Lecture 1: What is MATLAB?

Dr. Mohammed Hawa
Electrical Engineering Department
University of Jordan

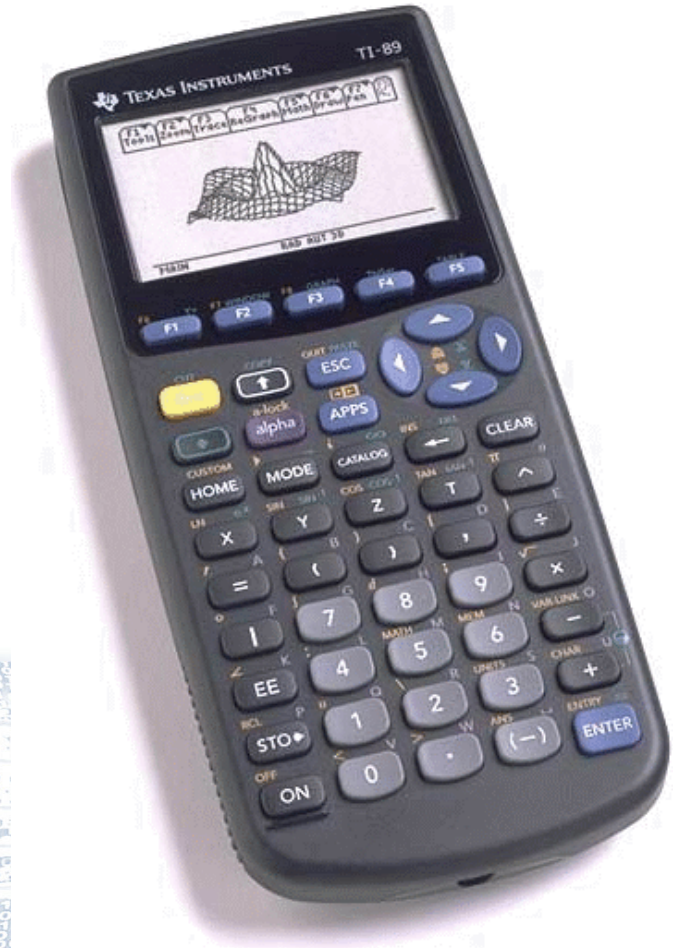
MATLAB

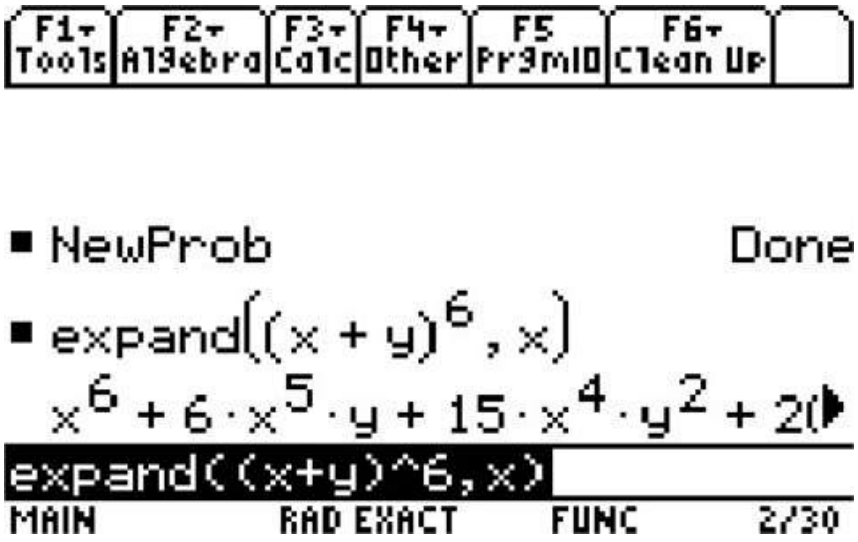
- MATLAB (MATrix LABoratory) is a numerical computing environment and programming language.
- Developed by MathWorks.
- MATLAB is widely used to solve engineering and science problems in academic and research institutions as well as the industry.
- In MATLAB, problems are expressed in familiar mathematical notation.
- MATLAB is an interactive system whose basic data element is a matrix (remember C/C++ arrays!).
- Open-source alternative is: GNU Octave.
- Paid alternative: LabVIEW MathScript

MATLAB can be used for:

- Matrix manipulations (math computations).
- Data analysis, exploration, and plotting.
- Implementation of algorithms.
- Creation of user interfaces.
- Data acquisition.
- Interfacing with programs written in other languages, (e.g., C, C++, Java, and Fortran).
- An optional toolbox (with MuPAD symbolic engine) allows accessing symbolic computing.
- An additional package, Simulink®, adds graphical simulation and model-based design.

Like a *VERY* advanced calculator





Would you go to an engineering exam without a calculator?

Solving Simultaneous Equations

- Find the values of x and y that satisfy the following equations simultaneously :

$$2x + y = 4$$

$$x - y = -1$$

- Can be solved by hand to get: $x = 1, y = 2$

- Remember how?

