Basics of Matlab Programming

Dr.Geoffrey Andima (Ph.D)
Busitema University
gandima.sci@busitema.ac.ug

MATLAB:- MATrix LABoratory

- Developed by Mathworks, Inc. http://www.mathworks.com
- It is an interactive, integrated, environment
- for numerical/symbolic, scientific computations and other Apps
- shorter program development and debugging time than other languages such as FORTRAN and C
- slow compared to FORTRAN or C
- easy to use
- automatic memory management; no need to declare arrays
- etc

Getting Started with MATLAB

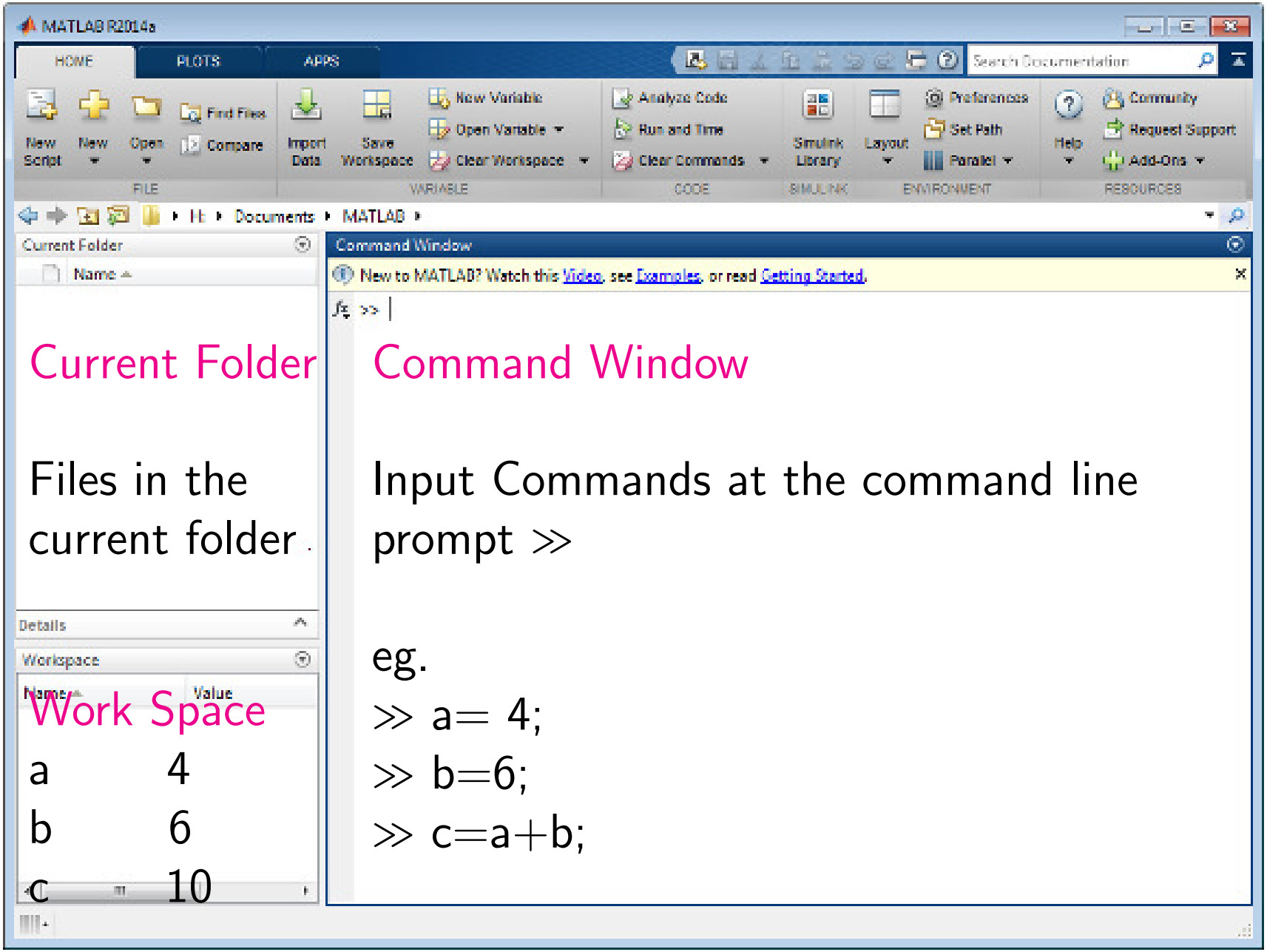
- Windows

  double click MATLAB icon

- Linux Cluster

  cd to bin folder in the installation dir
  then ./matlab

- Either case it opens a MATLAB window with >> prompt
- To stop, type quit or exit at the command prompt >>



Current Folder

Files in the current folder .

Command Window

Input Commands at the command line prompt ≫

eg.

