Introduction to Matlab

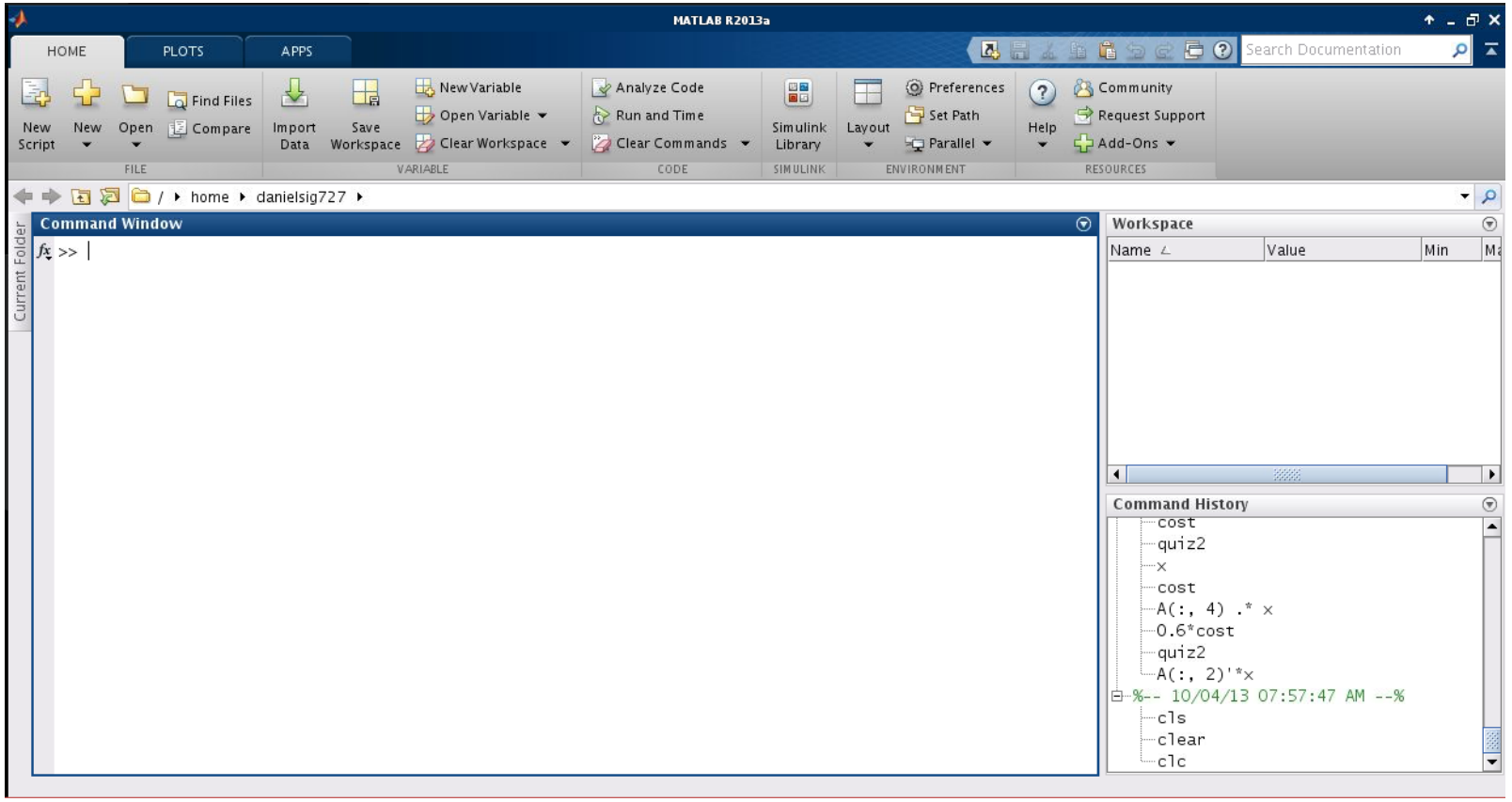
TA: Yi-Chung Wu

(Courtesy: Che-Wei Kuo)

What is Matlab?