Simultaneous Equations

- Solving simultaneous equations:

$$\begin{aligned} 2x + y + 2z &= 4 \\ x - y - z &= -1 \\ y - 2z &= 4 \end{aligned}$$

- Can be solved by hand to get:

  $x = 1.2$, $y = 2.8$, $z = 0.6$

- How?

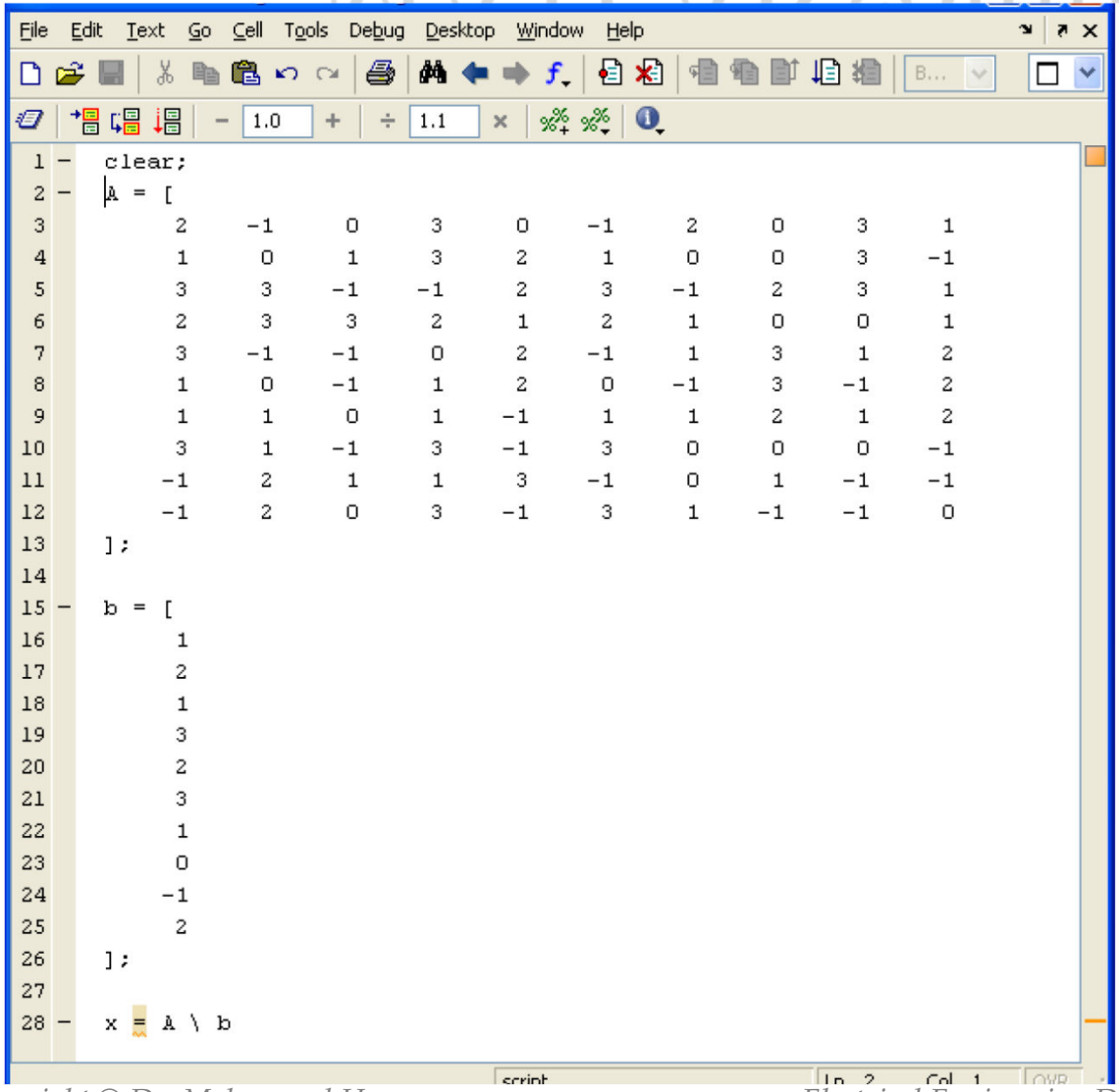
Solving Simultaneous Equations

- Many variables:

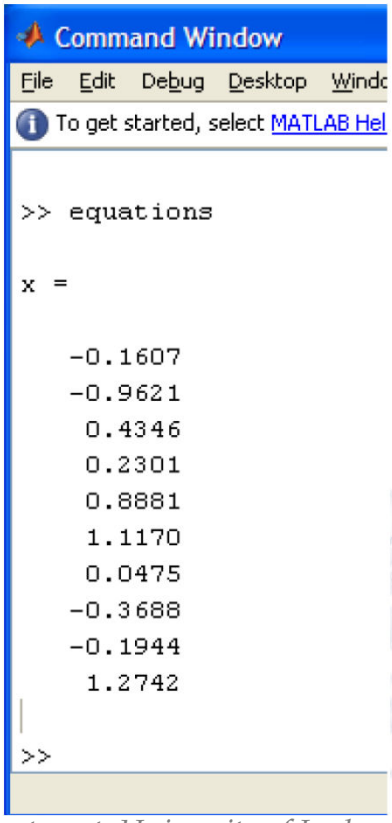
$$
\begin{array}{rrrrrrrrrr}
2x_1 & -x_2 & & +3x_4 & & -x_6 & +2x_7 & & +3x_9 & +x_{10} = 1 \\
x_1 & & +x_3 & +3x_4 & +2x_5 & +x_6 & & & +3x_9 & -x_{10} = 2 \\
3x_1 & +3x_2 & -x_3 & -x_4 & +2x_5 & +3x_6 & -x_7 & +2x_8 & +3x_9 & +x_{10} = 1 \\
2x_1 & +3x_2 & +3x_3 & +2x_4 & +x_5 & +2x_6 & +x_7 & & & +x_{10} = 3 \\
3x_1 & -x_2 & -x_3 & & +2x_5 & -x_6 & +x_7 & +3x_8 & +x_9 & +2x_{10} = 2 \\
x_1 & & -x_3 & +x_4 & +2x_5 & & -x_7 & +3x_8 & -x_9 & +2x_{10} = 3 \\
x_1 & +x_2 & & +x_4 & -x_5 & +x_6 & +x_7 & +2x_8 & +x_9 & +2x_{10} = 1 \\
3x_1 & +x_2 & -x_3 & +3x_4 & -x_5 & +3x_6 & & & & -x_{10} = 0 \\
-x_1 & +2x_2 & +x_3 & +x_4 & +3x_5 & -x_6 & & +x_8 & -x_9 & -x_{10} = -1 \\
-x_1 & +2x_2 & & +3x_4 & -x_5 & +3x_6 & +x_7 & -x_8 & -x_9 & = 2
\end{array}
$$

- Humans are note good at this.
  MATLAB (a computer software) is!

MATLAB solution



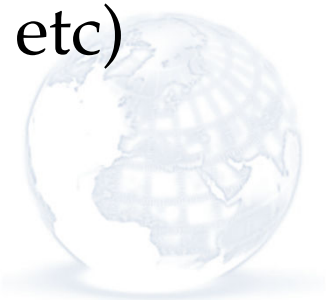
```
clear;
A = [
     2    -1     0     3     0    -1     2     0     3     1
     1     0     1     3     2     1     0     0     3    -1
     3     3    -1    -1     2     3    -1     2     3     1
     2     3     3     2     1     2     1     0     0     1
     3    -1    -1     0     2    -1     1     3     1     2
     1     0    -1     1     2     0    -1     3    -1     2
     1     1     0     1    -1     1     1     2     1     2
     3     1    -1     3    -1     3     0     0     0    -1
    -1     2     1     1     3    -1     0     1    -1    -1
    -1     2     0     3    -1     3     1    -1    -1     0
];

b = [
     1
     2
     1
     3
     2
     3
     1
     0
    -1
     2
];

x = A \ b
```



```
>> equations

x =

   -0.1607
   -0.9621
    0.4346
    0.2301
    0.8881
    1.1170
    0.0475
   -0.3688
   -0.1944
    1.2742

>>
```

MATLAB is powerful!

- We often need to solve systems with 10,000 or 100,000 simultaneous equations (could be non-linear or differential equations too)
- Can be done very quickly using a computer
- This is common in engineering
  - Electrical circuits
  - Image recognition
  - Communication systems (MIMO, OFDM, etc)
  - Operations research
  - Mechanics and dynamics, etc



