Matlab Programming
Introduction\textsuperscript{1} \textsuperscript{2}

Mili I. Shah

August 10, 2009

\textsuperscript{1}Matlab, An Introduction with Applications, 2\textsuperscript{nd} ed. by Amos Gilat
\textsuperscript{2}Matlab Guide, 2\textsuperscript{nd} ed. by D. J. Higham and N. J. Higham
Starting Matlab

Go to

http://www.loyola.edu/moresoftware/

and login with your Loyola name and password...

Matlab has eight main windows:

<table>
<thead>
<tr>
<th>Window</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Window</td>
<td>Main window, enter variables, runs programs</td>
</tr>
<tr>
<td>Figure Window</td>
<td>Contains output from graphic commands</td>
</tr>
<tr>
<td>Editor Window</td>
<td>Creates and debugs script and function files</td>
</tr>
<tr>
<td>Help Window</td>
<td>Provides help information</td>
</tr>
<tr>
<td>Launch Pad Window</td>
<td>Provides access to tools, demos, and documentation</td>
</tr>
<tr>
<td>Command History Window</td>
<td>Logs commands entered in the Command Window</td>
</tr>
<tr>
<td>Workspace Window</td>
<td>Provides information about the variables that are used</td>
</tr>
<tr>
<td>Current Directory Window</td>
<td>Shows the files in the current directory</td>
</tr>
</tbody>
</table>
To type a command the cursor must be placed next to the command prompt (>>).

Press **Enter** for the command to be executed. Multiple commands can be typed by typing a comma (,) between them.

A semicolon (;) at the end of a command suppresses the screen output.

Upper and lower case characters are not equivalent.

The up and down arrow keys can be used to scroll through previous commands. Also an old command can be recalled by typing the first few characters followed by the up arrow.

Type **help topic** to access online help on the command, function, or symbol topic.

Type **clc** to clear the screen

Type **exit** or **quit** to quit Matlab.
### Built-In Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sqrt(x)</code></td>
<td>Square root</td>
</tr>
<tr>
<td><code>exp(x)</code></td>
<td>Exponential (e^x)</td>
</tr>
<tr>
<td><code>abs(x)</code></td>
<td>Absolute value</td>
</tr>
<tr>
<td><code>log(x)</code></td>
<td>Natural logarithm</td>
</tr>
<tr>
<td><code>sin(x)</code></td>
<td>Sine of (x)</td>
</tr>
<tr>
<td><code>cos(x)</code></td>
<td>Cosine of (x)</td>
</tr>
<tr>
<td><code>tan(x)</code></td>
<td>Tangent of (x)</td>
</tr>
<tr>
<td><code>cot(x)</code></td>
<td>Cotangent of (x)</td>
</tr>
<tr>
<td><code>pi</code></td>
<td>(\pi)</td>
</tr>
</tbody>
</table>

Mili Shah  
MA304: Introduction
## Built-In Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqrt(x)</td>
<td>Square root</td>
</tr>
<tr>
<td>exp(x)</td>
<td>Exponential ( e^x )</td>
</tr>
<tr>
<td>abs(x)</td>
<td>Absolute value</td>
</tr>
<tr>
<td>log(x)</td>
<td>Natural logarithm</td>
</tr>
<tr>
<td>sin(x)</td>
<td>Sine of ( x )</td>
</tr>
<tr>
<td>cos(x)</td>
<td>Cosine of ( x )</td>
</tr>
<tr>
<td>tan(x)</td>
<td>Tangent of ( x )</td>
</tr>
<tr>
<td>cot(x)</td>
<td>Cotangent of ( x )</td>
</tr>
<tr>
<td>pi</td>
<td>( \pi )</td>
</tr>
</tbody>
</table>

```matlab
>> sqrt(4)
an = 2
```

```matlab
>> pi
ans =
3.1416
```
Defining Scalar Variables

Variable = Numerical value or computable expression

- = is the assignment operator which assigns a value to a variable
- Left-hand side can include only one variable name
- Right-hand side can be a number or an expression made up of numbers and/or variables previously assigned numerical values
- Variables must begin with a letter
- Press Enter to make the assignment
- ans is the value of the last expression that is not assigned

