Input/Output

Assis. Prof. Dr. MOHAMMED ELAMASSIE.
IEEE Senior Member | Optica Senior Member | Optica Travelling Lecturer.
Executive Co-Director of OKATEM.
Executive Co-Director of CT&T Research Group.

Spring 2024-25

Basic Input and Output

Write

Write (1/5)

MATLAB provides several functions for basic input and output operations. The most common ones are fprintf and fscanf.

fprintf: Writes formatted data to a file or the command window.

Example 1 fprintf('Hello, world!\n’);

Example 2

```
fileID = fopen('example.txt', 'w’);   %open a file named 'example.txt' for writing

% Define data to write
name = 'John Doe’;     age = 30;     height = 175.5;

% Write formatted data to the file
fprintf(fileID, 'Name: %s\n', name);  %s is a placeholder for a string
fprintf(fileID, 'Age: %d\n', age);        %d for an integer
fprintf(fileID, 'Height: %.2f\n', height); %.2f for a floating-point number

% Close the file
fclose(fileID);      %fclose is used to close the file once writing is complete.
```

Write (2/5)

Example 3

```
clear all; close all; clc
% Open a file for writing
fileID = fopen('data.csv', 'w');

% Define data to write
names = {'Alice', 'Bob', 'Charlie'};
ages = [25, 30, 35];
scores = [85.5, 90, 78.2];

% Write headers
fprintf(fileID, 'Name, Age, Score\n');

% Write data rows
for i = 1:numel(names)
    fprintf(fileID, '%s, %d, %.1f\n', names{i}, ages(i), scores(i));
end
% Close the file
fclose(fileID);
```

%.1f: This specifier formats the floating-point number with one decimal place. For example, if you have a number like 12.345, it would be formatted as 12.3.

%.2f: This specifier formats the floating-point number with two decimal places. Using the same example, 12.345 would be formatted as 12.35, as it rounds up the second decimal place.

Write (3/5)

Example 4

```
clear all; close all; clc

% Open a file for writing
fileID = fopen('Word_Doc.doc', 'w');
% Define data to write
names = {'MOHAMMED', 'SALMA'};
ages = [41, 40];
scores = [180, 160];


% Write headers
fprintf(fileID, 'Name, Age, Score\n');
% Write data rows
for i = 1:numel(names)
    fprintf(fileID, '%s, %d, %.1f\n', names{i}, ages(i), scores(i));
end
% Close the file
fclose(fileID);
```

Write (4/5)

Example 5 (1/2)

```
clear all;   close all;  clc

% Create a Word application object
word = actxserver('Word.Application');

% Add a new document
doc = word.Documents.Add;

% Define data to write
name = ‘MOHAMMED ELAMASSIE';
age = 40;
height = 181.5;

% Write formatted data to the document
selection = word.Selection;
selection.TypeText(sprintf('Name: %s\n', name));
selection.TypeText(sprintf('Age: %d\n', age));
selection.TypeText(sprintf('Height: %.2f\n', height));
```

Write (5/5)

Example 5 (2/2)

```
% Save the document
currentDir = pwd; % Get the current directory
fprintf('Current directory: %s\n', currentDir); % Print current directory
doc.SaveAs2(fullfile(currentDir, 'example3.docx')); % Save the document in
the current directory


% Close the document
doc.Close;

% Quit Word application
word.Quit;


disp('Document successfully created and saved.');
```

Read

Read (1/3)

To read data from an existing file in MATLAB, you can use various functions depending on the format of the file.

