BIL 108E
Introduction to Scientific and Engineering Computing

ASST. PROF. DR. İPEK AKIN

Starting with MATLAB

SCIENTIFIC COMPUTING

Scientific computing is the field concerned with

- **constructing mathematical models and numerical solution techniques**
- **using computers to analyse and solve scientific, social scientific and engineering problems.**

Scientists and engineers develop computer programs, application software, that model systems being studied and run these programs with various sets of input parameters.

APPLICATIONS OF SCIENTIFIC COMPUTING

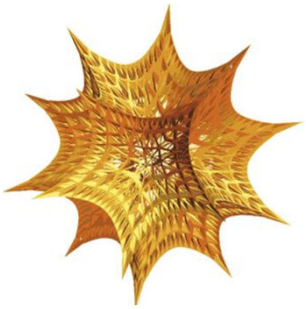
Numerical simulations

- Reconstruct and understand known events (e.g., earthquake, tsunamis and other natural disasters).
- Predict future or unobserved situations (e.g., weather, sub-atomic particle behaviour).

Model fitting and data analysis

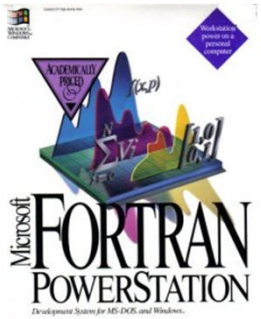
Optimization of processes

PROGRAMMING LANGUAGES FOR SCIENTIFIC COMPUTING

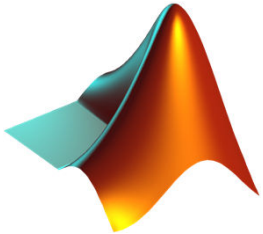


MATLAB

Mathematica



Scilab



GNU Octave

Matcad

Maple

WHY MATLAB ?

Advantages

- Interpreter
- Many many toolboxes
- Visualization
- Friendly environment
- Wide usage

Disadvantages

- Slower than C and Fortran
- Not freeware/open source

INTRODUCTION

- MATLAB is a powerful language for **technical computing**.

- The name **MATLAB** stands for **MATrix LABoratory**, because the system was designed to make matrix computations particularly easy.

MATLAB

Computer Programming Language

- Matrices
- Linear Algebra
- Numerical Analysis

Interactive Software Environment

- Manage variables
- Import and export data
- Perform calculations
- Generate plots
- Develop and manage files

