

Intro Math Problem Solving

October 3

plot() needs x and y lists

linspace() can create an x list

MATLAB functions: y lists from x lists

Special "dot" format for array arithmetic

Plotting formulas involving lists

Taylor Polynomials

A Reference

Most of today's discussion is covered in Chapter 4, Section 1, of "Insight Through Computing".

A PDF, "insight_chapter4.pdf" of this information is in Canvas class directory of files for "10_03".

Exercise

Our goal is a nice plot of the function:

$$y = \sin(5x) \exp(-x/2) / (1+x^2)$$

over the interval $[-2,3]$.

I encourage you to try to set up the `xlist` and `ylist` data, and the plot commands for this goal, even though you probably don't know enough to get it to work completely.

We should be able to do this by the end of the class.

We will also know how to customize the plot with thicker lines of our chosen color, grid lines, titles and so on.

The plot() command uses x and y lists

Think of the plot command this way:

```
plot ( xlist, ylist )
```

where xlist and ylist represent x values and the corresponding y values.

The actual names of the lists don't matter, but the lists **must** be the same length.

Typically, the x values are listed in order (unless you're doing something really interesting!)

Short lists can be created using "[*,*]"

If we don't have much data to enter, we can create lists by listing the data in square brackets:

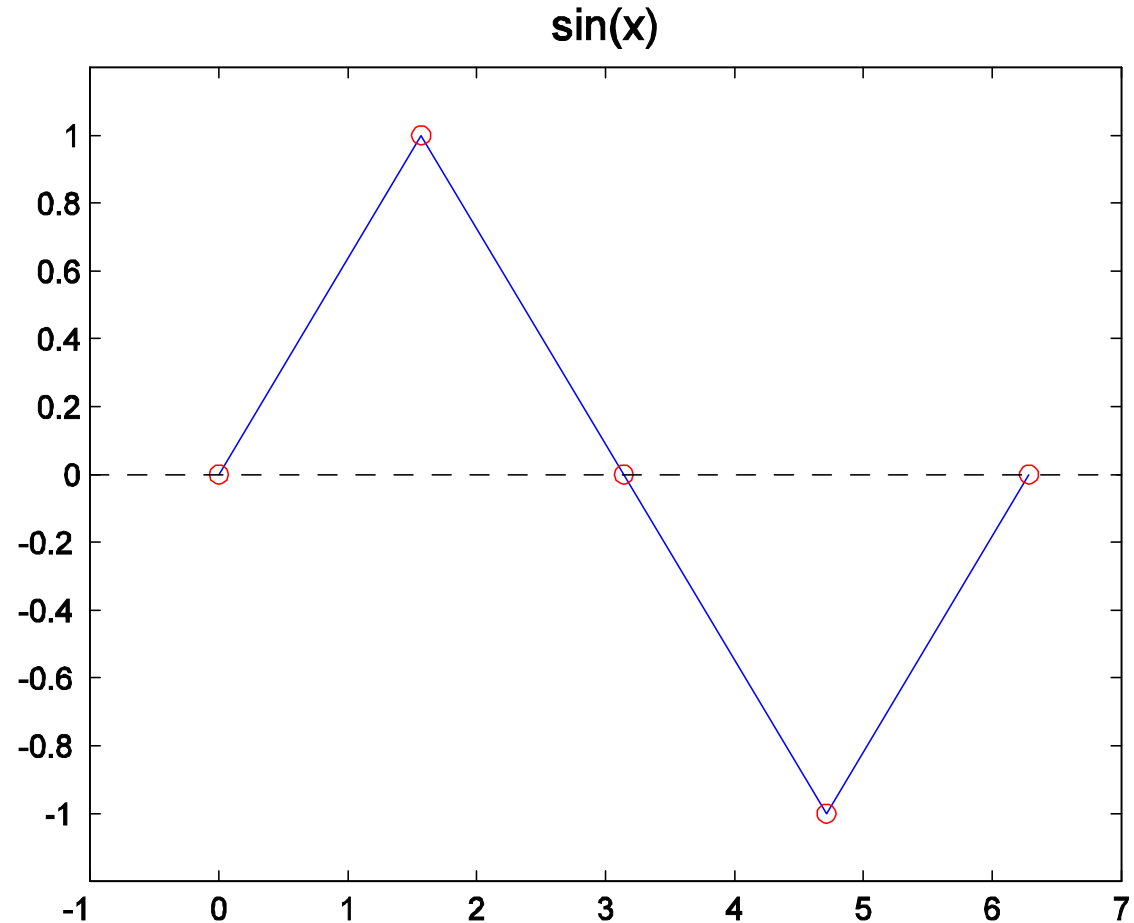
```
xlist = [ 0.0, 1.57, 3.14, 4.71, 6.28 ];
```

```
ylist = [ 0.0, 1.0, 0.0, -1.0, 0.0 ];
```

```
plot ( xlist, ylist );
```

