

# Matlab: The Basics

Shelley L. Knuth  
Research Computing  
CU-Boulder

# Outline

- What is Matlab? Who Uses it?
- Basic Matlab
  - Syntax
  - Variables
  - Reading in Data
  - Plotting
- Future: Running Matlab on Janus
  - Serial Applications
  - Running in Parallel
  - Interactively
- Matlab/Python throw down

# What Is Matlab?

- A high performance tool for technical computing
  - Integrates computation, visualization, and programming
  - Analyze data, develop algorithms, create applications  
[www.mathworks.com](http://www.mathworks.com)
- Uses many specialized toolboxes to make our lives easier
- Used by many in the sciences, engineering, math
  - Academia and industry

# Key Features

- Really easy to use
- Lots of help available online
- Can easily read in multiple data formats
- Good parallel computing capabilities
- Faster than C or Fortran to generate crude programs and run them to test quick results
- Drawbacks: Cost/Licensing
  - Fortunately CU has a license for those at the university

# Matlab Desktop

The screenshot shows the Matlab desktop interface with several open windows:

- Current Folder:** Shows the current directory path: Users/knuths/Documents/MATLAB.
- Command Window:** Displays the following command history:

```
1   2   3   4
5   6   7   8
9  10  11  12

>> y=[5 6 7 8]
y =
    5   6   7   8
>> transpose(y)
ans =
    5
    6
    7
    8

>> z=90*567
z =
    51030

>> z*y
ans =
    255150    306180    357210    408240
```
- Workspace:** Shows variables and their values:

Name	Value	Min	Max
ans	[255150,306180,...	255...	408
x	3x4 double	1	12
y	[5,6,7,8]	5	8
z	51030	51030	510
- Command History:** Shows the history of commands entered in the Command Window.

Annotations in red text and arrows highlight specific features:

- Current Directory** (highlighted in red) points to the Current Folder pane.
- Command Window - Insert commands** (highlighted in red) points to the Command Window pane.
- Workspace - View variables** (highlighted in red) points to the Workspace pane.
- Workspace - Double click** (highlighted in red) points to the Workspace pane.
- Command History - Previously entered commands** (highlighted in red) points to the Command History pane.
- Command History - Can save** (highlighted in red) points to the Command History pane.

# Common Commands

- **matlab** – open matlab
- **exit**- exit matlab
- **Clear** – clears all variables (can also specify)
- **Load** – loads a binary (.mat) file into workspace
- **Help [func]** – get help on a particular function
- **Length** – length of a vector
- **Size** – array dimensions

# Common Syntax

- % - comment
- “ ” - encloses strings
- [] – encloses a matrix
- ; - suppresses output
- Indices begin from 1

# Operators

- Equal to: ==
- Not equal: ~=
- Less than: <
- Greater than: >
- Less than or equal to: <=
- And: &
- Or: ||

# Matlab Variables

- Do not need to specify your variable type, like in Fortran, etc.
- For example, DO NOT need to say:  
**int x;**  
**float y;**  
**double z;**
- You can just simply define your variable  
`x=1;`

# Vectors and Matrices

- Vector

```
>> x=[1 2 3 4]
```

x =

1	2	3	4
---	---	---	---

OR:

```
>> x=1:4
```

x =

1	2	3	4
---	---	---	---

- Matrix

```
>> x=[1 2 3 4;5 6 7 8;9 10 11  
12]
```

x =

1	2	3	4
5	6	7	8
9	10	11	12

# Transpose

- Transpose

```
>> x'
```

```
ans =
```

```
1
```

```
2
```

```
3
```

```
4
```

- Transpose

```
>> transpose(x)
```

```
ans =
```

```
1
```

```
2
```

```
3
```

```
4
```

