Introduction to Matlab

Weichung Wang

2003 NCTS-NSF Workshop on Differential Equations,
Surface Theory, and Mathematical Visualization

NCTS, Hsinchu, February 13, 2003
Main Goals

- A brief Matlab introduction for beginners
- Warm-up for Prof. Karcher’s courses
Start Matlab in Windows
Start Matlab in Unix shell

➤ In command line window, type

> matlab
> matlab -nojvm
What is Matlab?

MATLAB == MATrix LABoratory

Mathworks: http://www.mathworks.com

Major software characteristics:
- matrix-based numeric computation
- high-level programming language
- graphics & visualization
- toolboxes provide application-specific functionality
What Is Matlab? (Cont.)

Multi-platform support (PC / Macintosh / UNIX)

Interfaces to other systems.
- Custom C, Fortran (MATLAB is callable)
- Extensive data I/O facility

Matlab is case sensitive (mtxA ~= MTXA)
As a Calculator

\begin{verbatim}
>> 2 + 6 - 4
ans =
 4

>> ans/2
ans =
 2
\end{verbatim}

\begin{verbatim}
>> a = 5
a =
 5

>> b = 6
b =
 6

>> c = b/a
c =
 1.2000
\end{verbatim}
Built-in Variables

\[ \pi (\equiv \pi) \text{ and } \text{ans are a built-in variables} \]

\[
\begin{align*}
\gg \pi \\
\text{ans} &= \\
3.1416
\end{align*}
\]

\[
\begin{align*}
\gg \sin(\text{ans}/4) \\
\text{ans} &= \\
0.7071
\end{align*}
\]
Built-in Functions

>> log(256)
ans =
5.5452

>> log10(256)
ans =
2.4082

>> log2(256)
ans =
8
Looking for Functions

>> lookfor cosine

produces

ACOS    Inverse cosine.
ACOSH   Inverse hyperbolic cosine.
COS     Cosine.
COSH    Hyperbolic cosine.
>> help log

produces

LOG    Natural logarithm.
LOG(X) is the natural logarithm of the elements of X. Complex results are produced if X is not positive.

See also LOG2, LOG10, EXP, LOGM.
Suppress Output With Semicolon

```
>> x = 5;
>> y = sqrt(59);
>> z = log(y) + x^0.25
z =
    3.5341
```
Multiple Statements Per Line

```
>> a = 5;    b = sin(a),  c = cosh(a)
b =
   -0.9589

c =
   74.2099
```
Vectors and Matrices

\[ \text{v} = [7 \ 3 \ 9] \]

\[ \text{w} = [2; 6; 1] \]

\[ \text{v'} \]

\[ \text{ans} = [2 \ 4 \ 1 \ 7] \]

\[ \text{A} = [1 \ 2 \ 3; \ 5 \ 7 \ 11; \ 13 \ 17 \ 19] \]
Matrix

\[
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{bmatrix}
\]

\[
\begin{align*}
\text{>> } & A = \begin{bmatrix} 1 & 2 & 3; 4 & 5 & 6; 7 & 8 & 9 \end{bmatrix} \\
\text{>> } & b = A(3,2) \\
& b = 8 \\
\text{>> } & c = A(1,1) \\
& c = 1 \\
\text{>> } & A(1,1) = c/b \\
& A = \\
& \begin{bmatrix}
0.2500 & 2.0000 & 3.0000 \\
4.0000 & 5.0000 & 6.0000 \\
7.0000 & 8.0000 & 9.0000
\end{bmatrix}
\end{align*}
\]
Colon Notation

```matlab
>> s = 1:4
s =
    1    2    3    4

>> t = 0:0.1:0.4
 t =
    0   0.1000   0.2000   0.3000   0.4000
```
Colon Notation (Cont.)

```matlab
>> A = [1 2 3; 4 5 6; 7 8 9];
>> A(:,1)
an =
    1
    4
    7

>> A(2,:)
an =
    4
    5
    6

>> A(2:3,1)
an =
    4
    7

>> A(1:2,2:3)
an =
an =
    2
    3
    5
    6
```
Colon Notation (Cont.)

```matlab
>> A = ones(8,8);
>> A(3:6,3:6) = zeros(4,4)
A =
    1     1     1     1     1     1     1     1     1     1
    1     1     1     1     1     1     1     1     1     1
    1     1     0     0     0     0     0     1     1     1
    1     1     0     0     0     0     0     1     1     1
    1     1     0     0     0     0     0     1     1     1
    1     1     0     0     0     0     0     1     1     1
    1     1     1     1     1     1     1     1     1     1
    1     1     1     1     1     1     1     1     1     1
```
Complex Numbers

```
>> sqrt(-4)
an =
   0 + 2.0000i

>> x = 1 + 2*i
x =
   1.0000 + 2.0000i

>> y = 1 - 2*i
y =
   1.0000 - 2.0000i

>> z = x*y
z =
   5
```
(or, \( x = 1 + 2j \))
## Complex Number (Cont.)