Plot $\sin(x)$ using 5 values: sine_5.m

x	sin(x)
0.00	0.0
1.57	1.0
3.14	0.0
4.71	-1.0
6.28	0.0



5 Points Not Enough, Try 9

To improve our plot, use more x values.

```
xlist = [ 0.0      1.57      3.14      4.71      6.28 ];
```

```
xlist = [ 0.0 0.78 1.57 2.35 3.14 3.92 4.71 5.49 6.28 ];
```

To update our ylist, we could compute the sine of each new value and insert it into our ylist:

```
ylist = [ 0.0, (compute this), 1.0, (compute this), ... ];
```

OR we could try this:

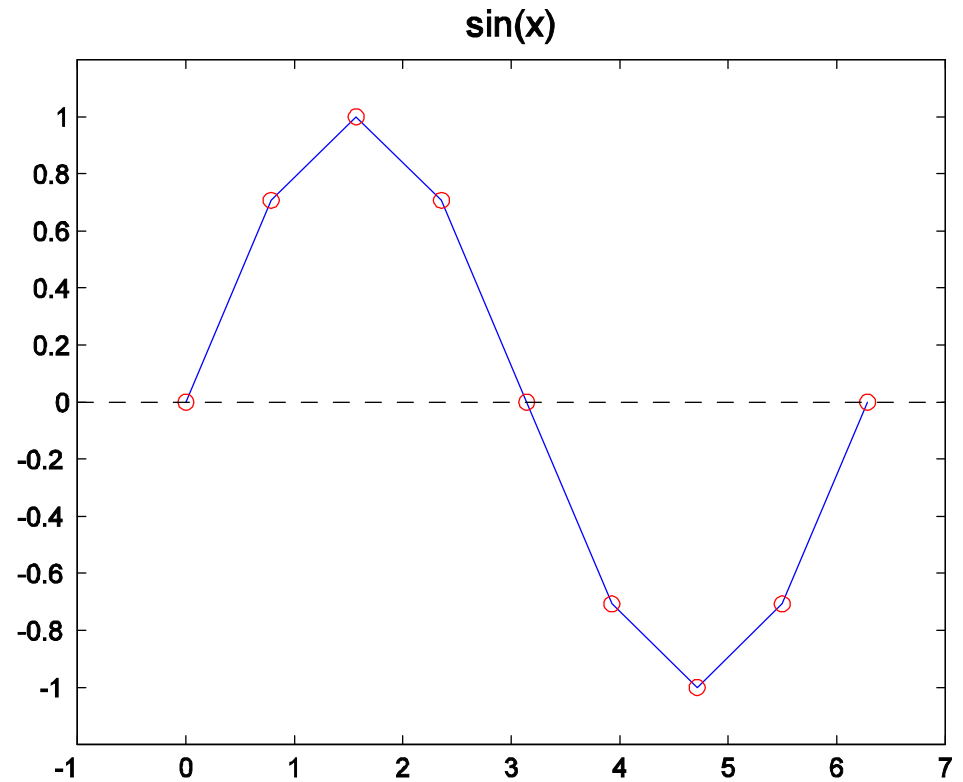
```
ylist = [ sin(0.0), sin(0.78), sin(1.57), ... ];
```

OR BETTER YET:

```
ylist = sin ( xlist );
```

Plot $\sin(x)$ using 9 values: sine_9.m

x	sin(x)
0.000	0.000
0.784	0.707
1.571	1.000
2.357	0.707
3.142	0.000
3.927	-0.707
4.712	-1.000
5.498	-0.707
6.283	0.000



Don't Enter Values By Hand!

