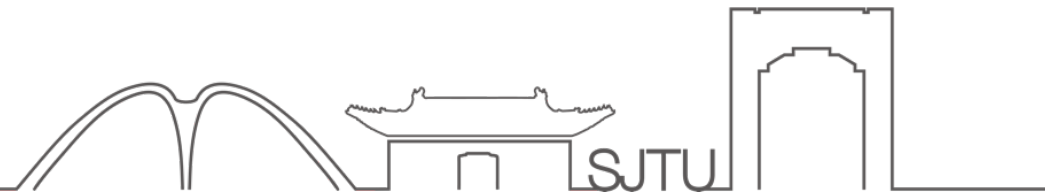
JOINT INSTITUTE
交大密西根学院

VG101/ENGR1010J: Introduction to Computers and Programming

Lecture 02

Xiaodong Wei



Outline

- Introduction to MATLAB
- MATLAB Development Environment
- Expressions

Introduction to MATLAB

What is MATLAB?

- MATLAB is one of the commercially available, sophisticated mathematical computation tools.
- Others include
  - Maple
  - Mathematica
  - MathCad
- Open source: GNU Octave
- About Python

MATLAB Excels at

- Numerical calculations
  - Especially involving matrices
- Graphics
- Toolboxes
- Simulink
- MATLAB stands for Matrix Laboratory

Why MATLAB?

- Easy to use
- Versatile
- Built-in programming language
- Many useful toolboxes

History of MATLAB

- MATLAB was originally written in Fortran, then later rewritten in C

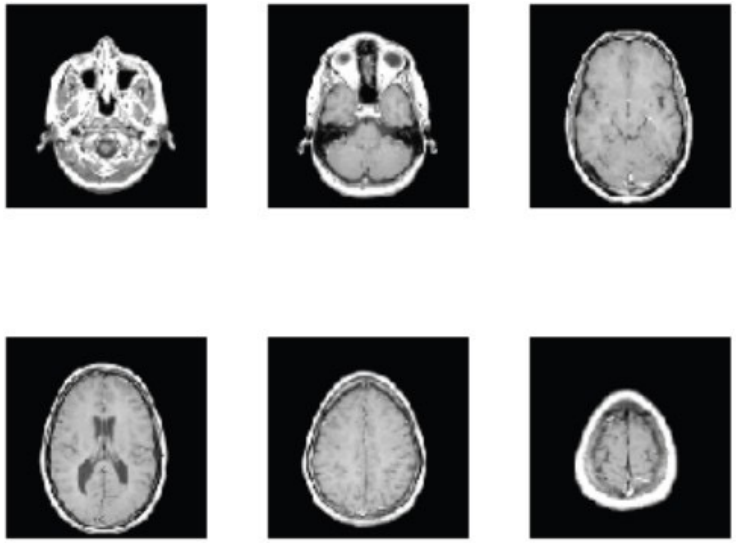
How is MATLAB used in Industry?

- Widespread, especially in the signal processing field
- Tool of choice in academia for most engineering fields
- Some examples....

Electrical Engineering

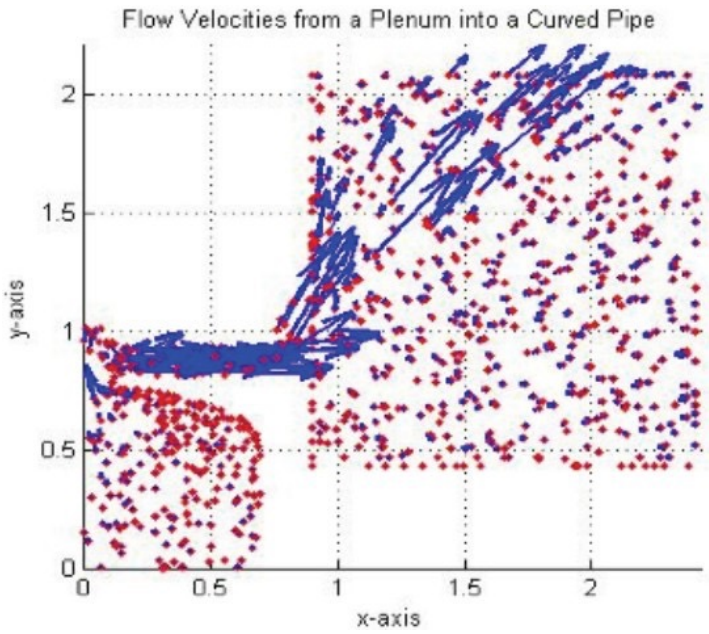
- These images simulate the visual system used in a housefly brain to detect collisions. The techniques developed are being used in autonomous robot systems that depend upon vision for navigation. The data was processed using MATLAB.

