

# Computing Methods for Experimental Physics and Data Analysis

## Data Analysis in Medical Physics

Lecture 2: intro to MATLAB, read and display medical images

Alessandra Retico

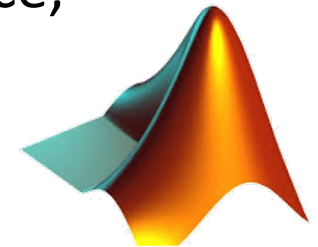
[alessandra.retico@pi.infn.it](mailto:alessandra.retico@pi.infn.it)

INFN - Pisa

# Brief introduction to MATLAB

---

- MATLAB (MATrix LABoratory) integrates computation, visualization, and programming in an easy-to-use environment.
- MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.
- MATLAB users come from various backgrounds of engineering, science, and economics.
- First of all: download and install on your laptop:  
<https://start.unipi.it/personale-t-a/strumenti-di-lavoro/strumenti-informatici/software-e-servizi-cloud/software-matlab/>



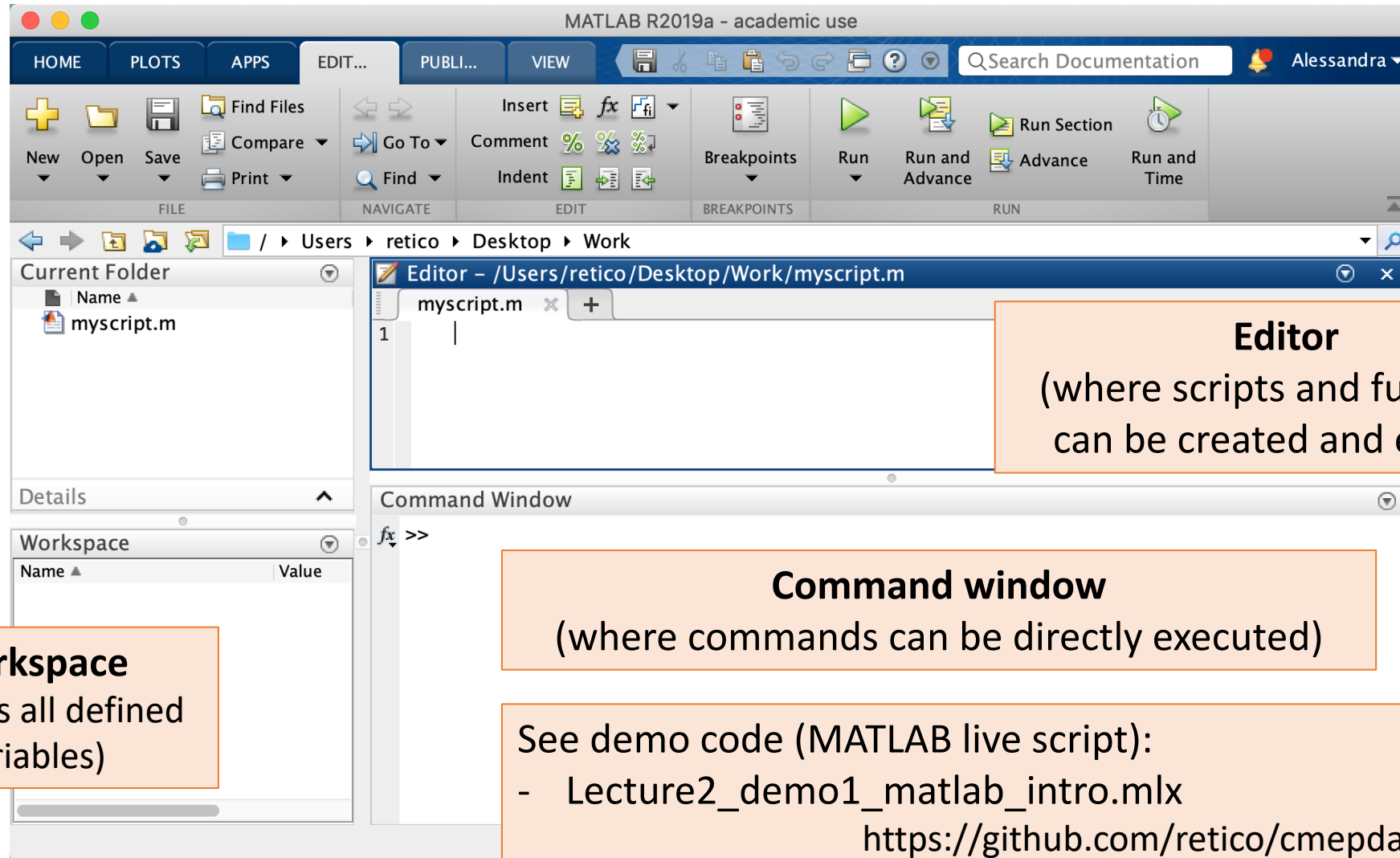
Current stable release  
is R2022b

# System requirements and useful MATLAB toolboxes

---

- System requirements for installation:
  - Administrative rights
  - **Processors – Minimum:** any Intel or AMD x86-64 processor
  - **RAM – Minimum:** 4 GB. **Recommended:** 8 GB
  - **DISK – Minimum:** 3.3 GB of HDD space for **MATLAB** only, 5-8 GB for a typical installation. **Recommended:** An SSD is recommended
- During the installation you have to specify the products to be installed, i.e. the **MATLAB toolboxes**.
