Introduction To Scientific Computing

Basics of MATLAB

Dr. Fintan Healy
Room 1.31, Queens Building
fintan.healy@bristol.ac.uk

University of BRISTOL

Lecture 1

Introduction

Unit Aims

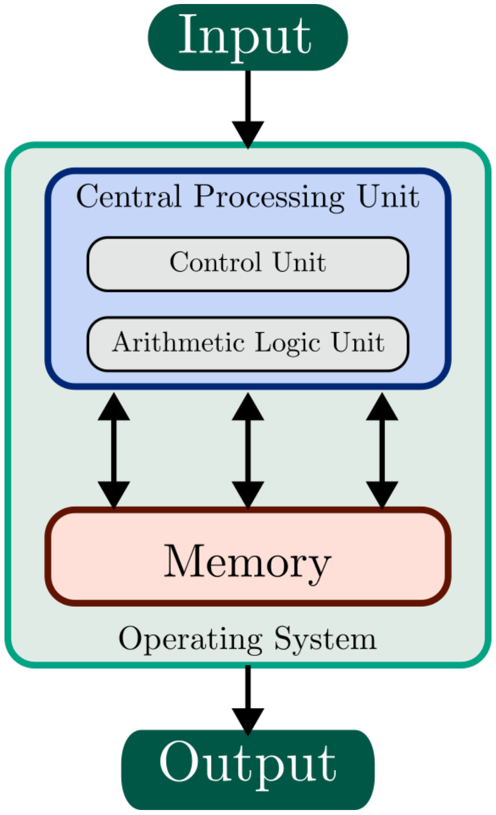
- To familiarise you with programming in MATLAB
- (Re-)Introduce you to core programming constructs
- Exposure to LaTeX for documentation and reporting

University of BRISTOL

Landscape of Scientific Computing

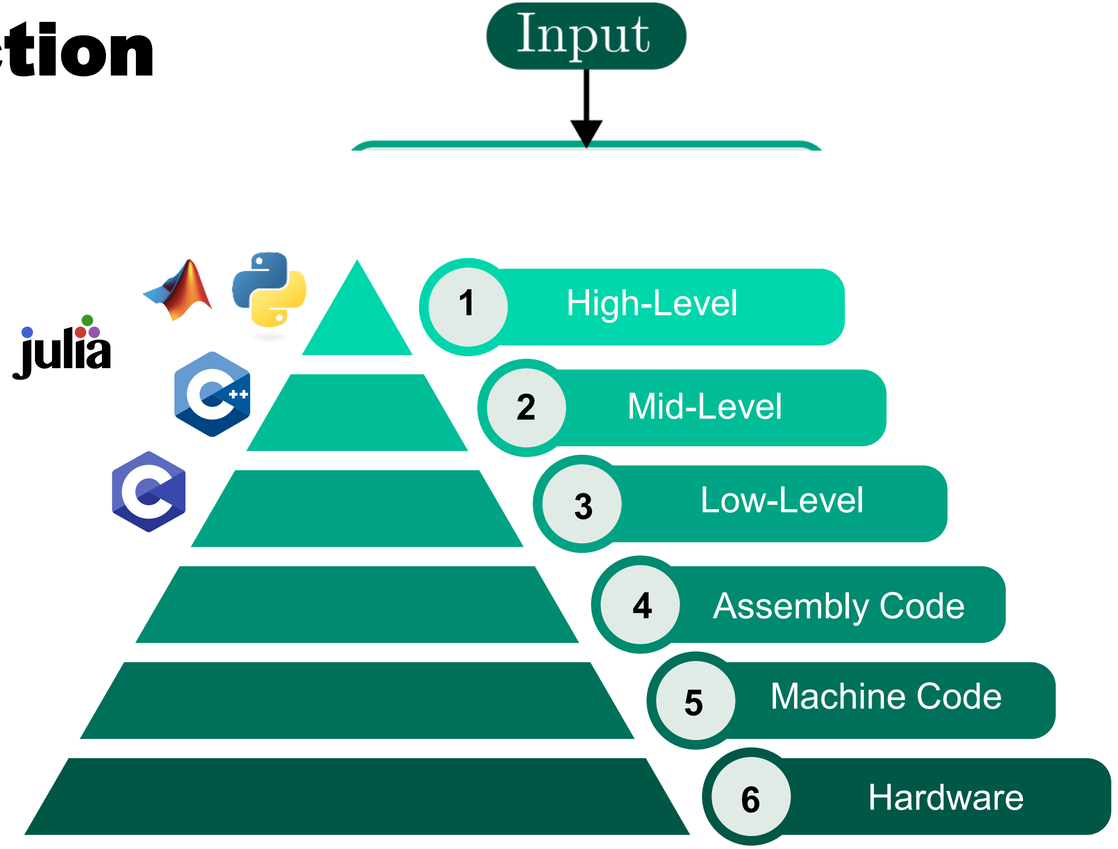
What is a Computer?