Biomedical Engineering



- These images were created from MRI scan data using MATLAB. The actual data set is included with the standard MATLAB installation, allowing you experiment with manipulating the data yourself.

Fluid Dynamics



- Results from a finite element analysis code were post processed using MATLAB to create this image.

Problem Solving in Engineering and Science

- State the Problem
- Describe the input and output
- Develop an algorithm
- Solve the problem
- Test the solution

State the Problem

- If you don’t have a clear understanding of the problem, it’s unlikely that you’ll be able to solve it
- Drawing a picture often helps you understand the system better

Describe the Input and Output

- Be careful to include units
- Identify constants
- Label your sketch
- Group information into tables

Develop an Algorithm

- Identify any equations relating the knowns and unknowns
- Work through a simplified version of the problem by hand or with a calculator
- Developing a flow chart is often useful for complicated problems

Solve the Problem

- Create a MATLAB solution
- Be generous with comments, so that others can follow your work. And without comments, you will forget what you wrote in the future.

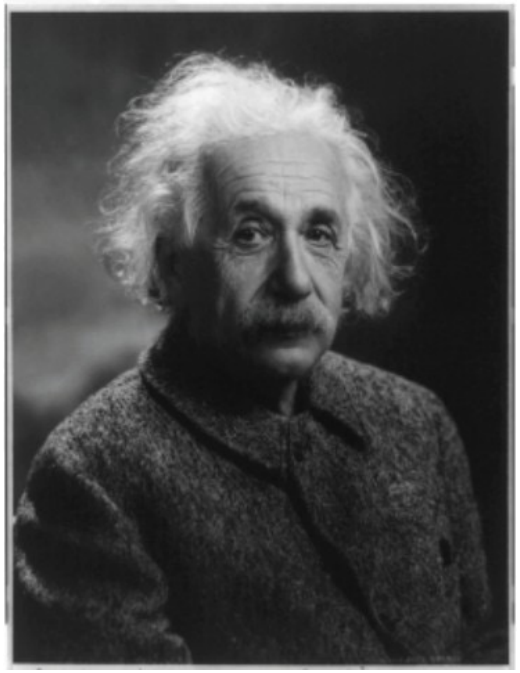
Test the Solution

- Compare to the hand solution
- Do your answers make sense physically?
- Is your answer really what was asked for?
- Graphs are often useful ways to check your calculations for reasonableness

Using Engineer Way of Thinking

- If you use a consistent problem solving strategy you increase the chance that your result is correct
- Here's an example....

Example



- Albert Einstein
- $E=mc^2$
- The sun is fueled by the conversion of matter to energy
- How much matter does the sun consume every day?

State the Problem

- Find the amount of matter necessary to produce the amount of energy radiated by the sun everyday.