Example 1

```
clear all; close all; clc
% Read data from the CSV file
data = open('data.csv')

% Display the read data
disp(data);
```

Example 2

```
clear all; close all; clc
% Open the doc file
open('Word_Doc.doc’)
```

Example 3

```
clear all; close all; clc
uiopen
```

Read (2/3)

Example 4

```
clear all; close all; clc
% Read the content of the file as text
fileContent = fileread('Word_Doc.doc');

% Split the content by newline characters
lines = strsplit(fileContent, '\n');

% Display each line of the file
for i = 1:numel(lines)
    disp(lines{i});
end
```

numel Number of elements in an array or subscripted array expression

Example 5

```
clear all; close all; clc
% Read starting from cell (2nd row (index 2-1), 3rd column (index  3-1)) using csvread
data = csvread('data.csv', 1, 2); % Assuming data starts from row 2 and column 3
disp(data);
```

Read (3/3)

Example 6

```
clear all; close all; clc
% Or using readmatrix
data = readmatrix('data.csv', 'Range', 'B2:C2'); % Range is specified using Excel-style notation
disp(data);
```

Example 7

```
clear all; close all; clc
% Or using readtable
% Read specific cell (2nd row, 'Age' column) using readtable
dataTable = readtable('data.csv');
age = dataTable.Age(2); % Assuming you want the age from the second row
disp(age);
```

Read/Write

Read/write (1/5)

Example 1

```
clear all; close all; clc
% Read dates from Excel file
dataTable = readtable('data.csv');

% Extract dates from the table
Names = dataTable.Name; % Assuming the column name is Var1

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

% Write dates to the text file
for i = 1:numel(Names)
    fprintf(fileID, 'Name: %s\n', char(Names(i))); %s is a placeholder for a string
end
% Close the text file
fclose(fileID);
```

Read/write (2/5)

Example 2

```
clear all; close all; clc
% Generate sinusoidal signal data
Fs = 1000; % Sampling frequency (Hz)
t = 0:1/Fs:1-1/Fs; % Time vector (1 second duration)
f = 10; % Frequency of the sinusoid (Hz)
A = 1; % Amplitude of the sinusoid
signal = A * sin(2*pi*f*t); % Generate sinusoidal signal


% Add noise to the signal
noise = 0.1 * randn(size(t)); % Gaussian noise with standard deviation 0.1
noisy_signal = signal + noise;


% Save the noisy signal to a text file
filename = 'noisy_signal.txt';
dlmwrite(filename, noisy_signal, 'precision', '%.6f');
```

Read/write (3/5)

Example 2 (Continue)

```
% Read the noisy signal from the text file
read_noisy_signal = dlmread(filename);


% For demonstration, let's just plot the read data
figure;
subplot(2,1,1);
plot(t, noisy_signal, 'b', 'LineWidth', 1.5);
xlabel('Time (s)');
ylabel('Amplitude');
title('Original Noisy Signal');
grid on;


subplot(2,1,2);
plot(t, read_noisy_signal, 'r', 'LineWidth', 1.5);
xlabel('Time (s)');
ylabel('Amplitude');
title('Read Noisy Signal from File');
grid on;
```

Read/write (4/5)

Example 3

```
clear all; close all; clc
% Generate a signal
Fs = 1000; % Sampling frequency (Hz)
t = 0:1/Fs:1-1/Fs; % Time vector (1 second duration)
f1 = 10; % Frequency of the first sinusoid (Hz)
f2 = 20; % Frequency of the second sinusoid (Hz)
signal = sin(2*pi*f1*t) + 0.5*sin(2*pi*f2*t); % Original signal

% Compute Fourier transform
N = length(signal); % Length of the signal
frequencies = (-N/2:N/2-1)*Fs/N; % Frequencies corresponding to the Fourier transform
signal_fft = fftshift(fft(signal)); % Compute the Fourier transform and shift frequencies

% Save Fourier transform data to a file
fft_filename = 'signal_fft.txt';
dlmwrite(fft_filename, [frequencies; abs(signal_fft)]', 'delimiter', '\t', 'precision', '%.6f');

% Perform inverse Fourier transform to reconstruct the signal
reconstructed_signal = ifft(ifftshift(signal_fft)); % Inverse Fourier transform

% Save the reconstructed signal to a file
reconstructed_filename = 'reconstructed_signal.txt';
dlmwrite(reconstructed_filename, reconstructed_signal', 'precision', '%.6f');
```

Read/write (5/5)