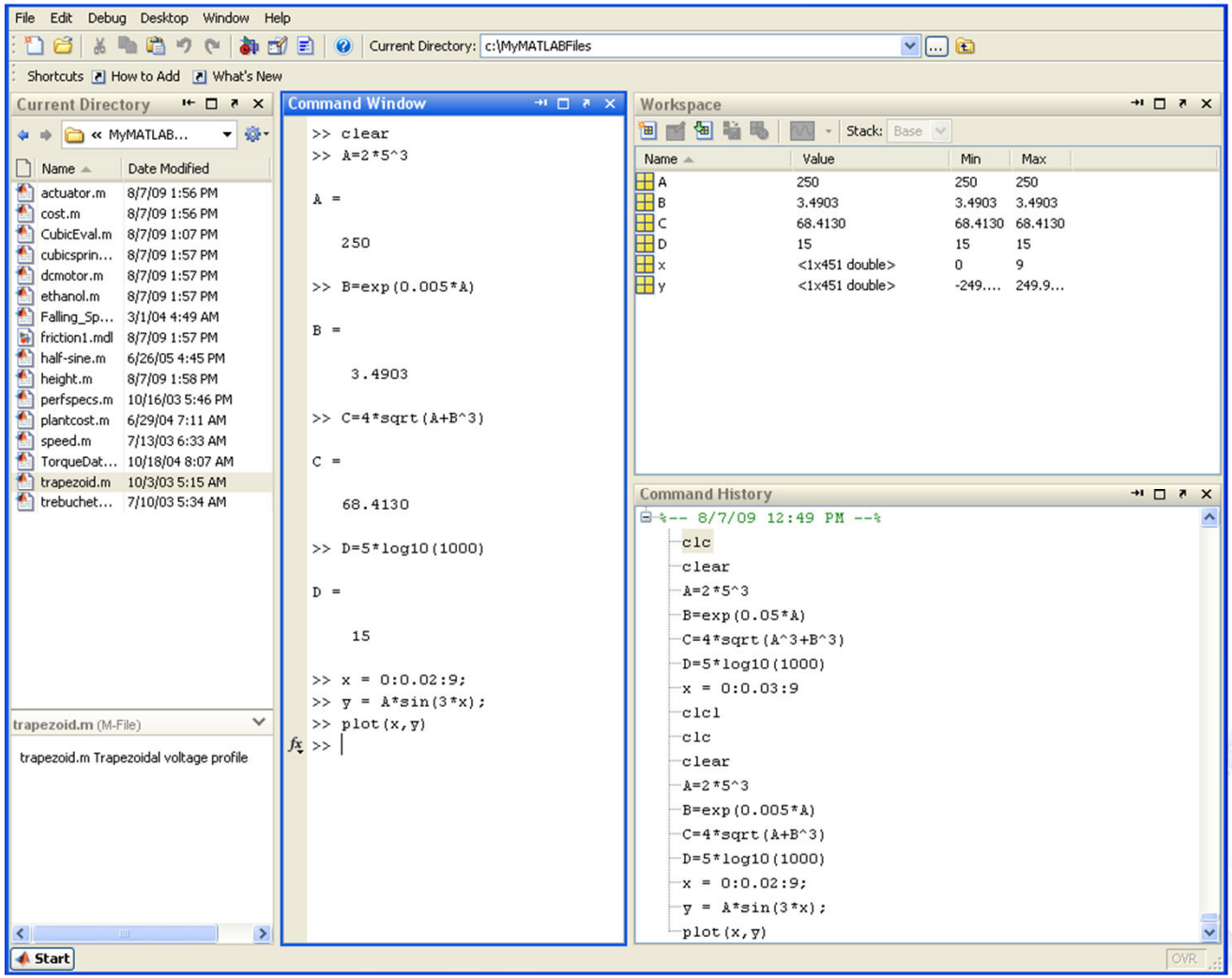
MATLAB vs. Programming languages

- MATLAB is a vector-based numerical analysis language:
  - Can be used as an advanced calculator and graphing tool
  - Also can be used as a programming language
- This is different than the programming languages you are familiar with (C, C++, …)
  - Can be a little frustrating since it takes time and effort to write code in MATLAB
  - But the code is very effective and can be refined gradually