Describe the Input and Output

- Input
  - Rate of energy radiation
    $E = 385*10^{24}$ Joules/second
  - Speed of light
    $c = 3.0*10^{8}$ meters/second
- Output
  - Mass in kilograms

Develop an Algorithm – Hand Example

- The energy radiated in one day is:

$$385 * 10^{24} \frac{J}{\text{sec}} * 3600 \frac{\text{sec}}{hour} * 24 \frac{hours}{day} * 1day = 3.33 * 10^{31} J$$

- Rearrange $E=mc^2$ and solve for m: $m=E/c^2$

$$m = \frac{3.33 * 10^{31} J}{\left(3.0 * 10^{8} m / \text{sec}\right)^2} = 3.7 * 10^{14} \frac{J}{m^2 / \text{sec}^2}$$

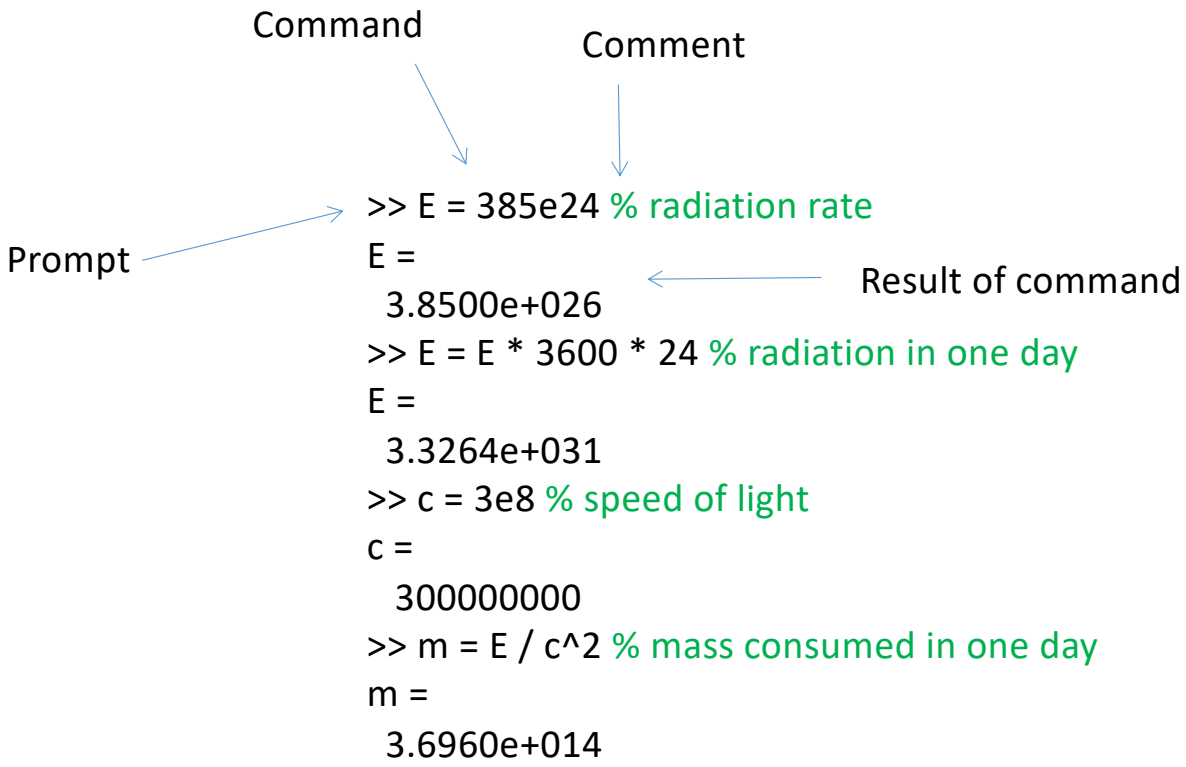
Double Check the Unit

$$3.7 * 10^{14} \frac{J}{m^2 / \sec^2}$$

$$1\ J = 1\ kg\ m^2/sec^2$$

$$= 3.7 * 10^{14} \frac{kg\ m^2 / \sec^2}{m^2 / \sec^2} = 3.7 * 10^{14}\ kg$$

Develop a MATLAB Solution – Command Window



Develop a MATLAB Solution – M-File

SunRadiation.m

```
1 —    E = 385e24; % radiation rate
2 —    E = E * 3600 * 24; % radiation in one day
3 —    c = 3e8; % speed of light
4 —    m = E / c^2; % mass consumed in one day
5 —    disp('mass consumed in one day is');
6 —    disp(m);
```

Command Window

```
mass consumed in one day is
   3.6960e+14

>>
```

Test your solution

- Matches the hand solution
- Is it reasonable?
- Consider...
  - Mass of the sun = $2*10^{30}$ kg
  - How long would it take to consume all that mass?

