



SIS Scuola Interdipartimentale
delle Scienze, dell'Ingegneria
e della Salute



Laurea Magistrale in IA (ML&BD)

Scientific Computing (part 2 – 6 credits)

prof. Mariarosaria Rizzardi

Centro Direzionale di Napoli – Bldg. C4

room: n. 423 – North Side, 4th floor

phone: 081 547 6545

email: mariarosaria.rizzardi@uniparthenope.it

The background features a large, faint watermark of the University of Naples Federico II logo. The logo is circular and contains a central figure of a woman (Minerva) holding a cornucopia. The text around the logo includes "1920 - 2020" at the top, "UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II" around the inner circle, and "100° ANNIVERSARIO" at the bottom.

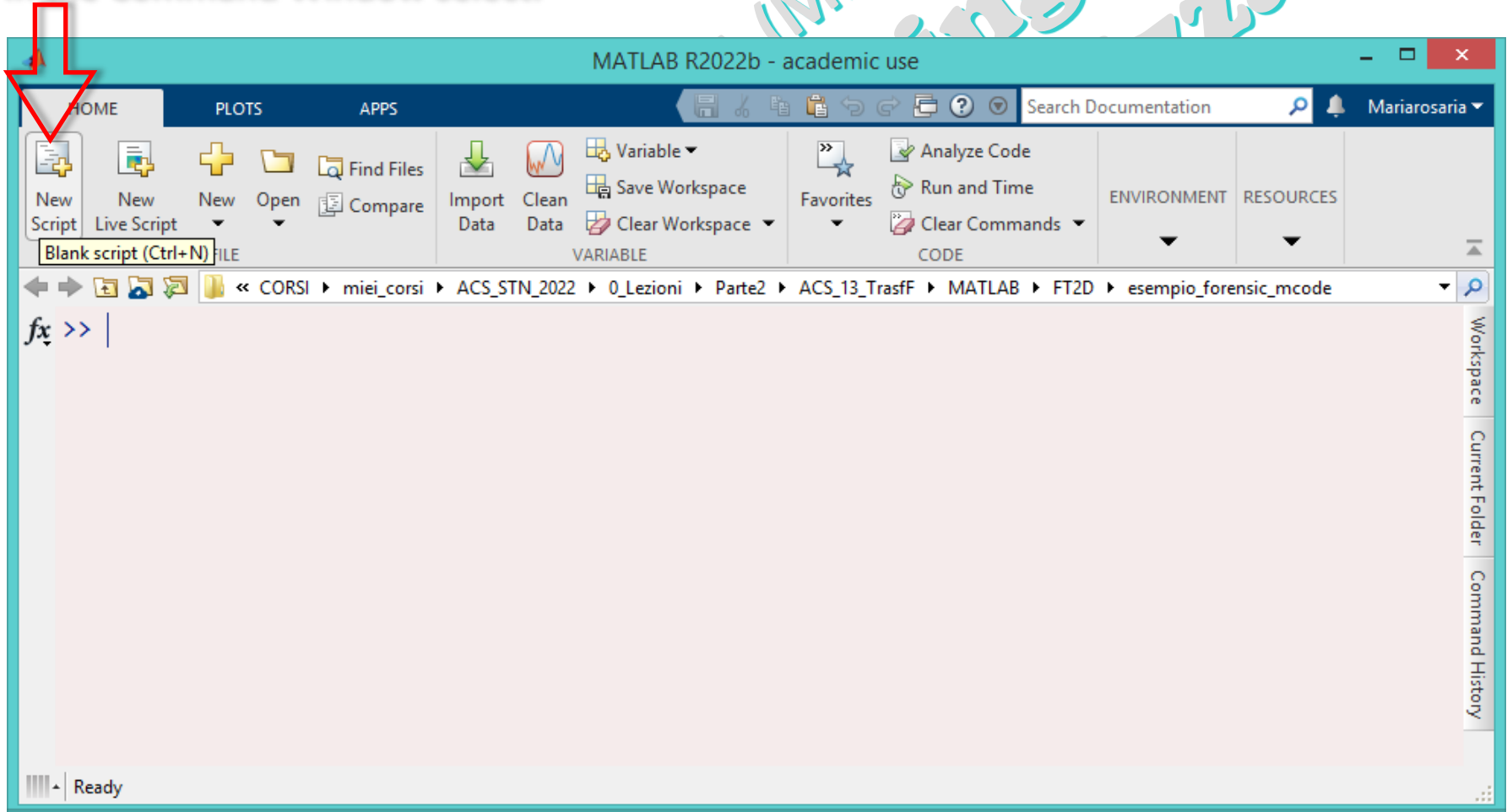
Contents

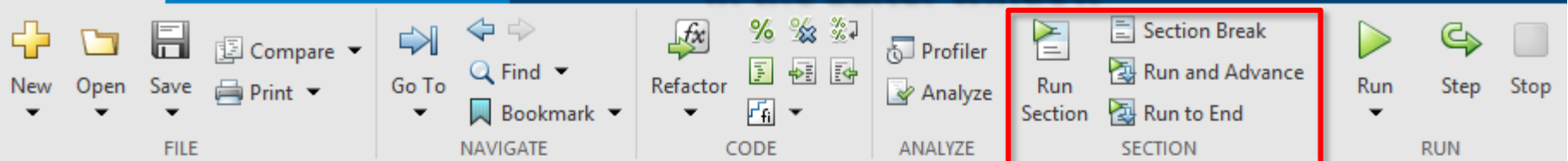
- **Recap: the MATLAB Editor (m-file).**
- **Advanced use of the MATLAB Live Editor (mlx-file).**

MATLAB Editor

We want to write a MATLAB program (m file) to display the plot of a function $y=f(x)$ in an interval.

In the Command Window select:





function_plot.m

file name: an '*' denotes that the file has not been saved

