Welcome to the MATLAB minicourse!

Instructor: Anila Yadavalli
anilayadavalli@gmail.com

What is MATLAB?

From website: MATLAB® is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world. The heart of MATLAB is the MATLAB language, a matrix-based language allowing the most natural expression of computational mathematics.

the edge program

What is MATLAB?

From website: MATLAB® is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world. The heart of MATLAB is the MATLAB language, a matrix-based language allowing the most natural expression of computational mathematics.

Translation: computational software based on mathematics + easy to use environment

What is MATLAB?

From website: MATLAB® is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world. The heart of MATLAB is the MATLAB language, a matrix-based language allowing the most natural expression of computational mathematics.

Translation: computational software based on mathematics + easy to use environment

Regardless of your field of mathematics, you will probably use MATLAB at some point!

What is MATLAB?

From website: MATLAB® is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world. The heart of MATLAB is the MATLAB language, a matrix-based language allowing the most natural expression of computational mathematics.

Translation: computational software based on mathematics + easy to use environment

Regardless of your field of mathematics, you will probably use MATLAB at some point!

Exercise 1: Basic Commands

1. Compute 3×5 by typing `3*5` and hitting enter.

Exercise 1: Basic Commands

1. Compute 3×5 by typing `3*5` and hitting enter.
2. Store the calculation 3×5 in a variable called `m`.

the edge program

Exercise 1: Basic Commands

1. Compute 3×5 by typing `3*5` and hitting enter.
2. Store the calculation 3×5 in a variable called `m`.
3. Enter the command `m = m+1` and see what happens.

the edge program

Exercise 1: Basic Commands

1. Compute 3×5 by typing `3*5` and hitting enter.
2. Store the calculation 3×5 in a variable called `m`.
3. Enter the command `m = m+1` and see what happens.
4. Create a variable named `y` that has the value `m/2`.

Exercise 1: Basic Commands

1. Compute 3×5 by typing `3*5` and hitting enter.
2. Store the calculation 3×5 in a variable called `m`.
3. Enter the command `m = m+1` and see what happens.
4. Create a variable named `y` that has the value `m/2`.
5. Enter `k = 8-2;` (notice the semicolon). What happens?

Exercise 1: Basic Commands

1. Compute 3×5 by typing `3*5` and hitting enter.
2. Store the calculation 3×5 in a variable called `m`.
3. Enter the command `m = m+1` and see what happens.
4. Create a variable named `y` that has the value `m/2`.
5. Enter `k = 8-2;` (notice the semicolon). What happens?
6. Use the UP arrow key to return to the command `m=3*5`. Change this line to `m=3*k`

Exercise 1: Basic Commands

1. Compute 3×5 by typing `3*5` and hitting enter.
2. Store the calculation 3×5 in a variable called `m`.
3. Enter the command `m = m+1` and see what happens.
4. Create a variable named `y` that has the value `m/2`.
5. Enter `k = 8-2;` (notice the semicolon). What happens?
6. Use the UP arrow key to return to the command `m=3*5`. Change this line to `m=3*k`
7. Check the value of `y`. Did it recalculate when `m` was changed?

the edge program

Exercise 1: Basic Commands

1. Compute 3×5 by typing `3*5` and hitting enter.
2. Store the calculation 3×5 in a variable called `m`.
3. Enter the command `m = m+1` and see what happens.
4. Create a variable named `y` that has the value `m/2`.
5. Enter `k = 8-2;` (notice the semicolon). What happens?
6. Use the UP arrow key to return to the command `m=3*5`. Change this line to `m=3*k`
7. Check the value of `y`. Did it recalculate when `m` was changed?
8. Use the UP arrow to rerun your code to return the updated value of `y`.

the edge program

Exercise 1: Basic Commands

1. Compute 3×5 by typing `3*5` and hitting enter.
2. Store the calculation 3×5 in a variable called `m`.
3. Enter the command `m = m+1` and see what happens.
4. Create a variable named `y` that has the value `m/2`.
5. Enter `k = 8-2;` (notice the semicolon). What happens?
6. Use the UP arrow key to return to the command `m=3*5`. Change this line to `m=3*k`
7. Check the value of `y`. Did it recalculate when `m` was changed?
8. Use the UP arrow to rerun your code to return the updated value of `y`.

the edge program

Exercise 2: Assigning Variables

1. Assign the value 8 to the variable `a`, and the value -2 to the variable `A`.

Exercise 2: Assigning Variables

1. Assign the value 8 to the variable `a`, and the value -2 to the variable `A`.
2. Create a new variable `avg` that stores the average of `a` and `A`.