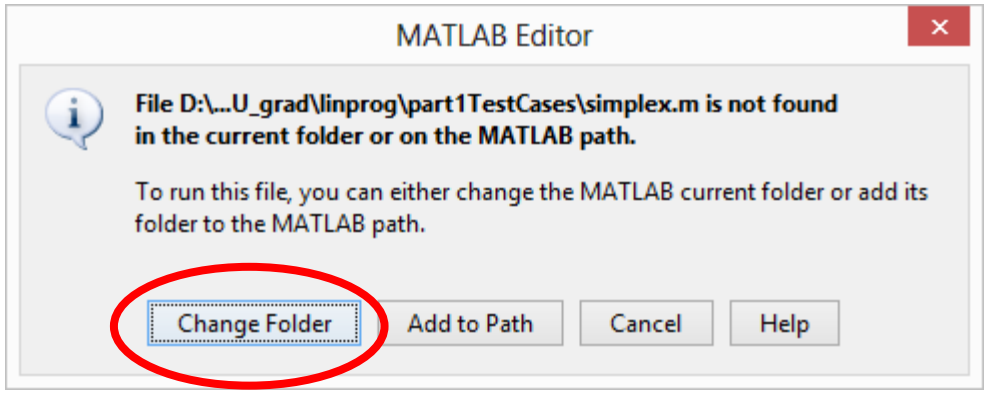
- A strong math calculator (matrix, equation, waveform…)
- A C-like programming language
- Include many set of tools (toolboxes) for many fields: communication, computer vision, biology, statistic…etc.

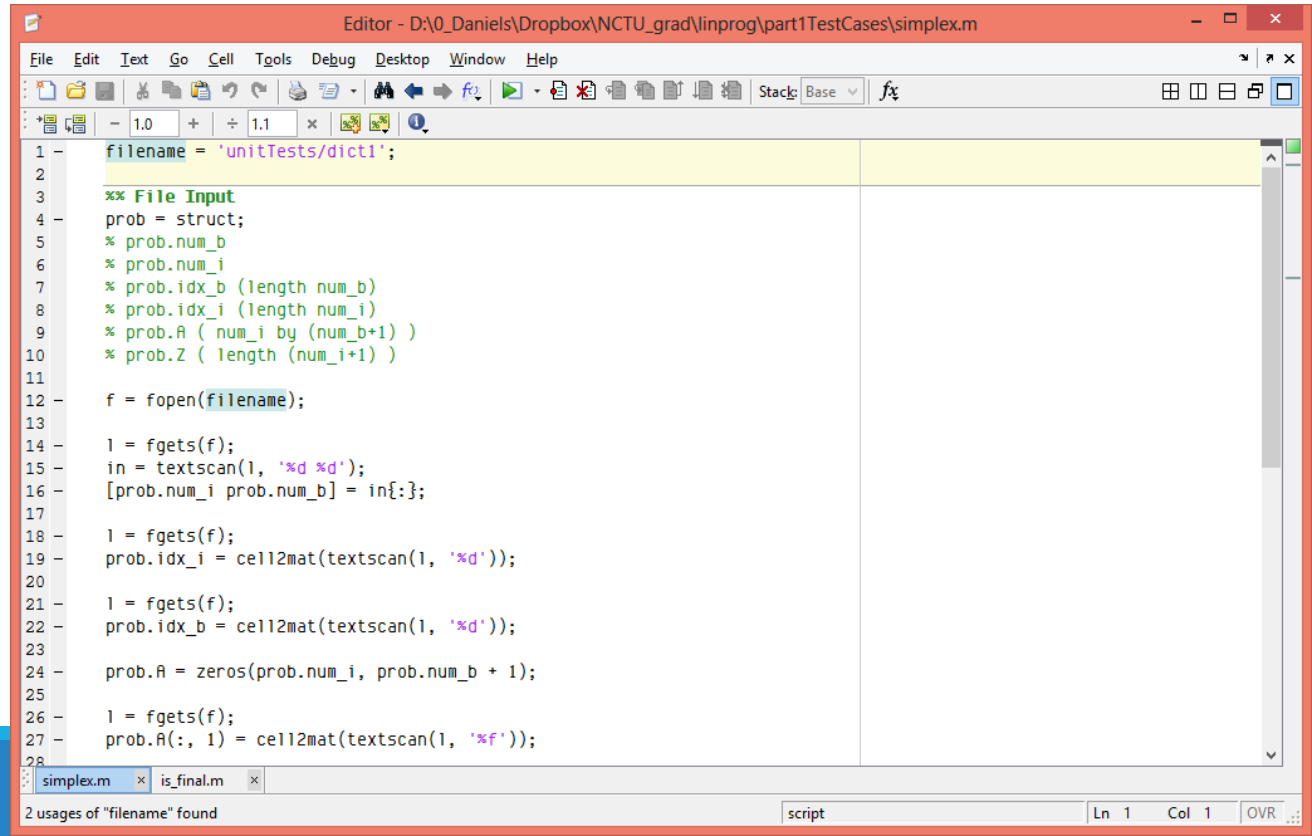
Matlab Interface



Two ways of running scripts

- Directly type the commands in the command window
- Use editor to save commands to a file
- Run (F5) and change folder