```
clear; clc
```

download the script: [function_plot.m](#)

%% input data ← **%% identifies a section of code**

```
N=input('Enter the number of points: N = ');
a=input('Enter the left endpoint of the interval: a = ');
b=input('Enter the right endpoint of the interval: b = ');
```

input a string

```
%%
fstr=input('Enter the function to be evaluated: f(t) = ', 's');
pf=['@(t)' fstr]; add @(t) before the string to form an anonymous function
pf=str2func(pf); % construct function handle from character vector
convert the string to anonymous function
```

```
%% display
```

```
x=linspace(a,b,N)';
y=pf(x);
```

```
figure(1); clf
```

```
plot(x,y)
```

```
set(gca, 'FontSize', 14)
```

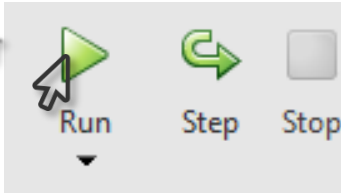
```
xlabel('x'); ylabel('y')
```

```
title(['Plotted function: f(t) = ' fstr], 'FontWeight', 'normal')
```

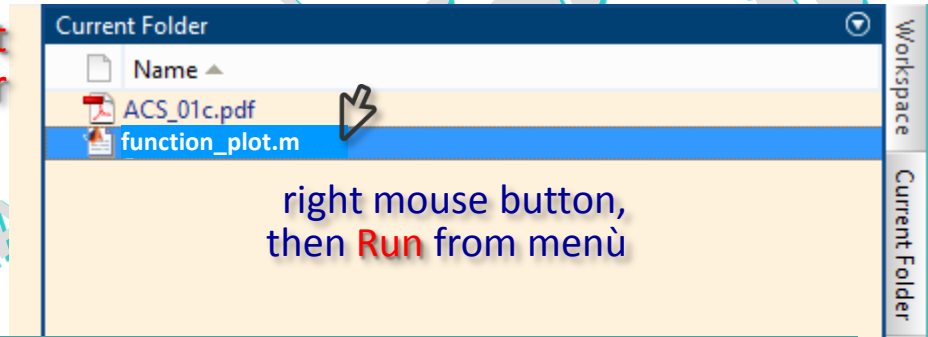
MATLAB Editor

To execute the script: function_plot.m

In the Editor window



In Current Folder



Then in the Command Window

A screenshot of the MATLAB R2022b interface. The Command Window shows the following text:

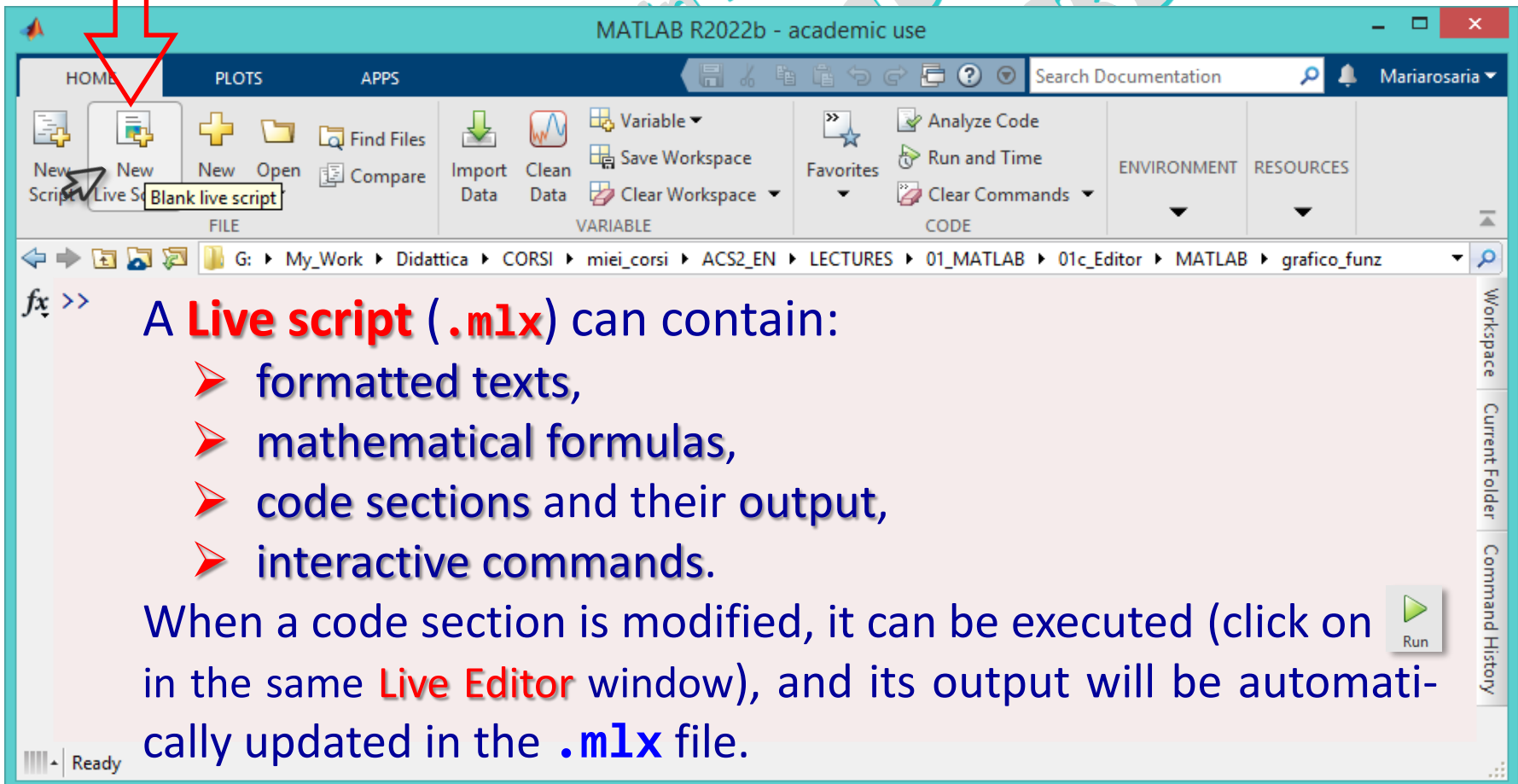
```
Enter the number of points: N = 101  
Enter the left endpoint of the interval: a = -2*pi  
Enter the right endpoint of the interval: b = +2*pi  
Enter the function to be evaluated: f(t) = cos(t) + 0.5*sin(10*t)  
fx >> |
```

A red arrow points from a red-bordered box containing the word 'string' to the function definition line. The Plots area shows a graph titled 'Plotted function: f(t) = cos(t) + 0.5*sin(10*t)'. The x-axis is labeled 'x' and ranges from -6 to 6. The y-axis is labeled 'y' and ranges from -1.5 to 1.5. The plot shows a blue line representing the function.