<table>
<thead>
<tr>
<th>Function</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs</td>
<td>Compute the magnitude of a number</td>
</tr>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td>angle</td>
<td>Angle of complex number in Euler notation</td>
</tr>
<tr>
<td>exp</td>
<td>If $x$ is real, $e^x$</td>
</tr>
<tr>
<td></td>
<td>If $z$ is complex, $e^{\text{Re}(z)}(\cos(\text{Im}(z)) + i\sin(\text{Im}(z)))$</td>
</tr>
<tr>
<td>conj</td>
<td>Complex conjugate of a number</td>
</tr>
<tr>
<td>imag</td>
<td>Extract the imaginary part of a complex number</td>
</tr>
<tr>
<td>real</td>
<td>Extract the real part of a complex number</td>
</tr>
</tbody>
</table>
Complex Number (Cont.)

\[ z = \zeta e^{i\theta} \]

\[ x = \Re(z) = |z| \cos(\theta) = \zeta \cos(\theta) \]

\[ y = i \Im(z) = i|z| \sin(\theta) = i\zeta \sin(\theta) \]

\[ \zeta = 5; \quad \theta = \pi/3; \]
\[ z = zeta \times \exp(i \times theta); \]
\[ z = 2.5000 + 4.3301i \]

\[ \text{abs}(z) \]
\[ \text{ans} = \]
\[ 5 \]

\[ \sqrt{z \times \text{conj}(z)} \]
\[ \text{ans} = \]
\[ 5 \]

\[ \text{real}(z) \]
\[ x = \]
\[ 2.5000 \]

\[ \text{imag}(z) \]
\[ y = \]
\[ 4.3301 \]
Manipulation of Matrices and Vectors

\[
\begin{bmatrix}
10 & 9 & 8 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 2 & 3 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
11 \\
11 \\
11 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
10 & 20 & 30 \\
9 & 18 & 27 \\
8 & 16 & 24 \\
\end{bmatrix}
\]
Vectorization

```
>> x = 0:pi/4:pi
   (define a row vector)
   x =
       0    0.7854   1.5708   2.3562   3.1416

>> y = cos(x)
   (evaluate cosine of each x(i))
   y =
       1.0000    0.7071     0  -0.7071  -1.0000
```

```
dx = pi/4.0
do 10 i=1,5
    x(i) = (i-1)*dx
    y(i) = sin(x(i))
10 continue
```
Vectorization (Cont.)

\[
\text{>> A = pi*[ 1 2; 3 4]}
\]
\[
A = \\
\begin{bmatrix}
3.1416 & 6.2832 \\
9.4248 & 12.5664
\end{bmatrix}
\]

\[
\text{>> S = sin(A)}
\]
\[
S = \\
\begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix}
\]

\[
\text{>> B = A/2}
\]
\[
B = \\
\begin{bmatrix}
1.5708 & 3.1416 \\
4.7124 & 6.2832
\end{bmatrix}
\]

\[
\text{>> T = sin(B)}
\]
\[
T = \\
\begin{bmatrix}
1 & 0 \\
-1 & 0
\end{bmatrix}
\]
## Array Operators

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>.*</td>
<td>element-by-element multiplication</td>
</tr>
<tr>
<td>./</td>
<td>element-by-element “right” division</td>
</tr>
<tr>
<td>.\</td>
<td>element-by-element “left” division</td>
</tr>
<tr>
<td>.^</td>
<td>element-by-element exponentiation</td>
</tr>
</tbody>
</table>
Array Operators (Cont.)

```
>> u = [1 2 3];
>> v = [4 5 6];
>> w = u.*v
  w =
       4   10   18

>> x = u./v
  x =
       0.2500   0.4000   0.5000
```

(element-by-element product)

(element-by-element division)
Array Operators (Cont.)