- A computer consists of hardware, software and an operating system. The `machine'
- An Operating System (OS) is the mechanism to access the computer's hardware - this is the `environment'
- A programming language is the language used to define a set of commands/tasks the programmer wishes the computer to perform



Levels of Abstraction

- Instructions are passed to the CPU/GPU as machine code
- There are different levels of abstraction, each with different coding languages
  - **Low Level**
    - ➢ minimal abstraction
    - ➢ direct hardware control
    - ➢ memory management
    - ➢ rapid execution
  - **High Level**
    - ➢ lots of abstraction (focus on logic and functionality)
    - ➢ minimal memory management
    - ➢ slower execution



Core Coding Constructs

- Different languages, different syntax. Same constructs.

Python

```python
val = 0
for i in range(1, 5):
    val = val + i
print(val)
```

MATLAB

```matlab
val = 0;
for i = 1:4
    val = val + i;
end
disp(val);
```

C++

```cpp
int val = 0;
for (int i = 1; i <= 4; i++) {
    val = val + i;
}
cout << val << endl;
```

Key Constructs:

- Variables
- Control Flow (loops, conditions)
- Functions
- Operators (+ - / * etc…)

Core Coding Constructs

- Different languages, different syntax. Same constructs.

Python

```python
val = 0
for i in range(1, 5):
    val = val + i
print(val)
```

MATLAB

```matlab
val = 0;
for i = 1:4
    val = val + i;
end
disp(val);
```

C++

```cpp
int val = 0;
for (int i = 1; i <= 4; i++) {
    val = val + i;
}
cout << val << endl;
```

Assembly Code

```asm
section .data
    fmt db "%d", 10, 0
section .text
    global main
    extern printf
main:
    mov eax, 1
.loop:
    cmp eax, 4
    jg .end
    add eax, 1
    jmp .loop
.end:
    mov esi, eax
    lea rdi, [rel fmt]
    xor eax, eax
    call printf
    ret
```

Machine Code

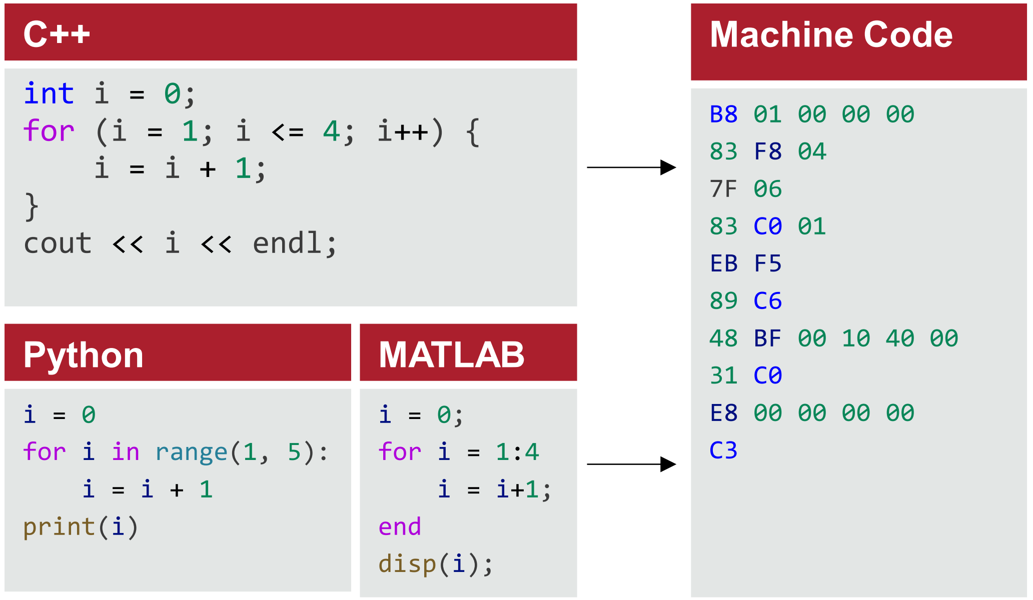
```
B8 01 00 00 00
83 F8 04
7F 06
83 C0 01
EB F5
89 C6
48 BF 00 10 40 00 00 00 00 00
31 C0
E8 00 00 00 00
C3
```