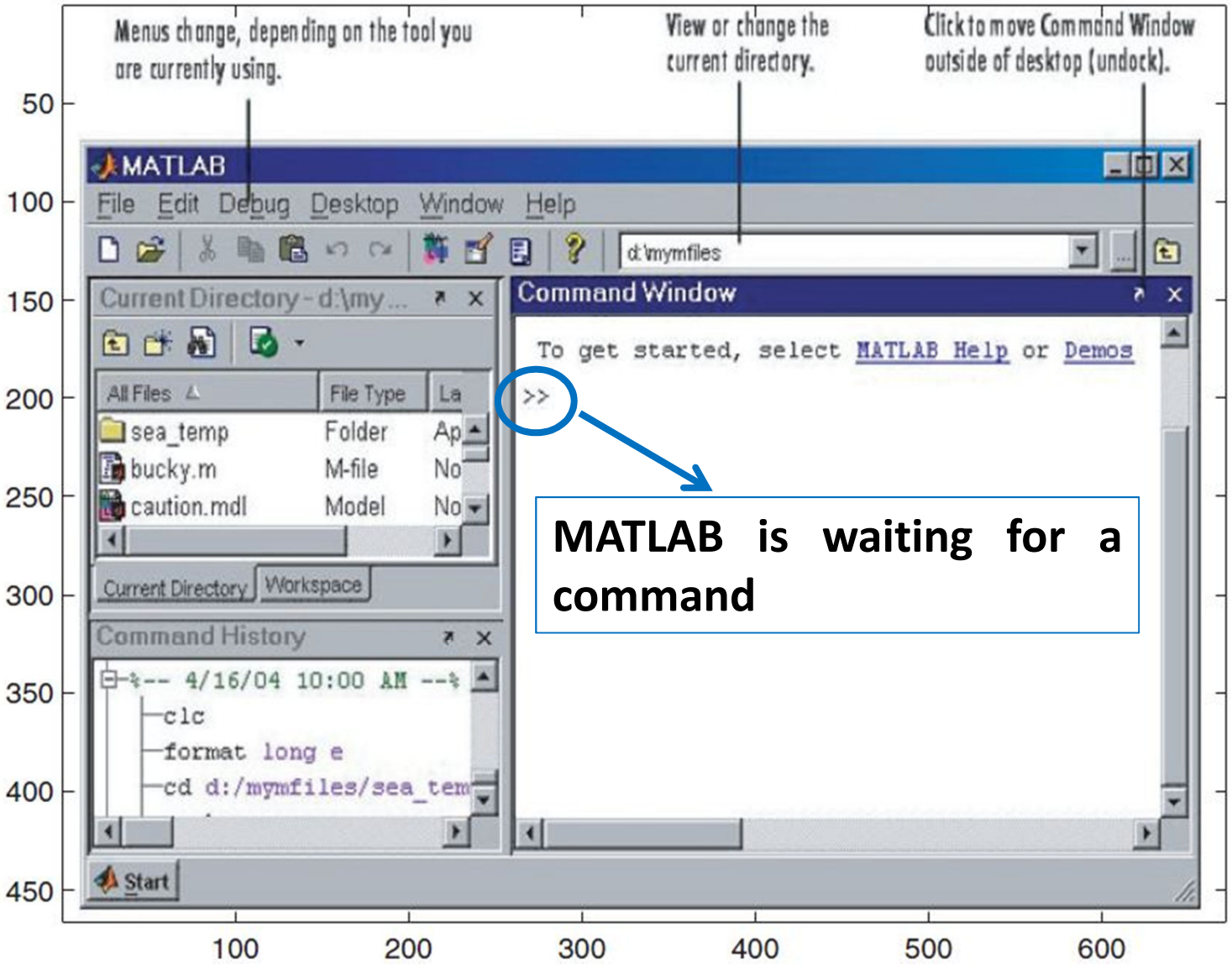
INTRODUCTION

- MATLAB can be used interactively
    - Type some commands at the special MATLAB prompt, and get the answers immediately.
    - The problems can be solved easily (e.g. finding a square root, or the solution to a system of differential equations)

INTRODUCTION

Two essentials requirements for successful MATLAB programming:

1. Learn the exact rules for writing MATLAB statements.
2. Develop a logical plan of attack for solving particular problems



QUIT → 1) Select Exit MATLAB from the desktop File menu

2) Enter quit or exit at the Command Window prompt

Do *not* click on the close box in the top right corner of the MATLAB desktop

INTRODUCTION

1. Command window
2. Workspace window
3. Command history window

3 of 8 different windows

MATLAB Windows

Window	Purpose
Command Window	Main window, enters variables, runs programs.
Figure Window	Contains output from graphic commands.
Editor Window	Creates and debugs script and function files.
Help Window	Provides help information.
Launch Pad Window	Provides access to tools, demos, and documentation.
Command History Window	Logs commands entered in the Command Window.
Workspace Window	Provides information about the variables that are used.
Current Directory Window	Shows the files in the current directory.

