



ECE 2409

Fundamentals of MATLAB



Fall 98





What is MATLAB?

- MATLAB (**MAT**rix **LAB**oratory) is a numerical simulation software.
- Its genius was the adoption of a vector/matrix structure for inputting data.
- Multidisciplinary: science, engineering, finance
- MATLAB runs on a variety of platforms, e.g., PC, Mac, Sun etc.



What Can MATLAB Do?

- **Any problem that can be cast in a matrix form is amenable to an efficient MATLAB-based solution.**
 - **Matrix manipulations**
 - **System of equations, e.g. circuits.**
 - **Interpolation and curve fitting.**
 - **Numerical integration/differentiation; Diff. Eq.**
 - **Signal processing, image processing, filtering**
 - **Control systems**
 - **Communications systems**





Course Outlook...

- **First 6 weeks:**

- **Basic and advanced matrix operations**
- **Basic and advanced plotting**
- **Characters and strings**
- **Color control**
- **Handle graphics**





...Course Outlook

- **Next 4 weeks**

- **Multimedia and MATLAB**

- **scripts and functions**

- **linear equations**

- **interpolation and curve fitting**



...Course Outlook

- **Last 2 weeks**

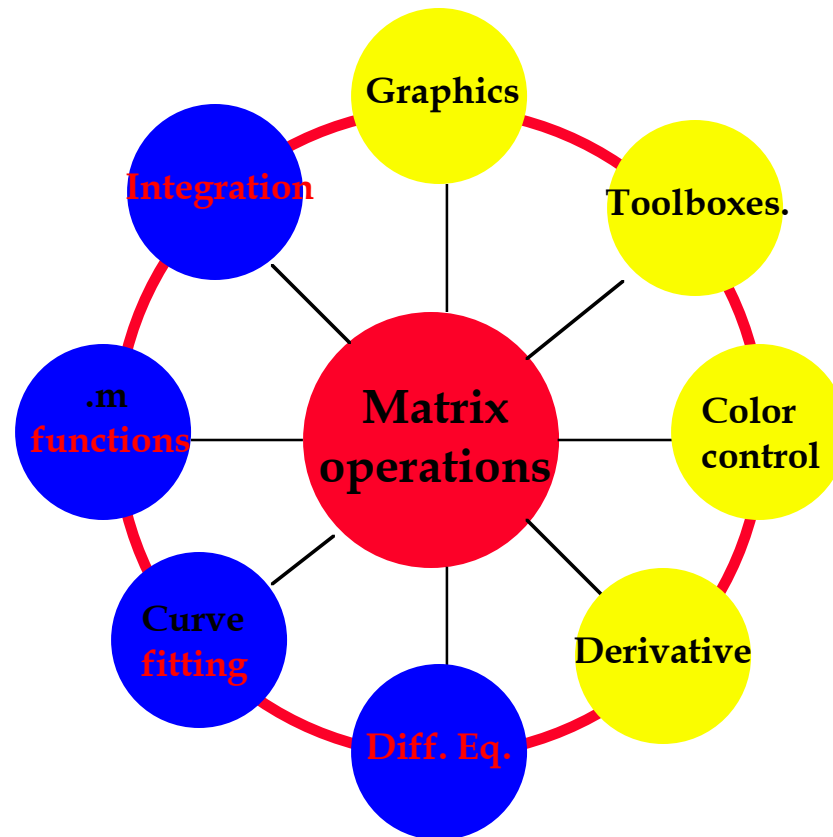
- integration, differentiation, differential equations

- MATLAB toolboxes





ANATOMY OF MATLAB





Net Resources on MATLAB

- **MathWorks Inc., the developers of MATLAB have their own site with many related links**


<http://www.mathworks.com>

- **There is a user's newsgroup, closely followed by MathWorks folks:**

comp.soft-sys.matlab



Running MATLAB

- Find the MATLAB folder
- Double click on MATLAB icon 
- You will then be presented with the “command” window in which you can issue your instructions

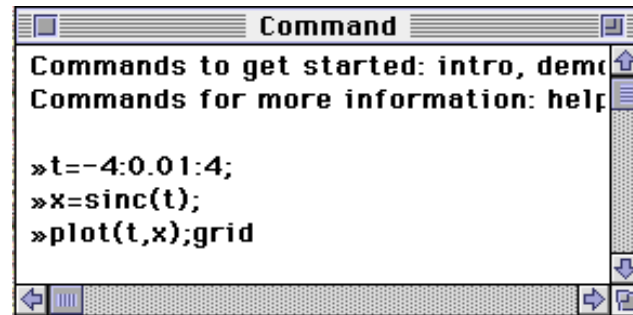
MATLAB Look

- In a typical MATLAB session there are 3 active windows:

→ Command window

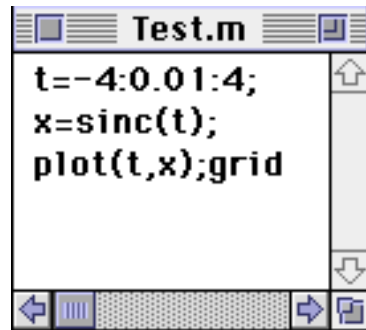
→ Figure window

→ .m file window



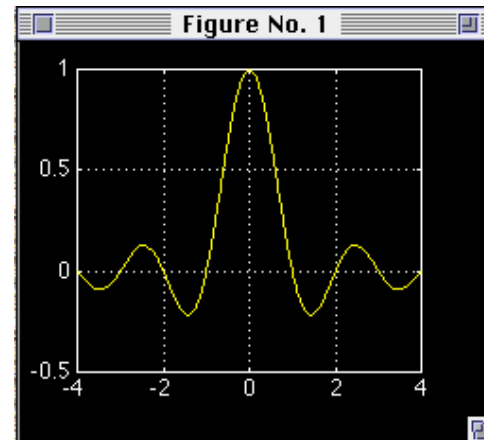
Command

```
Commands to get started: intro, dem  
Commands for more information: help  
  
»t=-4:0.01:4;  
»x=sinc(t);  
»plot(t,x);grid
```



Test.m

```
t=-4:0.01:4;  
x=sinc(t);  
plot(t,x);grid
```





Interacting with MATLAB

- **There are two ways to interact with MATLAB**
 - **Command window**
 - **.m file**
- **Instructions in the Command window are executed immediately. Also, any error messages are displayed in there.**





Generating a .m File

- From the file menu, choose **New**. An untitled window appears in which you can enter instructions.
- When finished, select **Save and Execute**. You have to assign a name and location to the .m file. MATLAB will then run your code





MATLAB SEARCH PATH

- MATLAB needs to know where to look for .m files and other functions
- Every folder and subfolder under MATLAB directory is on the search path
- To find the the existing *search path* type 'path'





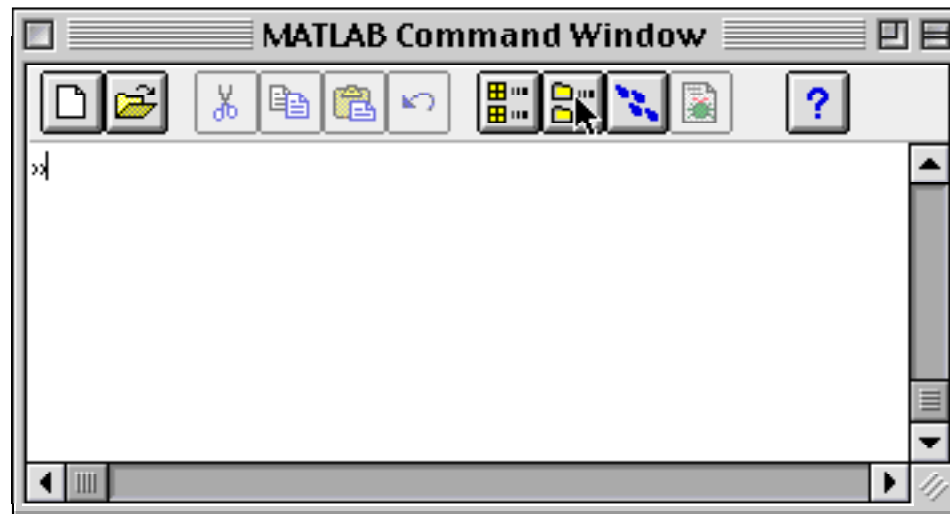
SETTING SEARCH PATH

- To add a new path
 - `path(path,'macintosh hd: desktop folder')`
- For PC and Unix machines the syntax is slightly different. Type 'help path' for additional information
- Selective directories can be added and removed using
 - `addpath(path,'macintosh hd:myfolder')`
 - `rmpath(path,'macintosh hd:myfolder')`