MATLAB Live Editor

We want to write a MATLAB program to display the plot of a function $y=f(x)$ in an interval, using the **Live Editor** with the addition of **interactive commands**.

In Command Window select:



A **Live script (.mlx)** can contain:

- formatted texts,
- mathematical formulas,
- code sections and their output,
- interactive commands.

When a code section is modified, it can be executed (click on **Run** in the same **Live Editor** window), and its output will be automatically updated in the **.mlx** file.

Example of a simple "Live Script" in Live Editor

The screenshot displays the MATLAB Live Editor interface. At the top, the menu bar includes LIVE EDITOR, INSERT, FIGURE, and VIEW. The toolbar contains icons for file operations (New, Open, Save, Print, Export), navigation (Go To, Find, Bookmark), text formatting (Normal, Bold, Italic, Underline, Monospace), and execution (Run, Step-Step, Run all sections (F5)).

The script content is as follows:

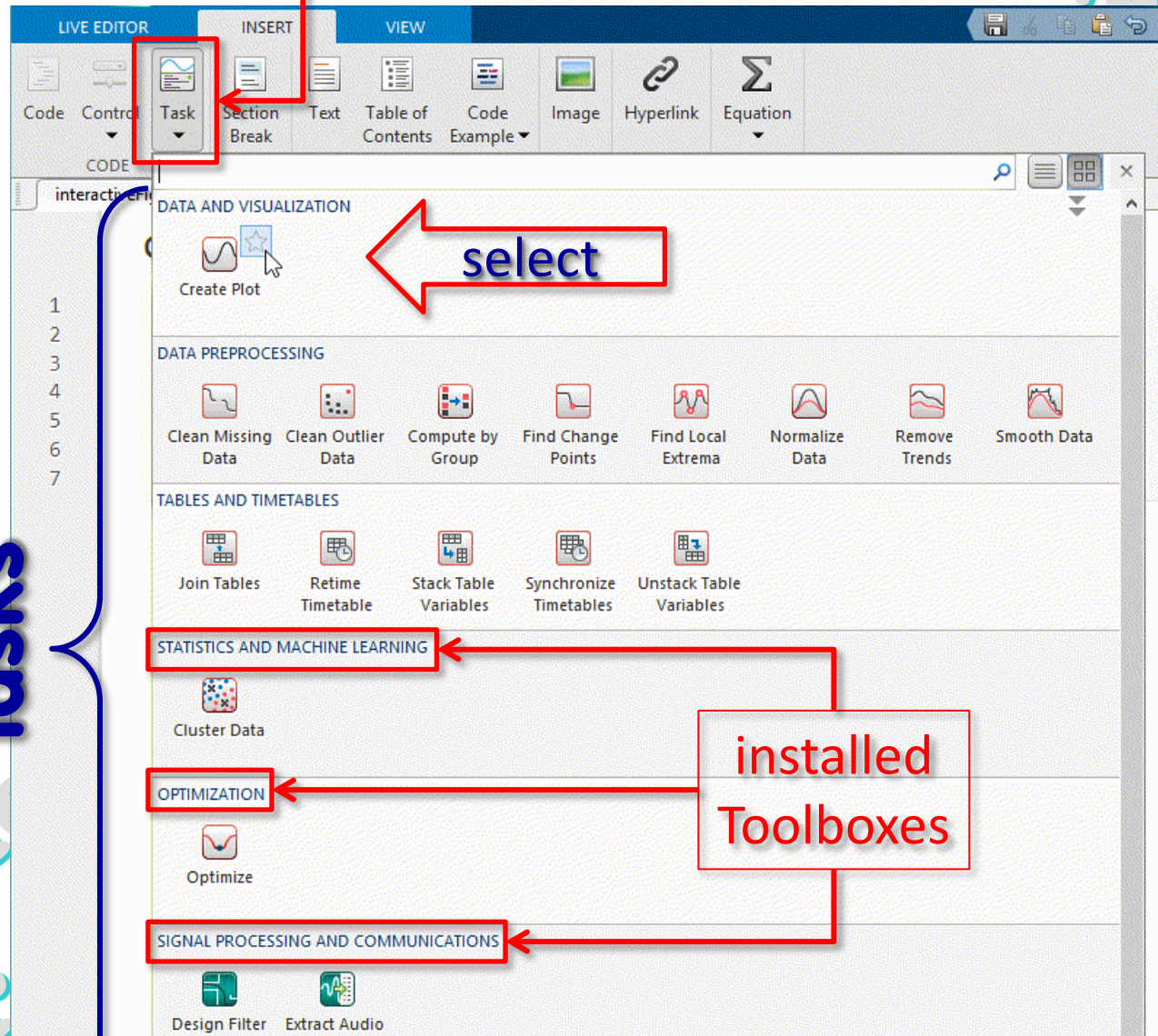
```
1 N = 201;  
2 a = -2*pi;  
3 b = +2*pi;  
4 pf=@(t) cos(t) + 0.5*sin(10*t);  
5 x = linspace(a,b,N)';  
6 y = pf(x);  
7 plot(x,y)
```

