Introduction to MATLAB
Learning Objectives

familiarity with:
- MATLAB operations
- Simple Plotting
- MATLAB functions, scripts
- Complex numbers
- Matrices, vectors
Here is what the MATLAB screen looks like when you start up.

Using Toolbox Path Cache. Type "help toolbox_path_cache" for more info.

To get started, select "MATLAB Help" from the Help menu.
If we generate a vector in the command window, it shows up in the workspace window.

```matlab
>> x = linspace(0, 5, 100);
```

To get started, select "MATLAB Help" from the Help menu.
we can see the files in the current directory by choosing Current Directory
we can bring up an editor window for making new functions or scripts

MATLAB Editor for generating functions, scripts
example of generating a plot

```matlab
x = linspace(0, 5, 100);
y = x.^2;
plot(x, y)
```
## MATLAB

### Built in constants and variables

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ans</td>
<td>most recent answer</td>
</tr>
<tr>
<td>eps</td>
<td>small constant ~ 10^{-16}</td>
</tr>
<tr>
<td>i or j</td>
<td>imaginary unit</td>
</tr>
<tr>
<td>inf</td>
<td>infinity</td>
</tr>
<tr>
<td>pi</td>
<td>3.14159 ...</td>
</tr>
</tbody>
</table>

### Some common functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sin(x)</td>
<td>sine</td>
</tr>
<tr>
<td>cos(x)</td>
<td>cosine</td>
</tr>
<tr>
<td>exp(x)</td>
<td>exponential</td>
</tr>
<tr>
<td>sqrt(x)</td>
<td>square root</td>
</tr>
<tr>
<td>log(x)</td>
<td>natural log</td>
</tr>
<tr>
<td>log10(x)</td>
<td>log to base 10</td>
</tr>
<tr>
<td>abs(x)</td>
<td>absolute value, magnitude of complex quantity</td>
</tr>
<tr>
<td>angle(x)</td>
<td>phase angle</td>
</tr>
<tr>
<td>real(x)</td>
<td>real part of</td>
</tr>
<tr>
<td>imag(x)</td>
<td>imaginary part of</td>
</tr>
</tbody>
</table>

### Standard mathematical operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>±</td>
<td>addition, subtraction</td>
</tr>
<tr>
<td>*</td>
<td>multiplication</td>
</tr>
<tr>
<td>/</td>
<td>division</td>
</tr>
<tr>
<td>^</td>
<td>exponentiation e.g. ( y^n = y^n )</td>
</tr>
</tbody>
</table>
Generation of (row) vectors

>> x = 0:0.1:0.5
x =
   0   0.1000   0.2000   0.3000   0.4000   0.5000

>> y = linspace(0, 0.5, 4)
y =
   0   0.1667   0.3333   0.5000

>> z = [ 1 2 3 4 5]
z =
   1   2   3   4   5

Suppression of echoing of output (put semi-colon at end of line)

>> z = [ 1 2 3 4 5] ;
>>
>> z
z =
   1   2   3   4   5

Comment line

>> % This is a comment
>>
Functions can have vector (matrix) arguments

```matlab
>> x = [ 1 2 3 4 5];
>> y = exp(x)
y =
 2.7183  7.3891  20.0855  54.5982  148.4132
```

Element by element operations on vector-valued functions

- addition, subtraction
- multiplication
- division
- exponentiation

```matlab
>> x = [ 1 2 3 4]
x =
 1  2  3  4
>> y = x + 2
y =
 3  4  5  6
>> z = x .^2
z =
 1  4  9 16
>> f = x .*x .^2
f =
 1  8 27 64
```
MATLAB does complex arithmetic with scalars and vectors

\[
\begin{align*}
\text{>> } & \quad z = 3 + 4*i \\
& \quad z = \\
& \quad 3.0000 + 4.0000i \\
\text{>> } & \quad y = i*z \\
& \quad y = \\
& \quad -4.0000 + 3.0000i \\
\text{>> } & \quad x = [1 + 3*i \ 2*i] \\
& \quad x = \\
& \quad 1.0000 + 3.0000i \ 0 + 2.0000i \\
\text{>> } & \quad y = [i \ i] \\
& \quad y = \\
& \quad 0 + 1.0000i \ 0 + 1.0000i \\
\text{>> } & \quad z = x .* y \\
& \quad z = \\
& \quad -3.0000 + 1.0000i \ -2.0000
\end{align*}
\]

