MATLAB Tutorial G377/277

Introduction
MATLAB – Matrix Laboratory created by The MathWorks, Inc. http://www.mathworks.com located in Natick, MA, is a commercially available numerical analysis package. MATLAB has 100’s of built-in functions and has the capability of creating new functions and developing graphical user interfaces (GUIs) to access these functions. MATLAB toolboxes are groups of functions that sold as separate packages. These special purpose toolboxes may be designed for signal processing, image processing, statistical analysis, database access, optimization, symbolic math, hardware programming... the list goes on and on. MATLAB also has the ability to create rich images and plots of data. Data graphics can be customized and annotated to display in research papers, proposals and presentations.
Getting Started
Start up MATLAB

Now what?
Getting help
At the MATLAB Command prompt type help then any of the words below.

general - General purpose commands.
ops - Operators and special characters.
lang - Programming language constructs.
elmat - Elementary matrices and matrix manipulation.
elfun - Elementary math functions.
specfun - Specialized math functions.
matfun - Matrix functions - numerical linear algebra.
datafun - Data analysis functions

More specific help… In newer versions of MATLAB the keyword doc can be used instead of help. The doc command followed by a function name will bring up the HTML supporting document for that function. Try it with the same word list above, for example doc general or doc fprintf.

Getting Started
Vector and Arrays
A vector is a single row or column of numbers.
An array is a collection of vectors or numbers.

Creating a vector:
\[ x = [1, 2, 3, 4] \], row vector
\[ x = [1 \ 2 \ 3 \ 4] \]
\[ x = [1; 2; 3; 4] \], column vector
\[ x = 1:1:4 \], Colon operator (start, increment, end)

Creating arrays:
\[ y = [1, 2, 3, 4; 5, 6, 7, 8] \]

Finding which variable are in the memory space
who - simple list
whos - detailed list

Clearing variables and memory
\[ \text{clear y} \]

Recalling last command
Press up arrow
<enter first letter of command>., press up arrow

Adding vectors
\[ z = x + x \]
\[ z = z - x \]

Transposing vector or array
\[ z = z' \]

Multiplying vectors
\[ w = x \times z \]
\[ v = x \times z \], this will produce an error

Multiplying vectors and arrays
\[ u = y \times z \]

Clear all variables
\[
\text{clear all}
\]

**Basic Matrix Operations**
Enter the following matrix
\[
A = \begin{bmatrix} 16 & 3 & 2 & 13; \\ 5 & 10 & 11 & 8; \\ 9 & 6 & 7 & 12; \\ 4 & 15 & 14 & 1 \end{bmatrix}
\]

Calculate the sum of the matrix
\[
\text{sum}(A)
\]

Transpose the matrix, find the sum and transpose the sum in one step
\[
\text{sum}(A')'
\]

Return the diagonals of A
\[
\text{diag}(A)
\]

Sum the diagonals of A
\[
\text{sum(diag}(A)\text{))}
\]

Using subscripts
\[
A = \\
\begin{bmatrix} 16 & 3 & 2 & 13 \\ 5 & 10 & 11 & 8 \\ 9 & 6 & 7 & 12 \\ 4 & 15 & 14 & 1 \end{bmatrix}
\]

Find the sum of the 4th column
\[
A(1, 4) + A(2, 4) + A(3, 4) + A(4, 4)
\]

What did you tell it to do? 13 + 8 + 12 + 1

Using the colon operator
\[
\text{sum}(A(1:4,4))
\]

**Important Matrices**
Zeros and Ones – Space holding
\[
B = \text{zeros}(4)
\]
\[
B = \text{ones}(3)
\]
\[
B = \text{ones}(3, 2) \text{ – non-square}
\]

**Concatenation**
Combining two or more matrices or vectors into one.
\[
B = [A \ A+32; A+48 \ A+16]
\]
\[
w = [x \ z']
\]

Adding new data to a matrix
\[
w(2,4) = 6
\]

