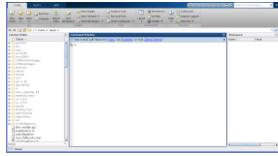


# CISC 3250

## Systems Neuroscience

### Matlab



Professor Daniel Leeds  
 dleeds@fordham.edu  
 JMH 328A

## Access to Matlab

Laptop/home computer:

- Mathworks link on our course website
- Student license for \$49

Lab computer:

- Open terminal
- Type: matlab

2

## Commands

Symbols and keywords cause actions

- `b=2` *creates variable b with value 2*
- `d=b+5` *creates variable d with value computed by adding 5 to value of b*
- `exit` *closes program*

3

## Variables

**Variables store information**

- Letters  
`neuronType='purkinje';`
- Single Number  
`numberOfDendrites=1000;`
- Group of numbers, in `[]` brackets  
`potentials=[-65 -64 -63.9 -62.8 -61.6];`

4

## Variable names

- A variable name is any valid identifier
  - Starts with a letter, contains letters, digits, and underscores (`_`) only
  - Cannot begin with a digit
  - Case sensitive:  
`username#userName#UserName`

5

## Data

Data can be read from files

- `load('classExample.mat');`
- `save('classExample2.mat','c','d');`

Data is analyzed through commands

- `numberOfSpikes=sum(spike_record);`
- `plot(spike_record)`

6

## Saving results

Save graphics with:

- `print -dpng filename.png`
- or
- `print -djpg filename.jpg`

7

## Standard arithmetic

Operators

- Addition:  $5 + 2$  evaluates to 7
- Subtraction:  $5 - 2$  evaluates to 3
- Multiplication:  $5 * 2$  evaluates to 10
- Division:  $4 / 2$  evaluates to 2
- Exponent:  $5 ^ 2$  evaluates to 25

8

## Vector arithmetic

- Vector is list of numbers in between [ ]
- Can replace one of operands with a vector

`2+[3 4 1]` yields `[5 6 3]`

- Can place results into new variable

`Variable_Name=number*vector;`

- Both operands can be vectors, but special rules apply

9

## The semi-colon ;

- Each command can optionally end with ;
- semi-colon prevents result display

10

## Counting in Matlab

`a:b` creates a vector `[a a+1 ... b-1 b]`

- `3:6` -> `[3 4 5 6]`

`a:k:b` creates a vector `[a a+k a+2k ... b]`

- `3:4:15` -> `[3 7 11 15]`

11

## Accessing vector elements

`a=[2.2 1.4 -5 3.5 -7.8];`

- `name` accesses full vector

`a`

- `name(index)` accesses single element

`a(4)` returns `3.5`

- `name(index1:index2)` accesses set of elements

`a(2:4)` returns `[1.4 -5 3.5]`

- `name(end)` accesses final element

12

## Matrices: rows and columns

```
B=[2.2 1.4; -5 3.5; -7.8 4.3];
```

- Spaces/commas separate columns  $\begin{bmatrix} 2.2 & 1.4 \\ -5 & 3.5 \\ -7.8 & 4.3 \end{bmatrix}$
- Semi-colons (;) separate rows
- name (row, col) accesses single element  
B(2,1) returns -5
- name (:, col) accesses all elements in column  
B(:,2) returns [1.4; 3.5; 4.3]
- name (:) vector of all elements in name

13

## Functions

```
c=[0 3 -2 4];
```

Data are analyzed through functions

```
function_name(input_variable)
```

- sum(c) -> 5
- min(c) ->
- max(c) ->
- plot(spike\_record)

14

## Finding desired values

```
find(vector<number> find(c<2)
Return indices in vector that are less than number
```

### Comparisons

- d<2, d>2 strict inequality
- d<=2, d>=2 semi-inequality
- d==2 equality

### Logic combinations

- d>5 & d<8 the AND operation
- d<5 | d>8 the OR operation

15