Key Constructs:
- Variables
- Control Flow (loops, conditions)
- Functions
- Operators (+ - / * etc…)

Compiled Versus Interpreted

- There are two* ways to get a computer to perform an ‘operation’, either:
- **Compiled**
  1. Source code developed
  2. Compiler converts to machine code
  3. The binary code is run on the OS
- **Interpreted**
  1. Source code developed
  2. Code executed in interpreter environment
  3. Interpreter reads and compiles the code ‘line-by-line’

***JIT** – Hybrid method…



C++

```
int i = 0;
for (i = 1; i <= 4; i++) {
    i = i + 1;
}
cout << i << endl;
```

Python

```
i = 0
for i in range(1, 5):
    i = i + 1
print(i)
```

MATLAB

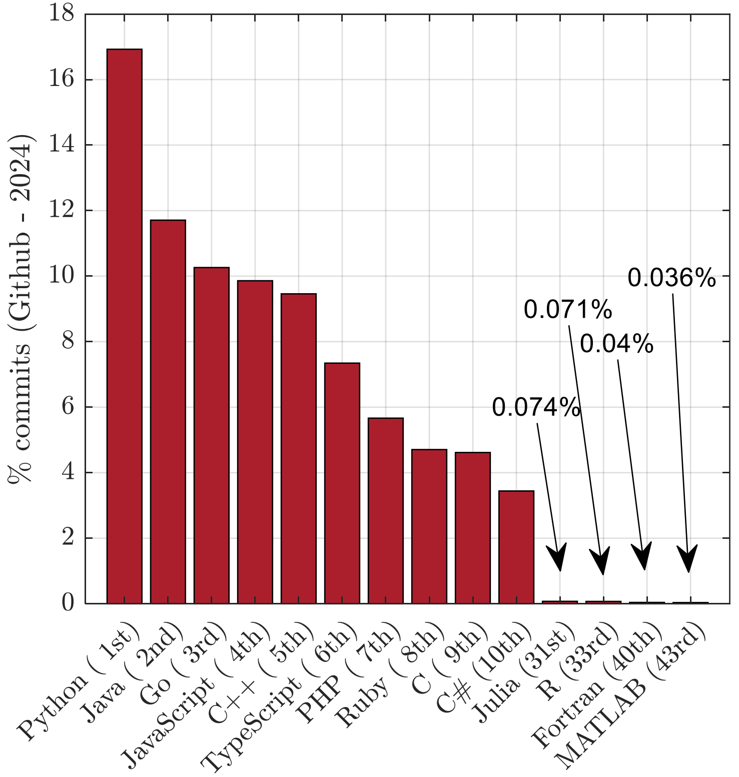
```
i = 0;
for i = 1:4
    i = i+1;
end
disp(i);
```

Machine Code

```
B8 01 00 00 00
83 F8 04
7F 06
83 C0 01
EB F5
89 C6
48 BF 00 10 40 00
31 C0
E8 00 00 00 00
C3
```