Command Window

- Main window

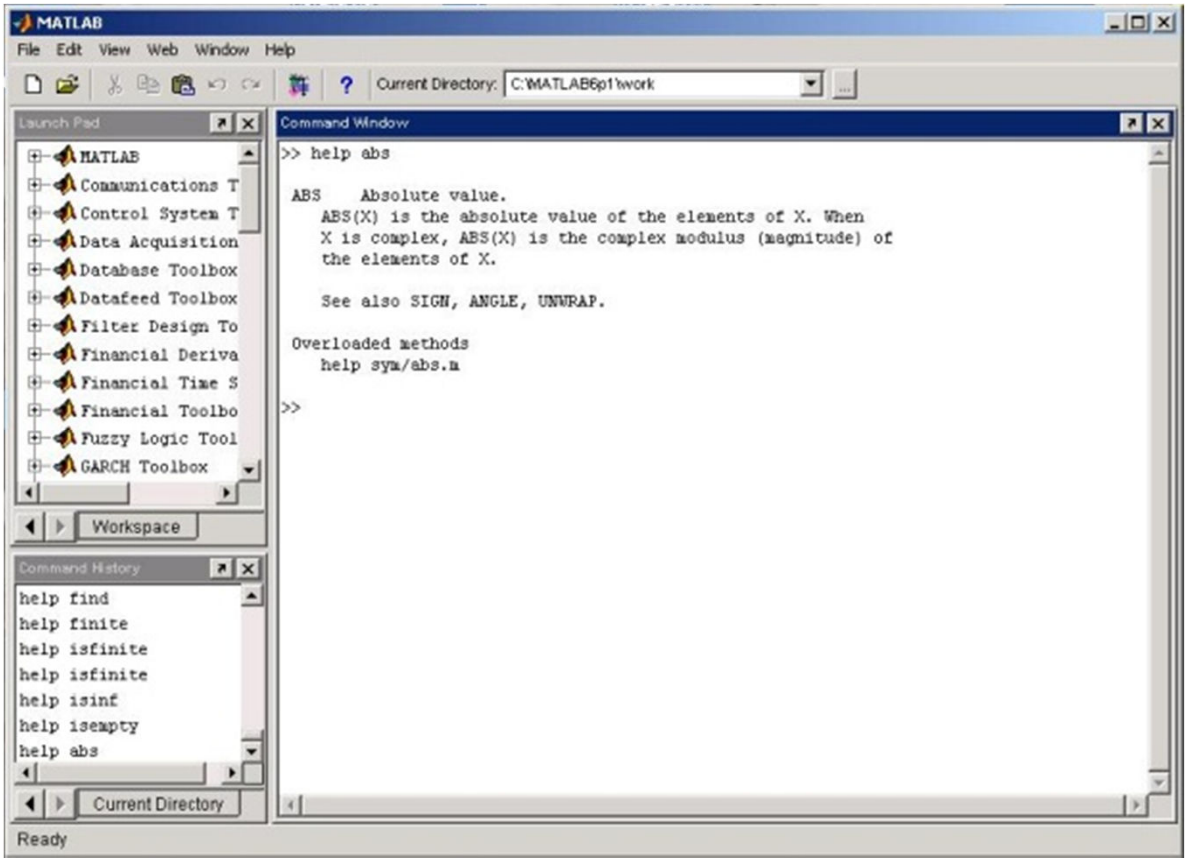
