# MATH 22AL
# Lab # 3 .

## Starting MATLAB and Recording your work

## 1 Objectives

**In this lab, you will explore the following topics using MATLAB:**

- **Using the plotting commands of MATLAB to create graphs.**
- **Creating and using m-files**
- **Exploring linear transformations in $\mathbb{R}^2$**
- **Exploring Matrix Multiplication.**

# Two computers involved: the server and the one runs graphical MATLAB

## Saving and Copying your work in LAB3.text file on MATH Dept Computers

**In this lab, you will plot and create some graphs. Textmatlab as we have used in previous sessions can not display graphs. For this reason you need to use MATLAB with graphical interface. It needs to be installed either on your own laptop (It is free for all UCD students), or already installed on computers in room 1118. But room 1118 is too small with only 16 computers for 520 students in this course. Using your own computer is a better way to do the LAB 3.**
**Either way, your lab work still need to be submitted through the server. Below are several methods for transferring your MATLAB output to math department server.**

### Methods for Copying or Saving MATLAB Results

**1. Record work using the `diary` Command**

To record all commands and outputs automatically start your work in the command window of MATLAB:

```
diary MySession.text    % Start recording
(Demostration only)
A = rand(3);
inv(A)
...
diary off                 % Stop recording
```

This creates a text file named `MySession.text` in the current MATLAB folder.
Open your LAB3.text file, your recorded work will show up in a new window within MATLAB called editor window.

**2. Copy from Editor Window of MATLAB**

1. Highlight the desired output in the **Editor Window**.
2. Press `Command + C` (Mac) or `Ctrl + C` (Windows) to copy.
3. Login to MATH Dept server by sending ssh username@point.math.ucdavis.edu to the terminal
4. At the terminal window, let server run pico LAB3.text
5. Use `Command + V` (Mac) or `Ctrl + V` (Windows) to paste your work.
6. Use Ctrl + O to save in pico
7. Use Ctrl + X to exit from pico
8. Submit using the server command submitm22al LAB3.text

Note: When copying text from a word processor into MATLAB or a LaTeX document, invisible characters such as non-breaking spaces or "smart quotes" can cause unexpected errors. Using a plain-text editor avoids these problems.
For MAC OS: **TextEdit (Plain Text Mode)**. Use `Format` → `Make Plain Text` to ensure no formatting.
For Windows : **Notepad** is the option.

## Starting MATLAB and Recording your work

### 2

**The following steps will help you start and record your work and save and submit it successfully.**

- **In Computer LAB (1118 MSB)**
  - **Start a MATLAB Session by clicking the MATLAB Icon.**
  - **In the command window (the one with >>) of MATLAB type diary LAB3.text**
- **Open MATLAB on your own computer. Type diary LAB3.text in the command window (the one with >>) then Press Enter**

---

```
type    % First Name:    Example: % Jennifer
type    % Last Name:     Example: % Brown
type    % Date:          Example: % 10/17/2025.
type    % Username:      m22als?-??. ( this your username replace "?")
```

# Plotting Graphs

## 3

**Note:Plot windows are not displayed when you connect remotely via `ssh`.**
**To view graphical plots, you must run MATLAB on your own computer using your locally installed version ( it is free)**
**OR use one of the terminals in the Mathematics Department computer lab (Room 1118 MSB).**

### 3.1 Generating 2-D Graphs

type vt1=datestr(now)

To graph a line segment passing through the points $(-1, 4)$ and $(2, -3)$,

type x = [-1 2]; y = [4 -3];
type plot(x, y)

Use the `axis` command to rescale the axes so that the line segment does not take up the entire window.

type axis([-3 6 -4 5])

You may chose a color for your lines by specifying it with a string.

type plot(x, y, 'r')

The line is now red! You can also make it dashed or dotted by specifying it with a string.

type plot(x, y, 'g- -')

The line is now dashed and green!
Later on you can visit
`http://www.mathworks.com/help/matlab/ref/plot.html`
for information on line styles.
You can also add a title and label to the graph.

type title('Plotting x and y')
type xlabel('x axis')
type ylabel('y axis')