Common functions take complex arguments (scalars or vectors)

\[
\begin{align*}
\text{>> } & \quad \text{exp}(i*pi) \\
& \quad \text{ans} = \\
& \quad -1 \\
\text{>> } & \quad x = [0 \ \text{pi}/2 \ \text{pi}] \\
\text{>> } & \quad \text{exp}(i*x) \\
& \quad \text{ans} = \\
& \quad 1.0000 \ 0 + 1.0000i \ -1.0000
\end{align*}
\]
Magnitude and phase of a complex number

\[ z = a + ib \]

\[ = Ae^{i\phi} \]

\[ A = |z| = \sqrt{a^2 + b^2} \]

\[ \phi = \tan^{-1}\left(\frac{b}{a}\right) \]

In MATLAB:

\[ >> z = 1 + 2*i; \]
\[ >> \text{abs}(z) \]
\[ \text{ans} = 2.2361 \]

\[ >> \text{angle}(z) * 180 / \pi \]
\[ \text{ans} = 63.4349 \]

convert from radians to degrees
Simple plotting

```matlab
>> x = linspace( 0, 2*pi, 100);
>> y = cos(x);
>> plot(x, y)
```

Multiple plots on same graph

```matlab
>> x = linspace( 0, 2*pi, 100);
>> y1 = cos(x);
>> y2 = sin(x);
>> plot(x, y1, x, y2)
```
or

```matlab
>> x = linspace(0, 2*pi, 100);
>> y1 = cos(x);
>> plot(x, y1)
>> hold on
>> y2 = sin(x);
>> plot(x, y2)
>> hold off
```
Adding a x-axis label, a y-axis label, and a title to a plot

```matlab
>> x = linspace(0, 2*pi, 100);
>> y = cos(x);
>> plot(x, y)
>> xlabel('x-axis text here')
>> ylabel('y axis text here')
>> title('title text here')
```

![Plot with labeled axes and title](image_url)
Plotting with different line styles

```
x = linspace( 0, 2*pi, 100);
y1 = cos(x);
y2 = cos(x+1);
y3 = cos(x+2);
plot(x, y1, '--', x, y2, ':', x, y3, '.-')
```

- **y1** dashed line
- **y2** dotted line
- **y3** dot-dash
Logical (0-1) vectors

```matlab
>> x = [ 1 2 3 4 5];
>> x >3
ans =
    0   0   0   1   1
>> x <= 4
ans =
    1   1   1   1   0
```

Use of logical vectors for defining piecewise function

```matlab
>> x = linspace(0, 1, 500);
>> y = (x .^2).* (x < 0.5) + (0.75 - x ).*(x >= 0.5);
>> plot(x, y)
```
Defining parts of vectors

```matlab
>> x = [ 5 6 3 8 7 9];
>> x(1:3)
ans =
   5   6   3
>> x(2:4)
ans =
   6   3   8
>> x(3:end)
ans =
   3   8   7   9
```

Other vector operations

Vector magnitude, length

```matlab
>> x = [3 4];
>> norm(x)
ans =
    5
>> length(x)
ans =
    2
```
Use of eps to avoid division by zero

```matlab
>> x=linspace(0, 4, 100);
>> y=sin(x)./x;

Warning: Divide by zero.

>> x = x + eps*(x == 0);
>> y=sin(x)./x;
>> plot(x,y)
```

Use of inf to evaluate expression when a variable goes to infinity

```matlab
function y = infinitytest(x)
x = x + eps*(x == 0);
y = 3/(1 + 4/x);

>> infinitytest(1)
an =
0.6000
>> infinitytest(0)
an =
1.6653e-16
>> infinitytest(inf)
an =
3```
MATLAB Functions

Functions are defined in the editor and saved as M-files. The name of the M-file is the name of the function with a .m extension, e.g. myfunction.m etc. As shown in the example below, functions can have multiple outputs (as well as multiple inputs).

