

MATLAB Starting Guide

Process Systems Engineering Labs – PSE-Lab – Politecnico di Milano

Giuseppe Pesenti

The PSE Labs require a basic familiarity with MATLAB programming.

As an indication, covering the following **basic concepts** is suggested to prepare for the PSE Labs:

- **Matlab interface:** the command line, the workspace and the variables in the workspace
- **Basic operations and command line:** using Matlab as a calculator, e.g. $2+2$, and using ";" to suppress output in the command line
- **Variables:** how to create variables and make assignments, e.g. $a=5$; $b=a$
- **Arrays and matrices:** how to create arrays and matrices, basic operations between them
- **Indexing:** how to access one or more values in an array/matrix, e.g. take the i -th element of an array $v(i)$, take all rows of the second column of a matrix $m(:,2)$
- **Logical operations:** testing equality, inequality ($==$, $\sim=$), logical and, or ($\&\&$, $|$), true, false (0, 1)
- **Control structures:** if elseif else, for loops, while
- **Matlab files:** how to create Matlab scripts and functions
- **Debugging:** how to read errors and how to debug your code (set breakpoints, make a step or continue till next breakpoint)
- **2D plots:** a basic idea of how to plot, e.g. $\text{plot}(x,y)$, changing line colors and adding xlabel, ylabel, xlim, ylim, legend
- **Useful commands:** for example `clc`, clear all, close all, help, docs, disp, find, length/size, and comments (the text after % is ignored)

We suggest [this tutorial](#) (in English) to cover the previous basic concepts – and many other tutorials can be found online. Learning to debug your code is very important. To learn about a MATLAB command, remember to use “*help command*” and “*docs command*”, and to check out the Mathworks documentation online.

Additionally, the following points are more specific to the specific problems of this course and Chemical Engineering in general, and concern the **use of built-in MATLAB solvers and integrators**:

- A basic idea of function handles (storing an association to a function inside a variable), e.g. `fiveorbigger=@(x) max(x,5)`
- A basic idea of anonymous functions (creating a function in one line of code), e.g. `squareminusone=@(x) x^2-1`
- **fsolve** and **fminsearch** - we will use them to solve numerically equations, or systems of equations
- The ode suite, e.g. **ode45** - these are built-in Matlab functions that integrate a user-defined function which contains a differential equation (or a system of differential equations)

In addition to the Mathworks specific documentation pages (e.g. [fsolve](#), [fminsearch](#), [ode45](#)), many tutorials can be found online covering these topics (e.g. [by Mathworks](#) about ode integrators).

You can also refer to the pdfs of a PSE-Lab bachelor course “Strumentazione e Controllo di Impianti Chimici” (in Italian, [link](#)). The first labs focus on providing MATLAB basics to 3rd year bachelor Chemical Engineering students and cover all the previous bullet points. In particular, [this presentation](#) covers the general MATLAB basics, [this presentation](#) covers anonymous functions and fsolve, [this presentation](#) covers ode solvers and [this guide](#) suggests a to-do list on using ode solvers.

Two other MATLAB tutorials (in Italian) prepared by the PSE-Lab are also available: [tutorial A](#), [tutorial B](#).