≫ a= 4;

≫ b=6;

≫ c=a+b;

Work Space

Variable and File Names

- case sensitive, e.g., fname and fName are different names
- can be a mix of letters, digits, and underscores (e.g., vector_A)
- variable namws always start with a letter e.g., xy or x2y
- maximum of 63 characters
- reserved characters: % = + - ; : ' [ ] ( ) , # $ & ^ ! ~ can NOT be used
- No space in the names e.g class marks not allowed

Uses of the characters % = ; ,

- % for adding comments, %% creates a shell in mfile
- = to asign a value to a variable e.g c=10
- ;
  (i) suppresses output in the command window
  (ii) delimits commands in the same line e.g clear;close all; a=4;b=5;
  (iii) seperate rows e.g. d=[2;3;4];
- ,
  (i) seperate columns e = [4,2,3,1];
  (ii) delimits commands in the same line e.g k = 4,m=12

Uses of the characters [] () :

- []
  - (i) for arrays e.g p =[12,3]
  - (ii) delete contents of arrays e.g x = [] deletes contents of x
- ()
  - (i) matrix indexing e.g t =[2,3,4], s=t(1,2)
  - (ii) inputs of functions e.g z = mean(t)
- :
  - (i) creating a row vector eg. A = 1:5; B = 0:2:10;
  - (ii) select in a range or all e.g. C= A(1,1:3); D = A(:,:)
  - (iii) reshape a matrix to a column eg. E = A(:);

Uses of the characters ... ' ! ~

- ... used to continue the code in the next line

```
1 if length(a)<5
2      a = [1,2,3,...
3           4,5];
4 end% end for the if condition
```

- '
  - (i) to transpose e.g. t =[2,3,4]; s=t';
  - (ii) create strings e.g. z = 'Hello';
- ~ means not e.g. is not equal ~=
- ! same as system

Creation of Arrays

```
1 % A row vector: use the column seperator
    , or space
2 A    = [1,2,3,4,5,6];
3
4 % A column vector: use the row seperator,
    ;
5 D1   = [2;4;5;6];
6
7 % Matrices of many columns and rows
8 C1 = [2,4,6;2,3,5;1,NaN,3];
9 C2 =[C1;C1];
```

Special Matrices

```
1  % Identity matrix of n by n
2  I = eye(4);
3
4  % Matrix of ones: ones(raws,columns)
5  D1  = ones(2,5);
6
7  % Matrix of zeros: zeros(raws,columns)
8  E1  = zeros(5,31);
9
10 % n by n magic square matrices
11 M = magic(3);
```

Matrix Operations

- Transpose of A - A'
- Inverse of A - inv(A)
- multiplication- A*A
- element-wise multiplication- A.*A
- For a linear equation, Ax = b;

```
1 x=A\b;
```

- Concatenation
  - Horz. Concatenation - Number of raws must be the same

    Hcon=[A,A] or [A A]
  - Vert. Concatenation - Number of columns must be the same

    Vcon=[A;A]

Matrix Indexing - access a particular element(s) in a matrix.
A = [10 21; 41 51; 32 19];

- to access an element
  new matrix = A(row index,column index)

  eg. A_new=A(3,2)

- select a number of rows and columns
  new matrix = A([row numbers],[column numbers])

  eg. B_new=A([2 3],[1 2])

- deleting a raw or column elements
  A([raw numbers],[column numbers])=[]

  A(1,:)=[]

Matrix Indexing ...

- Replace an element
  A(raw index,column index) = new number

  eg. A(2,3)=100;

- Replace a set of number eg numbers > 5
  A(A>5) = new number

  eg. A(A>5)=NaN;

- finding indices for numbers
  index=find(A>5)

  A(index)=5

Conditional Statements (if, elseif, else)

- if statements, expressions end

  eg. doy= 9
  if doy < 10
  DOY=['00' int2str(doy)];
  end

- if statements, expressions elseif statements, expressions else statements, expressions end

```
if doy<10
        DOY=['00' num2str(doy)];
elseif doy>9 && doy<100
        DOY=['0' num2str(doy)];
else
        DOY=int2str(doy);
end
```

Conditional statements ... (switch, case, otherwise)

- switch switch expression
  case case expression
  statements, expresions
  ...
  otherwise
  statements,expresions
  end

  eg. `igs_station='MBAR'`
  switch igs_station
  case 'MBAR'
  lon=30.74;lat=-0.6;
  case 'EBBE'
  lon=32.54;lat=0.04;
  otherwise
  lon=NaN; lat=NaN;
  end

Loops (for and while)