Test your solution

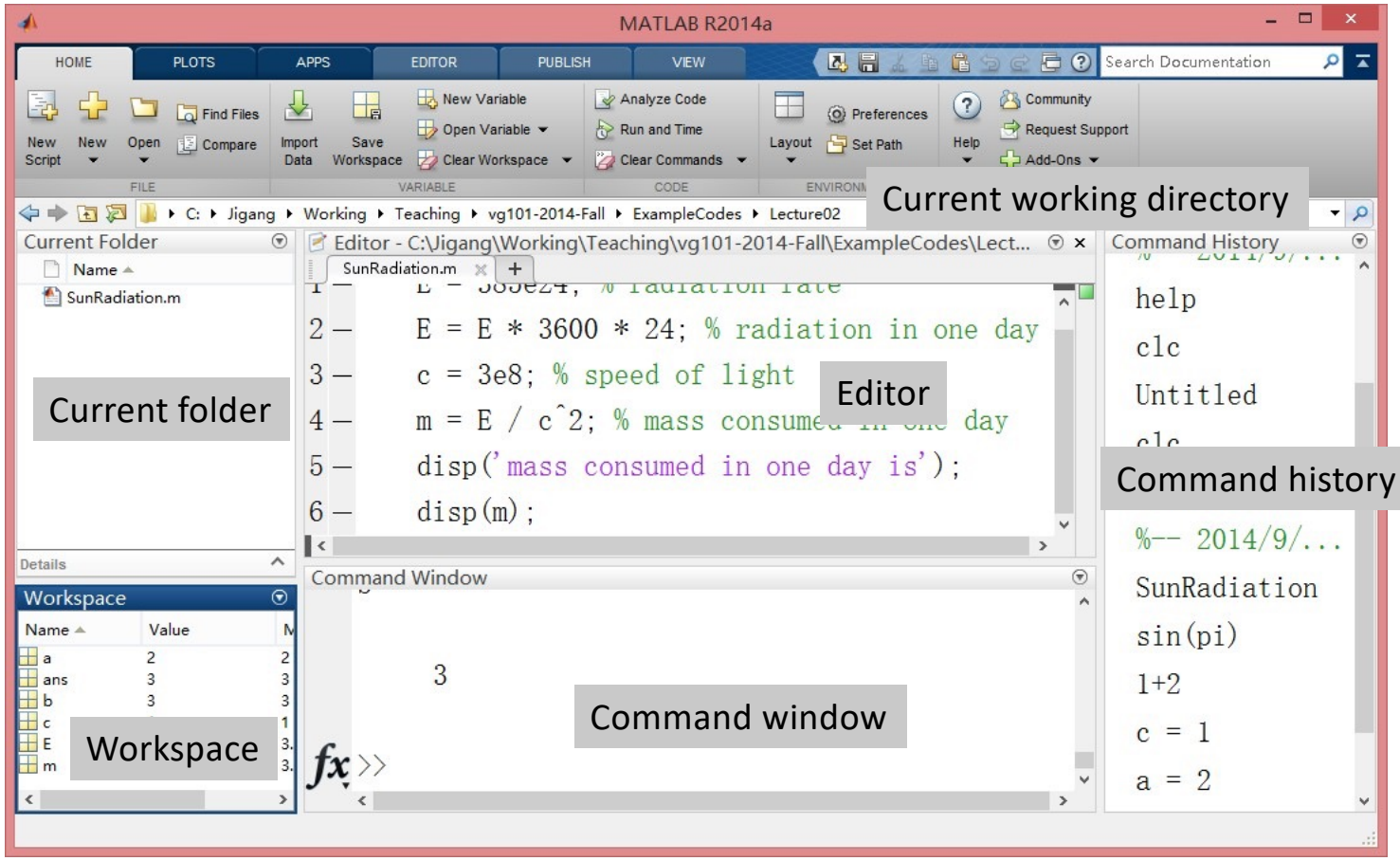
- time = (mass of the sun) / (rate of consumption)