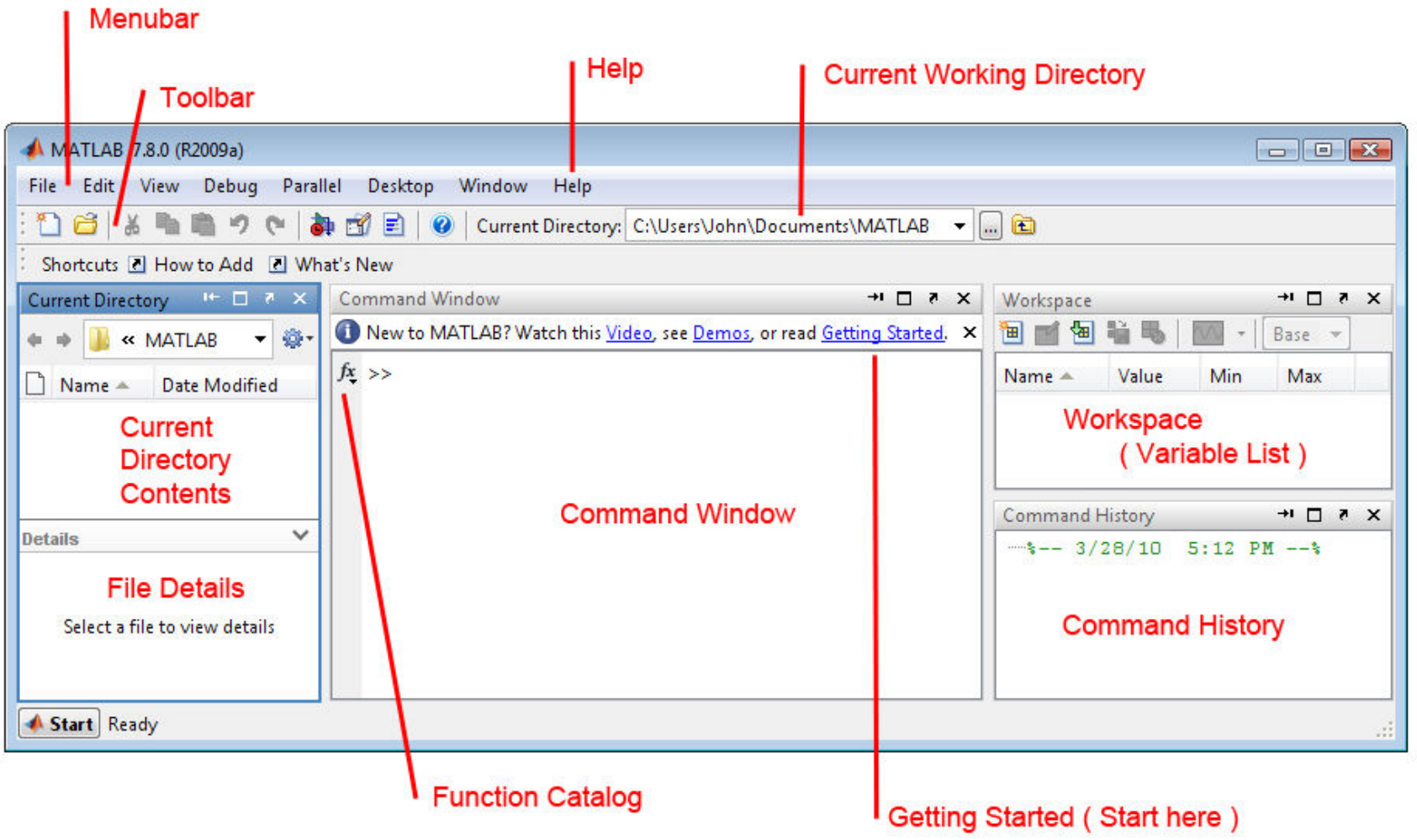
Know about MATLAB

- MATLAB is easy to begin with but needs hard work to master.
- MATLAB is optimized for performing matrix operations.
- MATLAB is interpreted
  - for the most part slower than a compiled language such as C++
  - but interactive and simplifies fixing errors
- Although primarily procedural, MATLAB does have some object-oriented elements.
- MATLAB is NOT a general purpose programming language
- MATLAB is designed for scientific computation and is not suitable for some things (such as parsing text)
- MATLAB is very useful for data analysis and rapid prototyping, but is not designed for large-scale system development.

Let us run MATLAB …

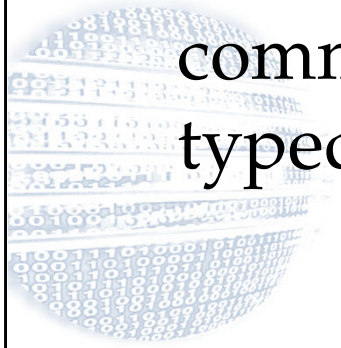


MATLAB Environment



MATLAB as a Calculator

- You can enter expressions at the command line and evaluate them right away.
- The >> symbols indicate where commands are typed.



previous command

next command

```
>> 3 + 5 * 8

ans =

     43

>>
```

Mathematical Operators

Operator	MATLAB	Algebra
+	+	5 + 4 = 9
–	-	5 - 4 = 1
×	*	5 * 4 = 20
÷	/	5 / 4 = 1.25
a^b	a^b	5^4 = 625

Order of Precedence (BEDMAS)

- B = Brackets
- E = Exponentials
- D = Division
- M = Multiplication
- A = Addition
- S = Subtraction

- Careful using brackets: check that opening and closing brackets are matched up correctly.

```
>> 3*4 + 2

ans =

    14

>> 3*(4+2)

ans =

    18
```

Order of Precedence

Precedence	Operation
First	Parentheses ( ), evaluated starting with the innermost pair.
Second	Exponentiation (power) ^ , evaluated from left to right.
Third	Multiplication * and division / with equal precedence, evaluated from left to right.
Fourth	Addition + and subtraction - with equal precedence, evaluated from left to right.

Exercise: Try it yourself

```
>> 8 + 3*5
ans =
    23

>> 8 + (3*5)
ans =
    23

>> (8 + 3)*5
ans =
    55

>> 4^2-12- 8/4*2
ans =
     0

>> 4^2-12- 8/(4*2)
ans =
     3
```

Entering Commands

- MATLAB retains your previous keystrokes.
- Use the ↑ key to scroll back through previous commands.
- Press the ↑ key once to see the previous entry, and so on.
- Use the ↓ key to scroll forward.
- Edit a line using the ← and → arrow keys, the Backspace key, and the Delete key.
- Press the Enter key to execute the command.
- You can copy (highlight & ctrl+c) from Command History window to the Command Window.

Built-in Math Constants

`pi`	π: ratio of circle's circumference to its diameter
`i`	$\sqrt{-1}$: Imaginary unit
`j`	$\sqrt{-1}$: Imaginary unit
`Inf`	∞: Infinity
`NaN`	Not-a-Number
`intmax`	Largest value of integer type
`intmin`	Smallest value of integer type
`ans`	Temporary variable containing the most recent answer
`eps`	The accuracy of floating point precision
	…

```
>> 2*pi
ans =
    6.2832

>> Inf+100000
ans =
   Inf

>> format long g

>> 2*pi
ans =

6.28318530717959

>> 1+ans
ans =

7.28318530717959
```

Exercise

```
>> 1/0
ans =
   ???

>> 0/0
ans =
   ???

>> 7/2*i
ans =
    ???

>> 7/2i
ans =
    ???
```

Exercise: Answers

```
>> 1/0
ans =
   Inf

>> 0/0
ans =
   NaN

>> 7/2*i
ans =
   0 + 3.5000i

>> 7/2i
ans =
   0 - 3.5000i
```