A good sine plot needs more x and y data, but this is not a job for humans!

Step 1: create equally spaced xlist in $[0, 2\pi]$:

```
xlist = linspace ( 0.0, 2*pi, 9 );
```

Step 2: evaluate the sine at each x in xlist:

```
ylist = sin ( xlist );
```

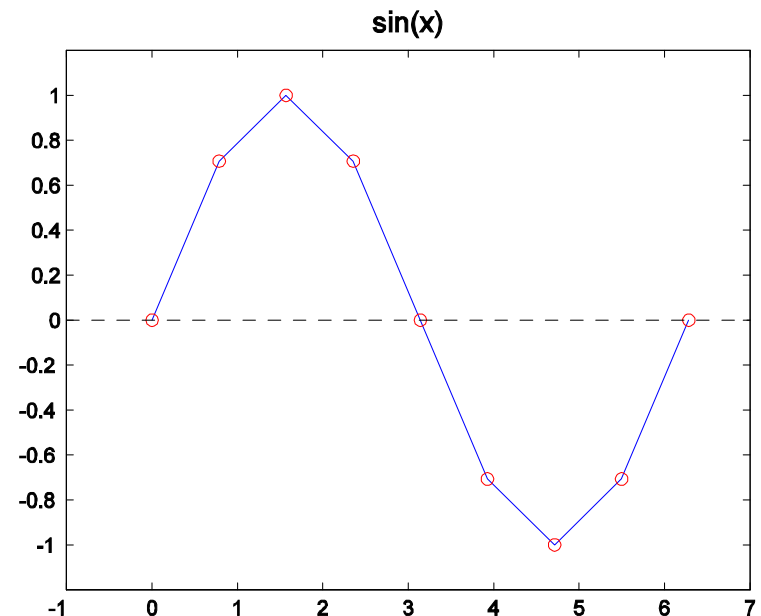
Step 3: Your plot is ready:

```
plot ( xlist, ylist );
```

A 9 Point Plot, "Automatically"

x	sin(x)
0.000	0.000
0.784	0.707
1.571	1.000
2.357	0.707
3.142	0.000
3.927	-0.707
4.712	-1.000
5.498	-0.707
6.283	0.000

```
x = linspace(0,2*pi,9);  
y = sin(x);  
plot(x,y)
```



Now Get a Smooth Plot

Once you can set up your data automatically, then it's easy to ask for more data to get a smoother curve.

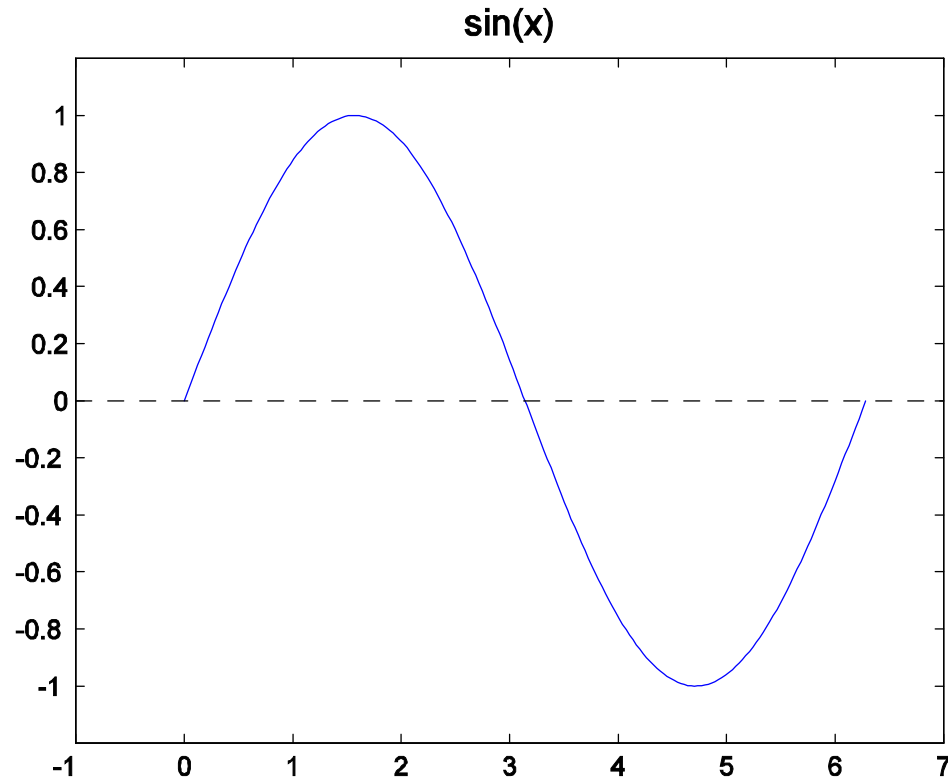
The only change we have to make is to ask `linspace()` for more points:

```
xlist = linspace ( 0.0, 2*pi, 200 );
```

```
ylist = sin ( xlist );
```

```
plot ( xlist, ylist );
```