```matlab
function [y, z] = test(x)
y = x.^2;
z = 10*exp(-x);
```

```matlab
>> x = linspace(0, 2, 100);
>> [s, t] = test(x);
>> plot(x, s, x, t)
```

All the variables appearing in a function are local to that function, i.e. they do not change any similarly named variables in the MATLAB workspace.
MATLAB Scripts

A MATLAB script is a sequence of ordinary MATLAB statements, defined in the editor and then saved as a file. The file has the name of the script followed by a .m extension, e.g. myscript.m. Typing the script name at the MATLAB prompt then causes the script to be executed. Variables in the script change any values of variables of the same name that exist in the current MATLAB session.

% testscript
x = linspace(0,2,100);
y = x.^2;
z = 10*exp(-x);
plot(x,y,x,z)

>> testscript
MATLAB Matrices

MATLAB was designed to perform operations with matrices very effectively. Entering matrices manually is as easy as vectors:

```matlab
>> matrix = [ 4 0 3;
0 3 5;
3 5 7 ]
matrix =
 4 0 3
0 3 5
3 5 7
```

Accessing individual components

```matlab
>> matrix(2,3)
ans =
 5
```

Accessing rows or columns

```matlab
>> matrix(:, 2)
ans =
 0
3
5
>> matrix(3, :)
ans =
 3 5 7
```
Some matrix functions

size(M) returns number of rows, nr, and number of columns, nc, as a vector [nr, nc]

trace(M) trace of M (sum of diagonal terms)

det(M) determinant of M

M' transpose of M (interchange rows and columns)

```matlab
>> size(matrix)
anst =
   3   3
>> trace(matrix)
anst =
    14
>> det(matrix)
anst =
   -43
>> matrix'
anst =
    4   0   3
    0   3   5
    3   5   7
```

(no change since matrix is symmetric)
Multiplying matrices and vectors

```matlab
>> v = [ 1 2 3];
>> v*matrix
ans =
    13   21   34

>> vt = v'
vt =
    1
    2
    3
>> matrix*vt
ans =
    13   21   34
    34

>> v*matrix*vt
ans =
    157
```
Special matrices

zeros(m,n)  matrix of all zeros with m rows and n columns
ones(m,n)  matrix of all ones with m rows and n columns
eye(m,n)  identity matrix with m rows and n columns

```matlab
>> zeros(3,3)
an =
      0     0     0
      0     0     0
      0     0     0
```

```matlab
>> ones(3,3)
an =
      1     1     1
      1     1     1
      1     1     1
```

```matlab
>> eye(3,3)
an =
      1     0     0
      0     1     0
      0     0     1
```
Solving a system of linear equations $Mx = b$

$$
>> M = \begin{bmatrix}
1 & 3 & 5 \\
2 & 1 & 1 \\
4 & 3 & 6 \\
\end{bmatrix}
$$

$M =$

\[
\begin{array}{ccc}
1 & 3 & 5 \\
2 & 1 & 1 \\
4 & 3 & 6 \\
\end{array}
\]

$$
>> b = \begin{bmatrix}
3 \\
2 \\
1 \\
\end{bmatrix}
$$

$b =$

\[
\begin{array}{c}
3 \\
2 \\
1 \\
\end{array}
\]

\[
\begin{align*}
1x_1 + 3x_2 + 5x_3 &= 3 \\
2x_1 + x_2 + x_3 &= 2 \\
4x_1 + 3x_2 + 6x_3 &= 1
\end{align*}
\]

$$
>> x = M \backslash b
$$

$x =$

\[
\begin{array}{c}
-0.0909 \\
3.9091 \\
-1.7273 \\
\end{array}
\]
Determining the eigenvectors and eigenvalues of a matrix $M$
(For a real symmetric matrix the eigenvalues are real and the eigenvectors are real
and orthogonal to each other)

$$[\text{eigenvects, eigenvals}] = \text{eig}(M)$$

$$\begin{bmatrix}
-0.8752 & 0.3507 & 0.3332 \\
0.4783 & 0.7300 & 0.4881 \\
0.0720 & -0.5865 & 0.8067 \\
\end{bmatrix}$$

Eigenvectors (in columns)

$$\begin{bmatrix}
3.7531 & 0 & 0 \\
0 & -1.0172 & 0 \\
0 & 0 & 11.2641 \\
\end{bmatrix}$$

Corresponding eigenvalues