Exercise 2: Assigning Variables

1. Assign the value 8 to the variable `a`, and the value -2 to the variable `A`.
2. Create a new variable `avg` that stores the average of `a` and `A`.
3. Try creating the variable `3sq = 9`. What happens?

the edge program

Exercise 2: Assigning Variables

1. Assign the value 8 to the variable `a`, and the value -2 to the variable `A`.
2. Create a new variable `avg` that stores the average of `a` and `A`.
3. Try creating the variable `3sq = 9`. What happens?
4. Which of the following variable names are valid?

`_ans_1` `ans_1` `ans1` `ans_`

the edge program

Variables: Summary

1. In MATLAB, variable names can contain any combinations of letters, numbers and the underscore (_) symbol.

Variables: Summary

1. In MATLAB, variable names can contain any combinations of letters, numbers and the underscore (_) symbol.
2. All variable names must start with a letter.

Variables: Summary

1. In MATLAB, variable names can contain any combinations of letters, numbers and the underscore (`_`) symbol.
2. All variable names must start with a letter.
3. Avoid using built-in function and variable names (more on this later). (i.e. avoid `pi, i, j, inf`)

Variables: Summary

1. In MATLAB, variable names can contain any combinations of letters, numbers and the underscore (`_`) symbol.
2. All variable names must start with a letter.
3. Avoid using built-in function and variable names (more on this later). (i.e. avoid `pi, i, j, inf`)
4. To clear a variable:
   - `clear variable_name1 variable_name2`
   - `clear all`
   - Right click on variable name in workspace and select Delete.

the edge program

Variables: Summary

1. In MATLAB, variable names can contain any combinations of letters, numbers and the underscore (`_`) symbol.
2. All variable names must start with a letter.
3. Avoid using built-in function and variable names (more on this later). (i.e. avoid `pi, i, j, inf`)
4. To clear a variable:
   - `clear variable_name1 variable_name2`
   - `clear all`
   - Right click on variable name in workspace and select Delete.
5. Types of variables
   - integer: e.g.) -1, 5, 10000
   - floating point: e.g.) 3.5000, `pi`, $\sqrt{10}$.
   - array: (later)
   - string: e.g.) `name = ‘‘Anila"`

the edge program

Exercise 3a: Saving and Loading Variables

1. Assign the following variables: `a = 5`, `b = 7`, `c = 3.14`

Exercise 3a: Saving and Loading Variables

1. Assign the following variables: `a = 5`, `b = 7`, `c = 3.14`
2. Enter the command `save variables.mat`

the edge program

Exercise 3a: Saving and Loading Variables

1. Assign the following variables: `a = 5`, `b = 7`, `c = 3.14`
2. Enter the command `save variables.mat`
3. Use the command `clear` to clear the workspace. What happened to the variables saved in the workspace?

Exercise 3a: Saving and Loading Variables

1. Assign the following variables: `a = 5`, `b = 7`, `c = 3.14`
2. Enter the command `save variables.mat`
3. Use the command `clear` to clear the workspace. What happened to the variables saved in the workspace?
4. Type `a` and hit `Enter`. What happens?

Exercise 3a: Saving and Loading Variables

1. Assign the following variables: `a = 5`, `b = 7`, `c = 3.14`
2. Enter the command `save variables.mat`
3. Use the command `clear` to clear the workspace. What happened to the variables saved in the workspace?
4. Type `a` and hit `Enter`. What happens?
5. Clear the command window with the command `clc`

the edge program

Exercise 3a: Saving and Loading Variables

1. Assign the following variables: `a = 5`, `b = 7`, `c = 3.14`
2. Enter the command `save variables.mat`
3. Use the command `clear` to clear the workspace. What happened to the variables saved in the workspace?
4. Type `a` and hit `Enter`. What happens?
5. Clear the command window with the command `clc`
6. Now enter the command `load variables.mat` and repeat #4. What happens?

the edge program

Exercise 3a: Saving and Loading Variables

1. Assign the following variables: `a = 5`, `b = 7`, `c = 3.14`
2. Enter the command `save variables.mat`
3. Use the command `clear` to clear the workspace. What happened to the variables saved in the workspace?
4. Type `a` and hit `Enter`. What happens?
5. Clear the command window with the command `clc`
6. Now enter the command `load variables.mat` and repeat #4. What happens?
7. Compute `a*b`.

the edge program

Exercise 3b: Saving and Loading Variables

Note

You can save variables in your workspace using the command `save filename`*. You can load them using* `load filename`*. You can also load/save just a subset of variables.*

the edge program

Exercise 3b: Saving and Loading Variables

Note

You can save variables in your workspace using the command `save filename`. You can load them using `load filename`. You can also load/save just a subset of variables.

