# Introduction to MATLAB

Larry Caretto
Mechanical Engineering 309

***Numerical Analysis of Engineering Systems***

February 3, 2014

California State University Northridge

## Outline

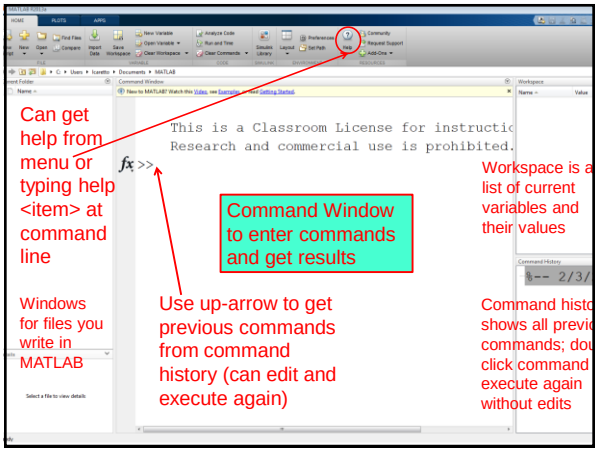
- MATLAB windows and Help
- Commands from command window
  - Formatting data and command window
  - Functions in MATLAB
  - Entering and using arrays of numbers
  - Introduction to plotting in MATLAB
  - Saving the workspace and diary files
  - Starting, saving, and opening m-files
  - Script files and MATLAB functions



## Starting MATLAB

- During class we will start working on the next programming assignment
- Start MATLAB and Word on your computers
  - Save Word file as <lastName>_pa2.docx
- Download second programming assignment from home page of course web site
  - www.csun.edu/~lcaretto/me309/pa2.pdf






## MATLAB Exercise

- Type `format compact` at MATLAB command prompt (>>)
- Use a wide command window to get as many results as possible on one line
- From time to time copy MATLAB commands, results, plots, and files to Word file
- Word file with MATLAB information and your discussion is only file required



## MATLAB Basics

- Variables are introduced without declaration of types or array size
- In general any variable can be an array of complex numbers
  - Can also be a simple number like 3
- Variables are case sensitive (A is not a)
- Type commands at >> prompt
- Results echoed after command unless you end command with semicolon (;)

## Command Window

- Examples of commands (following >>) and resulting output given below
- Copied from actual MATLAB session, but edited to eliminate blank rows
- Press **Enter** key after each command

```
>> %This is a comment
>> %Assign values to variables
>> x = 2
x =  2
>> y = 3
y =  3
>> %Default result is ans
>> x + y
ans =  5
```

```
>> %Assign result to variable
>> z = x/y
z =  0.6667
>> %Change output format
>> format ('long')
>> %Print value of z
>> z
z = 0.666666666666667
```

California State University Northridge



## Formatting Output

- **format** command sets output
- **format compact** and **format loose** set line spacing as narrow or wide
- Decimal display is controlled by format
  - **format short** gives five significant figures
  - **format long** gives fifteen significant figures
  - Enter **help format** for other choices
- All calculations done with ~15 significant figures regardless of format

California State University Northridge



California State University Northridge

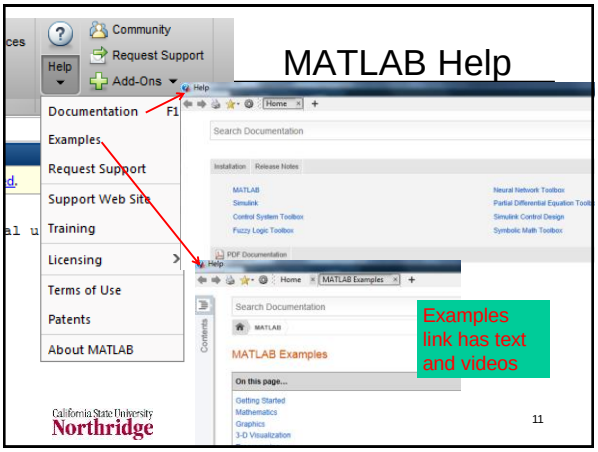


## Functions and Complex Numbers

- MATLAB has large number of functions
- Simple ones similar to other languages
  - Partial list: sin, sind, cos, tan, sqrt, asin, atan, sinh, tanh, log (natural log), log10, *etc.*
  - >> log(10) gives ans = 2.3026
- Can also have complex numbers
  - >> 2 + 3i gives ans = 2.0000 + 3.0000i
  - Can also use j in place of i for $\sqrt{-1}$
  - MATLAB does complex arithmetic and more