- You may add some toolboxes you like to the suggested ones (e.g. the Image Processing, Curve Fitting, Mapping, Wavelet and Deep Learning toolboxes we will use in the exercises)
- You can add more toolboxes, whenever you need, from the Add-Ons drop down menu from the MATLAB desktop HOME tab.

# Getting started with matlab



# Intro to Matlab for Python users

<https://blogs.mathworks.com/student-lounge/2021/02/19/introduction-to-matlab-for-python-users/>

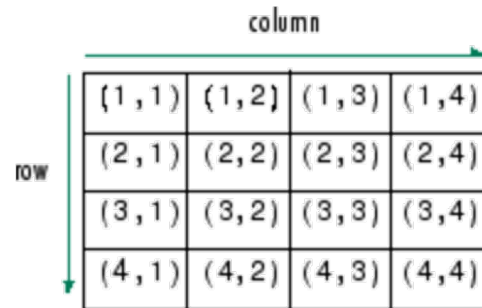
Python Syntax	MATLAB Syntax	Purpose	MATLAB Example
#	%	Comment	% hello
N/A	;	Suppress output	x(1,1);
\	...	Continue to next line	x = 1+ ... 2;
os	!	Operating system command	!echo hi
+ - * /	+ - * /	Mathematical operators	x = 1+2
**	^	Exponent	x = 1^2
* / **	.* ./ .^	Element-wise operators	x = [1 2].*[3 ; 4]
not, and, or	~ &	NOT, AND, OR logical operators	if x<2 & x>2
del	clear	Clear variable from memory	clear x y
clear	clc	Clear command window	clc

The basic data structure in MATLAB is a matrix, while Python treats everything as a general object.

<https://www.mathworks.com/content/dam/mathworks/fact-sheet/matlab-for-python-users-cheat-sheet.pdf>

# Multidimensional Matlab arrays

A multidimensional array in MATLAB® is an array with more than two dimensions. In a matrix, the two dimensions are represented by rows and columns.