Example 3 (Continue)

```
% Read data from both files
data_fft = dlmread(fft_filename);
frequencies_read = data_fft(:, 1);
signal_fft_read = data_fft(:, 2);

reconstructed_signal_read = dlmread(reconstructed_filename);

% Plot original and reconstructed signals
figure;   subplot(2,1,1);   plot(t, signal, 'b', 'LineWidth', 1.5);
xlabel('Time (s)’);   ylabel('Amplitude’);   title('Original Signal');
grid on;

subplot(2,1,2);
plot(t, real(reconstructed_signal_read), 'r', 'LineWidth', 1.5); % Real part of the
reconstructed signal
xlabel('Time (s)’);    ylabel('Amplitude’);   title('Reconstructed Signal');
grid on;
```

More Examples

Example - 1

```
clear all; close all; clc
% Generate a noisy signal
Fs = 1000; % Sampling frequency (Hz)
t = 0:1/Fs:1-1/Fs; % Time vector (1 second duration)
f1 = 10; % Frequency of the sinusoid (Hz)
signal = sin(2*pi*f1*t); % Original signal
noise = 0.5 * randn(size(t)); % Gaussian noise with standard deviation 0.5
noisy_signal = signal + noise; % Noisy signal

% Apply a simple moving average filter to the noisy signal
filter_order = 20;
filtered_signal = movmean(noisy_signal, filter_order);

% Save the original and filtered signals to a text file
filename_original = 'original_signal.txt';
filename_filtered = 'filtered_signal.txt';
dlmwrite(filename_original, [t' noisy_signal'], 'delimiter', '\t', 'precision', '%.6f');
dlmwrite(filename_filtered, [t' filtered_signal'], 'delimiter', '\t', 'precision', '%.6f');

% Read data from both files
data_original = dlmread(filename_original);
data_filtered = dlmread(filename_filtered);
t_read = data_original(:, 1);
noisy_signal_read = data_original(:, 2);
filtered_signal_read = data_filtered(:, 2);
```

Example – 1 (Continue)

```
% Plot original and filtered signals
figure;
subplot(2,1,1);
plot(t, noisy_signal, 'b', 'LineWidth', 1.5);
hold on;
plot(t_read, noisy_signal_read, 'r--', 'LineWidth', 1.5);
xlabel('Time (s)');
ylabel('Amplitude');
title('Original Noisy Signal');
legend('Original', 'Read from File');
grid on;


subplot(2,1,2);
plot(t, filtered_signal, 'b', 'LineWidth', 1.5);
hold on;
plot(t_read, filtered_signal_read, 'r--', 'LineWidth', 1.5);
xlabel('Time (s)');
ylabel('Amplitude');
title('Filtered Signal');
legend('Filtered', 'Read from File');
grid on;
```

Homework: You are given a dataset containing measurements of temperature over time. Your task is to perform the following steps:

1. Read the temperature data from a CSV file named "temperature_data.csv". The file contains two columns: "Time" (in seconds) and "Temperature" (in degrees Celsius).
2. Calculate the average temperature over the entire duration of the dataset.
3. Apply a simple moving average filter to smooth the temperature data. Use a window size of 10 samples.
4. Write the filtered temperature data to a new CSV file named "smoothed_temperature_data.csv".
5. Read the smoothed temperature data from the CSV file and plot both the original and smoothed temperature data on the same graph.

```
clear all; close all; clc
% Generate sample temperature data
time = (0:0.1:10)'; % Time vector (seconds)
temperature = 20 + 5 * sin(2*pi*0.5*time) + randn(size(time)); % Sinusoidal temperature with noise

% Write temperature data to CSV file
filename = 'temperature_data.csv';
csvwrite(filename, [time, temperature]);
```

Thanks for Your Attention…

Assis. Prof. Dr. Mohammed Elamassie
mohammed.elamassie@ozyegin.edu.tr

CENTER OF EXCELLENCE IN OPTICAL WIRELESS COMMUNICATION TECHNOLOGIES

ÖZYEĞİN UNIVERSITY