Programming In The Real World

Programming is a powerful tool for solving problems in every day industry settings. MATLAB is a program used for solving problems through numerical computation, plotting and logical programming. Devices such as robots are controlled with large programs that act as a set of instructions for completing a task.
Today's Learning Objectives

- After today’s class, students will be able to:
  - Demonstrate proper usage of basic MATLAB features (e.g., the Command Window, script files, other default windows, arithmetic operations, assigning variables and names, built-in functions, *help* command).
  - Identify need to translate mathematical notation into proper MATLAB syntax.
Today's Topics

- Intro to the MATLAB interface
- Variables: naming rules & calculations
- Functions
- Script files and other commands
Opening MATLAB

- Students, please open MATLAB now.

- CLICK on the shortcut icon →
  - Alternatively, select...
    ```
    start/All programs/MATLAB
    ```

- The following prompt should appear in the command window after a lengthy initialization process: `>>`
MATLAB Display

- Ribbon
  - Useful operations

- Current Directory
  - List of files

- Workspace
  - Defined variable values

- Command history
  - Displays what has been typed

![MATLAB Display with Ribbon, Current Directory, Workspace, and Command History]

- Command prompt

- Command Window
MATLAB’s Working Directory

- Current working directory is where files are saved and run from.
- When you first start MATLAB, it is a good practice to change the working directory to your Z: drive or USB device.
- Browse icon available.
Order of Precedence - important

- Higher-precedence operations are executed before lower-precedence operations
  - Similar to Excel or programmable calculators
- Two operations having the same precedence are executed from left to right

<table>
<thead>
<tr>
<th>PRECEDENCE</th>
<th>OPERATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>Parentheses (innermost pair first)</td>
</tr>
<tr>
<td>Second</td>
<td>Exponentiation</td>
</tr>
<tr>
<td>Third</td>
<td>Multiplication and division</td>
</tr>
<tr>
<td>Fourth</td>
<td>Addition and subtraction</td>
</tr>
</tbody>
</table>
Order of Precedence

- Example: Find the average of two numbers: 7 and 8

  \[\text{\textgreater\textgreater} \quad 7+8/2 \quad \text{\textgreater\textgreater} \quad (7+8)/2\]
Order of Precedence

- Example: Find the average of two numbers: 7 and 8

\[
\begin{align*}
\text{>> } & 7+8/2 \\
\text{ans} & = 11 \\
\text{Incorrect result}
\end{align*}
\]

\[
\begin{align*}
\text{>> } & (7+8)/2 \\
\text{ans} & = 7.500
\end{align*}
\]
Order of Precedence

- Example: Add the cube root of 27 to the 5\textsuperscript{th} root of 32

  \[27^{1/3} + 32^{0.2}\]
Order of Precedence

- Example: Add the cube root of 27 to the 5th root of 32

$$27^{1/3} + 32^{0.2}$$

ans =

11

Undesired result...press the UP arrow key, then add the parenthesis around the “1/3”

$$27^{(1/3)} + 32^{0.2}$$

ans =

5
Variable Definition – Key Points

```plaintext
>> x = 5
x = 5
>> x = x + 5
x = 10
```

This statement does not make mathematical sense. However in programming language, it changes the value of “x” to a new value based on its previously defined value.
Calculations with Variables

- Variables defined previously can be used to do calculations

- **Define:** \( a = 8, \) \( \) \( \text{my\_var} = 12 \)

\[
\begin{align*}
\text{>> a + my\_var} \\
\text{ans} = 20 \\
\text{>> a*my\_var^2} \\
\text{ans} = 1152
\end{align*}
\]
Rules in Naming Variables

- Variable names cannot exceed 63 characters
- Names **must** begin with a **letter**
- May contain **letters, digits, and the underscore character**
  - No spaces are allowed
- MATLAB **is** case sensitive, it distinguishes between uppercase and lowercase letters
- Avoid naming variables with currently defined MATLAB functions
  - Ex: exp, sin, cos, sqrt, length, mean, max, min etc.
Variable Name Examples

- **Allowed**
  - My_name
  - MyName
  - Myname32

- **Not allowed**
  - 1Name (starts with numeral)
  - My Name (no spaces allowed)
  - My-Name (no special characters allowed)
MATLAB Built-in Math Functions

- Pre-defined in MATLAB ready for use
  - \( \exp(x) \) – exponential \((e^x)\)
  - \( \log(x) \) – natural logarithm \((\log_e(x))\)
  - \( \log_{10}(x) \) – base 10 logarithm \((\log_{10}(x))\)
  - \( \sqrt{x} \) – square root \((\sqrt{x})\)
  - \( \text{abs}(x) \) – absolute value \(|x|\)

- For standard trigonometric functions, the units for angles are **radians**, e.g.
  - \( \sin(x), \cos(x), \text{asin}(x), \text{acos}(x), \tan(x), \cot(x) \)

- There are equivalent functions for **degrees**, e.g.
  - \( \text{sind}(x), \cosd(x), \text{asind}(x), \text{acosd}(x), \text{tand}(x), \cotd(x) \)
MATLAB Built-in Math Functions

- Ex: To calculate $\sin(\pi/2)$
  
  ```matlab
  >> sin(pi/2)  \quad \pi \text{ is defined as } \text{"pi"}
  ans =
  1
  ```

- Ex: To calculate $e^2$

  ```matlab
  >> e^2
  ??? Undefined
  function or variable ‘e’

  >> exp(2)
  ans =
  7.3891
  ```
MATLAB Built-in Math Functions