$$time = \frac{2*10^{30}\,kg}{3.7*10^{14}\,kg/day} * \frac{year}{365 days} = 1.5*10^{13}\,years$$

- This is 15 trillion year! – reasonable

MATLAB Development Environment

Default MATLAB Windows



Current Directory and Search Path

- MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.
- How to change current directory?
  - cd % display the current folder
  - cd new_folder_path

Command Window

- Use the Command Window to enter variables and to run functions and M-file scripts.
- Examples:

```
>> 1+2
>> 5-3
>> 6*9
>> 8/4
>> 2^8
>> sqrt(2)
>> sin(pi)
```

Command History

- You will see what you just typed in the Command Window.
- Hit the “up” key ↑ and the “down” key ↓ in the command window.

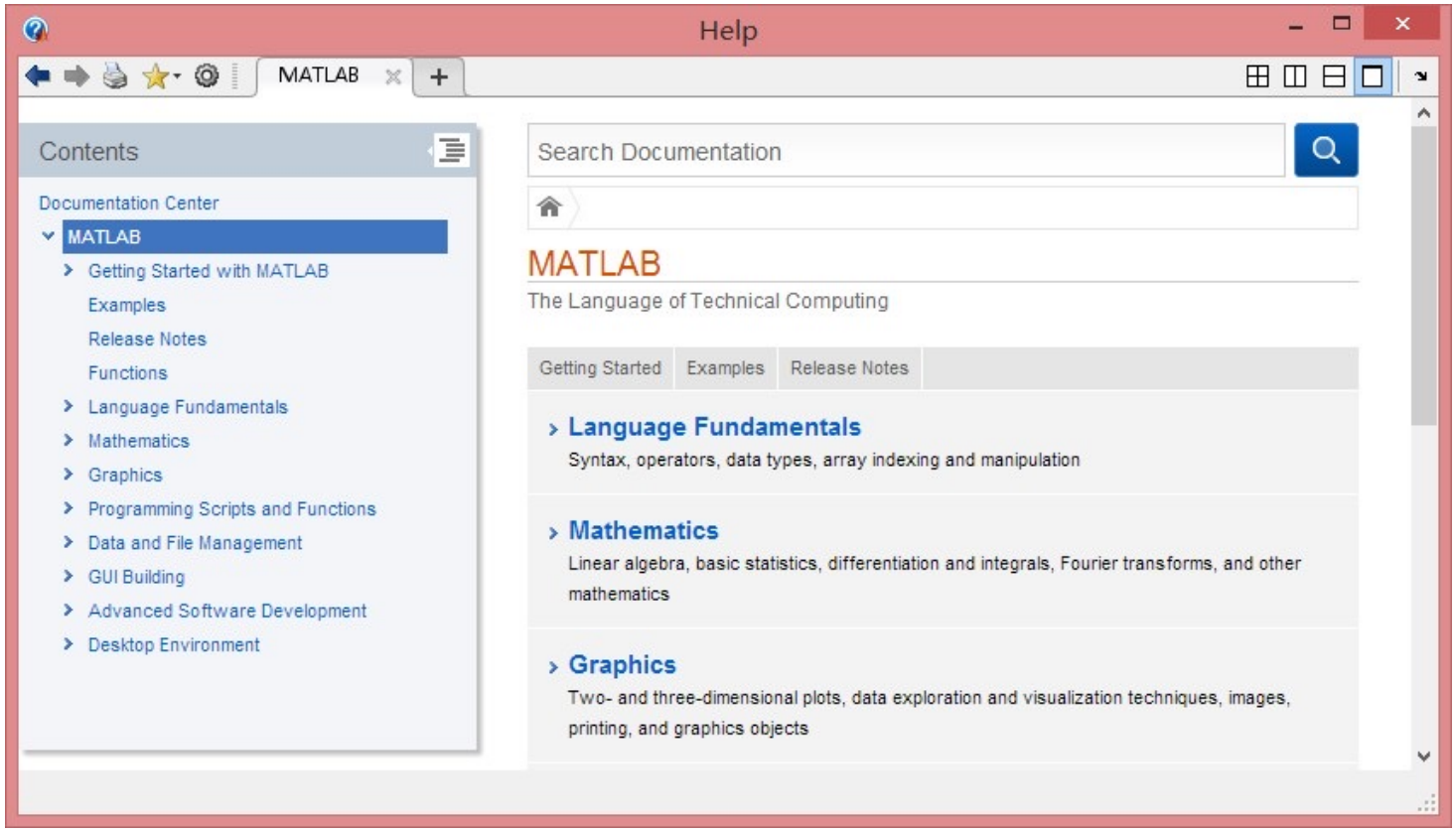
Workspace Window

- The MATLAB workspace consists of the set of variables built up during a MATLAB session and stored in memory.
- You add variables to the workspace by using functions, running M-files, and loading saved workspaces.
- Examples…

Editor/Debugger

- Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions.
- The Editor/Debugger provides a graphical user interface for text editing, as well as for M-file debugging.
- Example: create an M-file and run it in Command Window…

Matlab Help Documentation

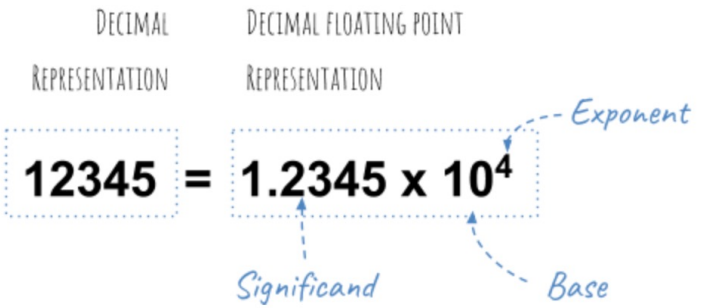


MATLAB Expressions

Variables

- When MATLAB encounters a new variable name, it automatically creates the variable. If the variable already exists, MATLAB changes its value.