Returns sine of argument in degrees

California State University Northridge




## MATLAB Help

California State University Northridge



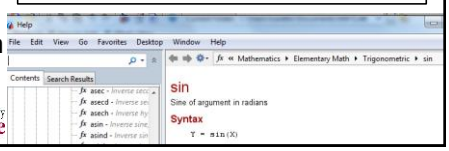
## More on Help

- Help example at right
- Click on doc sin for full html help on sine

```
>> help sin
 SIN    Sine of argument in radians.
    SIN(X) is the sine of the elements of X.

    See also asin, sind.

    Reference page in Help browser
       doc sin
```




California State University Northridge

## Arrays of Numbers

- We will say more about the following definitions when we cover matrix analysis
- An array of numbers in a row is called a row vector, *e.g.* r = [1 5 -4 10 3]
- An array of numbers in a column is called a column vector (see next slide)
- An two-dimensional array is called a matrix (see next slide)
  - Row/column vectors special case of matrix

California State University Northridge 13

## Matrix Examples

Row vector: r = [1 5 -4 10 3]

Column vector:

$$c = \begin{bmatrix} 8 \\ 17 \\ -21 \\ 14 \\ -3 \end{bmatrix}$$

A matrix with 3 rows and 5 columns is called a 3 by 5 (3 x 5) matrix

$$A = \begin{bmatrix} 12 & 8 & -9 & 0 & 45 \\ -7 & -16 & 23 & 87 & -12 \\ 0 & 3 & 5 & 6 & 16 \end{bmatrix}$$

The row vector above is a 1 by 5 matrix and the column vector is a 5 by 1 matrix

California State University Northridge 14

## Entering Arrays

– Enter a row vector by enclosing data in [ ] separated by a space

```
>> row = [12  -3  5  7  0]
row =
    12    -3     5     7     0
```

– Enter a column vector by enclosing data, separated by a semicolon (;) in [ ]

```
>> col = [-3; 6; 0; 4]
col =
    -3
     6
     0
     4
```

California State University Northridge 15

## Entering a Matrix

– Enter matrix data row by row

– Put spaces between data in the same row

– Put a semicolon to start data on next row

- MATLAB uses the … as a continuation signal
- After the … hit Enter and continue input of same command on a new line

```
>> A=[1 2; 3 4]
A =
     1     2
     3     4
>> B = [1 2 3; 4...
        5 6; 7 8...
        9]
B =
     1     2     3
     4     5     6
     7     8     9
```

California State University Northridge 16

## Entering a Matrix II

– Pressing enter after each row of data can be used to enter a matrix

– Using semicolons, all data can be placed on one row (see below)

– Continuation is only needed to start new line in the middle of data entry

```
B =      (Result)
     1     2     3
     4     5     6
     7     8     9
```

```
>> B = [1 2 3
        4 5 6
        7 8 9]
>> B = [1 2 3; 4...
        5 6; 7 8 9]
>> B = [1 2 3; 4 5 6; 7 8 9]
```

California State University Northridge 17

## Command History

- Can use up-arrow key to get old commands
- Can edit old commands
  - Use left- and right-arrow keys to move backward and forwards in command
  - Backspace or delete removes characters
  - Type new characters as desired
  - Hit enter to execute edited command
  - Repeat editing if second command is wrong

California State University Northridge 18

## Addressing Matrix Elements

- For row matrix, r, or column matrix, c, r(k) or c(k) is $k^{th}$ element in matrix
- If A is a matrix, A(m,n) is element at row m and column n
- If A is a matrix A(r1:r2,c1:c2) is a sub-matrix that includes all rows between r1 and r2 and all columns between c1 and c2
- If you want all columns or all rows enter A(r1:r2,:) or A(:,c1:c2), respectively