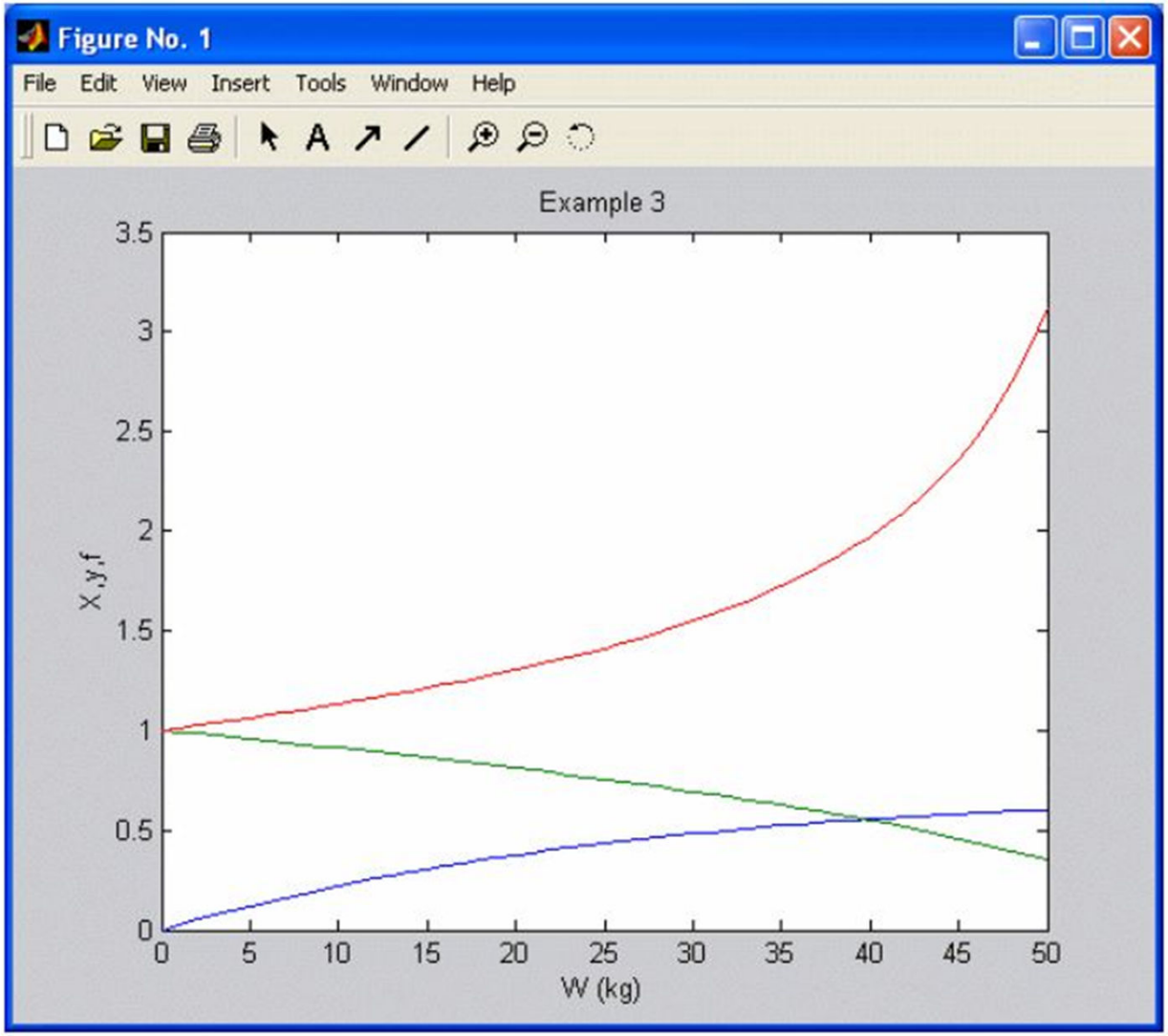