- Example: `a = 1`
- Variable names consist of a letter, followed by any number of letters, digits, or underscores.
- MATLAB is case sensitive: It distinguishes between uppercase and lowercase letters.
- Example: `A` and `a` are not the same variable.

Numbers



- MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers.
  - By default, stored as double-precision floating-point numbers
  - A floating-point number is represented with a fixed number of significant digits (the significand) and scaled using an exponent in some fixed base, e.g., 10.
  - Double precision: ~16 significant digits
- Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

```
3          -99          0.0001          9.6397238     1.60210e-20    6.02252e23
1i         -3.14159j    3e5i
```

Operators

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in the MATLAB documentation)
^	Power
’	Complex conjugate transpose
()	Specify evaluation order

Example: (1+1/1000)^1000, guess the result?

Functions

- MATLAB provides a large number of standard elementary mathematical functions.
- Elementary functions: `help elfun`

Some Useful Constants

Constant	Value
pi	3.14159…
i	$\sqrt{-1}$
j	$\sqrt{-1}$
Inf	Infinity
NaN	Not a number

Note that you can overwrite function names and constants. But try to avoid that unless you know what you are doing!

Examples of Expressions

- rho = (1+sqrt(5))/2
- a = abs(3+4i)
- b = sind(30)
- c = log(10)
- d = log10(10)