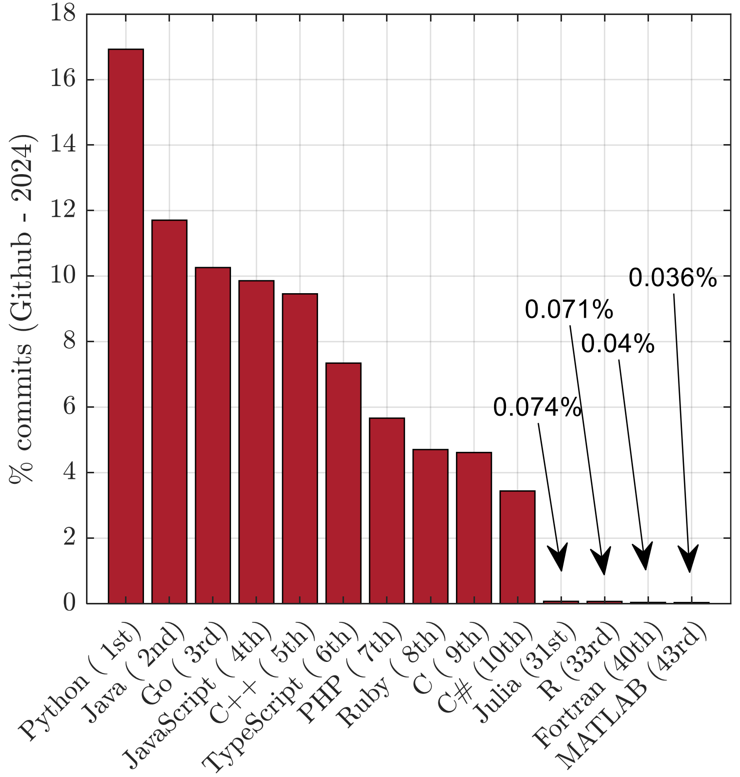
Most Popular Programming Langauges

- Python is the most ‘popular’ language.
  - Having it on your CV is excellent
  - It’s a “general-purpose” language and is widely used in web development and data science
- C++ performs well. It’s used in performance-critical applications
  - Operating systems
  - Game engines
  - Databases
  - Python & MATLAB packages….
- MATLAB is 43rd. Why are we learning this?



Most Popular Programming Langauges

- It's good to learn multiple languages!
- Statistics skewed by general programming
- But we're engineers, we don't care about*:
  - Web development
  - Front-end development (GUIs)
  - App development
  - Async software architecture

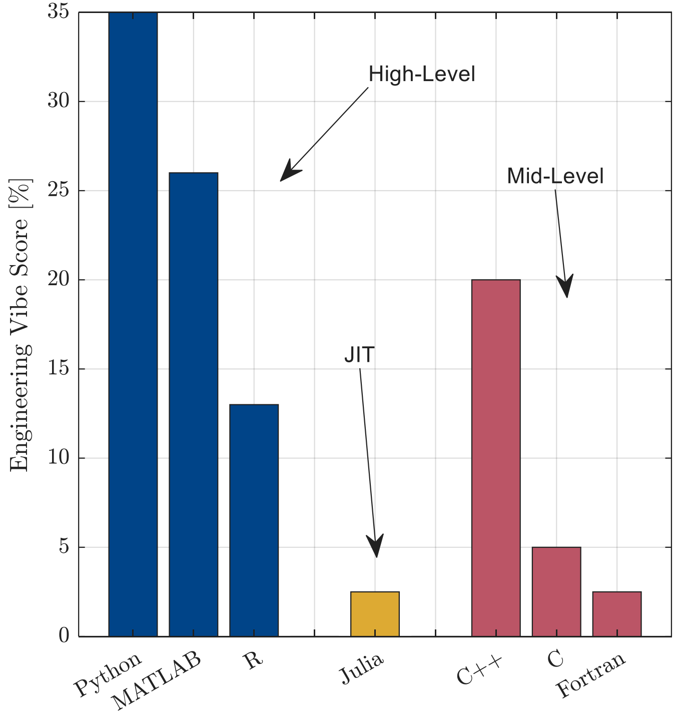


Scientific Programming

- A general-purpose language for scientific programming requires:

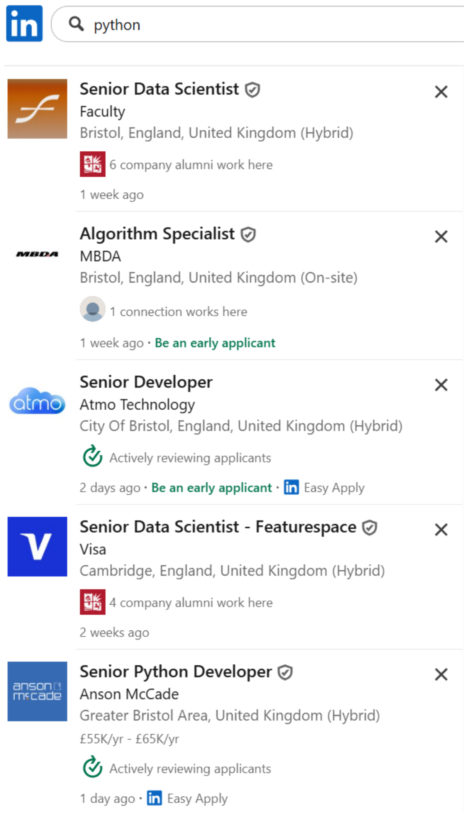
Scientific Programming

- A general-purpose language for scientific programming requires:
  - **Numerical Computing**
    - Matrix operations, linear algebra, signal processing
    - Simulation (numerical integration schemes)
  - **Data analysis + plotting**
  - **Interfacing with hardware**
    - For experiments
  - **A simple user interface**
    - Sometimes you *just* need a "fancy" calculator.
  - **Fast development / fast execution**
    - Good documentation
    - Industry adoption



Scientific Programming

- A general-purpose language for scientific programming requires:
  - **Numerical Computing**
    - ➢ Matrix operations, linear algebra, signal processing
    - ➢ Simulation (numerical integration schemes)
  - **Data analysis + plotting**
  - **Interfacing with hardware**
    - ➢ For experiments
  - **A simple user interface**
    - ➢ Sometimes you *just* need a “fancy” calculator.
  - **Fast development / fast execution**
    - ➢ Good documentation
    - ➢ Industry Adoption



python

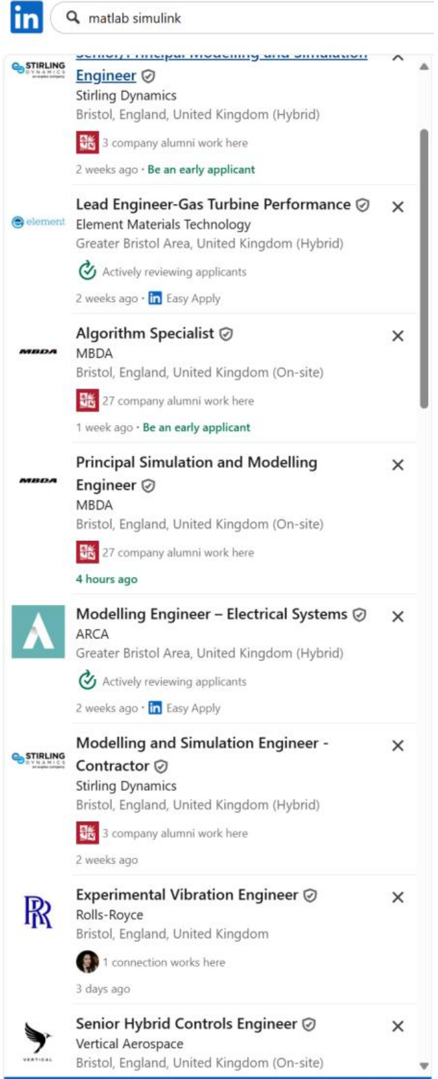
Senior Data Scientist
Faculty
Bristol, England, United Kingdom (Hybrid)
6 company alumni work here
1 week ago

Algorithm Specialist
MBDA
Bristol, England, United Kingdom (On-site)
1 connection works here
1 week ago · Be an early applicant

Senior Developer
Atmo Technology
City Of Bristol, England, United Kingdom (Hybrid)
Actively reviewing applicants
2 days ago · Be an early applicant · Easy Apply

Senior Data Scientist - Featurespace
Visa
Cambridge, England, United Kingdom (Hybrid)
4 company alumni work here
2 weeks ago