California State University Northridge 19

## Exercise: Enter the following

Row vector: r = [1 5 -4]

(3 x 2) matrix

$$B = \begin{bmatrix} 5 & 4 \\ 1 & 4 \\ 2 & 8 \end{bmatrix}$$

(3 x 3) matrix

$$A = \begin{bmatrix} 12 & 8 & 0 \\ -7 & -16 & 4 \\ 0 & 3 & 2 \end{bmatrix}$$

Column vector:

$$c = \begin{bmatrix} 8 \\ 3 \\ -2 \end{bmatrix}$$

Ask yourself, "what will the result of the following formulas be?" before entering them
Note that for any matrix, A, A' gives $A^T$, the transpose of A (the matrix A with rows and columns exchanged)
D = [A B] E = [B A] ct = c' F = [r; ct] G = [c A ct']
H = [G; D]

D and E results on slide 28

California State University Northridge 20

## Exercise: Saving the Workspace

- Select **File** | **Save Workspace As**
  - Use default file name (matlab.mat) or enter one of your own and click **Save**
  - Exit and restart MATLAB
  - Select **File** | **Open**, click on saved file name (matlab.mat) in dialog, and click **Open**
  - Type **whos** at command line to see that all your previous variables are there
  - Type H, then type H(2,3) at the command line. Is the result for H(2,3) correct?

California State University Northridge 21

## Exercise: SubArrays

- Recall that A(r1:r2,c1:c2) is a submatrix of A that includes all rows between r1 and r2 and all columns between c1 and c2
- Type the following and make sure that you believe the results (use <Enter>)
  - H(2:4,3:5)
  - H(:,2:4)
  - H(1,:)
  - H(2:4,3)

Type H(2:4,3) = [-1; -2; -3] at the command line then type H

Are the results what you expected?

California State University Northridge 22

## Other Array Definitions

- Use <variable> = <lower bound> : <increment> : <upper bound>
  - Can also use <lower bound> : <upper bound> with default <increment> = 1
  - Example creates array t

```
>> t = 0:pi/4:2*pi
t =
  Columns 1 through 7
         0    0.7854    1.5708    2.3562    3.1416    3.927    4.7124
  Columns 8 through 9
    5.4978    6.2832
```

California State University Northridge 23

## Array Functions

- If A is an array, a function like sin(A) is also an array giving the value of the sine for each element in the array, A

```
>> s=sin(t)
s =
    0    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.00
```

- We can use arrays like these to generate plots
  - Redo t and s to have more data points using `t = 0:pi/100:2*pi;`

Use semicolon(;) to suppress output

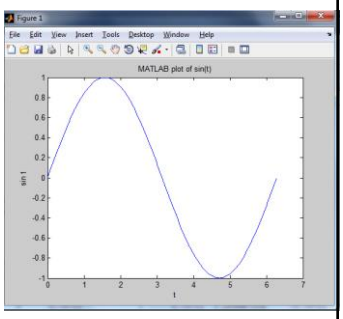
California State University Northridge 24

## MATLAB Plotting

– Commands for t and s below use semi-colon at end of line to suppress output

```
>> t=0:pi/100:2*pi;
>> s=sin(t);
>> plot(t,s)
>> ylabel('sin t')
>> xlabel('t')
>> title('MATLAB plot of sin(t)')
```

Single quotes define string constant




California State University Northridge 25

## Script Files

- MATLAB script files are like Excel macros: they are a set of commands that are executed by a single command
  - Type file name at command line to execute
- To start script file select **New** from home tab then select **Script** in submenu
- Copy the previous commands for sine plot from the command window to a file
- Can change sine to cosine

California State University Northridge 26

## Script File

```
%cosinePlot -- a script file
%       to plot the cosine
%Larry Caretto  31 Jan 14
%------------------------
t=0:pi/100:2*pi;
s=cos(t);
plot(t,s)
ylabel('cos t')
xlabel('t')
title('MATLAB plot of cos(t)')
```