Basic Operations

Built-in `help` function

- help <function-name>

```
>> help conv
 conv Convolution and polynomial multiplication.
    C = conv(A, B) convolves vectors A and B.

>> help plot
 plot   Linear plot.
    plot(X,Y) plots vector Y versus vector X.
```

Variables

- No declaration needed

```
>> x = 3
x =
     3
>> y = 4
y =
     4
```

```
>> x + y, x - y
ans =
     7
ans =
    -1
```

Matrix Input

- Use brackets “[]” to input matrices, vectors. “;” to change to next row
- Row vector:
  - `>> x = [1 2 3]`
  - `x =`
  - `1 2 3`
- Column Vector
  - `>> y = [1; 2; 3]`
  - `y =`
  - `1`
  - `2`
  - `3`
- Matrix
  - `>> A = [1 2 3; 4 5 6; 7 8 9]`
  - `A =`
  - `1 2 3`
  - `4 5 6`
  - `7 8 9`

Matrix Operations - Element extraction

- `A(i, j)` : extract A_{ij}
- Indexing: (indices in Matlab start at 1, note C start at 0!)
  - `:` → all
  - `a:b` → a-th to b-th element
  - `a:k:b` → element $a, a+k, a+2k, \ldots, b$
  - `end` → the last element's index
- Examples
  - `A(:, j)` : extract the j-th column
  - `A(i, :)` : extract the i-th row
  - `A(2:3, :)`, `A(2:end, :)`…etc

Matrix Operations - Element extraction

```
A =
     1     2     3
     4     5     6
     7     8     9

>> A(1, 2)
ans =
     2

>> A(1, 2:3)
ans =
     2     3

>> A(1, :)
ans =
     1     2     3
```

```
>> A(:, 1)
ans =
     1
     4
     7
>> A(1, 2:end)
ans =
     2     3
```

