Lab of COMP 406

Introduction of Matlab (I)

Teaching Assistant: Pei-Yuan Zhou
Contact: cspyzhou@comp.polyu.edu.hk
Lab 1: 12 Sep., 2014
Outline

- Matlab Description
  - Where is Matlab
  - What is Matlab
  - Matlab Desktop
  - MATLAB Math Operators

- Matrix and Array
  - Creation, Operation, Concatenation

- Workspace Variables

- Calling Functions
Where is Matlab?

• Find the Matlab under the folder
  – 1. Y:\Win32\Matlab\R2012a
  – 2. Double click it and open Matlab

• Or open Matlab on your computer
  – 1. Click 'Start'
  – 2. Click 'Run'
  – 3. Input 'nalwin32'
  – 4. Find the Matlab under the folder /Network Application Packages/Statistical & Mathematical/Matlab

• Send shortcut to your folder, for example: J:\starry

  *If you have any problem, please contact the technical staffs in PQ608. They are very nice and helpful.*
What is MATLAB?

- **MATLAB**
  - **MATrix LABoratory:** MATLAB is a program for doing numerical computation. It was originally designed for solving linear algebra type problems using matrices. It’s name is derived from **MATrix LABoratory**.
  - MATLAB has since been expanded and now has built-in functions for solving problems requiring data analysis, signal processing, optimization, and several other types of **scientific computations**. It also contains functions for 2-D and 3-D graphics and animation.
MATLAB Description

- Considering MATLAB at home
  - Standard edition
    - Available for roughly 2 thousand dollars
  - Student edition
    - Available for roughly 1 hundred dollars.
    - Some limitations, such as the allowable size of a matrix

http://www.mathworks.com/products/matlab/
Matlab Desktop (Cont.)

- **Launch Pad**
- **Workspace**
- **Current Directory**
- **Command Window**
- **History**

```
>> A = eye(4, 4)

A =

    1     0     0     0
    0     1     0     0
    0     0     1     0
    0     0     0     1
```
• Current Folder
  – Access your files

• Command Window
  – Enter commands at the command line, indicated by the prompt (>>).

• Workspace
  – view program variables
  – double click on a variable to see it in the Array Editor
  – Explore data that you create or import from files

• Command History
  – View or rerun commands that you entered at the command line

• Launch Pad
  – access help, tools, demos and documentation
The MATLAB environment is command oriented somewhat like UNIX. A prompt (>>) appears on the screen and a MATLAB statement can be entered. When the <ENTER> key is pressed, the statement is executed, and another prompt appears.

If a statement is terminated with a semicolon (;), no results will be displayed. Otherwise results will appear before the next prompt.

```matlab
» a=5;
» b=a/2

b =

2.5000
»
```
MATLAB Math Operators

Power
^ or .^ a^b or a.^b

Multiplication
* (matrix multiply) or .* (array multiply)
a*b or a.*b

Division
/ or ./ a/b or a./b
or
\ or \.\ b\a or b.\a

NOTE: 56/8 = 8\56

Addition
+ a + b

Subtraction
- a - b

Assignment
= a = b (assign b to a)

cos () Cosine value of a specific angle

sin() Sine value of a specific angle
Ex1

• As you work in MATLAB, you issue commands that create variables and call functions. For example, create a variable named a by typing this statement at the command line:

   \[ a = 1 \]

• MATLAB adds variable a to the workspace and displays the result in the Command Window.

• Try more:

   \[ b = 2 \]
   \[ c = a + b \]
   \[ d = \cos(a) \]
When you do not specify an output variable, MATLAB uses the variable ans, short for *answer*, to store the results of your calculation.

```
>> sin(a)
ans =
0.8415
```

If you end a statement with a semicolon, MATLAB performs the computation, but suppresses the display of output in the Command Window.

```
>> e = a*b;
```
Outline

- Matlab Description
  - Where is Matlab
  - What is Matlab
  - Matlab Desktop
  - MATLAB Math Operators

- Matrix and Array
  - Creation, Operation, Concatenation

- Workspace Variables

- Calling Functions
• MATLAB is an abbreviation for “matrix laboratory”. While other programming languages mostly work with numbers one at a time, MATLAB is designed to operate primarily on whole matrices and arrays.