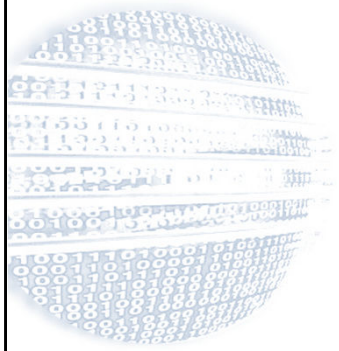
Possible Formats

Command	Description and example
`format short`	Four decimal digits (the default); 13.6745.
`format long`	16 digits; 17.27484029463547.
`format short e`	Five digits (four decimals) plus exponent; 6.3792e+03.
`format long e`	16 digits (15 decimals) plus exponent; 6.379243784781294e−04.
`format bank`	Two decimal digits; 126.73.
`format +`	Positive, negative, or zero; +.
`format rat`	Rational approximation; 43/7.
`format compact`	Suppresses some blank lines.
`format loose`	Resets to less compact display mode.

Built-in Functions

- Like a calculator, MATLAB has many built-in mathematical functions.

```
>> log2(131072)
ans =
    17

>> sqrt(4)
ans =
     2

>> abs(-3)
ans =
     3

>> exp(-1)
ans =

0.367879441171442
```



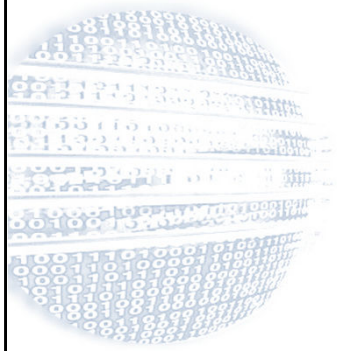
Common Built-in Functions

Function	MATLAB syntax*
e^x	`exp(x)`
$\sqrt{x}$	`sqrt(x)`
$\ln x$	`log(x)`
$\log_{10} x$	`log10(x)`
$\cos x$	`cos(x)`
$\sin x$	`sin(x)`
$\tan x$	`tan(x)`
$\cos^{-1} x$	`acos(x)`
$\sin^{-1} x$	`asin(x)`
$\tan^{-1} x$	`atan(x)`

*The MATLAB trigonometric functions listed here use radian measure. Trigonometric functions ending in `d`, such as `sind(x)` and `cosd(x)`, take the argument `x` in degrees. Inverse functions such as `atand(x)` return values in degrees.

Exercise: Discussed Later…

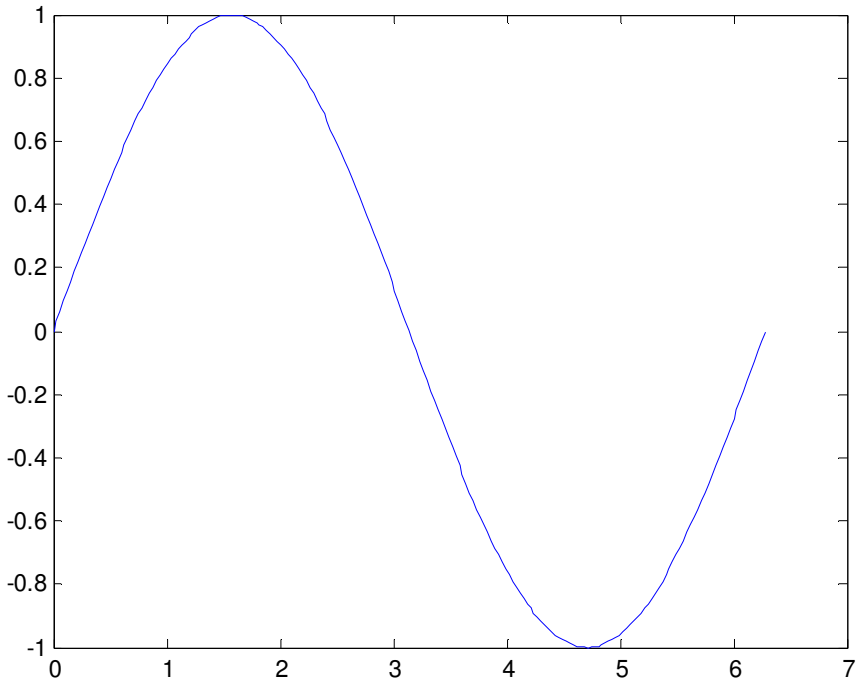
```
x = 0:pi/100:2*pi;
y = sin(x);
plot(x,y)
```

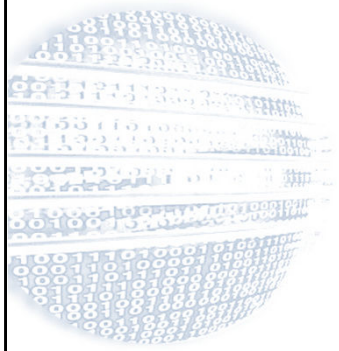


- By the way, what is the purpose of the semicolon at the end of the command?