# Concatenation

```
>> x=[1 2]
```

x =

1 2

```
>> y=[3 4]
```

y =

3 4

```
>> z=[x y]
```

z =

1 2 3 4

```
>> a=[x;y]
```

a =

1 2

3 4

# Matrix Operations

Given:

```
>> x=[1 2 3;4 5 6]
```

x =

1	2	3
---	---	---

4	5	6
---	---	---

```
>> y=[10 11 12; 20 21 22]
```

y =

10	11	12
----	----	----

20	21	22
----	----	----

Addition:

```
>> A=x+y
```

A =

11	13	15
----	----	----

24	26	28
----	----	----

Subtraction:

```
>> B=x-y
```

B =

-9	-9	-9
----	----	----

-16	-16	-16
-----	-----	-----

# Matrix Operations – Element by Element

Given:

```
>> x=[1 2 3;4 5 6]
```

x =

1	2	3
---	---	---

4	5	6
---	---	---

```
>> y=[10 11 12; 20 21 22]
```

y =

10	11	12
----	----	----

20	21	22
----	----	----

Multiplication:

```
>> C=x.*y
```

C =

10	22	36
----	----	----

80	105	132
----	-----	-----

Division:

```
>> D=x./y
```

D =

0.1	0.18	0.25
-----	------	------

0.2	0.24	0.27
-----	------	------

# If statements

- If statement example

```
if (x==1)
```

...

```
elseif (x>3)
```

...

```
else
```

...

```
end
```

- If statement example

```
if (x==1 || x>=8)
```

...

```
elseif ((x-1)<17)
```

...

```
else
```

...

```
end
```

# Loops

## For loops

```
for i=1:12
```

...

```
end
```

```
for i=1:0.5:5
```

...

```
end
```

## While loops

```
while ((x>30) & (b==40))
```

...

```
end
```

# .m files

The screenshot shows the MATLAB Editor window with two tabs open: "matlab\_introTutorial.m" and "netCDF\_matlabTutorial.m". The "netCDF\_matlabTutorial.m" tab is active, displaying the following MATLAB script:

```
1 % Matlab and NetCDF
2 % Written by: Shelley Knuth
3 % Affiliation: Research Computing, CU-Boulder
4 % Date: 28 February 2014
5 % Purpose: The purpose of this program is to work with NetCDF data in
6 % Matlab, including both reading and writing.
7 % The file we will use is sample_data.nc, which was a sample of data
8 % downloaded from Unisys that contains model precipitation data at various
9 % levels.
10
11 %% Inputs
12
13 % Set a variable defining the location of our test data
14 source='/Users/knuths/Documents/MATLAB/test_data/sample_data.nc';
15
16
17 %% Get info about NetCDF data
18 %% Display basic information on the netCDF file
19
20 % Includes information on the structure, number of
21 % dimensions and variables, etc.
22 finfo=ncinfo(source);
23 disp(finfo)
24
25 %% Dimensions
26
27 % What are the names of the dimensions?
28 dimNames={finfo.Dimensions.Name}
29 % This result shows that there are five dimensions: lat, lon, bnds, plev,
30 % and time.
31
32
33 % To find more information on each dimension, type:
```

The status bar at the bottom right indicates "script" and "Ln 27 Col 44".

# Matlab – Read in Data and Generate a simple plot

- Sample program to read in data and make some simple plots
- Read in txt, csv, xls data
- Make a line graph, histogram, and bar graph
- `matlab_intro_tutorial.m`

# NetCDF

- Network Common Data Form (netCDF)
- Allows you to create, access, and share array-oriented data
- Self-describing
  - Includes information defining the data it contains
- Portable
  - Accessed in many ways and forms
  - Can convert to netCDF from many formats

# NetCDF

- Network Common Data Form (netCDF)
- It is an interface to a library of data access functions for storing and retrieving data in the form of arrays
- Allows you to create, access, and share array-oriented data
- Self-describing
  - Includes information defining the data it contains
- Portable
  - Accessed in many ways and forms
  - Can convert to netCDF from many formats
- [www.unidata.ucar.edu](http://www.unidata.ucar.edu)

# Matlab and NetCDF

- Read in netCDF data and attributes
- `netCDF_matlab_tutorial.m`
- Display netCDF data
  - Matlab has no supported functions or toolboxes
  - Several options – MexEPS, nctoolbox