The plot below the code shows a blue line graph of a complex periodic function. The x-axis ranges from -8 to 8, and the y-axis ranges from -1.5 to 1.5. The function oscillates between approximately -1.5 and 1.5 with a period of about 2 units. A red arrow points to the plot with the label "output".

Annotations in the image include:

- A red box around the text "Plot of the function $f(x)$ for x in $[a,b]$ " with an arrow pointing to the plot.
- A red box around the code block with an arrow pointing to the code and the label "code".
- A red box around the text "plain text" with an arrow pointing to the text in the script editor.
- A red box around the "Run" button with an arrow pointing to it.

In the **Live Editor**, instead of writing code to display the plot, we can insert a **Task**



... other Tasks

Sci
Dr
Tasks


```
4 pf=@(t) cos(t) + 0.5*sin(10*t);
5 x = linspace(a,b,N)';
6 y = pf(x);
7 %plot(x,y
```

Task: Create Plot

Create Plot

h3 = plot of x and y

Select visualization

Search for a visualization Filter by Category All

Creates a 2-D line plot of the data in Y versus the corresponding values in X

Select data

X ▼

Y ▼

Select optional visualization parameters

select ▼ +

plot Add

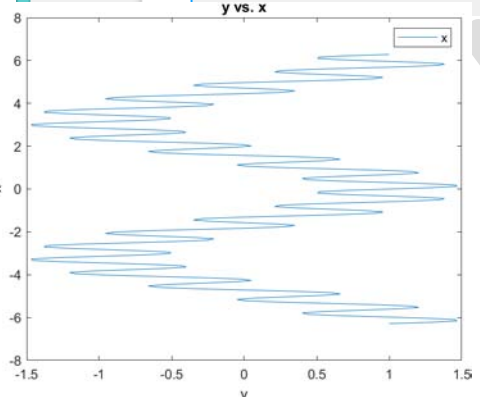
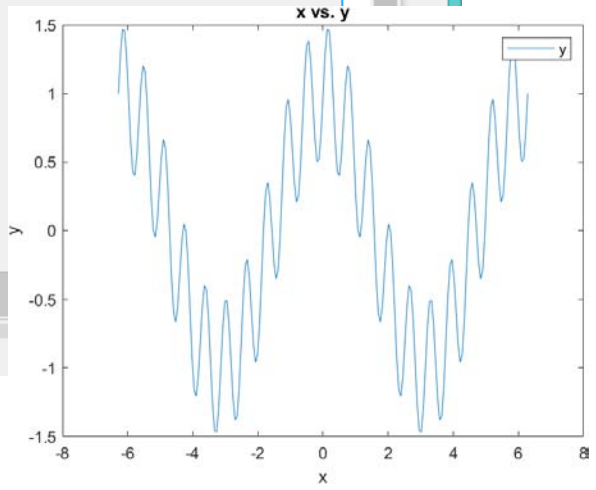
fill in the form fields

