Lesson 5: Simple plots and user-defined functions

Topics:

1. Simple plots and axis controls
2. User-defined functions returning a single value

Book Sections: 3.5 and 3.7

1. Simple Plots

MATLAB provides tools to visualize data or mathematical functions.

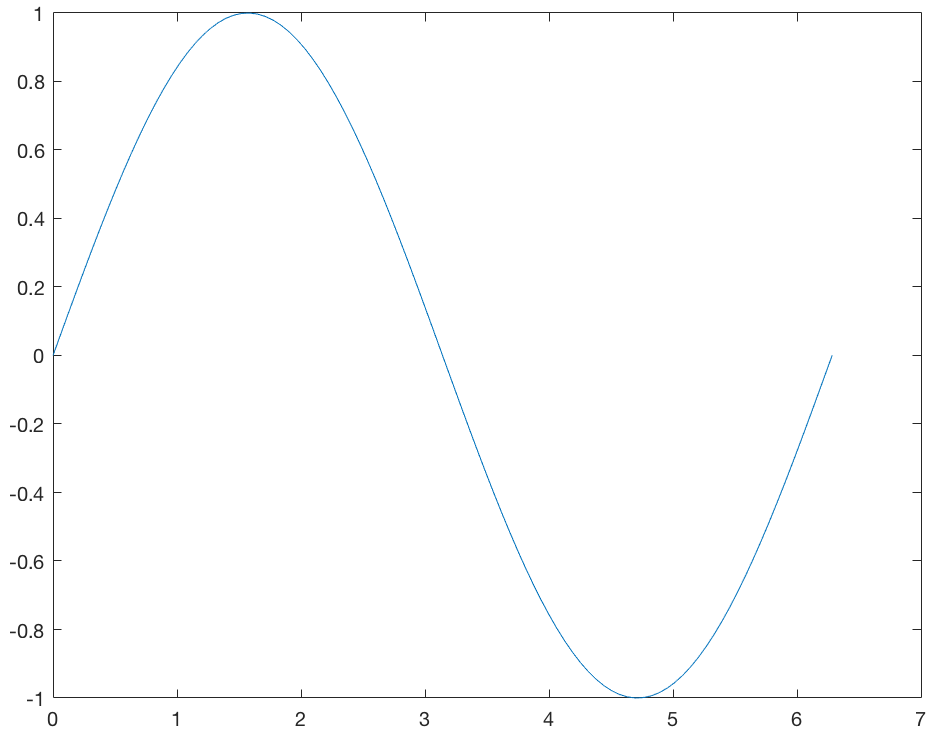
```
% You should read the help file
help plot
```

```
  plot   Linear plot.
     plot(X,Y) plots vector Y versus vector X. If X or Y is a matrix,
     then the vector is plotted versus the rows or columns of the matrix,
     whichever line up.  If X is a scalar and Y is a vector, disconnected
     line objects are created and plotted as discrete points vertically at
     X.

     plot(Y) plots the columns of Y versus their index.
     If Y is complex, plot(Y) is equivalent to plot(real(Y),imag(Y)).
     In all other uses of plot, the imaginary part is ignored.

     Various line types, plot symbols and colors may be obtained with
     plot(X,Y,S) where S is a character string made from one element
     from any or all the following 3 columns:
```

1.1 Plotting a function

```
% Example: Plot y = sin(x)

% Generate the data points
```

```
% Generate the data points
x=linspace(0,2*pi);
y=sin(x);

% Plot the function
plot(x,y)
```