Exercise: Discussed Later…

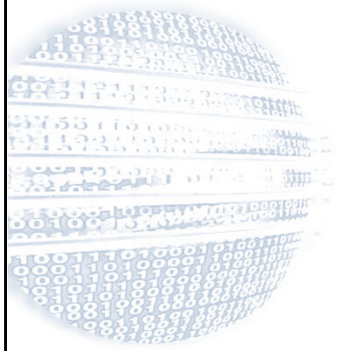
```
x = 0:pi/100:2*pi;
y = sin(x);
plot(x,y)
```





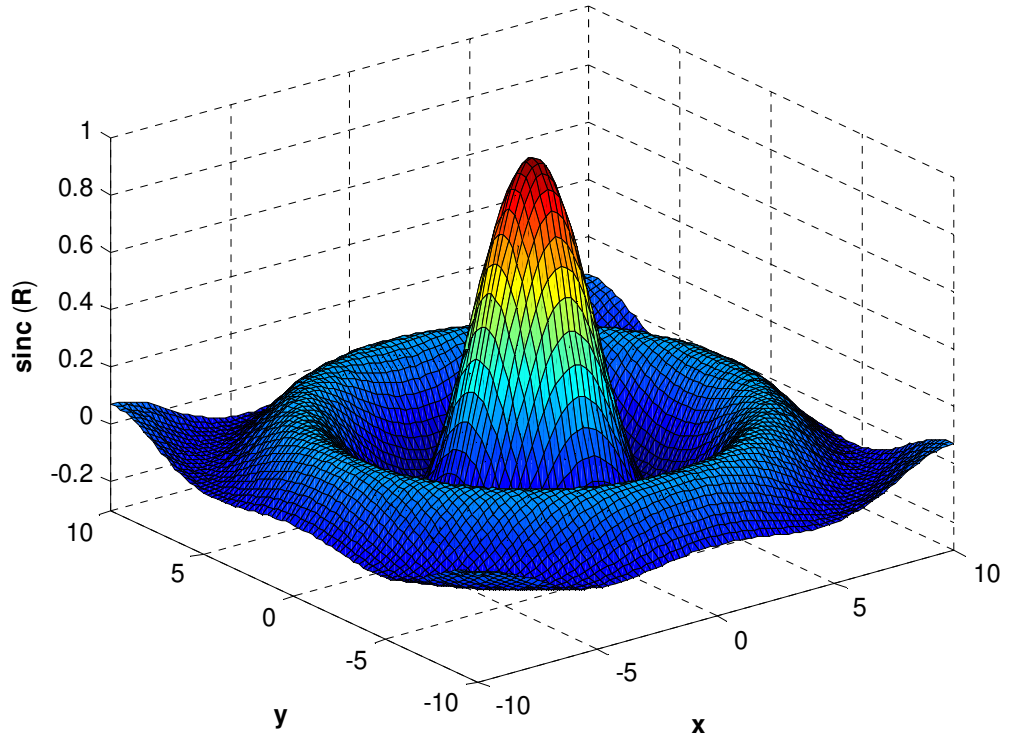
Exercise 2: Discussed Later…

```
[X,Y] = meshgrid(-10:0.25:10,-10:0.25:10);
f = sinc(sqrt((X/pi).^2+(Y/pi).^2));
surf(X,Y,f);
axis([-10 10 -10 10 -0.3 1])
```



Exercise 2: Discussed Later…

```
[X,Y] = meshgrid(-10:0.25:10,-10:0.25:10);
f = sinc(sqrt((X/pi).^2+(Y/pi).^2));
surf(X,Y,f);
axis([-10 10 -10 10 -0.3 1])
```



To Know More: `help`

```
>> help
HELP topics:

matlab\general          - General purpose commands.
matlab\ops              - Operators and special characters.
matlab\lang             - Programming language constructs.
matlab\elmat            - Elementary matrices and matrix manipulation.
matlab\randfun          - Random matrices and random streams.
matlab\elfun            - Elementary math functions.
matlab\specfun          - Specialized math functions.
matlab\matfun           - Matrix functions - numerical linear algebra.
matlab\datafun          - Data analysis and Fourier transforms.
matlab\polyfun          - Interpolation and polynomials.
matlab\funfun           - Function functions and ODE solvers.
matlab\sparfun          - Sparse matrices.
matlab\scribe           - Annotation and Plot Editing.
matlab\graph2d          - Two dimensional graphs.
matlab\graph3d          - Three dimensional graphs.
matlab\specgraph        - Specialized graphs.
matlab\graphics         - Handle Graphics.
matlab\uitools          - Graphical User Interface Tools.
matlab\strfun           - Character strings.
matlab\imagesci         - Image and scientific data
matlab\plottools        - Graphical User Interface Tools.
fuzzy\fuzzy             - Fuzzy Logic Toolbox
images\images           - Image Processing Toolbox
signal\signal           - Signal Processing Toolbox
wavelet\wavelet         - Wavelet Toolbox
...
```