- for MATLAB commands end

eg. s= 0; triglenumber=[];

```
for i1=1:20
    s=s+i1;
        if mod(s,2)==0
        continue
        end
     triglenumber=[triglenumber;s];
end
```

while loop

```
1  z = 'Mbar009-2000-12-13.txt';
2  x = [];
3  i = 1;
4  while isempty(x)
5       pt = z(1:i);
6       x= strfind(pt,'.');
7       i=i+1;
8
9  end
10 sprintf(pt)
```

Data Properties

- properties of the workspace variables are accessed by typing whos at command the prompt >>

e.g >> whos

Name	Size	Bytes	Class	Attributes
data1	2x3	50	single	
data2	100x345	5000	double	
data3	1x35	55	cell	
data4	12x34	400	char	
data5	10x5	58	struc	

string - characters enclosed in single quotes.

eg. mytext='Hello world';

- Concatenate strings like in Matrices

  eg. text=[mytext(1:5) 'my friend'];

- convert numeric values to strings - useful in labeling plots

  use int2str(numeric value) or num2str(numeric value)

  eg. datadir=['/home/andima/data/' int2str(2008) '/Cmn/']

- Format data into string using sprintf

  yr=2008;doy=3
  eg. daydir=sprintf('igs/%i/00%i',yr,doy);

Cell Arrays - Each element may point to a scalar, an array, or another cell array.

eg. C=cell(2, 3)`%create 2x3 empty cell array`

M = magic(2);

a = 1:3; b = [4;5;6]; s = 'This is a string';

C{1,1} = M;

C{1,2} = a;

C{2,1} = b;

C{2,2} = s;

- Some of the useful functions for cells include

  iscell, cell2mat

Structure - good for grouping arrays that are related.

eg. name(1).last = 'Smith';name(2).last = 'Hess';

name(1).first = 'Mary'; name(2).first = 'Robert';

name(1).sex = 'female'; name(2).sex = 'male';

- Alternatively

  name = struct('last','Smith','Hess',...,
  'first','Mary','Robert','sex','female','male');

- Related utilities: isstruct, fieldnames, getfield, isfield

File types

- script m-files (.m): commands that reside in the base of workspace
- function m-files (.m): memory access controlled; parameters passed as input, output arguments; reside in own workspace
- mat files (.mat): binary or text files handled with save and load
- mex files (.mex): runs C/FORTRAN codes from m-file;
- eng files (.eng): runs m-file from C/FORTRAN code
- C codes (.c): generated by MATLAB compiler
- P codes (.p): converted m-files to hide source for security

Script m-file

If you have a group of commands that are expected to be executed repeatedly, it is convenient to save them in a file

- enter commands in editor
- Save as a *.m file
- A script shares the same scope with that which it operates.

Function m-files

- It is declared with the key word function, with optional input parameters on the right and optional output on the left of $=$ sign
- all other parameters within function reside in function's own workspace.
- created in the MATLAB editor. e.g.

```
1 function M = my_mean(x)
2 M=sum(x)/numel(x);
3 end
```

- funtions may be called from a script, another function, or on command line

Importing data into MATLAB

Many functions are available to import data into matlab. The choice depends on the nature of the data. Some of these functions include

- load:
  - used to import data without text in it. eg

1	2457056.794792	7.07500	1	333.60	9.25	9.61	25.62	99.67	36.28	-99.000
2	2457056.795139	7.08333	1	333.73	9.40	9.55	25.68	99.81	36.43	-99.000
3	2457056.795486	7.09167	1	333.85	9.54	9.49	25.74	100.07	36.63	-99.000
4	2457056.795833	7.10000	1	333.98	9.69	9.43	25.79	100.29	36.81	-99.000
5	2457056.796181	7.10833	1	334.10	9.83	9.37	25.85	100.52	37.00	-99.000
6	2457056.796528	7.11667	1	334.23	9.98	9.31	25.91	100.61	37.14	-99.000
7	2457056.796875	7.12500	1	334.35	10.12	9.25	25.97	100.81	37.33	-99.000
8	2457056.797222	7.13333	1	334.48	10.27	9.19	26.02	100.92	37.47	-99.000
9	2457056.797569	7.14167	1	334.60	10.42	9.13	26.08	100.98	37.61	0.128
10	2457056.797917	7.15000	1	334.72	10.56	9.07	26.13	101.29	37.84	-99.000
11	2457056.798264	7.15833	1	334.84	10.71	9.01	26.19	101.53	38.04	-99.000
12	2457056.798611	7.16667	1	334.97	10.86	8.95	26.24	101.51	38.16	-99.000
13	2457056.798958	7.17500	1	335.09	11.01	8.89	26.29	101.57	38.30	-99.000
14	2457056.799306	7.18333	1	335.21	11.16	8.83	26.35	101.61	38.44	-99.000
15	2457056.799653	7.19167	1	335.33	11.31	8.77	26.40	101.61	38.56	-99.000
16	2457056.800000	7.20000	1	335.45	11.46	8.71	26.45	101.58	38.68	-99.000
17	2457056.800347	7.20833	1	335.57	11.61	8.66	26.50	101.69	38.84	-99.000
18	2457056.800694	7.21667	1	335.69	11.76	8.60	26.55	101.65	38.96	-99.000
19	2457056.801042	7.22500	1	335.81	11.91	8.54	26.60	101.64	39.08	0.173
20	2457056.801389	7.23333	1	335.93	12.07	8.48	26.65	101.62	39.21	-99.000
21	2457056.801736	7.24167	1	336.05	12.22	8.43	26.70	101.57	39.32	-99.000
22	2457056.802083	7.25000	1	336.16	12.37	8.37	26.75	101.43	39.41	-99.000
23	2457056.802431	7.25833	1	336.28	12.53	8.31	26.79	101.41	39.53	-99.000
24	2457056.802778	7.26667	1	336.40	12.68	8.26	26.84	101.39	39.66	-99.000
25	2457056.803125	7.27500	1	336.51	12.84	8.20	26.89	101.31	39.77	-99.000
26	2457056.803472	7.28333	1	336.63	12.99	8.14	26.93	101.28	39.90	-99.000
27	2457056.803819	7.29167	1	336.75	13.15	8.09	26.98	101.24	40.03	-99.000

```
1 filename = 'mbar034-2015-02-03.txt';
2 Data = load(filename);
```

