Overview, MATLAB Syntax

**Computation**

1 + 2 = 3

\[
\begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix}
\begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix} =
\begin{bmatrix}
7 & 10 \\
15 & 22
\end{bmatrix}
\]

\[\sin(30) = 0.5\]

**Programming**

X = 0

\[X = X + 0.1\]

\[X + 3 < 0.1\]

Output x

No → Yes

Cite as: Peter So, course materials for 2.003J / 1.053J Dynamics and Control I, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
Basic MATLAB functions

MATLAB Help Window

Info Window

MATLAB Tool Boxes

Courtesy of The MathWorks, Inc. Used with permission.

Cite as: Peter So, course materials for 2.003J / 1.053J Dynamics and Control I, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
MATLAB Data Types

- Logical
- Character
- Numeric
  - Integers: int8, uint8, int16, uint16, int32, uint32, int64, uint64
  - Floating point: single, double
- Cell array
- Structure
- Function handle
- User classes
- Java classes
MATLAB Data Structure

Everything in MATLAB are matrix!

A=5; A is a 1x1 matrix
A=[ 1 3 4 5]; A is a 1x4 matrix
A = [12 62 93 -8 22; 16 2 87 43 91; -4 17 -72 95 6]; A is a 5x3 matrix

Note: (1) Putting “;” behind a statement suppresses output
(2) Rows in a matrix is separated by “;” inside
(3) MATLAB command “whos” gives all the defined variables

What is a variable?
Matrix indexing: A(2,3) = 87

Cite as: Peter So, course materials for 2.003J / 1.053J Dynamics and Control I, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
Basic Linear Algebra

\[ m \cdot \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} ma & mc \\ mb & md \end{bmatrix} \]

\[ \begin{bmatrix} a & c \\ b & d \end{bmatrix} \pm \begin{bmatrix} e & g \\ f & h \end{bmatrix} = \begin{bmatrix} a \pm b & c \pm g \\ b \pm f & d \pm h \end{bmatrix} \]

\[ \begin{bmatrix} a & c \\ b & d \end{bmatrix} \cdot \begin{bmatrix} e & g \\ f & h \end{bmatrix} = \begin{bmatrix} ae + cf & ag + ch \\ be + df & bg + dh \end{bmatrix} \]

\[ A = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \]

\[ A \cdot A^{-1} = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]

\[ A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -c \\ -b & a \end{bmatrix} \]

\[ \begin{bmatrix} a & b & c \end{bmatrix}' = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \]

Scalar multi

Add/sub

Matrix multi

Inverse

Transpose
### MATLAB Operators – numeric

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Addition</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
</tr>
<tr>
<td>.*</td>
<td>Multiplication</td>
</tr>
<tr>
<td>./</td>
<td>Right division</td>
</tr>
<tr>
<td>.\</td>
<td>Left division</td>
</tr>
<tr>
<td>+</td>
<td>Unary plus</td>
</tr>
<tr>
<td>-</td>
<td>Unary minus</td>
</tr>
<tr>
<td>:</td>
<td>Colon operator</td>
</tr>
<tr>
<td>.^</td>
<td>Power</td>
</tr>
<tr>
<td>.'</td>
<td>Transpose</td>
</tr>
<tr>
<td>.'</td>
<td>Complex conjugate transpose</td>
</tr>
<tr>
<td>*</td>
<td>Matrix multiplication</td>
</tr>
<tr>
<td>/</td>
<td>Matrix right division</td>
</tr>
<tr>
<td>\</td>
<td>Matrix left division</td>
</tr>
<tr>
<td>^</td>
<td>Matrix power</td>
</tr>
</tbody>
</table>

\[
A = \begin{bmatrix} 1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad C = \begin{bmatrix} 5 & 6 \end{bmatrix}
\]

\[
A + C = \begin{bmatrix} 6 & 8 \end{bmatrix}
\]

\[
A \times C \quad \text{bad!}
\]

\[
A \times B = 11
\]

\[
A \times C = \begin{bmatrix} 5 & 12 \end{bmatrix}
\]

\[
A \div C = \begin{bmatrix} 0.200 & 0.333 \end{bmatrix}
\]

\[
A \div C = \begin{bmatrix} 5 & 3 \end{bmatrix}
\]
# MATLAB Operators – Relational, Logical

## Relational

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;</code></td>
<td>Less than</td>
</tr>
<tr>
<td><code>&lt;=</code></td>
<td>Less than or equal to</td>
</tr>
<tr>
<td><code>&gt;</code></td>
<td>Greater than</td>
</tr>
<tr>
<td><code>&gt;=</code></td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td><code>==</code></td>
<td>Equal to</td>
</tr>
<tr>
<td><code>~=</code></td>
<td>Not equal to</td>
</tr>
</tbody>
</table>