File must be saved as cosinePlot.m

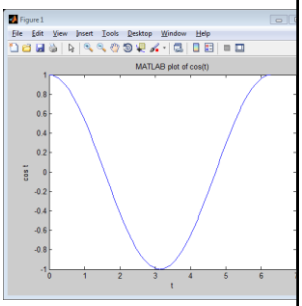
- Script file shown in box at left
  - Name is first entry after %
- First four lines are comments
  - Will show by help command
- Type file name at >> to execute

California State University Northridge 27

## Using Script File

```
>> help cosinePlot
cosinePlot -- a script file
       to plot the cosine
Larry Caretto  31 Jan 14
------------------------

>> cosinePlot
>>
```




- Commands above show results of help command and execution command

California State University Northridge 28

## Array Operations

- Can do mathematical operations on arrays providing they are compatible
- Two kinds of operations
  - Term-by-term operations act on each term in an array
    - Arrays must be same size (rows and columns)
  - Matrix operations will be covered after discussion of matrix analysis
  - Addition and subtraction are same for both types of operation

California State University Northridge 29

## Addition and Subtraction

```
D =
12   8  0  5  4
-7 -16  4  1  4
 0   3  2  2  8
```

```
E =
 5   4 12   8  0
 1   4 -7 -16  4
 2   8  0   3  2
```

```
D + E =
17  12 12   1  4
-6 -12 -3 -15  8
 2  11  2   5 10
```

D and E are shown on slide 18

```
D - E =
 7   4 -12 -3  4
-8 -20  11 17  0
-2  -5   2 -1  6
```

California State University Northridge 30

## Array Times a Scalar

- A single number, a = 3, is a scalar
- When an array is multiplied (or divided) by a scalar each term in the array is multiplied (or divided) by the scalar
- For r = [ 6 3 -1], 3r = [18 9 -3]
- For the A array defined previously, what is 4A? What is A/2?

$$A = \begin{bmatrix} 12 & 8 & 0 \\ -7 & -16 & 4 \\ 0 & 3 & 2 \end{bmatrix} \quad 4A = \begin{bmatrix} 48 & 36 & 0 \\ -28 & -64 & 16 \\ 0 & 12 & 8 \end{bmatrix} \quad \frac{A}{2} = \begin{bmatrix} 6 & 4 & 0 \\ -3.5 & -8 & 2 \\ 0 & 1.5 & 1 \end{bmatrix}$$



## Term-by-term Operations

- Have seen addition and subtraction
- Term-by-term multiplication and division of arrays use operators .* and ./
  - The period (.) distinguishes these from matrix operations to be covered later
- For term by term operations the arrays must be the same size
  - Same number of rows and same number of columns



## Term-by-term Multiplication

```
  D =
12   8 0 5 4
-7 -16 4 1 4
 0   3 2 2 8

  E =
 5  4 12   8 0
 1  4 -7 -16 4
 2  8  0   3 2
```

```
  D .* E =
 60    32     0    40     0
 -7   -64   -28   -16    16
  0    24     0     6    16
```



## Term-by-term Division

```
  D =
12   8 0 5 4
-7 -16 4 1 4
 0   3 2 2 8

  E =
 5  4 12   8 0
 1  4 -7 -16 4
 2  8  0   3 2
```

```
  D ./ E =
 2.400  2.000       0  0.6250   Inf
-7.000 -4.000 -0.5714 -0.0625 1.000
     0  0.375     Inf  0.6667 4.000
```



## Operations on Formulas

- How do you compute $e^{ax}\sin(bx)$ for an x array between 0 and 2 with $\Delta x = .01$?
  - Set the x array: >>**x = 0:0.01:2;**
  - Define constants, *e.g.*: >> **a = –1** >>**b = 12**
  - Use term by term multiplication for arrays of exponential and sine terms:
  - >> **y = exp(a*x).*sin(b*x);**
  - Do these steps yourself then execute command **plot(x,y)**