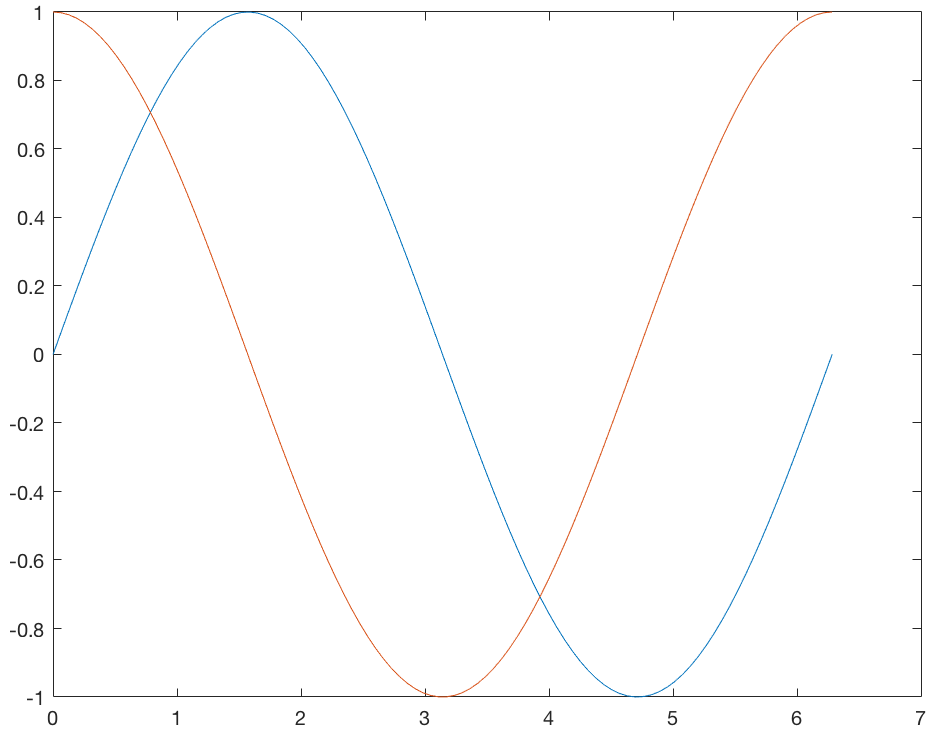


1.2 Plot two functions simultaneously

By default, MATLAB will make the two blue and red.

```
% Example: Plot y=sin(x) and y=cos(x)
x=linspace(0,2*pi);
y1=sin(x);
y2=cos(x);
```

```
plot(x,y1, x, y2)
```



1.3. Adding options (color, line style, and marker type)

Add options in single quotes.

```
x=linspace(0,2*pi);
y=sin(x);

% COLOR
plot(x,y,'r');    % plots y=sin(x) in red
plot(x,y,'k');    % plots y=sin(x) in black
```

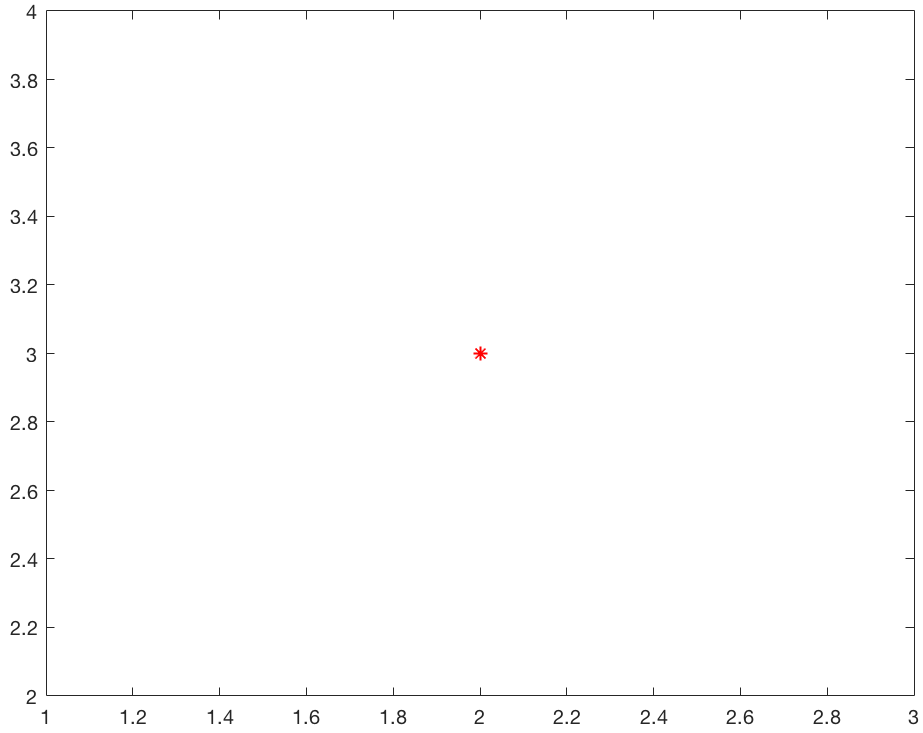
```
% LINE TYPE
plot(x,y,'-');    % plots y=sin(x) with solid line
plot(x,y,':');    % plots y=sin(x) with dotted line
plot(x,y,'--');   % plots y=sin(x) with dashed line

% MARKER TYPE
plot(x,y,'s');    % plots y=sin(x) with square points
plot(x,y,'*');    % plots y=sin(x) with star points
plot(x,y,'o');    % plots y=sin(x) with circle points
plot(x,y,'+');    % plots y=sin(x) with plus points

% See help file for more options
```