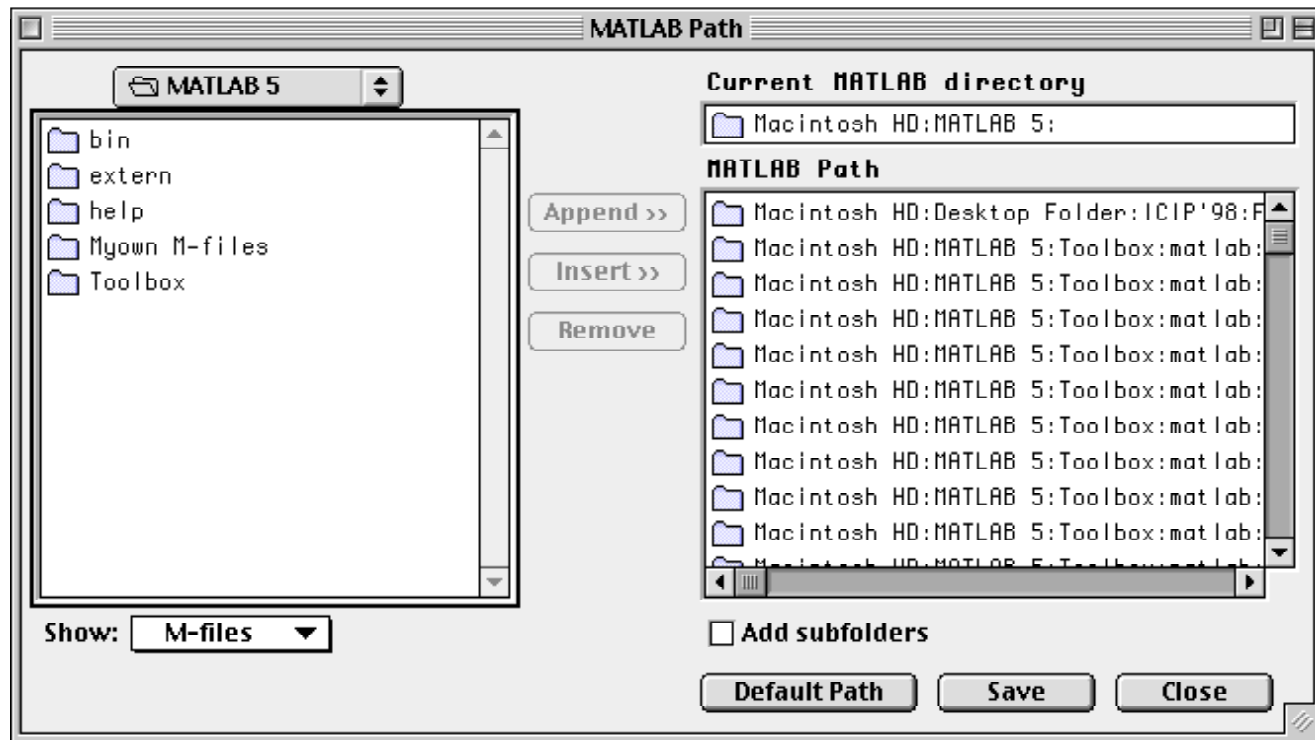
SETTING PATH IN MATLAB 5

- In the command view click the path button





HERE IS WHAT YOU GET





MAKING SETTINGS PERMANENT:*startup.m*

- Path and other settings will expire upon quitting MATLAB.
- To make them permanent, there is a 'startup.m' file.
- Place your path statement, or other settings, in 'startup.m'



