L2. Basics

Approach

Preview key concepts by first playing with Matlab as a calculator.

From formula to program.

Three Formulas

Surface Area Increase

Surface Area of a Sphere

Half-Angle

Quadratic Equation

Cosine(15 degrees)

X^2 + 5x + 6 = (x+2)(x+3)
Let’s Revisit the Key Ideas Above and Introduce Others...

A Script

% Quad1
% Solves \( x^2 + 5x + 6 = 0 \)

\[
\begin{align*}
a & = 1; \\
b & = 5; \\
c & = 6; \\
d & = \sqrt{(b^2 - 4*a*c)}; \\
r1 & = (-b - d)/(2*a) \\
r2 & = (-b + d)/(2*a)
\end{align*}
\]

Script

A sequence of instructions.

The order of the instructions is important.

A script is a program.

Comments

% Quad1
% Solves \( x^2 + 5x + 6 = 0 \)

\[
\begin{align*}
a & = 1; \\
b & = 5; \\
c & = 6; \\
d & = \sqrt{(b^2 - 4*a*c)}; \\
r1 & = (-b - d)/(2*a) \\
r2 & = (-b + d)/(2*a)
\end{align*}
\]

Comments and Readability

Start each program (script) with a concise description of what it does.

Define each important variable/constant.

Top a block of code for a specific task with a concise comment.

Comments

Begin with the "%" symbol. Goes to the end of the line.

Facilitate the reading and understanding of the script.
Arithmetic Expressions

% Quad1
% Solves  \( x^2 + 5x + 6 = 0 \)

\[
\begin{align*}
a &= 1; \\
b &= 5; \\
c &= 6; \\
d &= \sqrt{b^2 - 4*a*c}; \\
r1 &= \frac{-b - d}{2*a} \\
r2 &= \frac{-b + d}{2*a}
\end{align*}
\]

Arithmetic Expression

A recipe that results in the production of a number.

Built-In Functions

% Quad1
% Solves  \( x^2 + 5x + 6 = 0 \)

\[
\begin{align*}
a &= 1; \\
b &= 5; \\
c &= 6; \\
d &= \sqrt{b^2 - 4*a*c}; \\
r1 &= \frac{-b - d}{2*a} \\
r2 &= \frac{-b + d}{2*a}
\end{align*}
\]

Built-In Functions

These are "packagings" of more advanced calculations.

Some examples: \( \log, \exp, \sin, \cos, \ldots \)

Variables

% Quad1
% Solves  \( x^2 + 5x + 6 = 0 \)

\[
\begin{align*}
a &= 1; \\
b &= 5; \\
c &= 6; \\
d &= \sqrt{b^2 - 4*a*c}; \\
r1 &= \frac{-b - d}{2*a} \\
r2 &= \frac{-b + d}{2*a}
\end{align*}
\]

Variables

A variable is a "box" that holds a numerical value.

It has a name.

The name must begin with a letter.

Upper and lower cases are distinguished. Can use all letters and numbers and the underscore character.

Example: \( x1A_\text{New} \)
Assignment Statements

% Quad1
% Solves \( x^2 + 5x + 6 = 0 \)

\[
\begin{align*}
  a &= 1; \\
  b &= 5; \\
  c &= 6; \\
  d &= \sqrt{b^2 - 4ac}; \\
  r1 &= \frac{-b - d}{2a} \\
  r2 &= \frac{-b + d}{2a}
\end{align*}
\]
**Script Execution**

\[
\begin{align*}
  a &= 1; \\
  b &= 5; \\
  c &= 6; \\
  d &= \sqrt{b^2 - 4ac}; \\
  r1 &= (-b - d)/(2a) \\
  r2 &= (-b + d)/(2a)
\end{align*}
\]

**Remember...**

Instructions are executed in order.

In assignment statements, the right hand side is evaluated first and then the value is assigned to the variable named on the left hand side.

The variables on the right hand side must have values before they can be used in an expression.

**Question Time**

What is the value of \( X \) and \( Y \) after the following script is executed:

\[
\begin{align*}
  X &= 2; \\
  Y &= 7X; \\
  X &= Y; \\
  X &= X + 1;
\end{align*}
\]

A. \( X \) is 5 and \( Y \) is 14  
B. \( X \) is 15 and \( Y \) is 14  
C. \( X \) is 5 and \( Y \) is 21  
D. \( X \) is 15 and \( Y \) is 2

What is the final value of \( X \) and \( Y \)?

\[
\begin{align*}
  > X &= 8; \\
  > Y &= X; \\
  > X &= Y; \\
  > X &= 2X; \\
  > Y &= Y/2;
\end{align*}
\]

A. \( X \) is 16 and \( Y \) is 16  
B. \( X \) is 8 and \( Y \) is 8  
C. \( X \) is 16 and \( Y \) is 4  
D. \( X \) is 8 and \( Y \) is 4
Another Script

```matlab
% Quad2
% Solves ax^2 + bx + c = 0
% Assumes real roots.
a = input('Enter a: ');
b = input('Enter b: ');
c = input('Enter c: ');
d = sqrt(b^2 - 4*a*c);
r1 = (-b - d)/(2*a)
r2 = (-b + d)/(2*a)
```

The input Command

```
Variable Name = input(' Message');
```

The input Command

```
where to put it
```

Processed after the user hits the <enter> key.

Formatting Output

```
When leaving off the semicolon isn't good enough.
The tools: disp, fprintf
```

```
disp
```
Displays a string.
Example:
```
disp('This is a message')
```

```
fprintf
```
Used to format output. Example:
```
X = 1.23456789;
fprintf('x = %5.2f
',x))
```

Output line will look like
```
x =  1.23
```

The

```
The \n generates a carriage return
```

A Modification...

```
r1 = (-b - d)/(2*a);
r2 = (-b + d)/(2*a);
disp(' ')
fprintf('Root1 = %10.6f
',r1))
fprintf('Root2 = %10.6f',r2))
```