Week 11 MATLAB Activities

Contents

Time required: 120 minutes

1. Create a MATLAB script named **Wk11Lastname.m**
2. Save all programs in this script.
3. Include your name and date at the top of the script file as comments.
4. Put a Section Break between each program.

Reading

Reading

Matlab A Practical Introduction to Programming and Problem Solving (Stormy Attaway)

Sections 5.3, 5.4

Tutorial 1: Files

MAT files are a proprietary file format used by MATLAB. These files are binary files that store variables, data, and other information generated or manipulated within MATLAB.

- **Binary Format:** MAT files are binary files, meaning that they store data in a format that is not easily readable by humans. This binary format allows for efficient storage and retrieval of complex data structures.
- **Variable Storage**: MAT files can store MATLAB variables, including arrays, matrices, structures, cells, and more. These variables can be of various data types, such as numeric, character, or logical.
- **Usage**: MAT files are commonly used for saving and loading data between MATLAB sessions. They are useful for preserving the state of variables, sharing data with collaborators, or storing results for future analysis.
- **Functions for Handling .mat Files:** MATLAB provides functions like `save` and `load` to save variables to and load variables from MAT files. These functions allow users to store and retrieve data efficiently.

Saving Data:

- Syntax for Saving:
    - Use the save function.
    - Example: save('filename.mat', 'variable_name')
- Saving Variables:
    - Specify the variable to save in single quotes.
    - To save multiple variables, separate them with commas.
    - Example: save('data.mat', 'matrix_A', 'vector_B')
- Saving Specific Variables:
    - Save only specific variables by providing their names.
    - Example: save('selected_data.mat', 'variable1', 'variable2')

```
% Saving a single variable
matrix_A = rand(3, 3);
save('data.mat', 'matrix_A');

% Saving multiple variables
vector_B = [1, 2, 3];
save('data.mat', 'matrix_A', 'vector_B');

% Saving specific variables
variable1 = 'Hello';
variable2 = [4, 5, 6];
save('selected_data.mat', 'variable1', 'variable2');
```

Loading Data:

- Syntax for Loading:
    - Utilize the load function.
    - Example: load('filename.mat')
- Loading Specific Variables:
    - Load specific variables from a file.
    - Example: load('selected_data.mat', 'variable1', 'variable2')
- Checking Loaded Variables:
    - After loading, check the workspace for the loaded variables.
    - Example: whos

```
%%
% Loading all variables from a file
load('data.mat');

% Loading specific variables from a file
load('selected_data.mat', 'variable1', 'variable2');

% Checking loaded variables in the workspace
whos
```

Format Specifier

%s - print a string.

%c - print a single character.

%d - print a whole number.

%f - print a floating point number.

\n - print a new line (go to the next line to continue printing)

\t - print a tab.

\\ - print a slash.

%% - print a percent sign.

Text File Format:

- Use fprintf and fscanf for saving and loading text files.
- Example: fprintf('output.txt', '%s', 'Hello, MATLAB!')

Saving text to a file.

```
%% Saving text to a file

text_to_save = 'MATLAB rules!.';

% Open file for writing
fileID = fopen('output.txt', 'w');

% Save text to the file
fprintf(fileID, '%c', text_to_save);

% Close the file
fclose(fileID);

disp('Text saved to output.txt.');
```

Loading text from a file.

```
%% Loading text from a file

% Open file for reading
fileID = fopen('output.txt', 'r');

% Load text from the file
loaded_text = fscanf(fileID, '%c');
% Close the file
fclose(fileID);

disp('Text loaded from output.txt:');
disp(loaded_text);
```

Tutorial 2: Save and Load Vectors from Files

A MATLAB vector can be saved to a text file.

```
%% Save a vector to a text file
% Example vector
myVector = [3.14, 2.71, 1.618, 0.577];

% Open the file for writing
fileID = fopen('myVector.txt', 'w');

% Check if the file is successfully opened
if fileID == -1
    error('Error opening the file for writing.');
end

% Specify the format with '%.4f' to display up to 4 decimal places
fprintf(fileID, '%.4f ', myVector);

% Close the file
fclose(fileID);
disp("Vector saved to text file")
```