\[
\begin{align*}
\text{>> } A &= [1 \ 2 \ 3 \ 4; 5 \ 6 \ 7 \ 8]; \\
\text{>> } B &= [8 \ 7 \ 6 \ 5; 4 \ 3 \ 2 \ 1]; \\
\text{>> } A.*B \\
\text{ans} &= \\
&= \begin{bmatrix}
8 & 14 & 18 & 20 \\
20 & 18 & 14 & 8
\end{bmatrix} \\
\text{>> } A*B' \\
\text{ans} &= \\
&= \begin{bmatrix}
60 & 20 \\
164 & 60
\end{bmatrix} \\
\text{>> } A.^2 \\
\text{ans} &= \\
&= \begin{bmatrix}
1 & 4 & 9 & 16 \\
25 & 36 & 49 & 64
\end{bmatrix}
\end{align*}
\]

??? Error using ===> * 
Inner matrix dimensions must agree.
Plotting

```
>> x = linspace(0,3);
>> y = 10*exp(-2*x);
>> plot(x,y);
```
Plotting (Cont.)

```
plot(x,y,'kd--')
```

<table>
<thead>
<tr>
<th>Color</th>
<th>Symbol</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>point</td>
<td>solid</td>
</tr>
<tr>
<td>m</td>
<td>circle</td>
<td>dotted</td>
</tr>
<tr>
<td>c</td>
<td>x-mark</td>
<td>dashdot</td>
</tr>
<tr>
<td>r</td>
<td>plus</td>
<td>dashed</td>
</tr>
<tr>
<td>g</td>
<td>star</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>square</td>
<td></td>
</tr>
<tr>
<td>w</td>
<td>diamond</td>
<td></td>
</tr>
<tr>
<td>k</td>
<td>triangle(down)</td>
<td></td>
</tr>
</tbody>
</table>
Case Study

Examples by Prof. Hermann Karcher

Set 1: FirstConventions.m (script)

Change current directory to the directory containing the file FirstConventions.m
Give Orders by Command, Script, and Function

_script_

- Standard ASCII text files containing a sequence of normal MATLAB commands/statements.

- The command **PlotSine** causes the statements in the file named PlotSine.m to be parsed & executed in order. (Interpreter, not compiler.)

- % Comments start with "%" character

- Variables in a script file are **global**.
Function

- Function files provide extensibility to MATLAB.

- Usually contains input and output.

- Variables in a function file are by default local.

- However, you can declare a variable to be global.
Flow Control Constructs

Logic Control: if/switch

```matlab
if I == J
    A(I,J) = 2;
elseif abs(I-J) == 1
    A(I,J) = -1;
else
    A(I,J) = 0;
end

switch algorithm
    case 'ode23'
        str = '2nd/3rd order';
    case {'ode15s', 'ode23s'}
        str = 'stiff system';
    otherwise
        str = 'other algorithm';
end
```
Flow Control Constructs (cont.)

Iterative Loops: for / while

\[ N = 10; \]
\[ \text{for } I = 1:N \]
\[ \quad \text{for } J = 1:N \]
\[ \quad \quad A(I,J) = 1/(I+J-1); \]
\[ \quad \text{end} \]
\[ \text{end} \]
function y = mean(x)
% MEAN Average or mean value.
% For vectors, MEAN(x) returns the mean value.
% For matrices, MEAN(x) is a row vector
% containing the mean value of each column.
[m,n] = size(x);
if m == 1
    m = n;
end
y = sum(x)/m;
end
Workspaces or Stacks in MATLAB

MATLAB (or Base) Workspace:
For command line and script file variables.

Function Workspaces:
Each function has its own workspace for local variables.
Communicate to Function Workspace via inputs & outputs.
(Promotes structured coding & prevents name conflicts.)

Global Workspace:
Global variables can be shared by multiple workspaces.
(Must be initialized in all relevant workspaces.)
Initialize global variables in all relevant workspaces:

```matlab
» global variable_name
```

Initialize global variables in the “source” workspace before referring to them from other workspaces.
Case Studies

Examples by Prof. Hermann Karcher

Set 2: ConCsqrt.m and Csqrt.m (both functions)
Some Suggestions

Write comments

```matlab
% s=1 if scalar product(w,last) >=0  else s=-1
s=2*(real(w.*conj(last))>=0)-1;
```
Some Suggestions (Cont.)

- Give meaningful variable names and avoid single character variable names

```matlab
a = b * c;
f = m * a;

force = mass * acceleration;
for ii = 1:3
    moment = ii^2;
end;
```
Some Suggestions (Cont.)

- Indent codes
- Use Emacs
  (http://www.gnu.org/software/emacs/emacs.html)
- Learn by doing!
If You Were Sleeping…

It’s OK. Just remember two commands:

```plaintext
>> lookfor keyword
```

```plaintext
>> help command
```
Happy coding!

:-)
Reference

鈦思科技股份有限公司 Matlab 簡介投影片

Slides by Gerald Recktenwald