Plot $\sin(x)$ using 200 values: sine_200.m



Although the graph looks smooth, we can ZOOM IN to see that it's really a sequence of straight line segments! Look at the peak!

Fact #1: Linspace makes lists easily

```
x = linspace(1, 3, 5)
```

```
x : 

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| 1.0 | 1.5 | 2.0 | 2.5 | 3.0 |
|-----|-----|-----|-----|-----|


```

"x is a list"

"x is a table of values"

"x is an array"

"x is a vector"

"x is a row vector"

"x is a 1 by 5 matrix"

For nice spacing, may need 1 extra value.

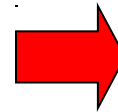
```
x = linspace(0,1,101)
```

```
x : 0.00 | 0.01 | 0.02 | ... | 0.99 | 1.00
```

If you want to take N steps from 0 to 1, you need to use $N+1$ points.
To take steps of size $1/10$, you want 11 points; steps of size $1/100$ require 101.

FACT #2: Built-In Functions Take Arrays

0.00	1.57	3.14	4.71	6.28
------	------	------	------	------



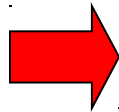
sin

x	y=sin(x)
0.00	?
1.57	?
3.14	?
4.71	?
6.28	?

And...

Return a new array of function values

sin



0.00 1.00 0.00 -1.00 0.00

x	y=sin(x)
0.00	0.0
1.57	1.0
3.14	0.0
4.71	-1.0
6.28	0.0

Examples

```
x = linspace(0,1,200);  
y = exp(x);  
plot(x,y)
```

```
x = linspace(1,10,200);  
y = log(x);  
plot(x,y)
```

Our goal: a plot of this formula

$$f(x) = \frac{\sin(5x) \exp(-x/2)}{1+x^2} \quad -2 \leq x \leq 3$$

It's Natural to Try This:

$$f(x) = \frac{\sin(5x) \exp(-x/2)}{1+x^2} \quad -2 \leq x \leq 3$$

We try this, but MATLAB refuses!

```
x = linspace(-2, 3, 200);  
y = sin(5*x) * exp(-x/2) / (1 + x^2)  
plot(x, y)
```

Why is MATLAB Confused?

Our formula makes perfect sense if x is just a number. But now x is a list. In order for "*" and "/" and "^" to do what we want, we need to put a "dot" in front of each of these commands.

When we operate on lists, "dots" are needed when "*", "/" or "^" are applied to lists.

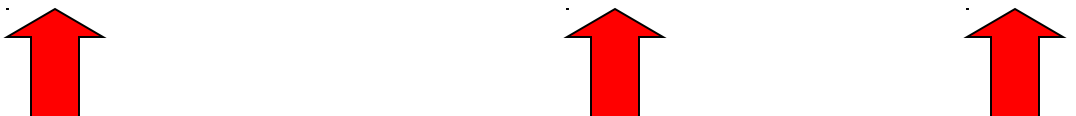
(The reason is that in linear algebra, "*", "/" and "^" represent other ways of combining lists.)