1.3.2. Combine all three options

```
% COMBINE OPTIONS
plot(x,y,'r:*');    % plots y=sin(x) in red, dashed, stars
```

1.4 Plotting a single point

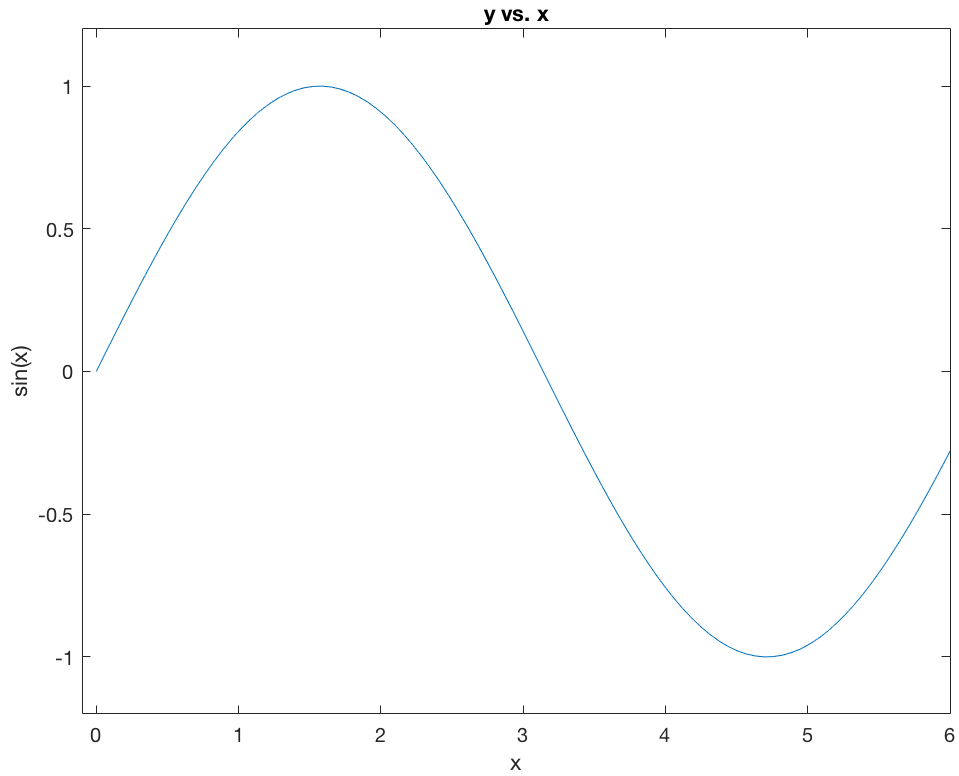
```
x=2;y=3;

plot(x,y,'r*')
```



1.5. Axis controls

```
% Define x and y
x=linspace(0,2*pi);
y=sin(x);

% PLOT
plot(x,y,'r')

% Add TITLE
title 'y vs. x'

% LABEL the X-AXIS
xlabel 'x'
```

```
% LABEL the Y-AXIS
ylabel 'sin(x)'

% SIZING
xmin=-0.1;xmax=6;ymin=-1.2;ymax=1.2;
axis([xmin xmax ymin ymax])
```