Deleting row and columns from a matrix
\[
X = A
\]
\[
X(:, 2) = []
\]

\[
X(1, 2) = [] \text{ – This is not allowed}
\]

Instead do the following:
\[
X(1, 2) = 0
\]

\[
\text{– or –}
\]
\[
X(1, 2) = \text{NaN} \text{ – NaN is a reserved word for Not a Number.}
\]

**Linear Algebra**
A matrix represents a linear transformation between spaces.

\[
A + A' \text{ – yields a symmetric matrix}
\]
\[
A' \times A \text{ – yields a symmetric matrix}
\]

Finding a determinant of a matrix
\[
d = \text{det}(A)
\]

Reducing the matrix to row echelon form
\[
r = \text{rref}(A)
\]

Finding the inverse of a matrix
\[
i = \text{inv}(A) \text{ – Sometimes it doesn’t work too well}
\]

Finding the eigenvalues
\[
e = \text{eig}(A)
\]
Basic Plotting In MATLAB

Creating some data to plot
\[ x = 0:\pi/100:2*\pi; \]
\[ y = \sin(x); \]

Generate a plot
\[ \text{plot}(x, y) \]

Now label the axes and add a title. The characters \pi creates the symbol for pi. You can use a subset of TeX commands embedded in the string to produce special characters such as Greek letters and mathematical symbols

Label the X and Y axes
\[ \text{xlabel}('x = 0:2\pi') \]
\[ \text{ylabel}('\text{Sine of x}') \]

Label the plot
\[ \text{title}('\text{Plot of the Sine Function}', 'FontSize', 12) \]
\[ \text{text}(1,-1/3,'\text{\it Note the odd symmetry.}') \]

Plotting multiple data sets on one figure
\[ y2 = \sin(x-.25); \]
\[ y3 = \sin(x-.5); \]
\[ \text{plot}(x, y, y2, x, y3) \]
- or -
\[ \text{plot}(x, y, 'b') \]
\[ \text{hold on} \]
\[ \text{plot}(x, y2, 'g') \]
\[ \text{plot}(x, y3, 'r') \]

Creating a legend of your data
\[ \text{legend}('\sin(x)', '\sin(x-.25)', '\sin(x-.5)') \]

Color strings are 'c', 'm', 'y', 'r', 'g', 'b', 'w', and 'k'. These correspond to cyan, magenta, yellow, red, green, blue, white, and black.

Line style strings are '-' for solid, '--' for dashed, ':' for dotted, '-' for dash-dot. Omit the line style for no line.
The marker types are '+' , 'o', '*', and 'x', and the filled marker types are 's' for square, 'd' for diamond, '^' for up triangle, 'v' for down triangle, '>' for right triangle, '<' for left triangle, 'p' for pentagram, 'h' for hexagram, and none for no marker.

Try it!
\[ \text{plot}(x, y3, 'r-.d') \] – creates a red dash-dotted line with red diamonds on the data points.

Looks a little crowded
\[ x2 = 0:pi/10:2*pi; \]
\[ \text{plot}(x, \sin(x), 'r:', x2, \sin(x2), 'rd') \]

Creating multiple plots on one figure
\[ [X, Y, Z] = \text{cylinder}(4*\cos(x2)); \]
\[ \text{subplot}(2, 2, 1); \text{mesh}(X) \]
\[ \text{subplot}(2, 2, 2); \text{mesh}(Y) \]
\[ \text{subplot}(2, 2, 3); \text{mesh}(Z) \]
\[ \text{subplot}(2, 2, 4); \text{mesh}(X, Y, Z) \]

Setting the Axes limits
\[ \text{axis}([\text{xmin} \text{xmax} \text{ymin} \text{ymax}]) \]

Setting the Axis Aspect Ratio
\[ \text{axis equal} \]
\[ \text{axis tight} \]
\[ \text{axis square} \]

Grid lines
\[ \text{grid on} \] – toggle on
\[ \text{grid off} \] – toggle off

Saving figure
File -> Save As