Remember:
- Use semicolon (;) to suppress screen output
- Multiple commands can be typed by typing a comma (,) between them.
Example: Assign the number 3 to variable $a$ and 4 to variable $b$. Print out $\sqrt{a^2 + b^2}$ and assign the solution to the variable $c$. 
**Example**: Assign the number 3 to variable \( a \) and 4 to variable \( b \). Print out \( \sqrt{a^2 + b^2} \) and assign the solution to the variable \( c \).

```plaintext
>> a=3; b=4; c = sqrt(a^2+b^2)
c = 5
```
Defining Scalar Variables

**Example**: Assign the number 3 to variable \(a\) and 4 to variable \(b\). Print out \(\sqrt{a^2 + b^2}\) and assign the solution to the variable \(c\).

\[
\text{c = } \sqrt{a^2 + b^2}
\]

```matlab
>> a=3; b=4; c = sqrt(a^2+b^2)
c =
5
```

**Example**: Verify

\[
\cos^2 \frac{x}{2} = \frac{\tan x + \sin x}{2 \tan x}
\]

by calculating each side of the equation for \(x = \pi/5\).
Example: Assign the number 3 to variable \( a \) and 4 to variable \( b \). Print out \( \sqrt{a^2 + b^2} \) and assign the solution to the variable \( c \).

\[
>> \quad a=3; \quad b=4; \quad c = \text{sqrt}(a^2+b^2)
\]

\[
c = 5
\]

Example: Verify

\[
\cos^2 \frac{x}{2} = \frac{\tan x + \sin x}{2 \tan x}
\]

by calculating each side of the equation for \( x = \pi/5 \).

\[
>> \quad x = \text{pi}/5;
\]

\[
>> \quad \text{LHS} = \text{cos}(x/2)^2, \quad \text{RHS} = (\text{tan}(x)+\text{sin}(x))/(2*\text{tan}(x))
\]

\[
\text{LHS} = 0.9045
\]

\[
\text{RHS} = 0.9045
\]
**Arrays**

- Used to store and manipulate numbers
- Arranged in rows or columns
Arrays

- Used to store and manipulate numbers
- Arranged in rows or columns

**One-Dimensional Array (Vector)**
- Represents point in $n$-dimensional space
  - Ex: $(x, y)$ in 2D and $(x, y, z)$ in 3D
Arrays

- Used to store and manipulate numbers
- Arranged in rows or columns

One-Dimensional Array (Vector)

- Represents point in $n$-dimensional space
  - Ex: $(x, y)$ in 2D and $(x, y, z)$ in 3D
- **Row Vector** (Use space or comma between numbers)

```plaintext
>>> x = [1 2 3]
x =
1 2 3
```
Arrays

- Used to store and manipulate numbers
- Arranged in rows or columns

**One-Dimensional Array (Vector)**

- Represents point in \( n \)-dimensional space
  - Ex: \((x, y)\) in 2D and \((x, y, z)\) in 3D
- **Row Vector** (Use space or comma between numbers)

```
>> x = [1 2 3]
x =
1 2 3
```

- **Column Vector** (Use semicolon between numbers)

```
>> x = [1; 2; 3]
x =
1
2
3
```
Arrays

Constant Spaced Vectors:

From \( m \) spaced by \( q \) to \( n \)

\[
\text{variable} = [m : q : n]
\]
Constant Spaced Vectors:

From $m$ spaced by $q$ to $n$

variable = $[m : q : n]$

```plaintext
>> x = [1:2:7]
x =
1 3 5 7
```
Arrays

- **Constant Spaced Vectors:**
  - From $m$ spaced by $q$ to $n$
    
    ```
    variable = [m : q : n]
    ```
    
    ```
    >> x = [1:2:7]
    x =
    1 3 5 7
    ```

  - From $m$ to $n$ with $q$ elements
    
    ```
    variable = linspace(m,n,q)
    ```
Constant Spaced Vectors:

From \( m \) spaced by \( q \) to \( n \)

\[
\text{variable} = [m : q : n]
\]

\[
>> \ x = [1:2:7] \\
x = \\
1 \ 3 \ 5 \ 7
\]

From \( m \) to \( n \) with \( q \) elements

\[
\text{variable} = \text{linspace}(m,n,q)
\]

\[
>> \ x = \text{linspace}(0,1,5) \\
x = \\
0 \ 0.2500 \ 0.5000 \ 0.7500 \ 1.0000
\]
Two-Dimensional Array (Matrix)

- Can store information like a table
- Solve systems of equations such as

\[
\begin{align*}
2x + 3y + z &= 4 \\
x - 5y + 3z &= 3 \\
4x - 2y + 3z &= 2
\end{align*}
\]

variable = [1st row; 2nd row; ...; last row]

\[
\begin{bmatrix}
2 & 3 & 1 \\
1 & -5 & 3 \\
4 & -2 & 3
\end{bmatrix}
\]
**Vector:**
- \( \text{ve}(k) \) picks the \( k \)th element of \( \text{ve} \)
- \( \text{ve}(m:n) \) picks the \( m \)th through \( n \)th elements of \( \text{ve} \)

\[
\begin{align*}
\text{\texttt{>> ve = [1 5 2 6 8 7]}} \\
\text{ve =} \\
\text{1 5 2 6 8 7} \\
\text{\texttt{>> ve(5)}} \\
\text{ans =} \\
\text{8} \\
\text{\texttt{>> ve(2:4)}} \\
\text{ans =} \\
\text{5 2 6}
\end{align*}
\]
Matrix:

- `mat(m,n)` picks the \((m, n)\)th element of \(mat\)
- `mat(m:n,p:q)` picks the \((m : n) \times (p : q)\) submatrix of \(mat\)

```matlab
>> mat = [1 4 2 3; 3 6 9 2; 1 4 9 7; 2 5 1 8]
mat =
  1  4  2  3
  3  6  9  2
  1  4  9  7
  2  5  1  8
>> mat(2,3)
ans =
   9
>> mat(2:4, 1:3)
     3   6   9
ans =
  1   4   9
  2   5   1
```
Adding Elements

- Can add elements by using the variable within vector/matrix
- Must be of appropriate size

```
>> mat = [1 4 2 3; 3 6 9 2;1 4 9 7]
mat =
  1   4   2   3
  3   6   9   2
  1   4   9   7
>> [mat; 2 5 1 8]
ans =
  1   4   2   3
  3   6   9   2
  1   4   9   7
  2   5   1   8
```
Delete elements by assigning nothing to these elements

```matlab
>> ve = [1 5 2 6 8 7]
ve =
1 5 2 6 8 7
>> ve(2:4) = []
ve =
1 8 7
>> mat = [1 4 2 3; 3 6 9 2; 1 4 9 7]
mat =
1 4 2 3
3 6 9 2
1 4 9 7
>> mat(2:3,:) = []
mat =
1 4 2 3
```
### Helpful Tips for Arrays

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>length(A)</td>
<td>Returns number of elements in the vector A</td>
</tr>
<tr>
<td>size(A)</td>
<td>Returns size of matrix A</td>
</tr>
<tr>
<td>reshape(A,m,n)</td>
<td>Rearranges A to have m rows and n columns (arranged column-wise)</td>
</tr>
</tbody>
</table>
Strings

- Is an array of characters
- Created by typing characters within single quotes
- Can include letters, digits, symbols and spaces

```matlab
>> a = 'Matlab is AWESOME'
a =
Matlab is AWESOME
>> a(1) =
ans =
M
>> a(1:6)
ans =
Matlab
```
Strings

- Is an array of characters
- Created by typing characters within single quotes
- Can include letters, digits, symbols and spaces

```matlab
>> a = 'Matlab is AWESOME'
a =
Matlab is AWESOME
>> a(1) =
an =
M
>> a(1:6)
an =
Matlab
>> a(1:6) = 'M Shah'
a =
M Shah is AWESOME
```