• All MATLAB variables are multidimensional arrays, no matter what type of data. A matrix is a two-dimensional array often used for linear algebra.
Creation : Array / Matrix

• To create an array with four elements in a single row, separate the elements with either a comma (,) or a space.

```matlab
>> a = [1 2 3 4] / a=[1,2,3,4]
a =
   1 2 3 4
```

• This type of array is a row vector.

• To create a matrix that has multiple rows, separate the rows with semicolons.

```matlab
>> a = [1 2 3; 4 5 6; 7 8 10]
```
Another way to create a matrix is to use a function, such as `ones`, `zeros`, or `rand`. For example, create a 5-by-1 column vector of zeros.

```matlab
>> z = zeros(5,1)
```

```
z =
    0
    0
    0
    0
    0
```
Matrix Operations

• MATLAB allows you to process all of the values in a matrix using a single arithmetic operator or function.

• Practice more:

  \[
  \triangleright a + 10 \\
  \triangleright \sin(a)
  \]

• To transpose a matrix, use a single quote ('):

  \[
  \triangleright a'
  \]
Matrix Operations

• You can perform standard matrix multiplication, which computes the inner products between rows and columns, using the * operator. For example, confirm that a matrix times its inverse returns the identity matrix:

```
>> a = [1 2 3; 4 5 6; 7 8 10]
>> p = a*inv(a)
```

```
p=

1.0000   0    -0.0000
 0 1.0000   0
 0    0 1.0000
```
Matrix Operations

- Notice that p is not a matrix of integer values. MATLAB stores numbers as floating-point values, and arithmetic operations are sensitive to small differences between the actual value and its floating-point representation. You can display more decimal digits using the format command:

```
>> format long %16-bits
>> p = a*inv(a)
```

```
p=
  1.0000000000000000 0 -0.0000000000000000
  0 1.0000000000000000 0
  0 0 0.9999999999999998
```
Matrix Operations

- Reset the display to the shorter format using `format short`
- `format` affects only the display of numbers, not the way MATLAB computes or saves them.
- To perform **element-wise multiplication** rather than matrix multiplication, use the `.*` operator:

<table>
<thead>
<tr>
<th>p=a*a</th>
<th>q=a.*a</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 36 45</td>
<td>1 4 9</td>
</tr>
<tr>
<td>66 81 102</td>
<td>16 25 36</td>
</tr>
<tr>
<td>109 134 169</td>
<td>49 64 100</td>
</tr>
</tbody>
</table>
Matrix Operations

• The matrix operators for *multiplication, division, and power* each have a corresponding array operator that operates element-wise. For example, raise each element of $a$ to the third power:

\[
p = a^3 = a \times a \times a
\]

\[
q = a.^3
\]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>489</td>
<td>600</td>
<td>756</td>
</tr>
<tr>
<td>1104</td>
<td>1353</td>
<td>1704</td>
</tr>
<tr>
<td>1828</td>
<td>2240</td>
<td>2821</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>27</td>
</tr>
<tr>
<td>64</td>
<td>125</td>
<td>216</td>
</tr>
<tr>
<td>343</td>
<td>512</td>
<td>1000</td>
</tr>
</tbody>
</table>
Concatenation

- *Concatenation* is the process of joining arrays to make larger ones. In fact, you made your first array by concatenating its individual elements. The pair of square brackets [] is the concatenation operator.

\[
\text{>>A = [a,a]}
\]

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 1 & 2 & 3 \\
4 & 5 & 6 & 4 & 5 & 6 \\
7 & 8 & 10 & 7 & 8 & 10 \\
\end{array}
\]
Concatenation

- Concatenating arrays next to one another using commas is called horizontal concatenation. Each array must have the same number of rows. Similarly, when the arrays have the same number of columns, you can concatenate vertically using semicolons.

\[ A = [a; a] \]

\[
A = \\
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 10 \\
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 10
\]
Outline

- Matlab Description
  - Where is Matlab
  - What is Matlab
  - Matlab Desktop
  - MATLAB Math Operators
- Matrix and Array
  - Creation, Operation, Concatenation
- Workspace Variables
- Calling Functions
Workspace Variables