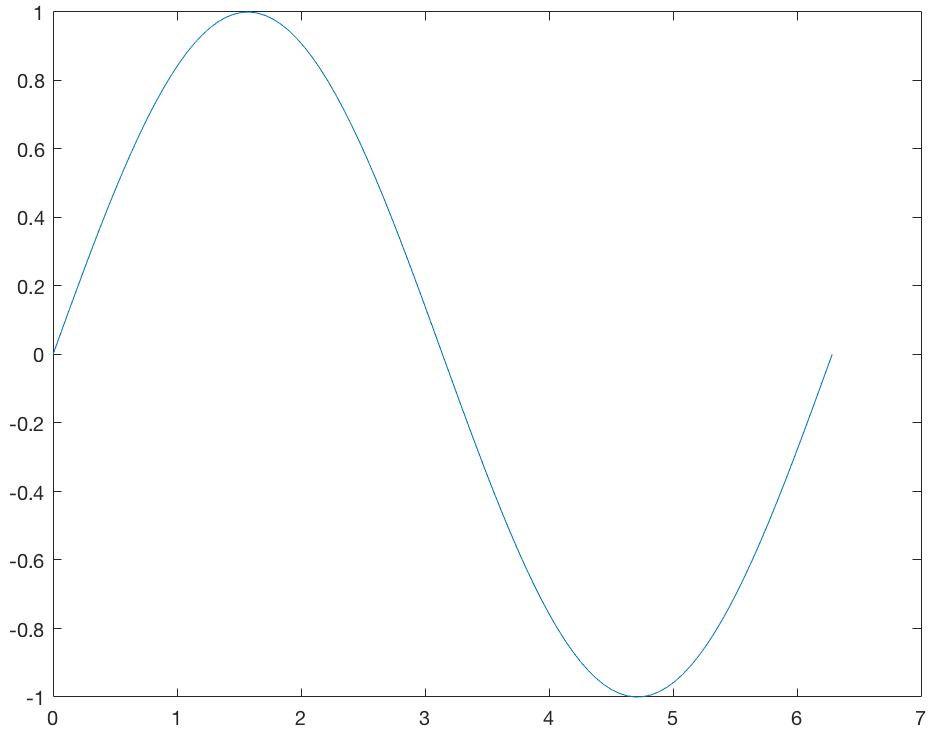


1.5.1 Clearing axis

NOTE: CLA and CLF do NOT work in LiveScript. Try typing these lines at the command prompt to view the effects of CLA and CLF.

```
% Define x and y
x=linspace(0,2*pi);
y=sin(x);

% PLOT
plot(x,y)
```

```
plot(x,y)
```




```
% CLEAR (only) AXIS
cla
% NOTE: Title is not cleared

% CLEAR FIGURE
clf
```

2. User-defined Functions

So far we have used SCRIPTS (or the command window). Now function files are introduced. You create a new function (function file) by clicking the dropdown arrow under "New" (or File --> New --> Function).

A MATLAB function is similar to a mathematical function.

NOTE: The file must be named exact the same as the function.

Example

```
% Get input from user regarding a key variable value
r=input('What is the radius of the circle? ');

% Funtion call (or "call to function")
A=circlearea(r);

% Display the output
fprintf('The area of the circle is %1.5f square units\n',A)
```

```
The area of the circle is 28.27433 square units
```

```

```

```
function A=circlearea(r)

    A=pi*r^2;

end
```

Return to MAT225