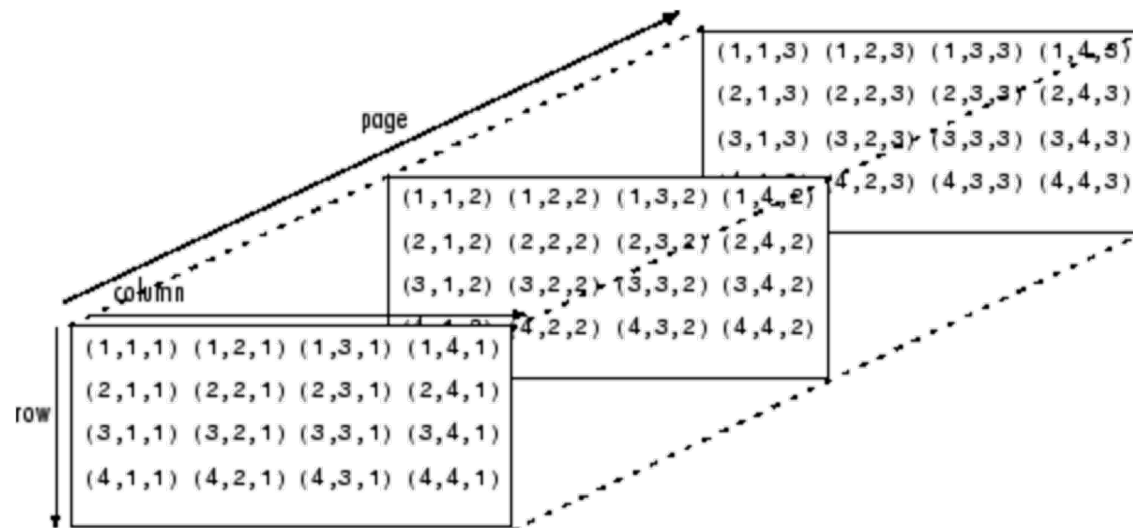


A 4x4 matrix representing a 2D array. The horizontal axis is labeled 'column' with a right-pointing arrow, and the vertical axis is labeled 'row' with a downward-pointing arrow. The elements are arranged in a grid:

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

In MATLAB, the basic type, is a multidimensional array. Array assignments in MATLAB are stored as 2D arrays of double precision floating point numbers, unless you specify the number of dimensions and type. Operations on the 2D instances of these arrays are modelled on matrix operations in linear algebra.

Each element is defined by two subscripts, the row index and the column index. Multidimensional arrays are an extension of 2-D matrices and use additional subscripts for indexing. A 3-D array, for example, uses three subscripts. The first two are just like a matrix but the third dimension represents *pages* or *sheets* of elements.



# Data type correspondence between Python and Matlab

---

Python	MATLAB
float	double, single
complex	complex single, complex double
int	(u)int8, (u)int16, (u)int32, (u)int64
float(nan)	NaN
float(inf)	inf
str	str, char
bool	logical
dict	struct
list, tuple	cell
pandas.dataframe	table

# Matlab m-files and user-defined functions

- Matlab programs can be written with any ASCII text editor, using the \*.m file extension.
- M-files can be executed from the matlab command line:

```
% plot_sin.m  
% this script plots the sinus function  
x=linspace(0,10*pi,200);  
y=sin(x);  
plot(y)
```

```
>> plot_sin           % it works both with and without specifying the .m extension
```

- User defined functions work just like commands in Matlab:

```
function [avg, st_dev]= show_stats(x)  
    % show_stats(x) returns the average (avg) and standard deviation (st_dev) of a vector (x)  
    n = length(x);  
    avg = sum(x) / n;  
    st_dev = sqrt(sum((x - avg).^2)/n);  
end
```

- Try:

```
>> help show_stats  
>> v1 = randn(100,1); [avg_v1,sd_v1]= show_stats(v1)
```



# Matlab & git

---

- The best place to share your MATLAB projects is [File Exchange](#) because of its popularity with the MATLAB user community
- [GitHub](#) is one of the most popular websites that host Git repositories
- Since R2014b [File Exchange is integrated with GitHub](#)
- In addition to what you already know about GitHub usage, you have to register your MATLAB file extensions such as .mlx, .mat, .fig, ... as binary formats. You can follow the instructions to **Register Binary Files with Git**:
  - [https://it.mathworks.com/help/matlab/matlab\\_prog/set-up-git-source-control.html](https://it.mathworks.com/help/matlab/matlab_prog/set-up-git-source-control.html)

You have to add some lines in the .gitattributes hidden file in your git repository

The .gitattributes file is already correctly set in the [https://github.com/retico/cmepda\\_medphys](https://github.com/retico/cmepda_medphys) repository

Add these lines to the .gitattributes file:

```
..  
*.fig binary  
*.mat binary  
*.mlx binary  
...
```