output

switch to display or hide the code

output

select

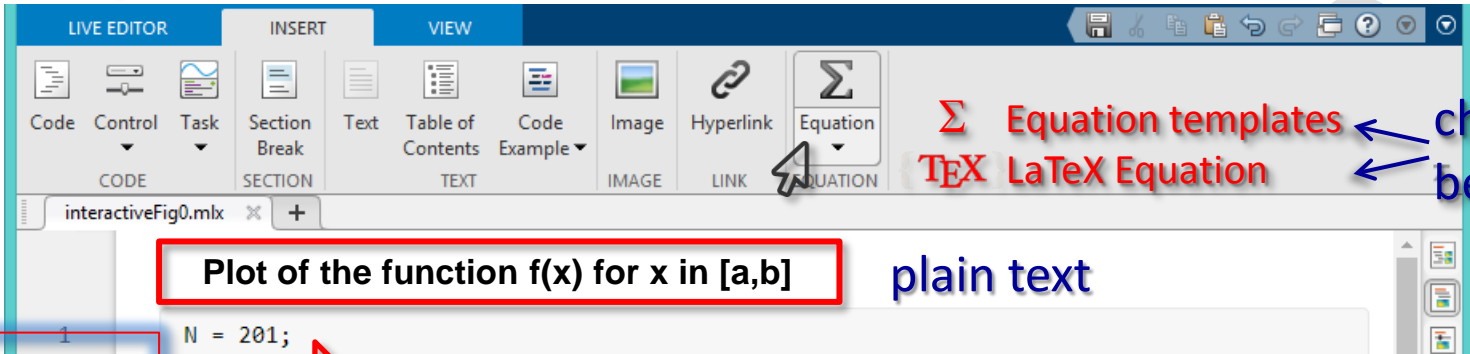


Select data

X ▼

Y ▼

A Live Script can contain **math formulas** in the text

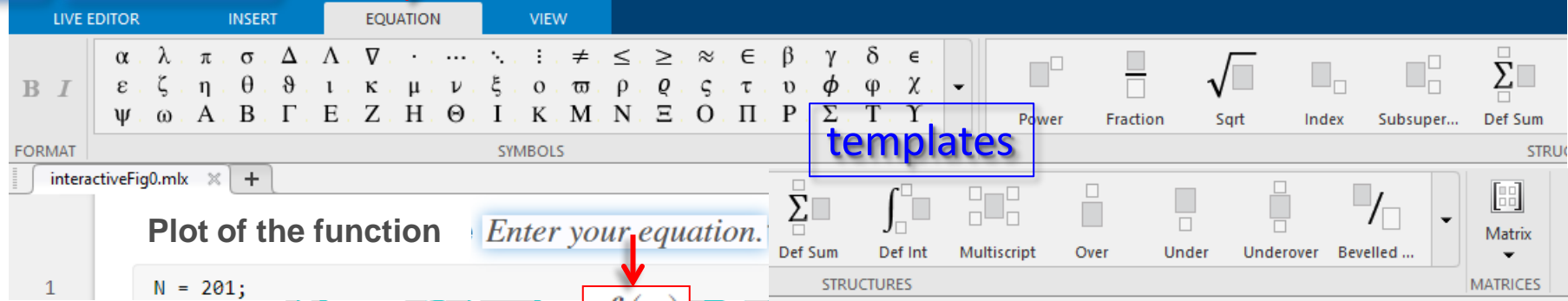
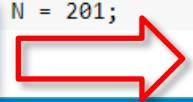


Σ Equation templates
 Σ TeX LaTeX Equation
 choose between

Plot of the function f(x) for x in [a,b]