- Ex: To calculate $\sin(2x)$ where $x = \pi/4$

```matlab
>> x = pi/4
x =
   0.7854

>> sin(2*x)
ans =
   1
```
MATLAB Built-in Math Functions

- Ex: Given the following function, find the value for \( x = 30 \) degrees.

\[
\frac{(\cos 2x)(\sin^2 x) + \tan\left(\frac{x}{2}\right)}{e^{3x}}
\]
MATLAB Built-in Math Functions

- Ex: Given the following function, find the value for \( x = 30 \) degrees.

\[
\frac{(\cos 2x)(\sin^2 x) + \tan \left( \frac{x}{2} \right)}{e^{3x}}
\]

>> x = 30

x =

30

>> x = (cosd(2*x)*sind(x)^2+tand(x/2))/exp(3*x)

ans =

0.0817
MATLAB Built-in Math Functions

- Ex: Given the following function, find the value for x = 30 degrees.

\[ \frac{(\cos 2x)(\sin^2 x) + \tan\left(\frac{x}{2}\right)}{e^{3x}} \]

OR

```matlab
>> x = 30
>> x = x*pi/180 % convert x to radians
x =
30
x =
0.5236

>> (cos(2*x)*sin(x)^2 + tan(x/2))/exp(3*x)
ans =
0.0817```
Saving a Script File

- Script files must be saved after completion
- In our class use **Save As** to your Z:\ or USB drive
  - This should be the same as the working directory you specified upon starting MATLAB
  - **SAVE YOUR WORK AS YOU GO!**
Saving a Script File

- The name of the script file is governed by the following rules:
  - No spaces are allowed
  - The name cannot start with a numeric
  - No special characters are allowed (except underscore)

<table>
<thead>
<tr>
<th>Allowed:</th>
<th>Not Allowed:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prob1a.m</td>
<td>Prob 1a.m (blank space)</td>
</tr>
<tr>
<td>Prob_1a.m</td>
<td>1aProb.m (can’t start with numeric)</td>
</tr>
<tr>
<td></td>
<td>Prob-1a.m (no special characters)</td>
</tr>
</tbody>
</table>
Useful Programming Commands

- To clear the command window, use “clc”
  
  \[ \text{>> clc} \]

- To clear all variables previously defined from memory, use “clear”

- Ex: define \( x = 5 \)
  
  \[ \text{>> x = 5} \quad \text{>> clear} \]

  \[ \begin{array}{l}
  x = \\
  \text{>> x} \\
  5 \\
  \end{array} \]

  \[ \text{??? Undefined function or variable “x”} \]
Useful Programming Commands

- To send a text message to the command window or display contents of a variable, use “disp(…)”

```matlab
>> disp('Brutus Buckeye')   NOTE: single quotes
Brutus Buckeye

>> x = 5
>> disp(x) No quotes for variable
x =
5
```

5
Useful Programming Commands

- To suppress the command window output, use a semicolon after the statement “;”

```matlab
>> x = 5
x =
  5
>> x = 6;
The value of x is changed but not displayed
>> disp(x)
>> x = 6
x =
  6
```
Goal: Create a script file that calculates hypotenuse of triangle based on Pythagoras theorem.

- Pythagoras theorem: \( a^2 + b^2 = c^2 \)
Script File Example

- Steps for creating script
  1. Create a script file and name it Pyth.m
  2. Clear the command window
  3. Clear all variables from memory
  4. Display your name and seat number
  5. Declare height of triangle as 3 units
  6. Declare base of triangle as 4 units
  7. Find the hypotenuse of the triangle
Script File Example

- Print of the script file.

```matlab
7/24/09 9:30 AM C:\Kuldeep_GTA\SU-09\Revisions\Pyth.m

clear % Clears the variables from memory
clc % Clears the command window
disp('Brutus Buckeye, Seat number 1')
disp('Example Problem 1')
height=3; % Note that we use ';' because we
don't want to display these 2 values
base=4; hypotenuse = sqrt((height^2)+(base^2))
```

- Print of the output (command window).

```
6/30/09 8:55 AM MATLAB Command Window 1 of 1

Brutus Buckeye, Seat number 1
Example Problem 1

hypotenuse =
```
Homework Example

- Script file to print

```matlab
clc
disp(‘Student, Joe’)
disp(‘ENGR 1181, Seat 99’)
disp(‘MAT – Introduction’)
disp(‘Problem 4’)
Prob4a = cos(5*pi ... 
...
```

- Command window to print

```
Student, Joe
ENGR 1181, Seat 99
MAT – Introduction

Problem 4
prob4a =
    0.2846
...
```
Important Takeaways

- Programming is a useful and powerful tool for problem solving in any engineering discipline
- MATLAB is a tool to solve these types of problems
- Basics of MATLAB interface
- Variable naming and rules
- Basic math operations and calculations
- Script files and other useful commands
Preview of Next Class

- Array Creation
  - Difference between scalar and vector
  - How to create a row/column vector in MATLAB
  - Examples of how to set constant spacing in vectors
  - Special commands and matrix examples
What’s Next?

- Review today’s Quiz #01
- Open the in-class activity from the EEIC website and we will go through it together.
- Then, start working on MAT-01 homework.
- Prepare for next class by reading about array creation in MATLAB - array creation is a useful way to store data to be analyzed or plotted.