## Logical

```matlab
A = [0 1 1 0 1];
B = [1 1 0 0 1];
```

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&amp;</code></td>
<td>Returns 1 for every element location that is <code>true</code> (nonzero) in both arrays, and 0 for all other elements.</td>
<td>( A &amp; B = 01001 )</td>
</tr>
<tr>
<td>`</td>
<td>`</td>
<td>Returns 1 for every element location that is <code>true</code> (nonzero) in either one or the other, or both arrays, and 0 for all other elements.</td>
</tr>
<tr>
<td><code>~</code></td>
<td>Complements each element of the input array, ( A ).</td>
<td>( \sim A = 10010 )</td>
</tr>
<tr>
<td><code>xor</code></td>
<td>Returns 1 for every element location that is <code>true</code> (nonzero) in only one array, and 0 for all other elements.</td>
<td>( \text{xor}(A,B)=10100 )</td>
</tr>
</tbody>
</table>

Cite as: Peter So, course materials for 2.003J / 1.053J Dynamics and Control I, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
One more MATLAB Operator – Sequence
“:” is the sequence operator that denote a range

\[
\begin{align*}
A &= 2:5 \quad A = \begin{bmatrix} 2 & 3 & 4 & 5 \end{bmatrix} \\
A &= 2:3:11 \quad A = \begin{bmatrix} 2 & 5 & 8 & 11 \end{bmatrix} \\
A &= \begin{bmatrix} 1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \end{bmatrix} \\
B &= A(2,:) \quad B = \begin{bmatrix} 4 & 5 & 6 \end{bmatrix} \\
C &= A(:,2) \quad C = \begin{bmatrix} 2 \\
5 \\
8 \end{bmatrix} \\
D &= A(2:3,:) \quad D = \begin{bmatrix} 4 & 5 & 6 \\
7 & 8 & 9 \end{bmatrix} \\
A(:,2) &= [] \quad A = \begin{bmatrix} 1 & 3 \\
4 & 6 \\
7 & 9 \end{bmatrix}
\end{align*}
\]

It is very useful to create, decimate, and generate submatrix

Cite as: Peter So, course materials for 2.003J / 1.053J Dynamics and Control I, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
Basic Graphic Output in MATLAB

\[
X = [1 2 3 4 5 6 7 8 9 10]
\]

\[
Y = [1 4 9 16 25 36 49 64 81 100]
\]

\textit{plot}(X, Y)
t=1:1:100;
plot(t,cos(t/10));
A couple more very useful graphic commands

(1) hold on/hold off – determines whether the next plot command overwrites or not

(2) figure – Creates new figure window

(3) From the figure window, under “edit menu”, the “copy figure” option allows you to copy the figure to the clipboard and then you can cut and paste it into other programs such as MSWord.

Cite as: Peter So, course materials for 2.003J / 1.053J Dynamics and Control I, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
Programming

What is programming?
Programming is the preparation of a step-by-step instruction for a computer to follow.

When is programming “profitable”
- repetitive computation
- automation/real time control
- reusable “code” – objects

Programming languages
C, C++, C#, java, m-lab script
Anatomy of a program

Flow chart – a graphic representation of the logical sequence of instructions

Algorithm – a sequence of instructions designed to solve a specific problem

Action

X=0

Decision

X+3<0.1

Terminal

Start
Conditionals

Conditional is a branching point in the program. Depending on specific condition, the program can take different actions.

Example: a simple program that add 1 to odd integer input and do nothing to even integer input.

```
start
input X
rem(X,2)==0
yes
X=X
output x
end
No
X=X+1
output x
end
```
Programming in MATLAB

Step 1: Create a m-file (xxx.m)  
[MATLAB Menu: file->new]

Step 2: Input sequence of MATLAB instructions

Step 3: Save (in working directory) and run  
[Editor Menu:debug->save & run]
x = input('input integer: '); if (rem(x,2) == 0) x=x; else x=x+1; end

output x

end
Conditional: If, else, end

if logic condition
  action1;
  action1;
else
  action2;
  action2;
end

Check out also elseif
Repetition

Example: fill a 1-D matrix $A$ with length 10 with 2s.

```
start

Create A

i=1;

A(i)=2;

i=i+1;

i>10

no

yes

end
```

Cite as: Peter So, course materials for 2.003J / 1.053J Dynamics and Control I, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].