Matrix Operations

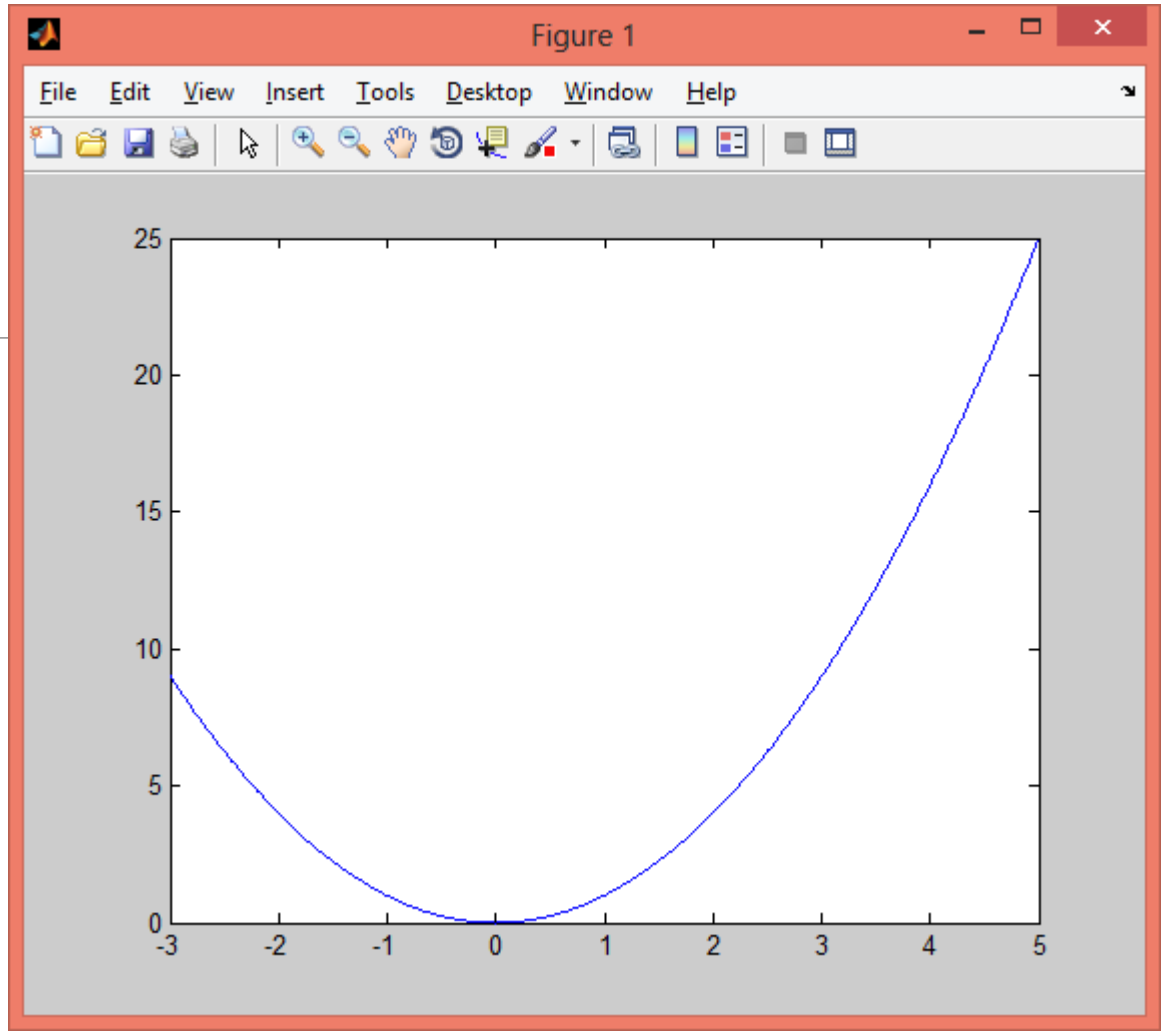
- Basic operators
  - +, -, * the same as defined in linear algebra
  - A^N : A multiplied to itself by N times
- Elementwise operators
  - `.*`, `./`, `.^`
  - Dimensions must agree
- Transposition
  - `transpose(A)`
  - Shortcut: `A'`
    (Hermitian for complex matrices)
- Inversion
  - `inv(A)`
  - `A^(-1)`
- Backslash/left devide
  - Shorthand of `inv(A) * B` → `A\B`

Some commonly used functions

- Trigonometric functions
  - `sin(x), cos(x), tan(x)`
- Absolute value
  - `abs(x)`
- Norms
  - `norm(x)`
- Statistics
  - `min(x), max(x)`
  - `var(x), mean(x), sum(x)`
- Getting size of matrices
  - `size(x), length(x)`
- Generating matrices
  - `zeros(m, n), ones(m, n), rand(m, n)`
- Exponential
  - `exp(i)`
- Constants
  - e: `exp(1)`
  - π: `pi`
  - `Imaginary: i`
  - `inf`
  - `NaN`

Plotting

- Plotting involves 3 steps
  - Generate X axis
  - Generate Y axis
  - Plot the graph
- Example: plot x^2 in [-3, 5]
  - `x = -3:0.01:5;`
  - `y = x.^2;`
  - `plot(x, y)`



- If you want to apply styles(dashed lines, colors…etc), type:

```
>> help plot
```

