Introduction to MATLAB®

Lecture 1: Variables, expressions and built in commands
What is MATLAB®?

MATLAB®

- Short for Matrix Laboratory
- Matrix data structures (arrays) are at the heart of programming in MATLAB
- We will consider arrays in \( n \)-dimensions where
  
  \[ 1D \iff \text{vectors} \quad \text{and} \quad 2D \iff \text{matrices}, \text{e.t.c.} \]

- In addition MATLAB is also a programming language
  - It is an interpreted language (i.e no need to compile)
  - Commands are executed line by line
Why MATLAB®?

1. Widely used by engineers and scientists worldwide to analyze and design products and systems.
2. Comes with built-in functions for performing a wide variety of mathematical operations, plotting and visualization of data.
3. Has additional packages that can be installed for specific subject areas, e.g., Symbolic Math Toolbox™ for performing calculus on functions, Deep Learning Toolbox™ for exploring AI algorithms.
4. The MATLAB programming environment allows one to customize programs to fit specifications.
Lecture 1 Objectives

- Familiarize yourself with the MATLAB programming environment
  
  *At the most basic level we can think of MATLAB as a very powerful graphing calculator! (with many built-in functions)*

- Learn how to use MATLAB to:
  
  1. execute simple commands.
  2. create variables and display their values
  3. Manipulate matrices and vectors
  4. Use built-in functions and commands
MATLAB WINDOWS

- MATLAB has 8 main windows, the first 4 are on the main screen by default

<table>
<thead>
<tr>
<th>Command window</th>
<th>Enter commands and variables, run programs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command history window</td>
<td>History of commands entered in window</td>
</tr>
<tr>
<td>Workspace window</td>
<td>Information about current variables</td>
</tr>
<tr>
<td>Current directory window</td>
<td>Shows files in the current folder/directory</td>
</tr>
<tr>
<td>Figure window</td>
<td>Output from graphic commands</td>
</tr>
<tr>
<td>Editor window</td>
<td>Create and debug scripts and functions</td>
</tr>
<tr>
<td>Help window</td>
<td>Get help on functions and scripts</td>
</tr>
<tr>
<td>Launch pad window</td>
<td>Access to tools, demos and documentation</td>
</tr>
</tbody>
</table>

Some commands
- `clc` - Clear command window
- `clf` - Clear figure
- `exit` or `quit` - Quit MATLAB
- `disp('text')` - Displays text
Variables and Expressions

Variable - A value that can change depending on conditions or information passed to the program.
In contrast, a program may have data values that are fixed constants.

Expressions - These are created using values, variables previously created, mathematical operators and built in functions e.g.

\[ 2 \cos(1) + \sqrt{2} \]
ans =
2.4948

Adding a semi-colon (;) at the end of the expression suppresses screen output
Expressions and formatting

- The default in MATLAB is to display numbers that have decimal points with 4 decimal places.
- There are many other options
  1. **format** `long` will result in 15 decimal places
     ```matlab
     >> format long
     >> 2*cos(1) + sqrt(2)
     ans =
     2.494818174109374
     ```
  2. **format** `shortEng` will result in 4 decimal places engineering format.
     ```matlab
     >> format shortEng
     >> 2*cos(1) + sqrt(2)
     ans =
     2.4948e+000
     ```
### Basic formats displaying $\pi$

<table>
<thead>
<tr>
<th>Format Style</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>format short</strong></td>
<td><em>default</em></td>
</tr>
<tr>
<td><code>format long</code></td>
<td>3.141592653589793</td>
</tr>
<tr>
<td><code>format rat</code></td>
<td>355/113</td>
</tr>
<tr>
<td><code>format bank</code></td>
<td>3.14</td>
</tr>
<tr>
<td><code>format short e</code></td>
<td>3.1416e+00</td>
</tr>
<tr>
<td><code>format long e</code></td>
<td>3.141592653589793e+00</td>
</tr>
<tr>
<td><code>format compact</code></td>
<td><em>(no blank lines)</em></td>
</tr>
</tbody>
</table>
Creating Variables

We create variables in MATLAB using an assignment statement

\[
>> \text{variablename} = \text{expression}
\]

**REMARKS**

1. The ‘\(=\)’ is an assignment operator (unlike the mathematics equal sign, does not mean equality)
2. The *expression* is evaluated at the result is stored in *variablename*

*Use variable names that make sense, e.g. if a variable to store the radius the name *radius* makes sense, \(y\) does not!*
Rules for variable names

1. Must begin with a letter of the alphabet, after that it may contain letters, digits, and the underscore but NO SPACES! e.g.
   interest_rate, my_num_1, e.t.c.