Use "DOT" on Times, Divide, Power example5.m

$$f(x) = \frac{\sin(5x) \exp(-x/2)}{1+x^2} \quad -2 \leq x \leq 3$$

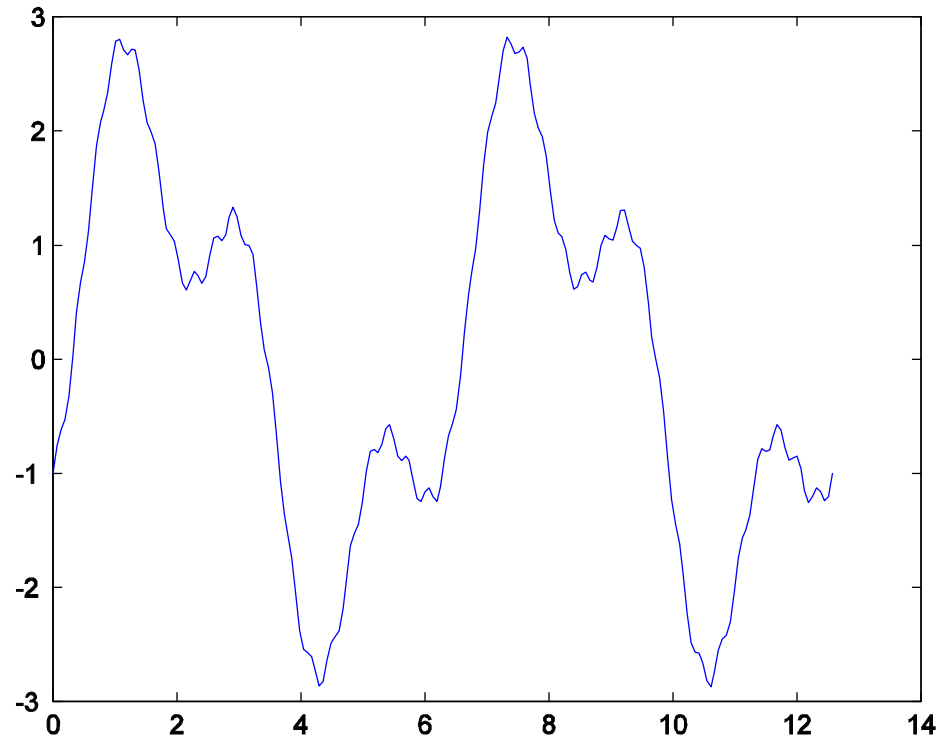
For arithmetic on lists, we need to use a "dot"

```
x = linspace(-2, 3, 200);  
y = sin(5*x) .* exp(-x/2) ./ (1 + x.^2)  
plot(x, y)
```



Array operations

example1.m



$$f(x) = 2\sin(x) - \cos(3x) + .1*\sin(20x)$$

Scale list by ONE number (*) No dot!

a:

10	8	-5
----	---	----

$$c = s * a$$

s:

2

c:

20	16	-10
----	----	-----

Add Two Lists: + (No dot needed)

a:

10	8	-5
----	---	----

$$c = a + b$$

b:

2	4	1
---	---	---

c:

12	12	-4
----	----	----

Subtract Lists: - (No dot)

a:

10	8	-5
----	---	----

$$c = a - b$$

b:

2	4	1
---	---	---

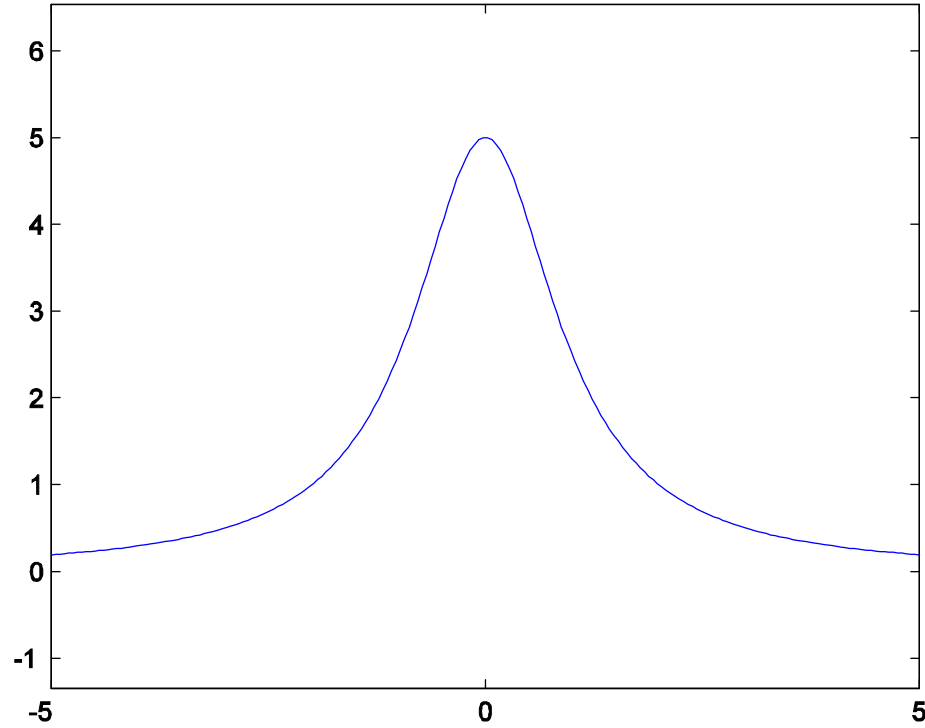
c:

8	4	-6
---	---	----

Example 1: No Special "Dot" Operations

```
x = linspace(0, 4*pi, 200);  
y1 = sin(x);  
y2 = cos(3*x);  
y3 = sin(20*x);  
y = 2 * y1 - y2 + 0.1 * y3;  
plot(x, y)
```

example2.m



$$f(x) = \frac{5}{1+x^2}$$

Powers of a List: "DOT ^"

a:

10

8

-5

$c = a.^s$

s:

2

c:

100

64

25



Add ONE Number to List: "+"

a:

10	8	-5
----	---	----

$$c = a + s$$

s:

2

c:

12	10	-3
----	----	----

Divide One Number by List: "DOT /"

a:

10	8	-5
----	---	----

$$c = 1 ./ a$$

c:

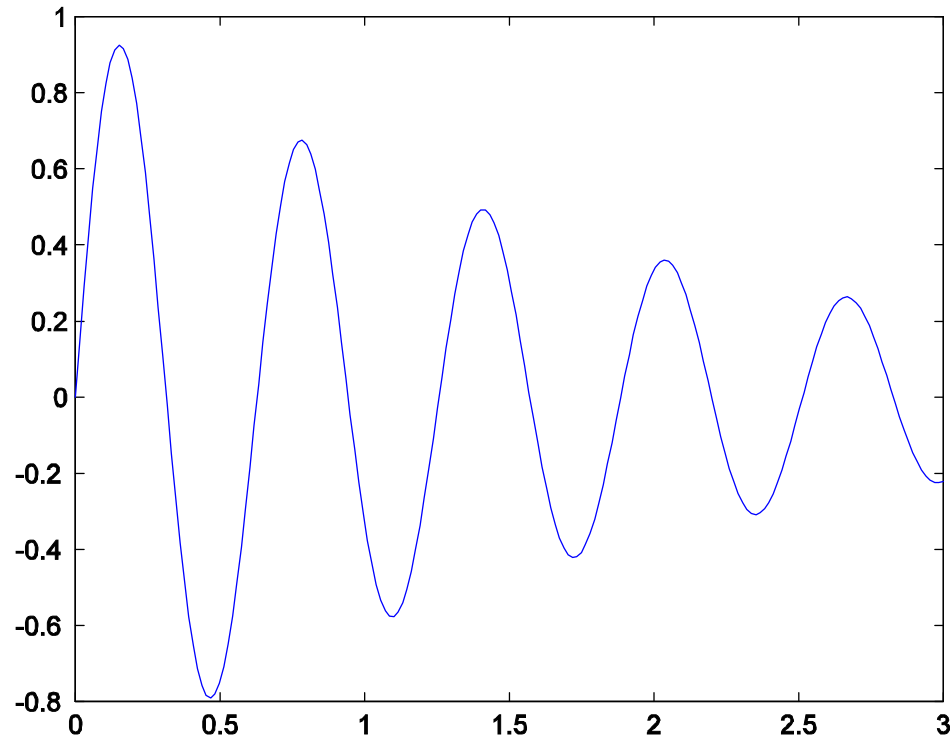
.1	.125	-.2
----	------	-----

Example 2 Solution:

The division and power operations must use a "dot":

```
x = linspace(-5,5,200);  
y = 5./(1+ x.^2);  
plot(x,y)
```

example3.m



$$f(x) = \exp(-x/2) \sin(10x)$$

Change Sign of a List: "-"

a:

10	8	-5
----	---	----

$$c = -a$$

c:

-10	-8	5
-----	----	---

A list divided by ONE number: "/"

a:

10	8	-5
----	---	----

$$c = a/s$$

s:

2

c:

5	4	-2.5
---	---	------

Multiply One List by Another: "DOT *"

a:

10	8	-5
----	---	----

c = a .* b

b:

2	4	1
---	---	---

.*

c:

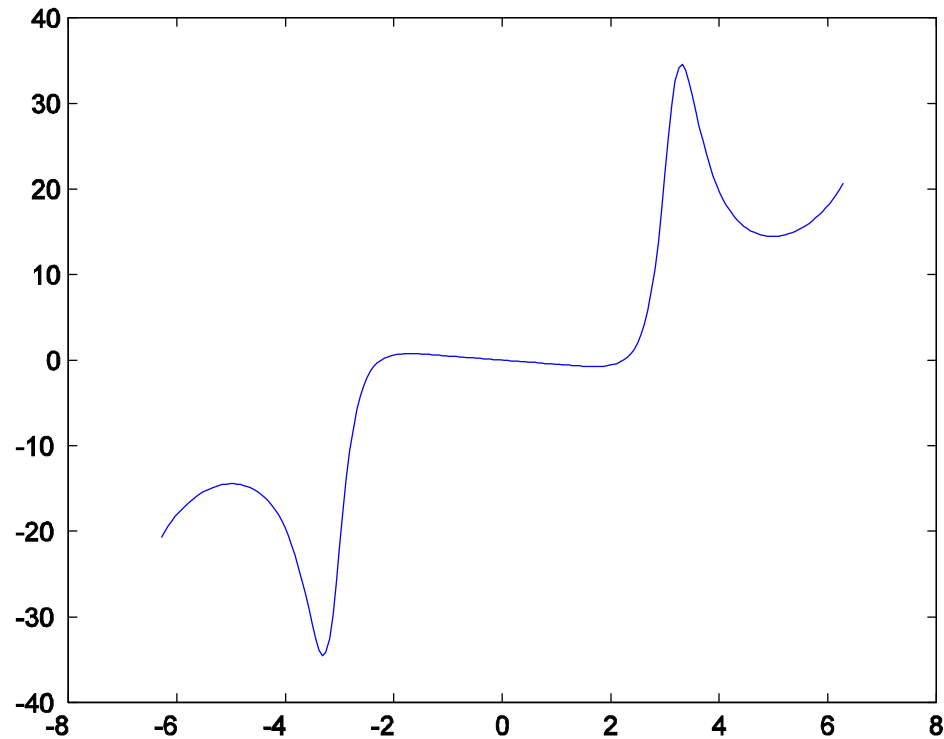
20	32	-5
----	----	----

Equation 3 Solution

We multiply one list by another.
For that, we need "DOT *":

```
x = linspace(0,3,200);  
y = exp(-x/2) .* sin(10*x);  
plot(x,y)
```

example4.m



$$f(x) = \frac{.2x^3 - x}{1.1 + \cos(x)}$$

One List Divided by Another: "Dot /"

a:

10	8	-5
----	---	----

$c = a ./ b$

b:

2	4	1
---	---	---

c:

5	2	-5
---	---	----



Equation 4 Solution

Power of a list: DOT ^

List divided by list: DOT /

```
x = linspace(-2*pi,2*pi,200);  
y = (0.2*x.^3 - x) ./ (1.1 + cos(x));  
plot(x,y)
```

When we need a DOT:

$a * b \rightarrow a .* b$ if a and b are BOTH lists;

$a / b \rightarrow a ./ b$ if b is a list;

$a ^ b \rightarrow a .^ b$ if a is a list
or b is a list (rare case).