Senior Python Developer
Anson McCade
Greater Bristol Area, United Kingdom (Hybrid)
£55K/yr - £65K/yr
Actively reviewing applicants
1 day ago · Easy Apply



matlab simulink

Senior/Principal Modelling and Simulation Engineer
Stirling Dynamics
Bristol, England, United Kingdom (Hybrid)
3 company alumni work here
2 weeks ago · Be an early applicant

Lead Engineer-Gas Turbine Performance
Element Materials Technology
Greater Bristol Area, United Kingdom (Hybrid)
Actively reviewing applicants
2 weeks ago · Easy Apply

Algorithm Specialist
MBDA
Bristol, England, United Kingdom (On-site)
27 company alumni work here
1 week ago · Be an early applicant

Principal Simulation and Modelling Engineer
MBDA
Bristol, England, United Kingdom (On-site)
27 company alumni work here
4 hours ago

Modelling Engineer – Electrical Systems
ARCA
Greater Bristol Area, United Kingdom (Hybrid)
Actively reviewing applicants
2 weeks ago · Easy Apply

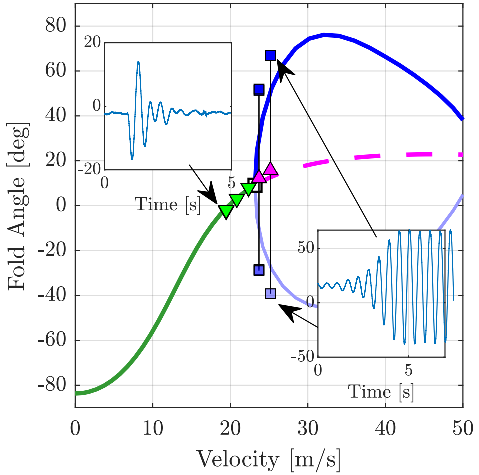
Modelling and Simulation Engineer - Contractor
Stirling Dynamics
Bristol, England, United Kingdom (Hybrid)
3 company alumni work here
2 weeks ago

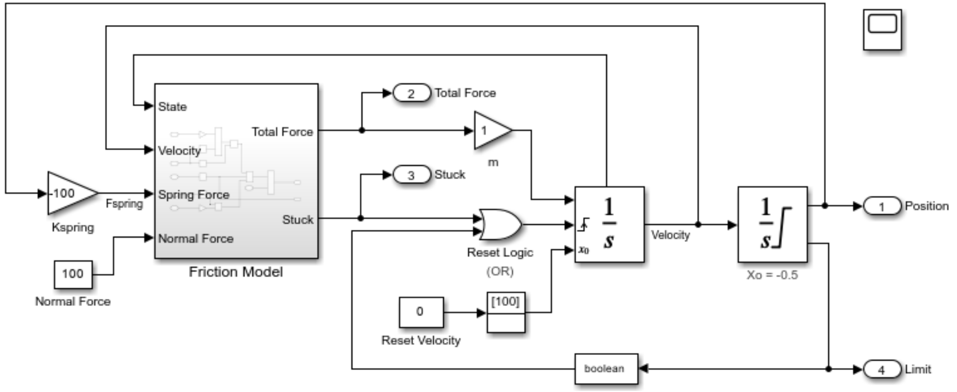
Experimental Vibration Engineer
Rolls-Royce
Bristol, England, United Kingdom
1 connection works here
3 days ago

Senior Hybrid Controls Engineer
Vertical Aerospace
Bristol, England, United Kingdom (On-site)

MATLAB

- MATLAB (Matrix Laboratory) is a development package produced by Mathworks **specifically for numerical, scientific and engineering calculations.**
- MATLAB is an interpreted language with similar syntax to C
- It comes packaged with a **mature IDE** (interactive development environment)
- It is an effective tool for initial development, data analysis, and **plotting**
- It has **broad industry adoption**:
  - Automotive, Aviation, F1 …
  - Particularly for controller development and data analysis
- Perhaps the best documentation of any language





MATLAB

- MATLAB (Matrix Laboratory) is a development package produced by Mathworks **specifically for numerical, scientific and engineering calculations.**
- MATLAB is an interpreted language with similar syntax to C
- It comes packaged with a **mature IDE** (interactive development environment)
- It is an effective tool for initial development, data analysis, and **plotting**
- It has **broad industry adoption**:
  - Automotive, Aviation, F1 …
  - Particularly for controller development and data analysis
- Perhaps the best documentation of any language