plain text

1 Equation

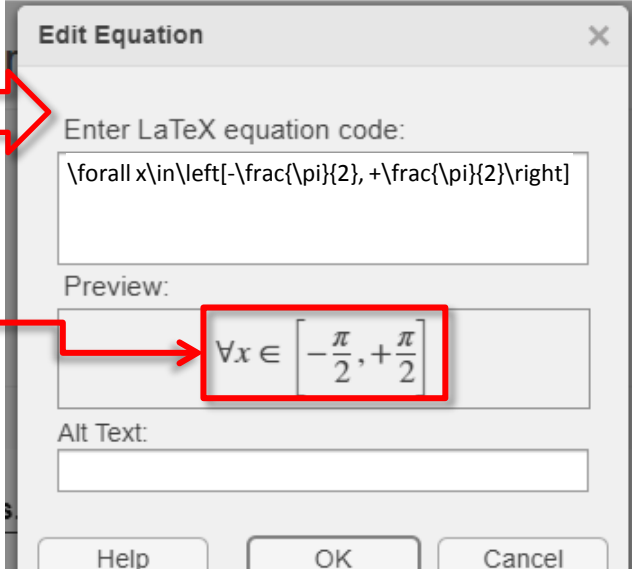


templates

Plot of the function *Enter your equation.*

$f(x)$

2 LATEX



result

$\forall x \in \left[-\frac{\pi}{2}, +\frac{\pi}{2}\right]$

LaTeX* (pronounced latek) is a software system for publishing documents. It is a **WYSIWYM** (What You See Is What You Mean) *markup language*, unlike Word which is **WYSIWYG** (What You See Is What You Get).

* **LaTeX**: short for Lamport's TeX, named after **Leslie Lamport** who in 1994 added a library of macros to TeX (from the Greek $\tau\epsilon\chi$ =technology and art), the typesetting system created by Donald Knuth in 1978.

download: [LaTeX_GreekLetters_SpecialCharacters_MATLAB.pdf](#), [1994_LaTeX_UserGuide_ReferenceManual.pdf](#), ...

We want to write a MATLAB program to display the plot of a function $y=f(x)$ in an interval, using the **Live Editor** with the addition of **interactive commands (Control)**.

The screenshot shows the MATLAB Live Editor interface. The 'CONTROL' button in the 'INSERT' tab is highlighted with a red box. Below it, a tooltip reads 'Insert slider, drop down, or other control'. The code editor contains the following code:

```
1 N = |  
2 a = -2*pi;  
3 b = +2*pi;  
4 pf=@(t) cos(t) + 0.5*sin(10*t);
```

A red box highlights the text 'Plot of the function $f(x) \forall x \in [-2\pi, +2\pi]$ '. A red arrow points from this text to the 'N =' line in the code, with the annotation 'insert **Control**: numeric slider here'.

fill in
the fields

The screenshot shows the 'Control' dialog box for the variable 'N'. The dialog has the following sections:

- LABEL**: 'Enter text to display when code is hidden'. The label is 'value of N:'.
- VALUES**: 'Enter value or select workspace variable'.
 - Min: 1
 - Max: 201
 - Step: 1
- DEFAULTS**: 'Enter or select from workspace'. The default value is 101.
- EXECUTION**:
 - Run On: Value changed
 - Run: Current section

LIVE EDITOR **INSERT** **VIEW**

Code Control Task Section Break Text Table of Contents Code Example Image Hyperlink Equation

CODE SECTION TEXT IMAGE LINK EQUATION

interactiveFig0.mlx* +

Plot of the function $f(x) \forall x \in [-2\pi, +2\pi]$

numeric slider

1 N = 133 1 201

display the value of N

N = 64 1 201

adding here a ';' the value of N is not displayed

2 a = -2*pi;

3 b = +2*pi;

4 pf=@(t) cos(t) + 0.5*sin(10*t);

5 x = linspace(a,b,N)';

6 y = pf(x);

7 %plot(x,y

insert **Control: drop down menu**

4 %pf = eval(@(t)cos(t)+0.5*sin(...))

pf = str2func(pf, @(t)cos(t)+0.5*sin(...))

the **eval()** function evaluates its string argument as a MATLAB code. **str2func()** converts its string argument into a function handle

insert **Control: edit field here**

1 N = 64 1 201

2 a = a -pi

3 LABEL

Enter text to display when code is hidden

Label Enter a:

4 TYPE

Data type MATLAB code

5 DEFAULTS

6 Enter or select from workspace

Default value -pi

7 EXECUTION

Run Current section

fill in the fields

6 pf = funct

7 @(t)co

8 x = linsp

y = pf(x)

%plot(x,y

▼ LABEL

Enter text to display when code is hidden

Label pf:

▼ ITEMS

Enter labels or values to add to drop down

h3 = plo

Item labels

@(t)cos(t)+0.5*sin(10*t)

@(t)cos(t/2)

Item values

"@(t)cos(t)+0.5*sin(10*t)"

"@(t)cos(t/2)"

Select a variable to add its content to drop down

Variable select

▼ DEFAULTS

Default item @(t)cos(t)+0.5*si...

▼ EXECUTION

Run Current section