To add a new graph to the same plot without removing the original graph, use the `hold` command.

type hold on
type y = -y
type plot(x, y, 'r')
type x = -x; plot(x, y, 'g')
type axis([-5 5 -5 5])

To stop adding graphs,

type hold off

## Plotting Graphs

**3.2**

**Of course, graphing three line segments is not very interesting.**
**Let us graph some more complicated functions.**

**type** t = linspace(0, 4, 100);
**type** y = exp(t); plot(t, y, 'r'), hold on
**type** y = -2*exp(t); plot(t, y, 'b')
**type** y = 3* exp(t); plot(t, y, 'm'), hold off

**The comma (in the second and forth line above) allows you to put multiple commands on the same line, unlike the semicolon which prevents printing output. If MATLAB printed a very long vector, then you forgot a semicolon somewhere.**

## 3-D Mesh Surface Plots

**3.3**

**If you have a function $Z(x, y)$, you can plot its surface in 3-D using the commands `mesh`, `meshgrid`, and `surf`. For more information, use the `help` command or see the documentation online. Here are some examples.**

**type** A = eye(9); mesh(A) **It's the identity matrix in 3-D!**
**type** [t,s] = meshgrid(-1:.2:4, -3:.2:3);
**type** z =sin(s.*t); mesh(z)

# M-Files

## 4

### How to create M-Files

MATLAB can execute a sequence of statements stored in an ordinary text file with a `.m` extension. Because of this extension, such files are called "m-files".
One type of m-file is the function m-file.
The first line of a function m-file must define the m-file as a function, specify its name, and specify its input and output variable names. A function m-file's function name and filename must be identical. For example, a function named "changerows" must be stored in a file named "changerows.m". When a function has more than one output variable, the output variables are enclosed by brackets: For example, `[m, n] = size(A)`. If your m-file is not in the working directory (this will default to your home directory), you must specify the path to it.

## Overview

The **Editor Window** in MATLAB is where you can write, save, and run your scripts (`.m`) and functions. It function like pico, but it is part of MATLAB. No terminal operation involed here. There are several ways to open it.

**Method 1: Using the Toolbar**

1. Open MATLAB.
2. On the **Home** tab, click **New Script**.
3. A new window `Untitled.m` opens -this is the Editor.

**Method 2: Using a Command**

1. In the **Command Window**, type `edit` and press Enter.
2. To open or create a specific file: `edit myscript.m`.

**Method 3: From the Current Folder**

1. In the **Current Folder** panel, double-click a `.m` file.
2. Or type `open myscript.m` in the Command Window.

**Method 4: Keyboard Shortcuts**

- Windows/Linux: **Ctrl + N**
- macOS: **Command + N**

### If you are using Math Dept computers, this is the instructions:

**Open a new editor window of MATLAB,**

Open a new editing window of MATLAB and save it as If you are using Math Dept computers, type the following exactly as you see

**Type the following in Editor window**

```
function rowchange(A, c, d)
display(A)
r = A(c,:);
A(c,:)  = A(d,:);
A(d,:)  = r;
display(A)
```

Save the file,
Go back to MATLAB session that you are working on.
And enter the following commands:

**(Note The `display` function at the end of the program file prints a variable to the screen.)**

**type** **rng shuffle**
**type** **A = round(10*rand(5))**
**type** **rowchange(A, 1, 4)**

**Try it for other rows. For example rows 2 and 3 by typing**

**type** **rowchange(A, 2, 3)**

## Linear Transformations

### 5

Create the following m-file and save it as `lintr2.m`
This means use MATLAB's editor to open a new editing window and save it as `lintr2.m`, then type the following in that window:

**Type the following in the file `lintr2.m`**

```
function lintr2(obj, A)
plot(obj(1,:), obj(2,:), 'k')
hold on
y = A * obj;
display(y)
plot(y(1,:), y(2,:), 'r')
hold off
```

This m-file plots the points in the matrix `obj`, and then plots the points in the matrix `A*obj`. Now in the first window that MATLAB session is running, Enter the following matrix; it describes the vertices in a drawing of a house.