A text file with a vector can be opened as numbers.

```
% Open a text file as a vector

% Open the file for reading
fileID = fopen('myVector.txt', 'r');

% Check if the file is successfully opened
if fileID == -1
    error('Error opening the file for reading.');
end

% Read the vector from the text file into floats
loadedVector = fscanf(fileID, '%f');

% Close the file
fclose(fileID);

% Display the loaded vector using a for loop and fprintf
fprintf('Loaded Vector elements: ');
for i = 1:length(loadedVector)
    fprintf('%.2f ', loadedVector(i));
end
fprintf('\n');
```

Example run:

```
Vector saved to text file
Loaded Vector elements: 3.14 2.71 1.62 0.58
```

Assignment 1: Sales Figures

The sales (in billions) for a division of the ABC Corporation for each of the four quarters of 2021 are stored in a file called "salesfigs.dat".

1.2 1.4 1.8 1.3

1. Create this file (Type the numbers in the Editor. Save As "salesfigs.dat").
2. Write a script that will load the data from the file into a vector.
3. Display the vector in a nice format.

Example run:

```
Sales Data
    1.2000    1.4000    1.8000    1.3000
```

Assignment 2: Wind Chill Factor

NOTE: A function can be created as a separate file, or as part of the same file. If the function is in the same file, it must be below wherever the function is called.

The Wind Chill Factor (WCF) measures how cold it feels with a given air temperature **T** (in degrees Fahrenheit) and wind speed **V** (in miles per hour).

One formula for WCF is

$$\boldsymbol{WCF = 35.74 + 0.6215T - 35.75 \times (V^{0.16}) + 0.4275T(V^{0.16})}$$

1. Write a function to receive the temperature and wind speed as input arguments, and return the WCF.
2. Create this function in the same file at the end of the file.
3. Create a **while true** loop that asks for input and calls the function. This loop will be around the entire program.
4. How to check for the exit command at the end of the loop.

Pseudocode

```
while true
    % Get user input for temperature in Fahrenheit

    % Check for exit command, -99 is unlikely number for temperature
    if temperature == -99
        disp('Exiting program.');
        break;
    end

    % Get user input for wind speed in miles per hour

    % Call the WCF function, return WCF

    % Display the result
end
```

Example run:

```
Enter temperature in Fahrenheit (type "-99" to end)70
Enter wind speed in mph: 25
Wind Chill: 69.50

Enter temperature in Fahrenheit (type "-99" to end)-99
Exiting program.
```

Check your numbers at the National Weather Service [Wind Chill Calculator]

Assignment 3: Beaufort Wind Scale

NOTE: A function can be created as a separate file, or as part of the same file. If the function is in the same file, it must be below wherever the function is called.

The Beaufort Wind Scale is used to characterize the strength of winds. The scale uses integer values and goes from a force of 0, which is no wind, up to 12, which is a hurricane.

Force	
0	There is no wind
1-6	There is a breeze
7-9	There is a gale
10-11	There is a storm

12	There is a hurricane

1. Generate a random integer force value between 0 and 12.
2. Create a function takes the random force value and returns the Beaufort Wind Scale description.
    a. Create this function in the same file at the end of the file.
3. Print a message regarding the type of wind that force represents.

Example run:

```
Beaufort Wind Scale: 8
There is a gale
Beaufort Wind Scale: 1
There is a breeze
Beaufort Wind Scale: 3
There is a breeze
Beaufort Wind Scale: 7
There is a gale
```

Assignment 4: conevol Function

This is a problem that you will have to solve yourself.

1. Write a script that will prompt the user for the radius and height.
2. Create a function named **conevol** to calculate the cone volume.
3. Return the volume.
4. Print the result in a nice sentence format.
5. The program will consist of a script and the **conevol** function that it calls.

Example run:

```
---   Calculate Volume of a Cone   ---
Please enter the radius: 5
Please enter the height: 6
The volume of a cone with radius 5.00 and height 6.00 units
is 157.08 cubic units.
```

Assignment Submission

1. Submit properly named and commented script files.

2. Attach a screenshot of the Command Window showing the successful execution of each script.
3. Attach all to the assignment in Blackboard.