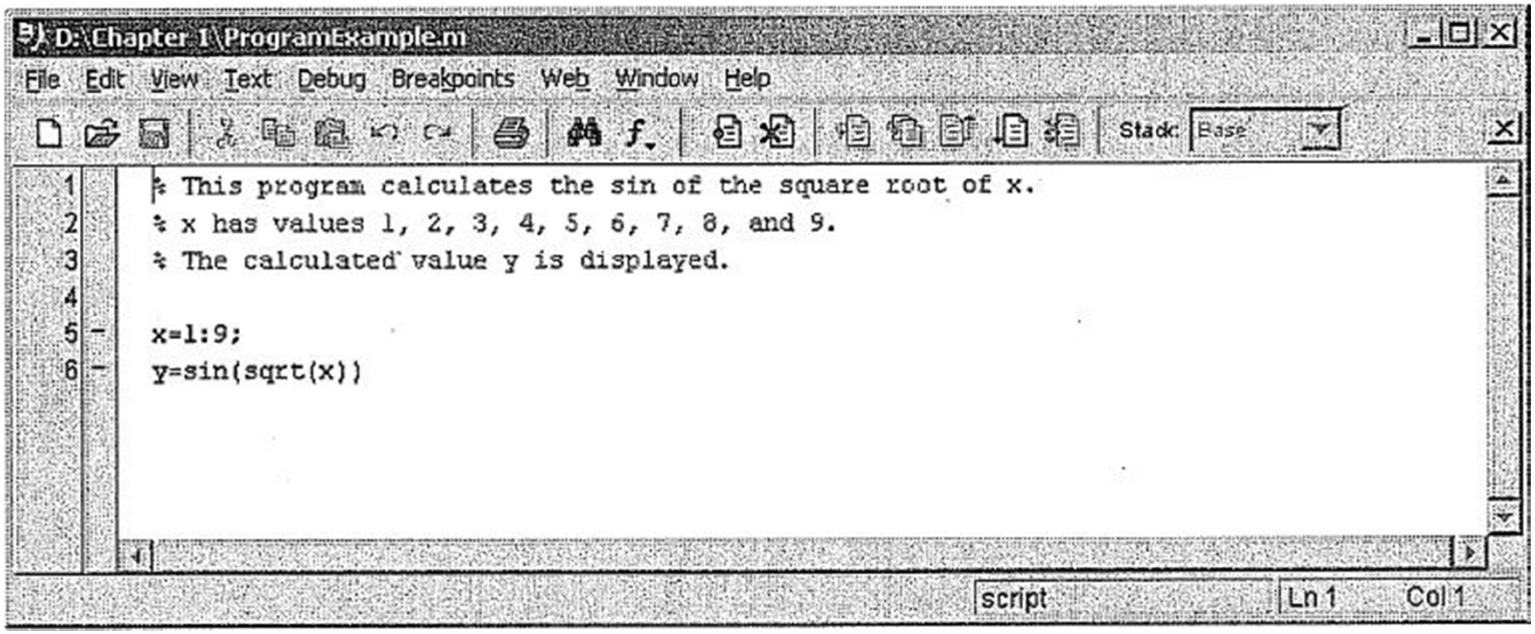
Figure Window

- Opens automatically when graphics commands are executed
- Contains graphs created by these commands



Editor Window

- Used for writing and editing programs
- Opened from the File menu in the Command Window
- Used for creating script files



Help Window

- Interactive and can be used to obtain any information on any feature of MATLAB.

The closed windows can be reopened by selecting them from the View menu

EXPRESSIONS

MATLAB is an expression based language.

It interprets and evaluates typed expressions.

- **Numbers**
- **Variables**
- **Operators**
- **Functions**

NUMBERS

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

VARIABLES

Variables are a fundamental concept in MATLAB.

ASSIGNMENT OPERATOR

- **The "=" sign in MATLAB is called the assignment or replacement operator.**
- **When you type x=3, you tell MATLAB to assign the value 3 to the variable x.**
- **In MATLAB, you cannot type 6=x**
- **It is not same MATLAB operator = and the equal sign in mathematics.**