type house = [0 0; 0 1; 1 2; 2 1; 2 0; 3 0; ...
type 1.75 0; 1.75 .5; 1.25 .5; 1.25 0; -1 0]'

type lintr2(house, eye(2))

Create a matrix that reflects across the $x$-axis
type reflectX = [1 0; 0 -1]
type lintr2(house, reflectX)

The house should be in black and its reflection about the $x$-axis in red.
Do the same thing for the $y$-axis:
type reflectY = [-1 0; 0 1]
type lintr2(house, reflectY)

To rotate the house 30 degrees counterclockwise, type the following.
type rot30 = [cos(pi/6) -sin(pi/6); sin(pi/6) cos(pi/6)]
type lintr2(house, rot30)
Write code and rotate the house 45 degrees .

## Linear Transformations

### 6

**Now we will consider shear.**

```
type    shearX = [1 2; 0 1]
type    lintr2(house, shearX)

type    shearY = [1 0; 0 3]
type    lintr2(house, shearY)

type    lintr2(house, reflectY * shearX)
```

**If the reflection is applied first,**
```
type    s5a1 = 'reflectY'
```

**If the shear is applied first,**
```
type    s5a2 = 'shearX'
```

**Now try the other order.**
```
type    lintr2(house, shearX * reflectY)
```
**In which order are the transformations being applied here?**

**If the reflection is applied first,**
```
type    s5a2 = 'reflectY'
```

**If the shear is applied first,**
```
type    s5a1 = 'shearX'
```

```
type    vtm=datestr(now)
```

## Matrix Multiplication

### 7

**Matrix multiplication can be expressed and observed in several ways; we will see a few of them.**

**type** **A = [-2 1 0; 2 0 1; 8 5 3; 1 2 -3]**
**type** **B = [1 5 -2; 0 -7 5; 1 3 -2 ]**

**Enter the following commands. Some will give errors; this is expected.**

**type** **A * B**
**type** **B * A**
**type** **inv(A)**
**type** **inv(B)**
**type** **inv(A*A')**
**type** **inv(A'*A)**
**type** **size(A'*A)**
**type** **size(A*A')**

**There should have been two errors. Explain each of them:**

**Comment:** `% [an explanation of why the first error occurred]`
**Comment:** `% [an explanation of why the second error occurred]`

**Let $A$ and $B$ be $\ell \times m$ and $m \times n$ matrices, respectively.**
**Then the matrix product $C = A \cdot B$ exists.**
**There are multiple ways to view matrix multiplication:**

**type** **C = A * B**

1. **The element in the $i$the row and $j$the column of $C$, $c_{ij}$, is equal to the dot product of the $i$the row of $A$ with the $j$th row of $B$. That is, $c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}$.**
2. **The $k$th column of $C$ is $A$ times the $k$the column of $B$.**
3. **The $k$the row of $C$ is the $k$the row of $A$ times $B$.**

## Matrix Multiplication

### 8

**Find the entry in position $(2,3)$ of the matrix $C$ in two ways.**

**type** **C(2,3)**

**type** **A(2,:) * B(:,3)**

**Find the second column of $C$ in two ways.**

**type** **C(:,2)**

**type** **A * B(:,2)**

**Find the second row of $C$ in two ways.**

**type** **C(2,:)**

**type** **A(2,:) * B**

**Find $C$ in two different ways.**

**type** **C = A * B**

**type** **A(:,1) * B(1,:) + A(:,2) * B(2,:) + A(:,3) * B(3,:)**

**Do you think you can justify why**

```
AB == A(:,1) * B(1,:)  + A(:,2) * B(2,:)  + A(:,3) * B(3,:)
```

**(You do not have to answer.)**

**type** **vt2=datestr(now)**

**type** **vt3=vt2-vt1**

**type** **vt4=vt2-vtm**

## This is the end of LAB 3

**You reviewed and learned about:**

1. **Plotting in MATLAB**
2. **Linear transformation**
3. **Matrix multiplication**

**Please edit your LAB3.text file using the editor window, remove all errors you typed and make sure to add needed comments then transfer, save and submit the LAB3.text file (see page 2 and 3 of this manual).**