reading data using fgetl

- fgetl: used to read each line of the data

```
1 mbar,   Uganda
2
3 -0.60147        30.73788        1337.65329
4
5 Jdatet           Time    PRN      Az      Ele     Lat     Lon     Stec    Vtec    S4
6 2457056.794792  7.07500  1       333.60  9.25    9.61    25.62   99.67   36.28   -99.000
7 2457056.795139  7.08333  1       333.73  9.40    9.55    25.68   99.81   36.43   -99.000
8 2457056.795486  7.09167  1       333.85  9.54    9.49    25.74   100.07  36.63   -99.000
9 2457056.795833  7.10000  1       333.98  9.69    9.43    25.79   100.29  36.81   -99.000
10 2457056.796181  7.10833  1       334.10  9.83    9.37    25.85   100.52  37.00   -99.000
11 2457056.796528  7.11667  1       334.23  9.98    9.31    25.91   100.61  37.14   -99.000
12 2457056.796875  7.12500  1       334.35  10.12   9.25    25.97   100.81  37.33   -99.000
13 2457056.797222  7.13333  1       334.48  10.27   9.19    26.02   100.92  37.47   -99.000
14 2457056.797569  7.14167  1       334.60  10.42   9.13    26.08   100.98  37.61   0.128
15 2457056.797917  7.15000  1       334.72  10.56   9.07    26.13   101.29  37.84   -99.000
16 2457056.798264  7.15833  1       334.84  10.71   9.01    26.19   101.53  38.04   -99.000
17 2457056.798611  7.16667  1       334.97  10.86   8.95    26.24   101.51  38.16   -99.000
18 2457056.798958  7.17500  1       335.09  11.01   8.89    26.29   101.57  38.30   -99.000
19 2457056.799306  7.18333  1       335.21  11.16   8.83    26.35   101.61  38.44   -99.000
20 2457056.799653  7.19167  1       335.33  11.31   8.77    26.40   101.61  38.56   -99.000
21 2457056.800000  7.20000  1       335.45  11.46   8.71    26.45   101.58  38.68   -99.000
22 2457056.800347  7.20833  1       335.57  11.61   8.66    26.50   101.69  38.84   -99.000
23 2457056.800694  7.21667  1       335.69  11.76   8.60    26.55   101.65  38.96   -99.000
24 2457056.801042  7.22500  1       335.81  11.91   8.54    26.60   101.64  39.08   0.173
25 2457056.801389  7.23333  1       335.93  12.07   8.48    26.65   101.62  39.21   -99.000
26 2457056.801736  7.24167  1       336.05  12.22   8.43    26.70   101.57  39.32   -99.000
27 2457056.802083  7.25000  1       336.16  12.37   8.37    26.75   101.43  39.41   -99.000
```

```
1 filename = 'mbar002-2015-01-02.Cmn';
     fid = fopen(filename);
2 eof = 'Jdatet';header =[];
3
4 while isempty(header)
5 line1    = fgetl(fid);
6 header = strfind(line1,eof);
7 end
```

reading data using textscan

- textscan: used to read each line of the data

```
1 filename = 'mbar002-2015-01-02.Cmn';
     fid = fopen(filename);
2 eof = 'Jdatet';header =[];
3
4 while isempty(header)
5 line1    = fgetl(fid);
6 header = strfind(line1,eof);
7 end
8 Data = textscan(fid);
```