VARIABLE NAMES

- **Variable names are case-sensitive.**

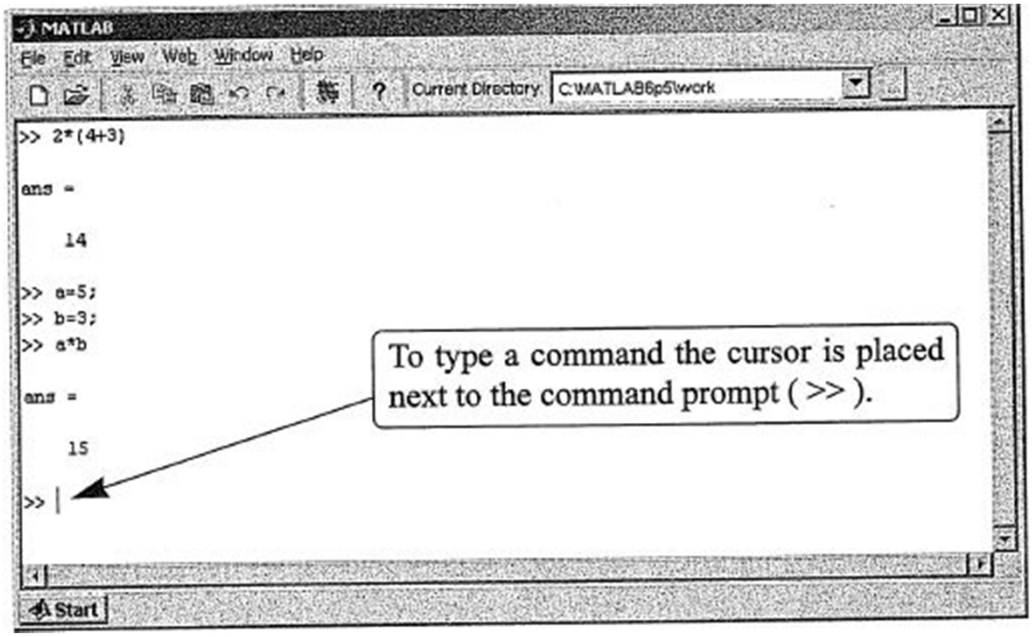
  Speed , speed, SPEED

- **Variable names must begin with a letter.**

- **Variable names must contain less than 32 characters.**

Working in the Command Window

- Used for executing commands, opening other windows, running programs written by the user, managing the software



- Command + Enter key → command is executed

- Only the last command is executed. Everything executed previously is unchanged.

Working in the Command Window

- Several commands can be typed in the same line by **typing a comma btw the commands**

  Enter → **commands are executed in order from left to right**

- Not possible to go back to a previous line in the Command Window, make a correction, and then re-execute the command

- A previously typed command can be recalled to the command prompt with the up-arrow key (↑).

- When the command is displayed at the command propmt, it can be modified and executed. The down-arrow key (↓) can be used to move down the previously typed commands.

Working in the Command Window

- If a command is too long to fit in one line, it can be continued to the next line by typing three periods **…** (called an ellipsis) and pressing **Enter** key

- The command can continue line after line up to 4096 characters

Working in the Command Window

The Semicolon (;)

- If a ; is typed at the end of a command the output of the command is not displayed
- Typing a ; is useful when the result is obvious or known, or when the output is very large
- If several commands are typed in the same line, the output from any of the commands will not be displayed if a ; is typed btw the commands instead of a ,

Working in the Command Window

Typing %

- When **%** is typed in the beginning of a line, the line is designated as a comment → line is not executed with **Enter** key

clc command

- Type **clc** + press **enter** → clears the Command Window

Arithmetic Operations with Scalars

Operation	Symbol	Example
Addition	+	5 + 3
Subtraction	–	5 – 3
Multiplication	*	5 * 3
Right division	/	5 / 3
Left division	\	5 \ 3 = 3 / 5
Exponentiation	^	5 ^ 3 (means $5^3 = 125$)

SPECIAL VARIABLES AND CONSTANT

Command	Description
ans	Temporary variable containing the most recent answer
eps	Specifies the accuracy of floating point precision
i, j	The imaginary unit $\sqrt{-1}$
Inf	Infinity
NaN	Indicates an undefined numerical result
pi	The number of π

Examples

```
>> 8/10
ans =
      0.8000
```

```
>> 5*ans
ans =
      4
```

```
>> r=8/10
r =
    0.8000
>> r
r =
    0.8000
>> s=20*r
s =
    16
```

Order of Precedence

Precedence	Operation
First	Parentheses, evaluated starting with the innermost pair.
Second	Exponentiation, 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.

- If 2 or more operations have the same precedence, the expression is executed from left to rights.

Examples of Precedence

```
>> 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
```

Examples of Precedence

```
>> 3*4^2 + 5
ans =
     53
```

```
>>27^(1/3) + 32^(0.2)
ans =
     5
```

```
>>(3*4)^2 + 5
ans =
     149
```

```
>>27^1/3 + 32^0.2
ans =
     11
```

Examples of Precedence

```
>> 4 + 5/3 + 2
ans =
   7.6667
>> 5^3/2
ans =
  62.5000
>> 27^(1/3) + 32^0.2
ans =
   5
>> 27^1/3 + 32^0.2
ans =
   11

>> 0.7854 - (0.7854)^3/(1*2*3) + 0.785^5/(1*2*3*4*5)...
- (0.785)^7/(1*2*3*4*5*6*7)

ans =
   0.7071
>>
```

5/3 is executed first.

