

# SAVING, IMPORTING AND EXPORTING

Presented by Tony Vo

Slides by Tony Vo



# SAVING VARIABLES TO DISK

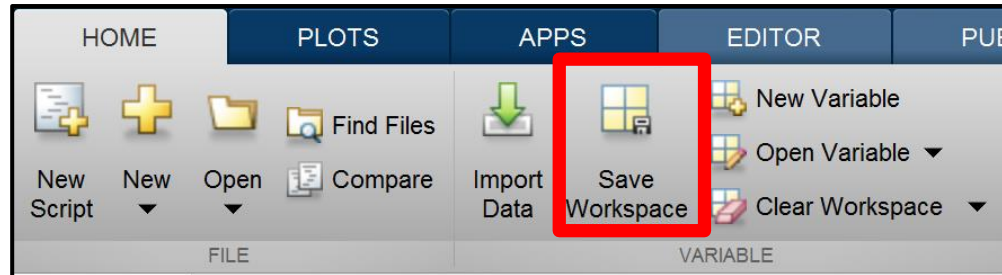
- The variables you create disappear when you close MATLAB
  - This is an issue if your m-files take a long time to run

- You can save and load the workspace using

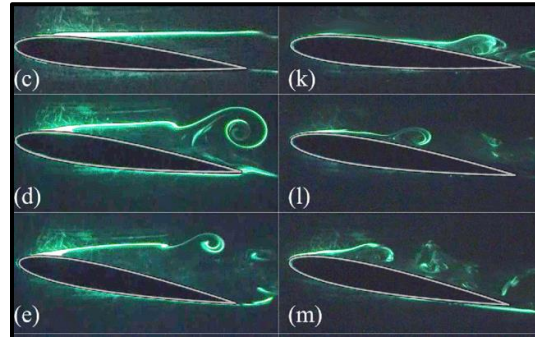
`save <filename>.mat`

`load <filename>.mat`

Workspace	
Name ^	Value
index	92
max_x_coord	5.1000
max_y_coord	24.0804
x	1x121 double
y	1x121 double



- Up to until this point, we have created data in MATLAB
  - E.g. `A = 1 : 20;`    `B = [ 1 2 3; 4 5 6 ];`    `C = linspace(1, 20, 5);`
- Many engineering problems use data collected from real-world sources
  - E.g. sensors or people
  - We load in data to avoid having to input it manually

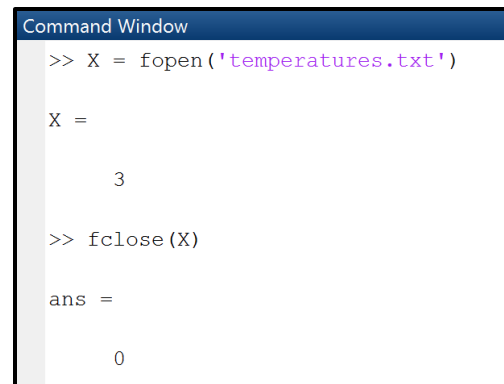
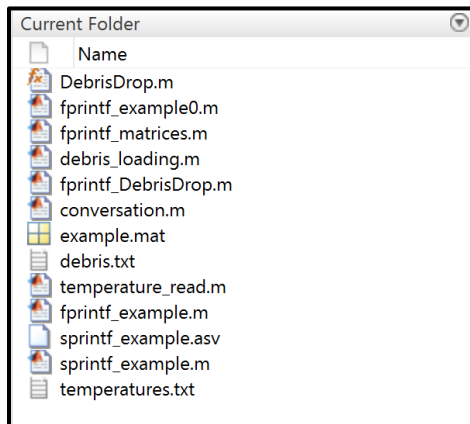


- There are two separate methods for importing data:
  1. Using the fopen, fclose and fgetl functions
  2. Using the importdata function
  
- Both methods have their advantages and disadvantages
  - Use the one that is most appropriate for your task

# FOPEN AND FCLOSE FUNCTIONS

- **fopen**: opens a file
  - Syntax: `file_id = fopen('filename')`
  - A file is assigned a file identifier once it is opened using `fopen`
  - Ensure that the file being opened is in the current directory

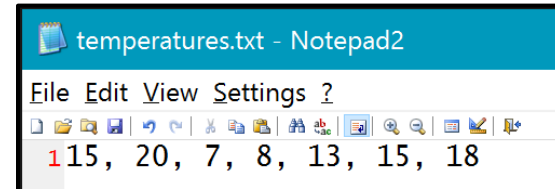
- **fclose**: closes a file
  - Syntax: `fclose(file_id)`