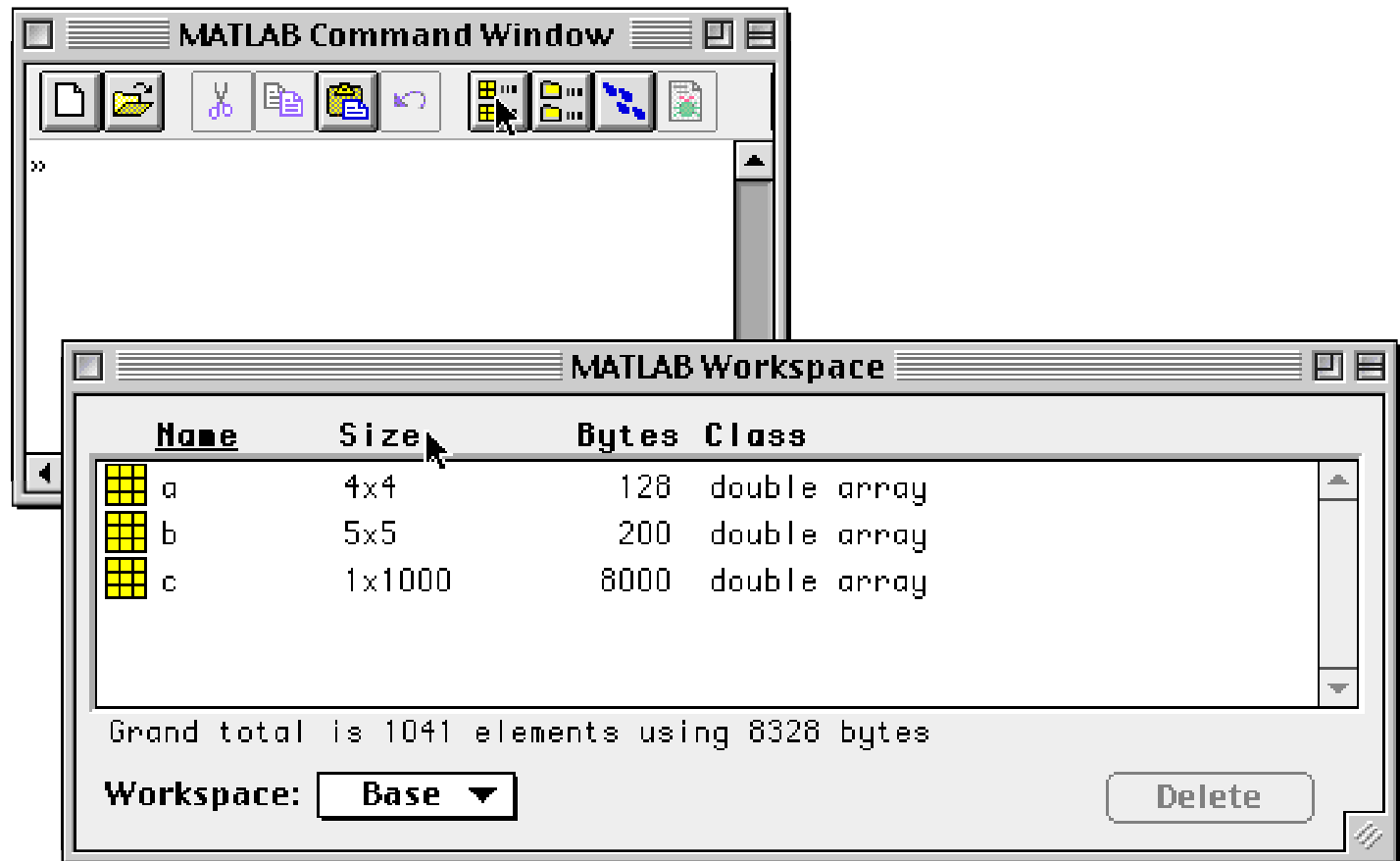


MATLAB WORKSPACE

- The place that holds your variables, arrays and matrices is called *workspace*
- If you want to see what variables you have defined, their size and storage requirement, workspace is where you look at
- type 'whos' at the prompt



VIEWING WORKSPACE IN MATLAB 5.1





GETTING HELP IN MATLAB

- There are a number of places you can turn to get help
 - Online help by typing 'help ...'
 - 'lookfor ...' searches by keyword
 - User's guide
 - Instructor/TA
 - MATLAB's web site
 - comp.soft-sys.matlab



IMPORTING DATA INTO MATLAB

- To have an output, we need an input. How do you make MATLAB read your data?
 - Manual
 - Generate data in an .m file (e.g. `s=cos(t)`)
 - Load data from an ASCII file
 - For image data use 'imread'

Basic MATLAB Input

- The way you interact with MATLAB is through scalars, vectors and matrices.
- You can define a vector of values using the following syntax:

→ `v=[1 5 6 8 9];`

- For a 2D array, the following syntax applies:

→ `V=[1 2 3;4 5 6;7 8 9];`

→ This is equivalent to

$$V = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$



Important operators: semi-colon(;) and colon(:)