- A working knowledge of MATLAB is key for many of your units:

- **Year 2:**
  - Aerodynamic: lab exercise
  - Dynamics/Control: coursework
  - AVDASI2: useful for repeated calcs
- **Year 3:**
  - RP3: many computational projects
  - Numerical aero: all examples in MATLAB
  - Control: Simulink
- **Year 4:**
  - AVDASI4: detailed design calcs
  - Many optional units either have c/w that needs Matlab, or Matlab is a useful tool

MATLAB Basics

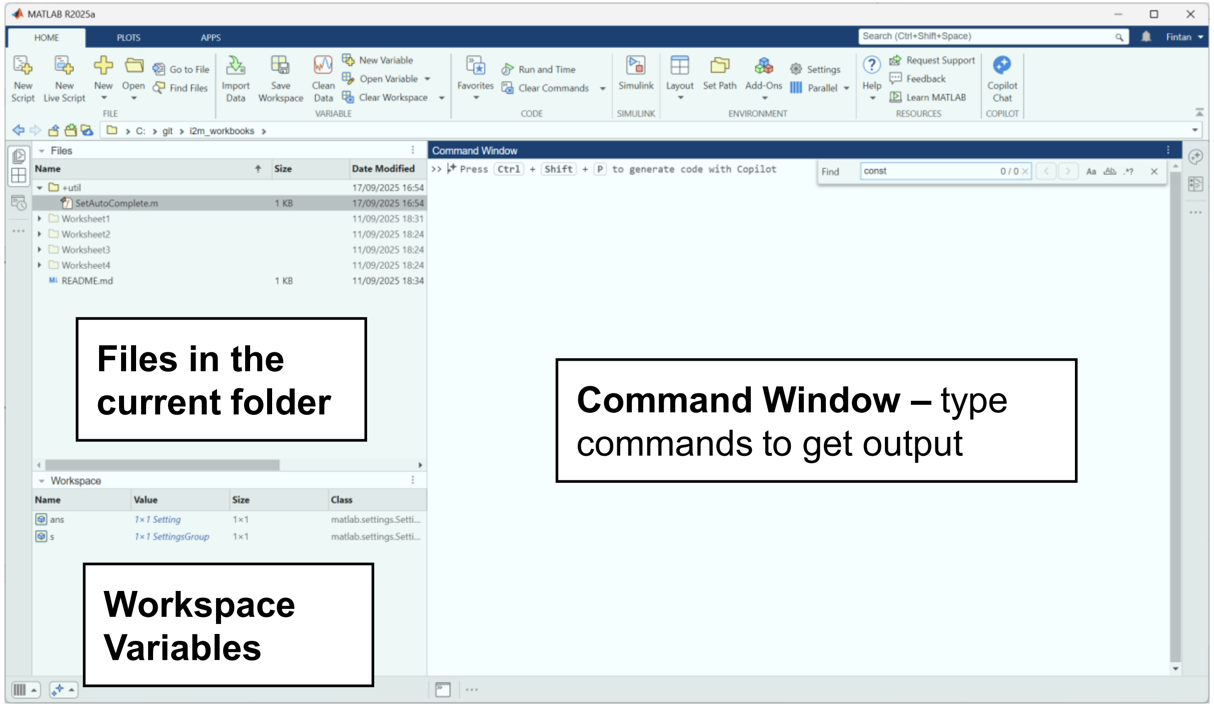
Installing MATLAB

- Follow the instructions at the following link to install MATLAB on your personal machines https://uob.sharepoint.com/sites/itservices/SitePages/matlab.aspx
  - As part of the process you will need to create a MathWorks account.
- The full installation, including all the packages, is available on all Engineering PCs

Note:

Minimum required Packages MATLAB + Simulink (if you are short of disk space, there is no need to install all of the extra packages)

The Interactive Development Enviroment (IDE)



Name	Size	Date Modified
+util		17/09/2025 16:54
SetAutoComplete.m	1 KB	17/09/2025 16:54
Worksheet1		11/09/2025 18:31
Worksheet2		11/09/2025 18:24
Worksheet3		11/09/2025 18:24
Worksheet4		11/09/2025 18:24
README.md	1 KB	11/09/2025 18:34