- Once we have opened a file with `fopen`, we can use the `fgetl` function to get lines of data from the file
  - Syntax: `data = fgetl(file_id)`
  - `fgetl` imports data as a string

```
% Opening file, importing data
filename = 'temperatures.txt';
fid = fopen(filename, 'r'); % Opening read-only
temperature = fgetl(fid); % Getting 1st line of text
num_temp = str2num(temperature); % Convert to number
fclose(fid);

% printing the mean temperature
fprintf('Mean temperature is %.1f\n', mean(num_temp));
```

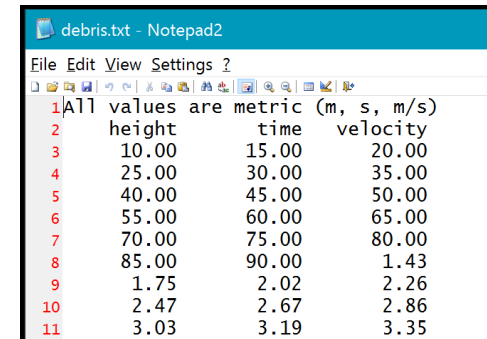


Workspace	
Name ^	Value
ans	0
fid	3
filename	'temperatures.txt'
num_temp	[15,20,7,8,13,15,18]
temperature	'15, 20, 7, 8, 13, 15, 18'

- Calling `fgetl()` N times will get N lines from the text
- What if we want to read a file with 1000 lines?

```
% Reading debris data using fgetl()
% Multiple lines
clear all; close all; clc;

% opening file
filename = 'debris.txt';
fid = fopen(filename, 'r'); % Opening read-only
D1 = fgetl(fid); % Getting 1st line of text
D2 = fgetl(fid); % Getting 2nd line of text
D3 = fgetl(fid); % Getting 3rd line of text
D4 = fgetl(fid); % Getting 4th line of text
fclose(fid);
```



```
debris.txt - Notepad2
File Edit View Settings ?
1 All values are metric (m, s, m/s)
2 height time velocity
3 10.00 15.00 20.00
4 25.00 30.00 35.00
5 40.00 45.00 50.00
6 55.00 60.00 65.00
7 70.00 75.00 80.00
8 85.00 90.00 1.43
9 1.75 2.02 2.26
10 2.47 2.67 2.86
11 3.03 3.19 3.35
12 3.78
13 4.16
14 8.58
15 12.13
16 14.86
17 17.16
18 19.18
19 21.01
```

Workspace	
Name ^	Value
ans	0
D1	'All values are metric (m, s, m/s)'
D2	' height time velocity'
D3	' 10.00 15.00 20.00'
D4	' 25.00 30.00 35.00'
fid	3
filename	'debris.txt'

# IMPORTDATA FUNCTION

- Use the `importdata` function to load files containing a lot of data
  - The `importdata` function is different to the import data wizardSyntax: `all_data = importdata('filename')`
- `importdata` command loads in all of the data in the file
  - No need for `fopen` and `fclose`
- More control using the `importdata` function
- The `importdata` function cannot write to files



# IMPORTDATA FUNCTION

- Imports the data in as a structure
  - Structure is comprised of data, text data and column headers
  - Data from structures can be accessed via: `content = <matrix>.<content_type>`

```
% importing the data
X = importdata('debris.txt');

% numerical data
num_data = X.data;

% text data
text_data = X.textdata;

% column headers
coltext_data = X.colheaders;
```

```
X =

      data: [17x3 double]
    textdata: {2x3 cell}
  colheaders: {'height'  'time'  'velocity'}
```