5^3 is executed first, /2 is executed next.

1/3 is executed first, 27^(1/3) and 32^0.2 are executed next, and + is executed last.

27^1 and 32^0.2 are executed first, /3 is executed next, and + is executed last.

Type three periods ... (and press **Enter**) to continue the expression on the next line.

The last expression is the first 4 terms of the Taylor series for $\sin(\pi/4)$.

Display Formats

- MATLAB has several other formats for displaying numbers. Details of these formats can be obtained by typing **help format** in the **Command Window**.

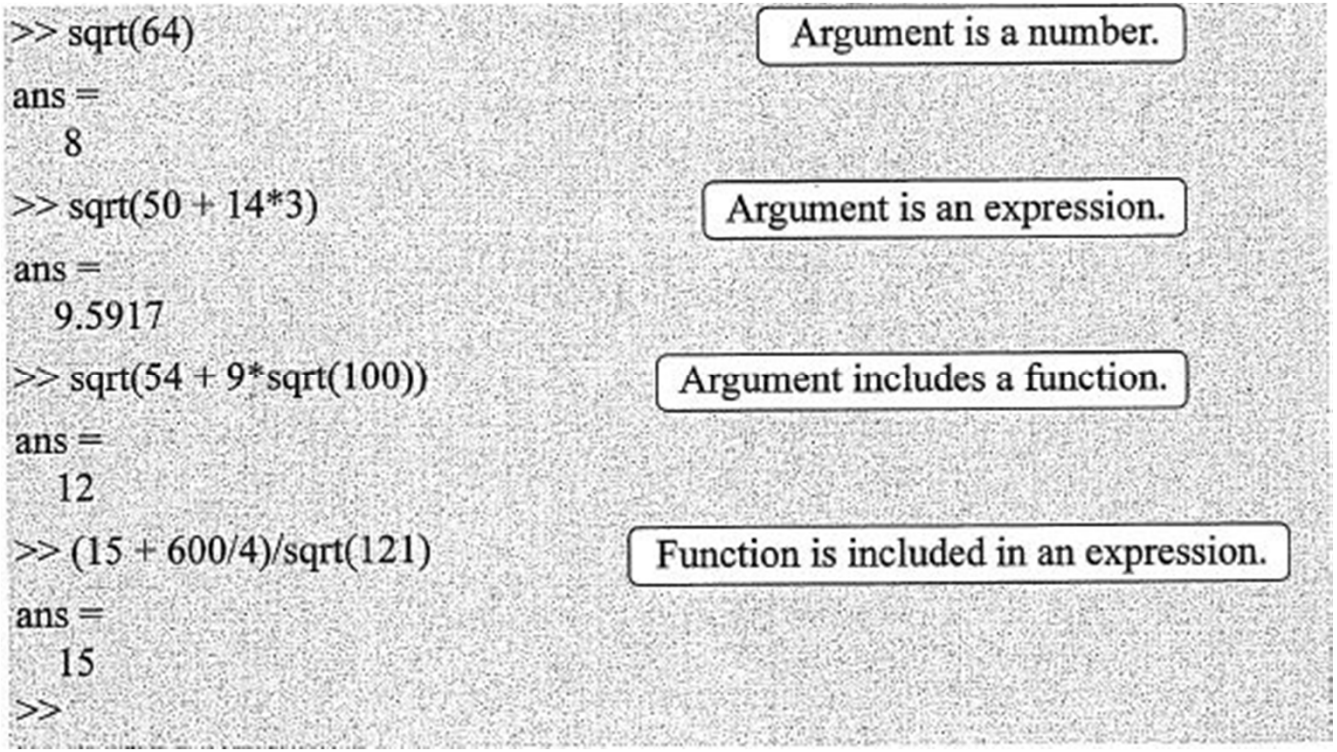
- The format in which numbers are displayed does not affect how MATLAB computes and save numbers.