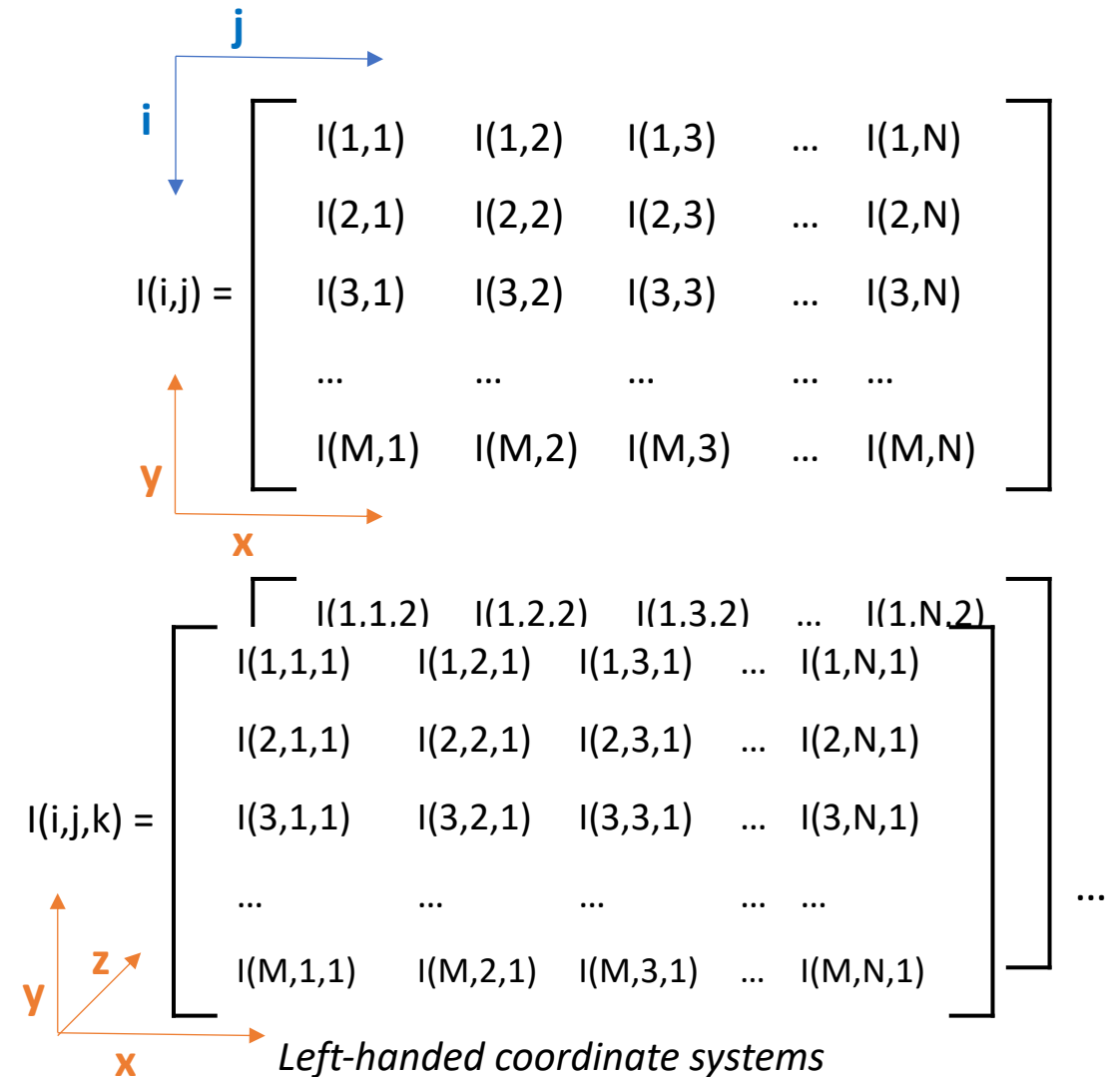
# Basic image processing

- Basic operations with images:
  - `imread()`, `imshow()`, `imwrite()`
  - Histogram operations
  - Morphological Operations
  - Image filtering
  - Contrast (C), noise (N), signal-to-noise (SNR) and contrast-to-noise (CNR) ratio
  - ...

