

Matlab

By the end of this lesson, you should be able to:

- *Create Code*
- *Process Data*
- *Produce plots*
- *Write nice Functions to solve Bioengineering problems*
- *Understand the concepts of functions, loops, conditional statements, and code segmentation*

1

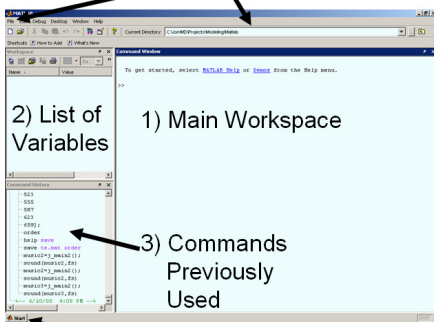
Outline

- User Interface
- Variables and Vectors
- If Statements and Loops
- Scripts and Functions
- Plotting and Homogeneous Transformation

2

User Interface

File Info and Directory Taskbar



2) List of Variables

1) Main Workspace

3) Commands Previously Used

Start Button

3

File Extensions

- *.m "M" file – stores code
- *.mat stores data

4

M files

- M files store lists of commands
- That way, you don't have to retype them over and over
- Always use an M file
- (Unless you're using Matlab as an expensive calculator)

5

Variables

- Data is stored in variables

Height = 200;

Diameter = 10;

Volume = Height*pi/4*Diameter^2

Matlab is case-sensitive. Make sure you are consistent with upper-lower case.

Usually put a semicolon at the end of the line

Don't use a semi colon if you want to see the answer

See tutorial 1.m


New Terms

=
;
num2str
disp
pi
pause
clc
help

6

Vectors and Matrices

- Use `[]` to create a matrix
- Use `[]` to append a matrix
- Use `()` to access a matrix
- `matrix(Row,Column)`

• $J =$ 

| | | | |
|---|---|---|---|
| A | B | C | D |
| E | F | G | H |
| I | J | K | L |

• $J(2,3) = G$

• Transpose: J'

• Matrix math:

$\mathbf{a \cdot b}$

• Multiply individual elements: $\mathbf{a \cdot b}$

See tutorial_2.m

New Terms

`[]`
`()`
`[x; y]`
 A'
length
size
rand
zeros
ones

7

Conditional Statements

```
if volume < 1
    disp('Volume is too small');
elseif volume < 2
    density = mass/volume;
    torque = q.*density.^2;
elseif volume == 2.1
    disp('Singularity exists')
else
    disp('Volume is too large');
end
```

See tutorial_3.m

New Terms

`<`, `<=`, `~=`
`==` equals?
`&&` and
`||` or
`if`
`elseif`
`end`
`mean`
`min`
`max`
`sort`
`Find`
`round`

8

Conditional Statement rules

- 1) No semi colon after the if statement
- 2) Use `==` otherwise you make it equal!
(Be very careful; this is often done)
- 3) Always `end` an if statement
- 4) Indent any consequence of if statement

9

Loops

```
for time = 1:n
    pos = x(time)
    vel = (pos(time) - pos(time-1))/1;
end
```

New Terms

```
1:10
q(end)
q(i-1)
mod
for
```

See tutorial_4.m

10

Scripts and Functions

- Scripts: m files that you frequently call.
- They are identical to inserting the text in the file
- Functions:
 - They are given variables
 - They return variables
 - They don't show anything about what they're doing inside

11

Script Example

```
for i = 1:5
    figure(i)
    set_up_figure; % This is the script
    plot(data)
end
```

(separate file named set_up_figure.m)

```
axis equal;
axis manual;
```

12

Function Example

```
for i = 1:3
    mass(i) = calc_mass(density, volume);
end
```

Either in the same file, or a separate file
named "calc_mass.m"
function m = calc_mass (v1,v2)
 m = v1*v2;

New Terms

dot
inv
sqrt

See tutorial_5.m

13

Why Should I Use Functions

- They save space (No need to repeat code)
- If you fix a mistake somewhere, you don't have to try to remember everywhere you made that same mistake
- They segment code into check-able portions (Very useful for debugging!)

14

How Many Functions should I make?

- Too many functions: you are spending all your code transferring variables
- Too few functions: they aren't useable for different applications. They aren't debuggable if they don't work.
- Where do you draw the line? Experience.

15

Structure (Indents and Comments)

- Always indent for:
 - Loops
 - If Statements
- Use Lots of comments % This is a comment
 - That way you'll remember what you were trying to do, 5 years from now.

16

Data Retrieval

- Loading and Saving Data

New Terms

clear
save
load

See tutorial_6.m

17

Plotting

```
x = 1:10; % Creates a vector  
y = 2*(1:10);
```

```
plot(x,y);  
help(plot)  
plot(x,y,'bo*');  
xlabel('X axis')  
ylabel('Y axis')  
title('This is a plot')
```

New Terms

sin, cos
exp
plot
close
subplot
title
xlabel, ylabel
axis equal
axis manual
hold on
legend
set

See tutorial_7.m

18

Homogeneous Transformation

- Useful for changing reference frames
- Also useful for graphics

$$H = \begin{bmatrix} \cos \theta & -\sin \theta & \Delta x \\ \sin \theta & \cos \theta & \Delta y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{matrix} x_1 & x_2 & \dots & x_n \\ data = y_1 & y_2 & \dots & y_n \\ 1 & 1 & \dots & 1 \end{matrix}$$

$$New_data = H * data$$

New Terms

See tutorial_8.m

gca

19

Copying

- Don't copy code:
 - From each other
 - From the internet
- I will use MOSS to check for copying
- Copied Code results in:
 - 0 for the assignment
 - 1 letter grade decrease in final grade
- Recommendations:
 - Talk before you write the code
 - Write the code by yourself. Not in groups.
 - If you have a specific error, ask a friend for help. Do not have him or her guide you through the entire project, or your code will end up being copied, even though you didn't mean to.

20
