA in ingresso, costituita da una serie di basi peptidiche di lunghezza massima pari a 1000 basi, dia in uscita la conta di ciascuna delle 4 basi possibili, in ordine 'A', 'C', 'G', and 'T'. La funzione deve essere in grado di trovare degli eventuali errori: se la stringa non fosse DNA, ma RNA, o altro, o se fosse troppo lunga (> 1000 basi). Si fornisca anche una rappresentazione grafica, a piacere (es. Istogramma, grafico a torta, ecc) del risultato. Testare la funzione sulle 5 stringhe salvate nella variabile 'Sample_Strings' presente nella cartella Teams, nella stessa variabile ci sono anche i risultati per ogni stringa, e gli eventuali errori.

Sample dataset:

```
>> s = 'AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC';
```

Sample output

```
>> 20  12  17  21
```

Esercitazione

2) Caricare in Matlab il file 'SampleEEG.mat' presente nel MS Team condiviso, contiene una variabile omonima di dimensione 9x15000, questa rappresenta un segnale EEG di esempio, misurato in μV , su 9 canali in 30 secondi. E' presente anche una variabile con il nome dei canali 'Chans_Name' e una con la frequenza di campionamento 'sampling_rate'.

Plottare in una figura con 9 subplot l'andamento della stima della potenza di ciascun canale, calcolata su finestre di un secondo non sovrapposte. Approssimare la potenza alla somma quadratica dei valori del segnale in ogni finestra, normalizzata per la lunghezza della finestra stessa.

Tarare opportunamente gli assi, le labels, intitolare ogni subplot con il nome del canale corrispondente. Fornire su schermo il nome del canale che raggiunge il picco massimo di potenza e il tempo, in secondi, in cui questo viene raggiunto.

Esempio di figura (scegliere liberamente tutte le opzioni grafiche):