Name	Value	Size	Class
ans	1×1 Setting	1×1	matlab.settings.Setti...
s	1×1 SettingsGroup	1×1	matlab.settings.Setti...

```
>> Press Ctrl + Shift + P to generate code with Copilot
```

Files in the current folder

Command Window – type commands to get output

Workspace Variables

MATLAB Commands

- Commands can either be typed directly into the command window or written into a script.
- Anything written by the user in the command window has the command prompt symbol in front of it:

MATLAB

```
>> x = "A String";
>> x = 3.2;
>> y = x * (sqrt(x) - 2)
y = -0.6757
```

- = is an assignment
  - a = b; "assign "b" to "a"
- **Semi-colon (;)** used to suppress output
- MATLAB is case-sensitive
- Variable types are dynamic

MATLAB Scripts

- A script in MATLAB is simply a text file that contains some MATLAB commands.
- The extension of any MATLAB script is .m (e.g. filename.m).

Script "example.m"

```
clear all
close all
% This is a comment
a=2;
% pi is a built-in number
b=tan(pi );
% output something by not having a semi-colon
c=a*b

%% This is a new section
c=sin(2*pi);
d=cos(2*pi);
disp([c,d])
```

- "clear all" clears the workspace
- "close all" closes any open figures
- Comments begin with "%"
- To run, either:
  - Press run in the editor
  - Type script name into the command window
    - e.g. ">> example"
  - Press "ctrl+enter" to run a section
  - Highlight code and press F9

MATLAB Documentation

- A powerful command in MATLAB is the "*help*" command.

```
>> help <somefunction>
```

- Returns the internal MATLAB documentation in the command window, e.g.

```
>> help plot
 plot - 2-D line plot
    This MATLAB function creates a 2-D line plot of the data in
    Y versus the corresponding values in X.
```

- For even more detail use the doc command in the command window:

```
>> doc <somefunction>
```

MATLAB versus Python

Python

```
import numpy as np
A = np.array([[2, 3], [1, 7]])
b = np.array([[4], [4]])
x = A @ b
print(x)
```

MATLAB

```
A = [2,3;1,7];
b = [4;4];
x = A*b;
disp(x)
```

Python

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

def pendulum(t, y):
   g = 9.81
   L = 1
   return [y[1], -g/L * np.sin(y[0])]

sol = solve_ivp(pendulum, (0, 10), [0, 0.1])

plt.figure()
plt.plot(sol.t, sol.y[0])
plt.xlabel('Time (s)')
plt.ylabel('Angle (rad)')
plt.show()
```

MATLAB

```
function out =pendulum(t,y)
 g = 9.81;
 L = 1;
 out = [y(2);g*sin(y(1))/L];
end

[t,y] = ode45(@pendulum,[0 10], [0; 0.1]);

figure;
plot(t, y);
xlabel('Time (s)');
ylabel('Angle (rad)');
```

- In MATLAB:
  - All functions pre-loaded, **no imports**!
  - Whitespace is not mission critical
  - MATLAB uses 1-based indexing
- In Python:
  - Open-source
  - Huge ecosystem
  - Easy integration with other languages
  - “Go-to” language for deep learning

University of BRISTOL

Course Schedule

Course Structure

Week	Lecture	Lab	
1	Introduction	(a-sync)	Workbook 1
2	Basic Syntax	(Lab Session)	Workbooks 2-4
3	Plotting, Functions, Tips & Tricks	(Lab Session)	
4	Latex	(a-sync)	

- Supervised labs in weeks 2 & 3
  - No new content in labs, I and other teaching staff will be there to support you in completing the workbooks

Summary

- Have introduced MATLAB

This Week

- Complete “Getting Started” Worksheet
- Attempt Worksheet 1

https://i2sc.fintanhealy.co.uk/

- Information can also be found on Blackboard:
  - Organisations -> CADE Student Handbook 2025-26 -> About Your Programme -> Aerospace Engineering Undergraduate -> Year 2

University of BRISTOL

ENJOY!

bristol.ac.uk