2. The limit to the number of characters in a variable name is 63 stored in namelengthmax.

3. MATLAB is case sensitive so INTEREST_RATE is different from Interest_rate.

4. Certain words are reserved so they should not be used as variable names.

5. Names of built in functions cannot be used as variable names.
Some reserved variable names

- You should not use these variables in your code
  - **ans**: The result of a previous calculation
  - **computer**: The type of computer you are on
  - **eps**: The smallest positive number $\epsilon$ that satisfies $1 + \epsilon > 1$
  - **i, j**: The imaginary unit ($\sqrt{-1}$)
  - **realmax, realmin**: The largest and smallest real numbers that can be represented on this machine.
Commands related to Variables

1. **who** shows variables that have been defined
2. **whos** shows more information about the type of variables
3. **clear** clears out all variables. You can also specify a specific variable to clear.
4. **save** saves all your variables in a file `matlab.mat`
Mathematical Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>multiplication</td>
</tr>
<tr>
<td>+</td>
<td>addition</td>
</tr>
<tr>
<td>-</td>
<td>negation, subtraction</td>
</tr>
<tr>
<td>/</td>
<td>division (division by, e.g. 10/5 = 2)</td>
</tr>
<tr>
<td>\</td>
<td>division (division into, e.g., 5 \ 10 = 2)</td>
</tr>
<tr>
<td>^</td>
<td>exponentiation (e.g., 5^2 = 25)</td>
</tr>
</tbody>
</table>

Operator precedence rules

- MATLAB follows mathematical rules for order of operations:

  \[
  \{ () \} \rightarrow \{ ^ \} \rightarrow \{ *, /, \} \rightarrow \{ +, - \}
  \]

- You can use parenthesis to change the precedence of an expression.
### MATLAB built-in functions and constants

- **Some basic functions and constants:**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sqrt(x)</code></td>
<td>Square root</td>
</tr>
<tr>
<td><code>exp(x)</code></td>
<td>Exponential ( e^x )</td>
</tr>
<tr>
<td><code>abs(x)</code></td>
<td>Absolute value</td>
</tr>
<tr>
<td><code>log(x)</code></td>
<td>Natural logarithm</td>
</tr>
<tr>
<td><code>sin(x)</code></td>
<td>Sine of ( x )</td>
</tr>
<tr>
<td><code>cos(x)</code></td>
<td>Cosine of ( x )</td>
</tr>
<tr>
<td><code>tan(x)</code></td>
<td>Tangent of ( x )</td>
</tr>
<tr>
<td><code>π</code></td>
<td>( \pi )</td>
</tr>
<tr>
<td><code>i</code></td>
<td>( \sqrt{-1} )</td>
</tr>
<tr>
<td><code>inf</code></td>
<td>Infinity ( \infty )</td>
</tr>
<tr>
<td><code>NaN</code></td>
<td>&quot;not a number&quot;, such as the result of 0/0.</td>
</tr>
</tbody>
</table>


Help and lookfor commands

There are many built-in functions in MATLAB:

- Use `lookfor` can be used to find a specific command to perform a task.
- Use `help` to find out information about a known command.

```
>> help elfun
```

lists all elementary functions in MATLAB.
diary function

- You can keep a record of your MATLAB sessions using the `diary` function.
- Calling

  ```matlab
  >> diary filename
  ```

  will save the commands you enter into the command window along with the output in a plain text file.
- `diary off` and `diary on` will pause and restart the recording, resp.
MATLAB script files a.k.a m-files

- MATLAB script files (also known as m-files) are useful for running commands multiple types without having to type them into the command line.
- Script files are named with an extension .m
- You can type in your commands into the m-file, edit, save them and execute them as many times as you want.
- The MATLAB editor is the best way to create m-files.
- A typical script file will contain MATLAB commands and comments
- Comments describe the purpose of the script file and any useful information about the commands executed
- Comments within m-files begin with (%)
Exercise

Write a MATLAB script that assigns the variables

\[ x = 4, \quad y = 3, \quad z = \frac{1}{8} \]

and calculates the following expressions:

1. \( \frac{5(y - x)}{4z - 19} \)
2. \( e^{\frac{z+y}{x}} + 6^{\sqrt{x}} \)
3. \( 2 \sin(x) \sec(y) \)