Nuts & Bolts guide to MATLAB

Adrian KC Lee ScD

MGH-HST Athinoula A. Martinos Center for Biomedical Imaging; Department of Radiology, Harvard Medical School, Boston, MA.

November 18 2010 | Why.N.How Tutorial Series
Overview

‣ Why do we use MATLAB?
‣ Navigating in MATLAB environment.
‣ Matrix Operations; Cell array; Structure.
‣ General programming tips.
‣ Simple MATLAB functions.
‣ Error Flow / Debugging mode.
‣ Plots / Graphs.
Why MATLAB?

MATLAB = MATrix LABoratory

- Basic data element is the matrix;
- Vectorized operations;
- Good for generating engineering graphics.

- Specialized Toolboxes.
- Interacts with Inputs and Output devices.
MATLAB environment

- Current directory
- Variables currently in Workspace
- Command prompt
- Command History
Matrix Manipulations

- **Matrix size: row x column**
  - Basic data element is the matrix;

- **Vectors: a degenerate matrix**
  - **Row Vector = matrix with only 1 row**
    - Separator used: , or <space>
  - **Column Vector = matrix with only 1 column**
    - Separator used: ; or <(Carriage) Return>
Matrix Manipulations (2)

- Element in matrix can be accessed:
  - $A(\text{row, column}) = \text{new\_value}$;

- Matrix Transpose
  - e.g., $A'$ (Note: Hermitian conjugate for Complex Field)
    - Separator used: , or <space>

- Sub-matrix:
  - Range specifies by <start>:<end>
  - Entire dimension selected by :
Matrix Manipulations (3)

- Math functions operate on matrices:
  - Beware of commands used
    - e.g., * vs .* (matrix vs array multiply)
  - Reminder: Matrices are not commutative in multiplication, i.e., $A*B \neq B*A$

- Solving system of linear equations:
  - $Ax = b \Rightarrow x = \text{inv}(A)*b$ or $x = A\backslash b$

- More math functions...
Array vs. Cell

- Array: dimensions and types are consistent

```matlab
B_array = ['abc','d';'ef','gh'];
B_error = ['abc';'defgh'];
B_mix = [50,'a';'c',70]
char(50) = "2" and char(70) = "F"
B_cell = {50,'a';'c',70}
```

```matlab
[50] 'a'
= 'c' [70]
```
### Array of Structures

- **Different data fields for each subject**

  ```matlab
data = {'KC', 30, 10; ...
      'Adrian', 25, 23; ...
      'Lee', 60, 50};

NAME = data(:,1)
AGE = cell2mat(data(:,2))
```

- **Array of structures**

  ```matlab
subject(1).name = 'KC';
subject(1).age = 30;
subject(1).mental_age = 10;
subject(1).HDR = S01;

subject(2).name = 'AA';
subject(2).age = 25;
subject(2).mental_age = 23;
subject(2).HDR = S02;
```

**Warning:** only an illustration. Not HIPPA kosher!
Control Statements

- Conditional:
  - Syntax

  ```
  if <expression>
      <statements>
  (elseif <expression>
      <statements>)
  else
      <statements>
  end
  ```
Control Statements (2)

Repetition:

• Looping with **while**

```plaintext
while <expression>
  <statements>
end
```

• Looping with **for**

```plaintext
for var = start:step:end
  <statements>
end
```
General Computing Tips

- Divide and Conquer.
- Comments (both header and body).
- Choose obvious (and not reserved) variable names.
- Keep trace of different editions.
- Concept of “Flags.”
- Control flow of program (error messages).
MATLAB functions

‣ Function declaration:

```matlab
function (output args) = ...

<function name> (input args)
```

‣ Save as an m-file (.m) using the **exact** function name for the filename.

‣ Make sure that the path is added to the workspace to call function.

factorial(2).m, comb(2).m
DEBUG mode

- Debug > Stop if Errors / Warnings >
  - Access to variables in workspace
  - Step / Step In / Step Out to trace loop
  - (if no MATLAB errors, make one yourself to have access to DEBUG mode for tracing)
  - Use `dbquit` to exit debug mode
MATLAB plots

- Basic 2-D MATLAB plotting command
  - \texttt{plot(x,y,s)}
  - \texttt{x} and \texttt{y} must be the same size
  - \texttt{s} is a string that can be used to denote color, symbol and line-type of the plot.
  - See also \texttt{semilogx}, \texttt{semilogy}, \texttt{loglog}, \texttt{polar}, \texttt{fill}, \texttt{bar}, \texttt{errorbar}, \texttt{hist}, \texttt{plotyy}, \texttt{area}, \texttt{pie}, \texttt{stem}, \texttt{stairs} etc.
Plotting (time domain)

- Define time vectors to generate tones and plots:
  - Syntax: `start:step:finish`
  - `close all hidden`, `hold`
  - `figure` creates a new figure.
  - `subplot`, `xlabel`, `ylabel`, `title`, `axis`, `strcat`, `stem`
References & other tips

- www.mathwork.com/matlabcentral
  - File Exchange
    - Shared programs (e.g., errorbarxy)
  - MATLAB helpdesk
- Linear Algebra Tutorial
  - MIT OCW 18.06 Linear Algebra