Command	Description	Example
`format short`	Fixed-point with 4 decimal digits for: $0.001 \le number \le 1000$ Otherwise display format `short e`.	>> 290/7 ans = 41.4286
`format long`	Fixed-point with 14 decimal digits for: $0.001 \le number \le 100$ Otherwise display format `long e`.	>> 290/7 ans = 41.42857142857143
`format short e`	Scientific notation with 4 decimal digits.	>> 290/7 ans = 4.1429e+001
`format long e`	Scientific notation with 15 decimal digits.	>> 290/7 ans = 4.142857142857143e+001
`format short g`	Best of 5-digit fixed or floating point.	>> 290/7 ans = 41.429
`format long g`	Best of 15-digit fixed or floating point.	>> 290/7 ans = 41.4285714285714
`format bank`	Two decimal digits.	>> 290/7 ans = 41.43
`format compact`	Eliminates empty lines to allow more lines with information displayed on the screen.	
`format loose`	Adds empty lines (opposite of `compact`).	

Elementary Math Built-In Functions

- Basic arithmetic operations + **functions**
- Function has a name and argument in parantheses

sqrt (x) → argument is x
name is sqrt



Elementary Math Functions

Function	Description	Example
sqrt (x)	Square root	>> sqrt(81) ans= 9
exp (x)	Exponential (e^x)	>> exp(5) ans= 148.4132
abs (x)	Absolute value	>> abs(-24) ans= 24
log (x)	Natural logarithm Base e logarithm (ln)	>> log(1000) ans= 6.9078
log10 (x)	Base 10 logarithm	>> log10(1000) ans= 3.0000
factorial (x)	The factorial function x! (x must be a + integer)	>> factorial(5) ans= 120

Trigonometric Math Functions

Function	Description	Example
`sin(x)`	Sine of angle x (x in radians).	>> sin(pi/6) ans = 0.5000
`cos(x)`	Cosine of angle x (x in radians).	>> cos(pi/6) ans = 0.8660
`tan(x)`	Tangent of angle x (x in radians).	>> tan(pi/6) ans = 0.5774
`cot(x)`	Cotangent of angle x (x in radians).	>> cot(pi/6) ans = 1.7321

- The inverse trigonometric functions are asin (x), acos (x), atan (x) and acot (x).
- The hyperbolic trigonometric functions are sinh (x), cosh (x), tanh (x) and coth (x).

Rounding Functions

Function	Description	Example
`round(x)`	Round to the nearest integer.	>> round(17/5) ans = 3
`fix(x)`	Round towards zero.	>> fix(13/5) ans = 2
`ceil(x)`	Round towards infinity.	>> ceil(11/5) ans = 3
`floor(x)`	Round towards minus infinity.	>> floor(-9/4) ans = -3
`rem(x,y)`	Returns the remainder after x is divided by y.	>> rem(13,5) ans = 3
`sign(x)`	Signum function. Returns 1 if $x > 0$, -1 if $x < 0$, and 0 if $x = 0$.	>> sign(5) ans = 1

Useful Commands for Managing Variables

Command	Outcome
`clear`	Removes all variables from the memory.
`clear x y z`	Removes only variables `x`, `y`, and `z` from the memory.
`who`	Displays a list of the variables currently in the memory.
`whos`	Displays a list of the variables currently in the memory and their size together with information about their bytes and class.

Example

A trigonometric identity is given by :

$$\cos^2 \frac{x}{2} = \frac{\tan x + \sin x}{2 \tan x}$$

Verify that the identity is correct by calculating each side of the equation, substituting x=π/5.

Solution

```
>> x = pi/5;                                  Define x.
>> LHS = cos(x/2)^2                           Calculate the left-hand side.
LHS =
    0.9045
>> RHS = (tan(x) + sin(x))/(2*tan(x))         Calculate the right-hand side.
RHS =
    0.9045
>>
```

MATLAB FILES

- **The Workspace can be saved to a data file**

  extension is .mat (ex: example1.mat)

- **.mat files can be reloaded by using open from file menu**