Lesson 1: Introduction to MATLAB

Learning Outcomes:

1. Become familiar with the MATLAB system, including key terms
2. Know where to find help (e.g., Documentation)
3. Use MATLAB as a scientific calculator
4. Know and use the most important MATLAB command
5. Generate random numbers
6. Describe rules of MATLAB variables
7. Execute expressions and assign to variables
8. Identify data types (e.g., double, strings)
9. Identify several programming tips and pitfalls

Book Sections: 1.1 - 1.5

1. MATLAB System

MATLAB consists of five components:

1. Desktop (Command Window, History, Workspace, Current Folder)
2. Mathematical Function Library
3. The Language
4. Graphics Package
5. External Interface/Libraries

1.2 Key Terms

prompt

scalar

strings

types

variable

help

double

single

casting

seed

incrementing

ellipsis

2. Help and Documentation

There are several ways to get help in MATLAB:

- Help by using Desktop Menus
- Help at the Command prompt, for example,

```
doc
help help
help sin
lookfor for
```

- Search Documentation in Toolbar
- Robust Community
- Web: Google is your friend
- Online documentation: https://www.mathworks.com/help/
- Watch demo

```
demo
```

3. Comments

A **comment** is an annotation or explanation in the source code that is not executed by the complier. It is by far one of the most important coding elements for making a program readable and easy to undertand. All programming languages have them. In particular, # in Python, R, SQL, and // in Java. MATLAB uses the percent symbol (%) to indicate a comment. *Use generously*.

```
% This is a comment.
% Here is another.
% Here is a third.
% Their purpose is for human readability and understanding.
% They are ignored by interpreters and compliers.
```

For example, here is a header section in a simulation program I wrote for an original research project.

```
% ----------------------------------------------------------------------- %
% Project name:          3D Reservoir Simulation
% Purpose:               Model subsurface fluid discharge
% Author:                James Quinlan
% Modified:              2018-06-09
% MATLAB Version:        9.4.0.885841 (R2018a) Update 3
% Sources:               http://www.coatsengineering.com/spe10.htm
% Workflow:
%         1. Load Permeability data (obtain dimensions and scale as needed)
%         2. Set input parameters (config file)
%         3. Define output results structure *** (needs revisited)
%         4. Solve global fine scale equation for comparison (rel. errors)
%         5. Solve coarse scale problem (per method)
%            a). Solve local (on blocks surround face) pressure
%            b). Average local pressure values to use in upscaling
%            c). Upscale transmissibilities by solving optimization
% ----------------------------------------------------------------------- %
```

4. Variables

Variables store values in memory that will be used or modified by the program later. Variable names must follow the rules below.

Naming Rules

- must start with a letter; after that can contain any letters, numbers, or the underscore character.
- can be very long (use many characters)
- are case sensitive (e.g., Pressure, pressure, preSsuRe, and preSSurE are all different)

Naming Conventions

- use mnemonic names (make sense in the context, e.g., pressure, volume, windSpeed)
- make readable by using (i) underscores (e.g., wind_speed) or (ii) camelCase (e.g., windSpeed)

Assignment

The easiest way to declare a variable is by using the **assignment operator**, the equal sign (=). For example,

```
my_variable = 5.3

x=5;
y=10;
z=12;

area=5;
```

```
Volume=10;
volumE=12;

who  % display variables

clearvars x       % clear variable x
who

clearvars y z
who

clearvars
who
```

Variable Types

MATLAB has the following basic data types:

- Integers (int8, int16, ...)
- Single precision
- Double precision (**default**)
- Logical
- Charcter (chr) or string

```
x=5;
whos % display variables, size, and type


% 1.7 Type casting
x=single(5)
y=uint8(5)
z=logical(1)             % True or False
whos

% Characters ('x') and Strings ("x")
Achar='hello world'      % Use single quotes
Astring="hello world"    % Use Double quotes
x='5';                   % Different than x=5;
X=5;
```

```
whos
```

5. Scientific Calculator

At the most basic level, MATLAB is a scientific calculator capible of performing simple scalar operations such as addition, subtraction, multiplication, division, and exponentiation. In particular,

```
2+3     % Addition
2-3     % Subtration
2*3     % Multiplication
2/3     % (Right) Division
2\3     % (Left) Division
2^3     % Exponentiation
```

Opertations can be combined to obtain more complex calculations. Parenthesis are used for groupings. The standard order of operations PEMDAS is followed.

```
5*(8-19)^3
5*4-2
5*(4-2)
19/3*4\(5-2)^11
```

5.1 Scientific functions

```
exp(1);         % Exponential function
log(10)         % Natural log
log2(10)        % Log base 2
log10(10)       % Log base 10
sin(10)         % sine function
abs(-10)        % Absolute value
floor(pi)       % Greatest Integer
ceil(pi)        % Least Integer
nthroot(8,3)    % Nth root
```

5.2 Matematical Constants

```
pi
```

```
i       % Square root of -1
j       % Square root of -1
Inf     % Infinity
NaN     % Not a number
```

5.3 Format output

```
help format

format short
format long
format bank

% Apply various formats to pi
pi
```

6. Random Number Generator

Random numbers are useful in simulations. Often data is not available so code can be tested using random data instead. Computers use a random, actually **pseudorandom**, number generators, which are **not** truly random at all, but give the appearence of randomness.

```
% Set the seed of the random number generator to default
rng('default');

% Generate Random Number
r=rand()    % Uniform
s=randn()   % Normal
t=randi(6)  % Random Integer btw 1 and 6

rng(5)  % Set seed
rand

rng(5)
rand
% Notice you get the same random numbers

% Random between a and b
```

```
a=3
b=21
r=a+(b-a)*rand

% Random integer between a and b
r=randi([a,b])


% See documentation for further details
help rng
help rand
help randi
help randn
```

7. Programming Tips

1. **Avoid Technical Debt:** Technical debt refers to ***deferred work***, typically based on a trade-off between "do it quick" and "do it right". Often, "do it quick (and dirty)" is choosen with intention of fixing errors later. However, next iteration, some other deadline approaches and the quick solution is again selected. DIRTFT = Do It Right The First Time
2. Use comments liberally
3. Use mnemonics for variable names

8. Exercises

1. Calculate $7 \times 3.1 + \frac{\sqrt{120}}{5} - 15^{5/3}$.
2. Calculate $\frac{3.8^2}{2.75 - 4.1 \times 25} + \frac{5.2 + 1.8^5}{\sqrt{3.5}}$
3. Gosper's Approximation for factorials is given by: $n! = \sqrt{(2n + \frac{1}{3})\pi}\, n^n e^{-n}$. Estimate $19!$
4. Calculate $\sin(4\pi) + \frac{2}{3}\sqrt[7]{591} - \ln(14)$

Solutions:

```
% 1.
7*3.1+(sqrt(120)/5)-15^(5/3)

% 2.
```

```
((3.8^2)/(2.75-41*25))+((5.2+(1.8^5)))/(sqrt(3.5))

% 3.
n=19;
Gosper=sqrt((2*n+1/3)*pi)*(n^n)*(exp(-n))
```

Resources

MATLAB Documentation: https://www.mathworks.com/help/matlab/