# IMPORTDATA FUNCTION

- Separate data into variables after you have imported it

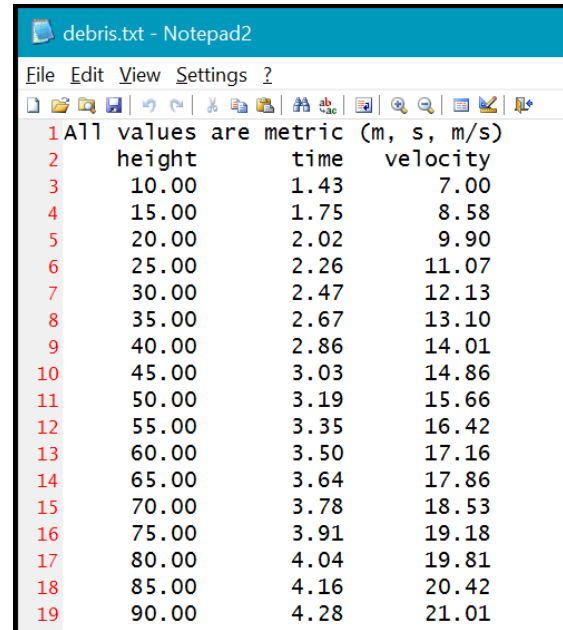
```
% importing the data  
X = importdata('debris.txt');
```

```
% numerical data  
num_data = X.data;
```

```
% text data  
text_data = X.textdata;
```

```
% column headers  
coltext_data = X.colheaders;
```

```
% sorting the data  
height = num_data(:,1);  
time = num_data(:,2);  
velocity = num_data(:,3);
```



debris.txt - Notepad2

File Edit View Settings ?

1 All values are metric (m, s, m/s)

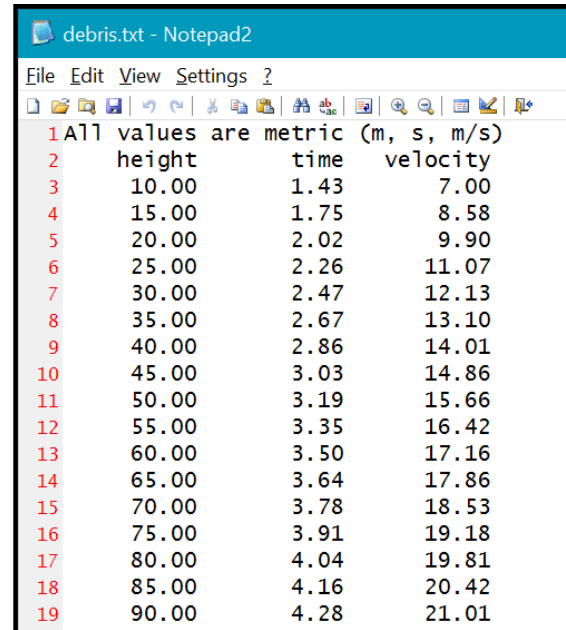
2	height	time	velocity
3	10.00	1.43	7.00
4	15.00	1.75	8.58
5	20.00	2.02	9.90
6	25.00	2.26	11.07
7	30.00	2.47	12.13
8	35.00	2.67	13.10
9	40.00	2.86	14.01
10	45.00	3.03	14.86
11	50.00	3.19	15.66
12	55.00	3.35	16.42
13	60.00	3.50	17.16
14	65.00	3.64	17.86
15	70.00	3.78	18.53
16	75.00	3.91	19.18
17	80.00	4.04	19.81
18	85.00	4.16	20.42
19	90.00	4.28	21.01

- `fprintf` can be used in conjunction with `fopen` to export data
  - The 'w' permission is used to write

```
% fprintf with matrix variables, prints to text file
% Requires DebrisDrop.m function
filename = 'debris.txt';

% Calculating fall time and velocity from height
h = 10:5:90; % height
[t, v] = DebrisDrop(h); % [time, velocity]

% Printing results to file as a table
file_id = fopen(filename, 'w'); % Opening to write text
fprintf(file_id, 'All values are metric (m, s, m/s)\n');
fprintf(file_id, '%10s %10s %10s\n', 'height', 'time', 'velocity');
fprintf(file_id, '%10.2f %10.2f %10.2f\n', [h' t' v]');
fclose(file_id); % Closing file since we are done
```



debris.txt - Notepad2

1	All values are metric (m, s, m/s)		
2	height	time	velocity
3	10.00	1.43	7.00
4	15.00	1.75	8.58
5	20.00	2.02	9.90
6	25.00	2.26	11.07
7	30.00	2.47	12.13
8	35.00	2.67	13.10
9	40.00	2.86	14.01
10	45.00	3.03	14.86
11	50.00	3.19	15.66
12	55.00	3.35	16.42
13	60.00	3.50	17.16
14	65.00	3.64	17.86
15	70.00	3.78	18.53
16	75.00	3.91	19.18
17	80.00	4.04	19.81
18	85.00	4.16	20.42
19	90.00	4.28	21.01

- Saving and loading workspaces
  - Importing data with fopen, fgetl, fclose functions
  - Importing data with importdata function
  - Exporting data with fprintf
- 
- Why does the file identifier start with 3 when using fopen?