Use the .* operator for term-by-term array operations



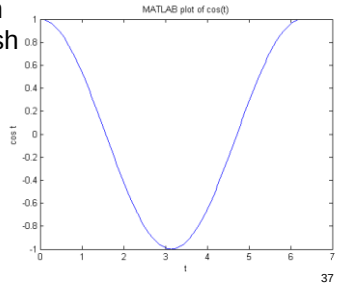
## The publish Command

- Used to publish reports in various formats: html, Word, Power Point, *etc*.
- Type help publish for full information
- Sample for cosinePlot script
- `>>publish('cosinePlot', 'html')`
- Result is ans= <path to html document>
- C:/Users/lcaretto/Documents/MATLAB/html/cosinePlot.html
- Copy this path and paste it into the URL box of your web browser

## Result

- File that results from using publish command shown on previous slide

```
%cosinePlot -- a script file
%           to plot the cosine
%Larry Caretto  31 Aug 11
%-------------------------
t=0:pi/100:2*pi;
s=cos(t);
plot(t,s)
ylabel('cos t')
xlabel('t')
title('MATLAB plot of cos(t)')
```






## MATLAB File Locations

- Stored as file in MATLAB subdirectory of your Documents folder
  - C:\Users\lcaretto\Documents\MATLAB
- Need to use the same computer in lab to have access to old files
  - Can use Save As command in MATLAB to save files to memory stick (or copy from your MATLAB directory)
  - Possible to save in other directories
    - Type **help userpath** to see how to do this



## Saving Files to Z: Drive

- Create matlab subdirectory on Z: drive (or U: drive) using Windows
  - Choose subdirectory appropriate for you
    - This slide assumes matlab subdirectory
- At MATLAB command prompt, enter the command: `userpath('z:\matlab')`
- To check that this worked, enter the command: `userpath`
  - Should get: `ans = z:\matlab;`



## The `diary` Command

- Keeps a record of your commands and MATLAB responses in a text file
  - Use **diary <filename>** to start and **diary off** to stop
    - *E.g.* **diary 20140203.out** will start recording
  - Can stop and restart in same session
  - Use command **diary on** to append new commands to previous ones
  - This command does not copy figures
  - Can edit diary file in text editor



## More About Plotting

- The plot command is plot(x, y, 'options')
  - x,y are variables to be plotted on x,y axes
  - Options is a string that can combine several options (color, line style, marker)
    - Default (no options) plot is solid blue line with no markers
    - Colors are blue(b), green(g), red(r), cyan(y), magenta(m), yellow(y), black(b), white(w)
    - Line styles: solid(–), dotted(:), dash-dot(–.), dashed(--), and no line(none)
    - Example: dashed red line option is 'r--'



## More Plot Options

- Marker options: point(.), circle(o), x-mark(x), plus(+), star(*), square('s'), *etc*.
  - What options does 'g:s' specify
- xlabel('time') writes time as label on x axis
- ylabel('function') for function as y axis label
- title('Chart Title')
  - Note use of character constants by placing text inside single quotation marks(')
  - See **help plot** or **doc plot** for more options including multiple lines on one plot, formatting, and mathematical symbols in labels

## Still More Plot Options

- Four plot comands: plot, semilogx, semilogy, and loglog
  - Give linear, semilog (x or y) or log-log plots
  - Place one or more lines on each plot
  - All commands have similar syntax with following arguments (x1, y1, options1, x2, y2, options2, … , xN, yN, optionsN)
    - xK and yK are arrays plotted on x and y axes
    - optionsK describe colors, line style, etc. for data set K

California State University Northridge 43

## Plot Legend and Axes

- Legend for multiple lines on one plot
  - legend ('title1', 'title2', … , 'titleN')
    - Provides title for each line in same order that lines are specified in plot commands
    - Legend in upper-right-hand corner
    - Shows line color and marker next to title provided in legend command
- axis command sets limits as array
  - axis( [xMin, xMax, yMin, yMax] )
  - Default limits show all data