- A large variety of functions for image processing are available in the MATLAB Image Processing toolbox

See demo code:

- `Lecture2_demo2_image_read_display.mlx`
- `Lecture2_demo3_read_dicom.m`



# Hands-on materials

---

On GitHub [https://github.com/retico/cmepda\\_medphys](https://github.com/retico/cmepda_medphys)

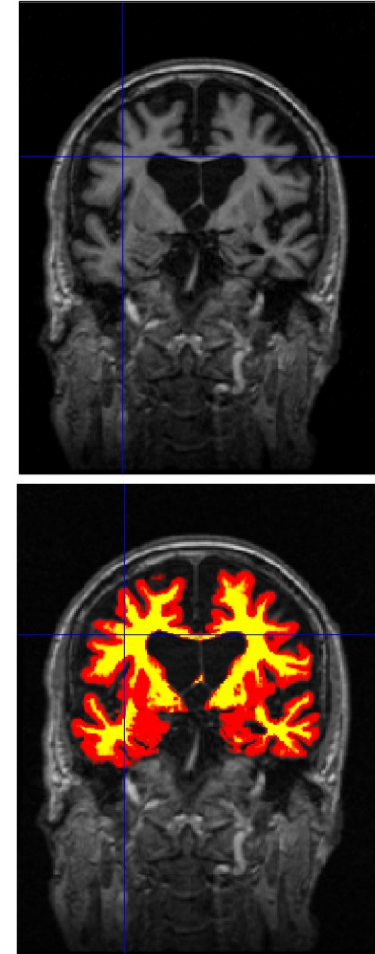
- Read and display 3D data (DICOM dir)
  - [L2\\_exercise1.m](#)
- Read and display 4D data (NIfTI file)
  - [L2\\_exercise2.m](#)
- Visualization of image overlays
  - [L2\\_exercise3.m](#)

Image display and image overlay in Mango:

- Visualize and make an overlay of the segmented brain tissues on a brain MRI with data on DATASETS/IMAGES/NIfTI\_Examples/Brain\_segment/

Check the exercise solutions on GitHub:

[L2\\_exercise1\\_solution.m](#), [L2\\_exercise2\\_solution.m](#),  
[L2\\_exercise3\\_solution.m](#)



# Sample DICOM/NIfTI images

---

- You can download the DATASETS directory, which contains a sample of publicly available medical images, from:
  - <https://pandora.infn.it/public/cmepda/DATASETS>
  - and on [https://drive.google.com/open?id=1YqK7ZkM-P2IrfD7Pj-SCmjz-GWd\\_1-Y](https://drive.google.com/open?id=1YqK7ZkM-P2IrfD7Pj-SCmjz-GWd_1-Y)
- Example of open repositories of medical data
  - <https://openneuro.org>
  - [http://fcon\\_1000.projects.nitrc.org](http://fcon_1000.projects.nitrc.org)
  - <https://nihcc.app.box.com/v/ChestXray-NIHCC>
  - <https://www.cancerimagingarchive.net>
  - <http://www.oasis-brains.org>
  - <http://adni.loni.usc.edu>
- Additional data samples (medical images/image features) for hands-on sessions will be distributed later during the course

# References and sources

---

- Books

- Digital Image Processing for Medical Applications, Geoff Dougherty
- Handbook of Medical Image Processing and Analysis, Isaac N. Bankman
- Image Processing and Acquisition using Python, Ravishankar Chityala & Sridevi Pudipeddi

- Sources

- <https://start.unipi.it/personale-t-a/strumenti-di-lavoro/strumenti-informatici/software-e-servizi-cloud/software-matlab/>
- <https://it.mathworks.com/help/matlab/getting-started-with-matlab.html>
- <https://it.mathworks.com/videos/>
- <https://blogs.mathworks.com/student-lounge/2021/02/19/introduction-to-matlab-for-python-users/>
- <https://numpy.org/doc/stable/user/numpy-for-matlab-users.html>
- <https://it.mathworks.com/help/matlab/external-language-interfaces.html>
- [https://it.mathworks.com/help/matlab/matlab\\_prog/set-up-git-source-control.html](https://it.mathworks.com/help/matlab/matlab_prog/set-up-git-source-control.html)
- <https://blogs.mathworks.com/community/2014/10/20/matlab-and-git/>