For Loops

- for <var> = <assignment-list>
    (do something using <var>)
  end

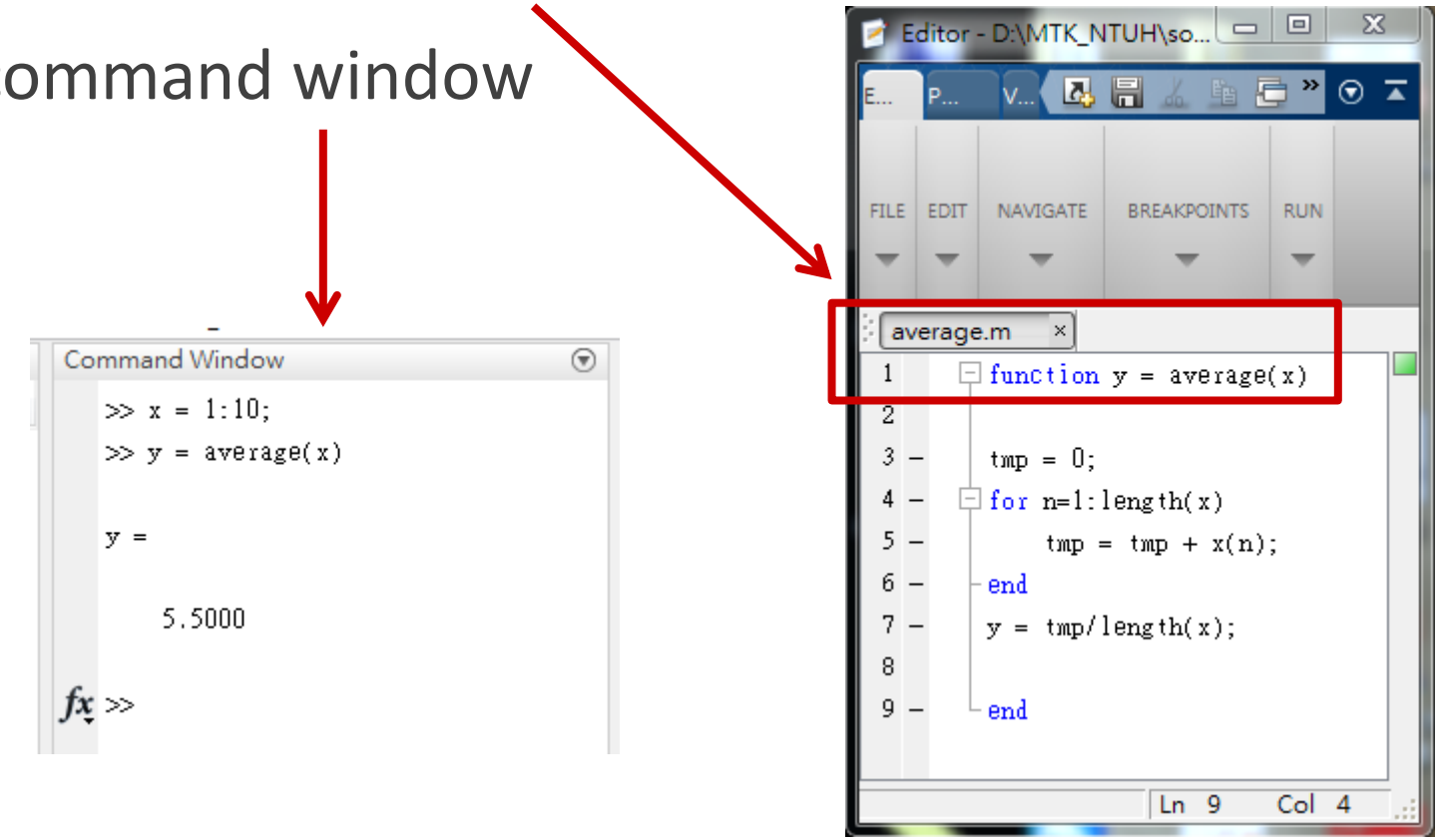
- Example (imaginary i is overwritten)
  - for i = 1:10
      disp(i);
    end
  - for i = [1:2:9] % [1 3 5 7 9]

While Loops

- while <condition>
  <do something>
  end
- Condition operators: >, >=, <, <=, ==, ~= (not equal to)
- Example
  - x = 0;
    while x ~= 10
    disp('haha');
    end
  - Ctrl+c to stop the infinite loop

Define your own function

- The function name should be the same as the file name
- Use the function in the command window



Save commands as .m file

- Type your commands in text editor software
- Save the file as <filename>.m
- Open your .m file with Matlab
- Run the command

Useful Links

- Matlab online manual on Mathworks website
  - http://www.mathworks.com/help/matlab
- Tutorial in Chinese
  - http://libai.math.ncu.edu.tw/bcc16/B/matlab/

Demo: Prob 4.38 (a)

$$x(v) = \sqrt{v} \times e^{-0.5v^{1.5}} u(v-2)$$

- Prob_4_38a.m

```
clc         % clear Command Window
clear all   % clear Workspace
close all   % close figures

% assign an array
v1=-6; dv=0.01; v2=6;
v=v1:dv:v2;
x=sqrt(v).*exp(-0.5*v.^1.5).*heaviside(v-2);
figure(1)
plot(v,x); xlabel('v'); ylabel('x(v)');
```

- Figure 1