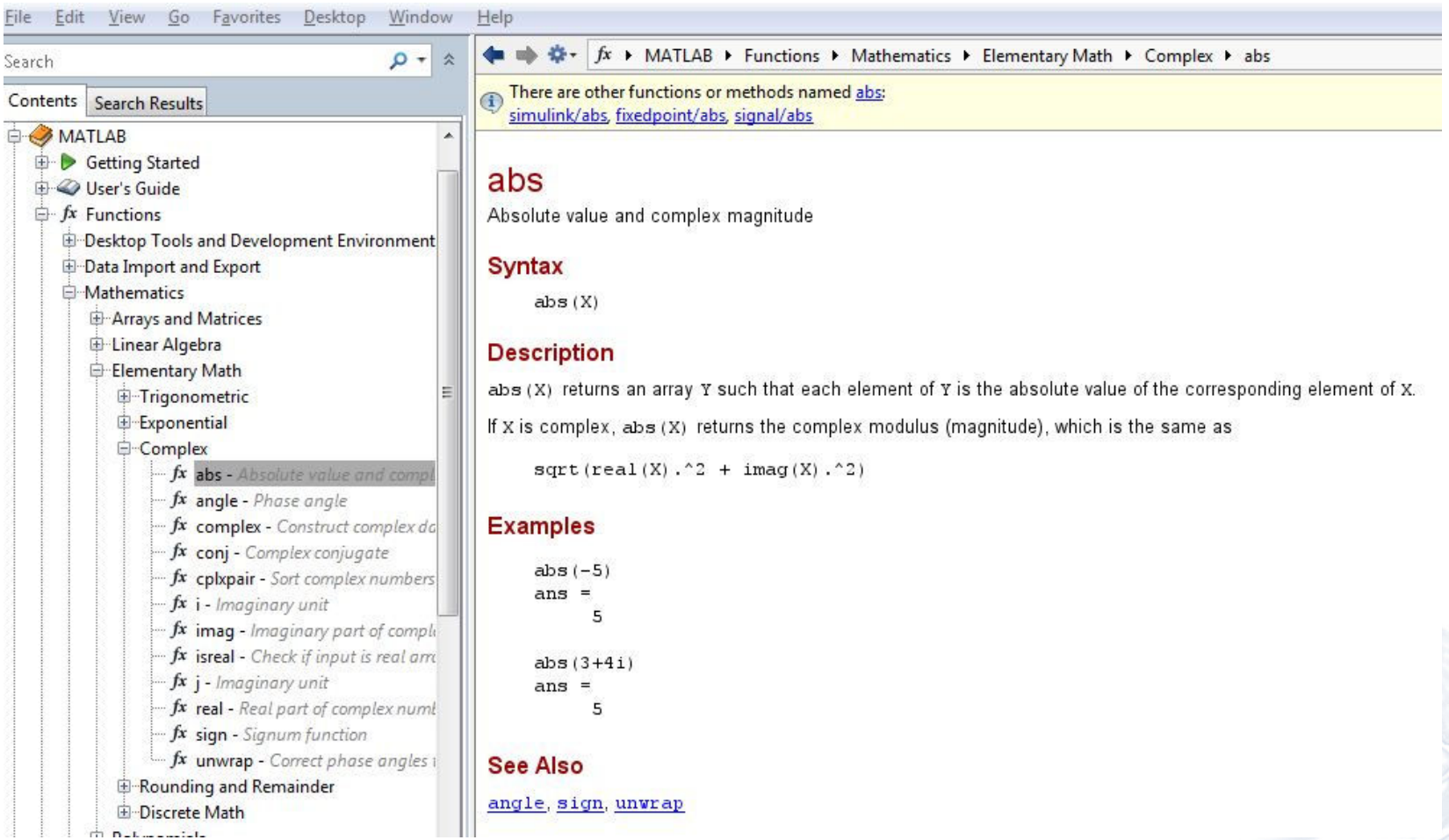
Go inside: `help`

```
>> help elfun
  Elementary math functions.

  Trigonometric.
    sin         - Sine.
    sind        - Sine of argument in degrees.
    sinh        - Hyperbolic sine.
    asin        - Inverse sine.
    asind       - Inverse sine, result in degrees.
    asinh       - Inverse hyperbolic sine.
    cos         - Cosine.
    ...

  Exponential.
    exp         - Exponential.
    expm1       - Compute exp(x)-1 accurately.
    log         - Natural logarithm.
    log1p       - Compute log(1+x) accurately.
    log10       - Common (base 10) logarithm.
    log2        - Base 2 logarithm and dissect floating point num.
    pow2        - Base 2 power and scale floating point number.
    realpow     - Power that will error out on complex result.
    reallog     - Natural logarithm of real number.
    ...

  Rounding and remainder.
    fix         - Round towards zero.
    floor       - Round towards minus infinity.
    ceil        - Round towards plus infinity.
    round       - Round towards nearest integer.
    mod         - Modulus (signed remainder after division).
    rem         - Remainder after division.
    sign        - Signum.
```

For a specific function: `help exp`

```
>> help exp
 EXP    Exponential.
    EXP(X) is the exponential of the elements of X, e to the X.
    For complex Z=X+i*Y, EXP(Z) = EXP(X)*(COS(Y)+i*SIN(Y)).

    See also expm1, log, log10, expm, expint.

    Overloaded methods:
       codistributed/exp
       fints/exp

    Reference page in Help browser
       doc exp
```

To Know More: `doc abs`



Where do you get more help?

- Read your textbook.
- Practice the end-of-chapter examples.
- References in the syllabus.
- MATLAB Central: http://www.mathworks.com/matlabcentral/
- Google
- YouTube