- A semi-colon separates rows of a matrix, i.e.

```
V=[1 2 3;4 5 6;7 8 9]
```

- Semi-colon suppresses MATLAB's response. Try the above with and without a semi-colon at the end of the line.
- The last use of semi-colon is to string along several commands on one line:

```
v=[1 2 3 4];plot(v);grid
```





Using Colon for Automatic Array Generation

- For large arrays sizes, manual generation is not practical.
- For example, want to fill a time array `t` from 0 to 1 second in increments of 1/100 of a second.

```
t=0:1/100:1;
```

- If `;` is omitted, you will see all the elements of array `t` appear on the screen



Selecting a Subset of a Matrix

- Colon is a more powerful operator. It allows you to pick subsets of a larger matrix
- For example, if V is a 5x5 matrix, then

$$W=V(1:3,5)$$

are rows 1 to 3 and column 5 of V .

- We can skip rows or columns,

$$W=V(1:2:5,4)$$

Examples

● $W=V(1:3,5)$

$$V = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$$



Examples

- $W=V(1:2:5,3)$

$$V = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$$

...Continued

- If you want all the rows but only a few columns use:

$$W=V(:,2:3)$$

- In the above, we are picking all 5 rows of V but only columns 2 and 3

$$V = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix} \quad \rightarrow \quad W = \begin{bmatrix} 2 & 3 \\ 7 & 8 \\ 12 & 13 \\ 17 & 18 \\ 22 & 23 \end{bmatrix}$$



Array Operations

- Array operations are defined on an **element-by-element** basis. For example,
- For example, if $A=[a_1, a_2, a_3]$ and $B=[b_1, b_2, b_3]$ then element-by-element multiplication is written in MATLAB by preceding the operator by a dot(.),

$$A.*B=[a_1b_1 \ a_2b_2 \ a_3b_3]$$

- Same holds for all other operations: division, exponentiation, addition subtraction



Examples

- $x=[1\ 3\ 5\ 2\ 9\ 7]$ and $y=[4\ 3\ 1\ 5\ 4\ 8]$. What is $x.*y$?
- $Z=x.*y=[4\ 9\ 5\ 10\ 36\ 56]$



Why term-by term multiplication?

- Term-by-term multiplication arises in evaluating functions.
- For example,

$$y = e^{-4t} \cos(10t)$$

- If t goes from 0 to 1 in increments of 0.01, then each term must be evaluated individually, then multiplied term by term



More Examples

- $v=[1:5]$ is equivalent to $v=[1\ 2\ 3\ 4\ 5]$.
- $v=[1:2:10]$ is equivalent to $v=[1\ 3\ 5\ 7\ 9]$
- $v=[5:-1:1]$ is equivalent to $v=[5\ 4\ 3\ 2\ 1]$
- Guess what $V=[1:3;4:6;7:10]$ is equivalent to?



Matrix Transposition

- To transpose a matrix use apostrophe. Try this on an array as well as a matrix

$$W=V'$$

- Try transposition on the transposed matrix again to see if you get the original matrix back.



Comment line

- MATLAB interprets any line with a % in front of it as a comment line
- It is a very good practice to comment every line of code, no matter how short.
- Usage
 - $V=W'$; %V is the transpose of W



Useful Functions

- size(x) returns the size of an array or matrix
- length(x) does a similar thing
- whos shows your “workspace”
- clear clears all variables
- clear x clears variable x only
- clc clears command window



Special Matrices

- [eye\(m\)](#) creates an $m \times m$ identity matrix
- [eye\(mxn\)](#) creates an $m \times n$ identity matrix
- [zeros\(m\)](#) creates an $m \times m$ all zero matrix
- [zeros\(mxn\)](#) creates an $m \times n$ all zero matrix
- [ones\(m\)](#) creates an $m \times m$ all 1's matrix
- [ones\(mxn\)](#) creates an $m \times n$ all 1's matrix



Exercises:In-class

- **In-class**

- Do the practice on the bottom of page 40 in Etter's book
- Evaluate and plot the function on slide #31 for $0 < t < 1$.
- Go through every special matrix on slide #36



Exercises:homework

- Read Chapter 2
- Do the second Practice on Page 54. First, get the answers manually then check them using MATLAB
- Evaluate and plot function f on page 53. For x , pick a range and increment that produces a nice plot



What to turn in

- You need to turn in your MATLAB m-file along with other material, as appropriate
- Top of your m-file should contain
 - Due date
 - Name
 - Section
- Every line should be commented