1. Assign the following variables: `k = 7`, `j = 10`, `avg= (k+j)/2`, `diff = j-k`
2. Save your variables using `save vars.mat`
3. Clear the workspace and command window.

the edge program

Exercise 3b: Saving and Loading Variables

4. Load just `avg` by entering the command `load vars avg`. Check that the value of avg was loaded.

Exercise 3b: Saving and Loading Variables

4. Load just `avg` by entering the command `load vars avg`. Check that the value of avg was loaded.
5. Clear the workspace and command window, and reload all of the variables.

Exercise 3b: Saving and Loading Variables

4. Load just `avg` by entering the command `load vars avg`. Check that the value of avg was loaded.
5. Clear the workspace and command window, and reload all of the variables.
6. Save just `avg` and `diff` using the command `save('avgdiff.mat', 'avg', 'diff')`

the edge program

Exercise 3b: Saving and Loading Variables

4. Load just `avg` by entering the command `load vars avg`. Check that the value of avg was loaded.
5. Clear the workspace and command window, and reload all of the variables.
6. Save just `avg` and `diff` using the command `save('avgdiff.mat', 'avg', 'diff')`
7. Clear the workspace and command window. Load `avgdiff`

the edge program

Exercise 3b: Saving and Loading Variables

4. Load just `avg` by entering the command `load vars avg`. Check that the value of avg was loaded.
5. Clear the workspace and command window, and reload all of the variables.
6. Save just `avg` and `diff` using the command `save('avgdiff.mat', 'avg', 'diff')`
7. Clear the workspace and command window. Load `avgdiff`
8. Compute `avg*diff`

the edge program

Exercise 3b: Saving and Loading Variables

4. Load just `avg` by entering the command `load vars avg`. Check that the value of avg was loaded.
5. Clear the workspace and command window, and reload all of the variables.
6. Save just `avg` and `diff` using the command `save('avgdiff.mat', 'avg', 'diff')`
7. Clear the workspace and command window. Load `avgdiff`
8. Compute `avg*diff`

Note

See more at https://www.mathworks.com/help/matlab/ref/save.html#d123e1196753

the edge program

Exercise 4: MATLAB Built-in Functions

the edge program

Exercise 4: MATLAB Built-in Functions

1. Type `pi` and hit `Enter`.

the edge program

Exercise 4: MATLAB Built-in Functions

1. Type `pi` and hit `Enter`.
2. Create a new variable called `x` with the value $\frac{\pi}{2}$.

Exercise 4: MATLAB Built-in Functions

1. Type `pi` and hit `Enter`.
2. Create a new variable called `x` with the value $\frac{\pi}{2}$.
3. Create a new variable called `y` with the value $\sin(x)$.

Exercise 4: MATLAB Built-in Functions

1. Type `pi` and hit `Enter`.
2. Create a new variable called `x` with the value $\frac{\pi}{2}$.
3. Create a new variable called `y` with the value $\sin(x)$.
4. Use the command `sqrt` to calculate $\sqrt{-9}$.

the edge program

Exercise 4: MATLAB Built-in Functions

1. Type `pi` and hit `Enter`.
2. Create a new variable called `x` with the value $\frac{\pi}{2}$.
3. Create a new variable called `y` with the value $\sin(x)$.
4. Use the command `sqrt` to calculate $\sqrt{-9}$.

the edge program

Exercise 4: MATLAB Built-in Functions

1. Type `pi` and hit `Enter`.
2. Create a new variable called `x` with the value $\frac{\pi}{2}$.
3. Create a new variable called `y` with the value $\sin(x)$.
4. Use the command `sqrt` to calculate $\sqrt{-9}$.

Note

MATLAB automatically rounds to 4 decimal places. You can change this using the commands `format long` and `format short`.

the edge program

Other Things: MATLAB Documentation

If you ever forget the command for something, or come across a command you don't know in someone else's code, you can always look up the documentation for that command.

the edge program

Other Things: MATLAB Documentation

Navigate to the documentation page for `randi`:
`https://www.mathworks.com/help/matlab/ref/randi.html`.

Other Things: MATLAB Documentation

Navigate to the documentation page for `randi`: `https://www.mathworks.com/help/matlab/ref/randi.html`.

Use it to create a variable `x` which has a random value between 1 and 60.

Other Things: MATLAB Documentation

Navigate to the documentation page for `randi`: `https://www.mathworks.com/help/matlab/ref/randi.html`.

Use it to create a variable `x` which has a random value between 1 and 60.

You can easily find the documentation page by entering the command `doc function_name` into the command window. Try opening the documentation page for the function `round` using this command.

the edge program

Other Things: Live Scripts

A live script contains formatted text, code, and section breaks. Unlike the command window, it allows you to run multiple lines of code at once.

You can also break it into sections and run individual sections.

From now on, we will work through exercises using the live script format. Let's begin!

Open the file Vectors_and_Arrays.mlx

the edge program

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$