California State University Northridge 44

## MATLAB Functions

- Contrast with script files that use any defined variable in the current session
- Function files receive inputs through argument list like other languages
- MATLAB functions can return more than one variable
- Function <name> ( <argument list> )
- Each variable returned by a MATLAB function can be a scalar or an array

California State University Northridge 45

## Trajectory Function

```
function [x, y] = traj(v0, theta, N)
%Computes frictionless trajectory
%Uses SI units (meters, seconds)
%V0 is initial speed in m/s
%theta is initial angle in degrees
%N is number of points computed
g = 9.80665;    %gravity in m/s^2
tMax = 2 * v0 * sind(theta) / g;
t = 0:tMax/(N-1):tMax;
x = v0*cosd(theta) * t;
y = v0*sind(theta) * t - g * t .^2/2;
plot(x,y);
end
```

Use file names the same as the function names (e.g. traj.m) to save functions

Array operation

Use semicolons to avoid intermediate output from function code

California State University Northridge 46

## Using Your MATLAB Functions

- Used as any MATLAB Function

```
>>v0 = 10;
>>theta = 60;
>>N = 100;
>>[x, y] = traj(v0, theta, N);
```

- Can use only part of return variables
  - >>= traj(v0,theta,N) returns x values in ans
  - >>x = traj(V0,theta,N) returns x values in x
  - >>[x y] = traj(V0,theta,N) for x and y values

California State University Northridge 47

## MATLAB if Statements

- Use the following format

```
if <expression1>
    <statements1>
elseif <expression2>
    <statements2>
<other elseif's here>
else    %optional
    <statements>
end
```

Same structure as VBA but "Then" not used

All keywords (if, else, elseif) in lower case

Final statement is end, not endif

California State University Northridge 48

## While Statement

- MATLAB has only one conditional looping command with a test before

```
while <condition>
    <statements>
end
```

MATLAB keywords (while and end) must be lower case

- The `<statements>` in the while loop continue to execute while the `<condition>` is true

California State University Northridge 49

## for Statement

- Similar to VBA For statement, but loop limits are a MATLAB array specification

```
for <index> = <MATLAB array>
    <statements>
end
```

- Examples of for statements

```
for T = [300, 500, 1000, 5000]
for x = 0 : 0.01 : 2
for k = 1 : 25
```

(Same as 1:1:25)

California State University Northridge 50

## Convert 2D Array to Column

- Convert two-dimensional array, x, to one-dimensional array, y, in MATLAB
- Can enter following lines from command prompt or write as script
  - Used in second programming exercise

```
n = size(x); %n(1) = rows; n(2) = columns
y = x(:,1);  %copy first x column into y
   %copy remaining x columns into y
for k = 2:n(2)   y = [y; x(:,k)];   end
```

California State University Northridge 51

## Publishing to Word

```
% Define the structure, options_doc_nocode,
% to include code in the output
% and publish to Microsoft Word format:
options_doc_nocode.format='doc'
options_doc_nocode.showCode=true
% Publish sine_wave.m:(put m-file name here)
publish('sine_wave.m',options_doc_nocode)
% View the published output file in Word:
winopen('html/sine_wave.doc')
```

Matlab creates *.doc name with same first part as m-file (cannot use docx)

California State University Northridge 52

## MATLAB Review 1/2

- User interface, command window, help
  - Commands echoed unless followed by ;
  - Use `format compact` for small spacing
  - Conventional mathematical expressions and functions
  - To enter arrays separate items on one row by spaces and separate rows by semicolon
  - Combine smaller arrays to get larger ones only if arrays are compatible

California State University Northridge 53

## MATLAB Review 2/2

- Get array components, A(row,column), or subarrays, A(r1:r2,c1:c2), A(:,c1:c2), A(:,c1), A(r1,:), A(r1:r2,c1), A(r1,c1:c1)
- Set array components equal to some values as in A(3,2:4) = [4 9 6] (or is it [4; 9; 6]?)
- Define array as <start>:<increment>:<last>
- Functions of arrays
- Term-by-term operations (+ - .* ./ .^)
- Plot commands
- Scripts and publish command

California State University Northridge 54