- The workspace contains variables that you create within or import into MATLAB from data files or other programs. For example, these statements create variables A and B in the workspace.

\[
A = \text{magic}(4);
B = \text{rand}(3,5,2);
\]

- You can view the contents of the workspace using whos.

```
whos
```

- The variables also appear in the Workspace pane on the desktop.
Workspace Variables

- Workspace variables do not persist after you exit MATLAB. Save your data for later use with the `save` command,

  ```matlab
  save myfile.mat
  ```

- Saving preserves the workspace in your current working folder in a compressed file with a `.mat` extension, called a MAT-file.

- To clear all the variables from the workspace, use the `clear` command.

- Restore data from a MAT-file into the workspace using `load`.

  ```matlab
  load myfile.mat
  ```
Exercise

\[ a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{bmatrix}; \]
\[ z = \text{zeros}(5,1); \]
\[ A = \text{magic}(4); \]
\[ B = \text{rand}(3,5,2); \]

\text{whos}

\text{save lab1.mat}

\text{load lab1.mat}
Outline

- Matlab Description
  - Where is Matlab
  - What is Matlab
  - Matlab Desktop
  - MATLAB Math Operators

- Matrix and Array
  - Creation, Operation, Concatenation

- Workspace Variables

- Calling Functions
Calling Functions

- MATLAB provides a large number of functions that perform computational tasks. Functions are equivalent to subroutines or methods in other programming languages.

- Suppose that your workspace includes variables A and B, such as

  \[
  A = [1 \ 3 \ 5]; \\
  B = [10 \ 6 \ 4];
  \]

- To call a function, enclose its input arguments in parentheses:

  \[
  \text{max}(A) \\
  \text{ans} = 5
  \]
Calling Functions

• If there are multiple input arguments, separate them with commas:

\[
\text{max}(A,B) \\
\text{ans} = \\
10 \quad 6 \quad 5
\]

• Return output from a function by assigning it to a variable:

\[
\text{maxA} = \text{max}(A);
\]

• When there are multiple output arguments, enclose them in square brackets:

\[
[\text{maxA},\text{location}] = \text{max}(A);
\]
• Enclose any character string inputs in single quotes:
  
  \texttt{disp ('hello world');}

• To call a function that does not require any inputs and does not return any outputs, type only the function name:

  \texttt{clc}

• The \texttt{clc} function clears the Command Window.

• \texttt{clear} removes all variables from the workspace.
Help and Documentation

• All MATLAB functions have supporting documentation that includes examples and describes the function inputs, outputs, and calling syntax.
• There are several ways to access this information from the command line:
  • Open the function documentation in a separate window using the doc command.
    
    \[
    \text{doc mean}
    \]
  • View an abbreviated text version of the function documentation in the Command Window using the help command.
    
    \[
    \text{help mean}
    \]
Exercises

• >> a = ones(2, 4)
• >> a = eye(3)
• >> cos(pi)
• >> sin(pi)
• >> 0/0
• >> 1/0
• >> sqrt(36)
• >> A = [1, 2; 3, 4];
• >> B = [-1, -2; 2,1];
• >> S = 3;
• >> A.*B
• >> A*B
• >> S.*A
• >> S*B
More Exercises

• How to assign \[
\begin{bmatrix}
2 & 5 & 3 \\
7 & 3 & 5 \\
9 & 1 & 2
\end{bmatrix}
\] to a variable named \(a\)? And what’s the transpose of \(a\)?

• What’s the summation of \[
\begin{bmatrix}
2 & 5 & 3 \\
7 & 3 & 5 \\
9 & 1 & 2
\end{bmatrix}
\] and \[
\begin{bmatrix}
2 & 7 & 3 \\
7 & 10 & 5 \\
4 & 1 & 15
\end{bmatrix}
\]?

• Type in \(1.8 == 1.8000000000000000000000000000001\) (24 zeros in total, see what happen)
What We have Learned?

1. Matlab Description and Desktop Basics
2. Matrices
3. Workspace Variables
4. * Character Strings
5. Calling Functions
Thank You

Q & A