Question Time

How many errors in the following statement given that

```
x = linspace(0,1,100) :
```

$$y = (3*x .+ 1) / (1 + x^2)$$

- A. 0 B. 1 C. 2 D. 3 E. 4

Question Time

How many errors in the following statement given that

```
x = linspace(0,1,100) :
```

$$y = (3*x .+ 1) / (1 + x^2)$$

$$y = (3*x + 1) ./ (1 + x.^2)$$

A. 0

B. 1

C. 2

D. 3

E. 4

Question Time

Does this assign to y the values $\sin(0^\circ), \sin(1^\circ), \dots, \sin(90^\circ)$?

```
x = linspace(0, pi/2, 90) ;  
y = sin(x) ;
```

A. Yes

B. No

Question Time

Does this assign to y the values $\sin(0^\circ), \sin(1^\circ), \dots, \sin(90^\circ)$?

```
%x = linspace(0, pi/2, 90) ;  
x = linspace(0, pi/2, 91) ;  
y = sin(x) ;
```

A. Yes

B. No

Nicer Plots

The default plot line thickness is 1, but you can change it:

```
plot ( xlist, ylist, 'LineWidth', 3 );
```

The default plot line color is blue; You can make it red(r) (or green(g), black(k), cyan(c), yellow(y), magenta(m):

```
plot ( xlist, ylist, 'r-' );
```

You can add grid lines, titles, and labels:

```
plot ( xlist, ylist );
```

```
grid ( on );
```

```
title ( 'This is my plot!' );
```

```
xlabel ( 'Centuries' );
```

```
ylabel ( '←- How many ants I saw-->' );
```

Saving Plots

A MATLAB plot appears in a separate figure window. Under its FILE menu is an option to "save as" which includes options:

Save As: specify a file name

Where: specify a directory

Format: a menu including JPEG and PNG

You can also use a command, such as:

```
print ( '-djpeg', 'myfile.jpg' ); <- JPEG
```

or

```
print ( '-dpng', 'myfile.png' ); <- PNG
```

Tricks with plot()

The `plot()` command just connects each pair of (x,y) values to the next one.

When we plot formulas, we naturally assume that the x values must be a list that goes from left to right, and the y values are determined by a formula.

But there are lots of other things we can do too.

Let me show you how to draw a triangle, a square, and a circle.

Try for a Triangle: triangle_plot.m

A triangle involves three points, so we might expect the following to work:

```
txlist = [ 0.0, 2.0, 1.0 ];
```

```
tylist = [ 0.0, 1.0, 3.0 ];
```

```
plot ( txlist, tylist );
```

...We almost got our triangle, but what's missing...and why?...and how do we fix it?

Can We add a Square?

Let's use different names for the data, so we still have our triangle data around:

```
sxlist = [ 0.0, 1.0, 1.0, 0.0, 0.0 ];  
sylist = [ 0.0, 0.0, 1.0, 1.0, 0.0 ];  
plot ( sxlist, sylist, 'r-' );
```

Hey, where did our triangle go?

Fix #1: Two plots in one call:

To plot the square and the triangle together, the plot command will let us list the two sets of data in one command:

```
plot ( txlist, tylist, 'b-', ...  
      sxlist, sylist, 'r-' );  
title ( 'Triangle and Square together!' );
```

Fix #2: Hold On/Hold Off

We can make several plot commands all appear on the same image, if we first say "hold on", then make our plots, and then say "hold off":

```
hold on
```

```
plot ( txlist, tylist, 'g-' );
```

```
plot ( sxlist, sylist, 'm-' );
```

```
[...MORE PLOT COMMANDS POSSIBLE..]
```

```
hold off
```

To erase all previous plot information before issuing a "hold on" command, you can type

```
clf % <- "Clear the figure"
```

The "fill" command: triangle_fill.m

MATLAB's fill() command:

```
fill ( xlist, ylist, color );
```

lets us list the corners of a shape, and fill it with a color. The easy colors are

'r', 'g', 'b' (red/green/blue),

'c', 'm', 'y' (cyan/magenta/blue)

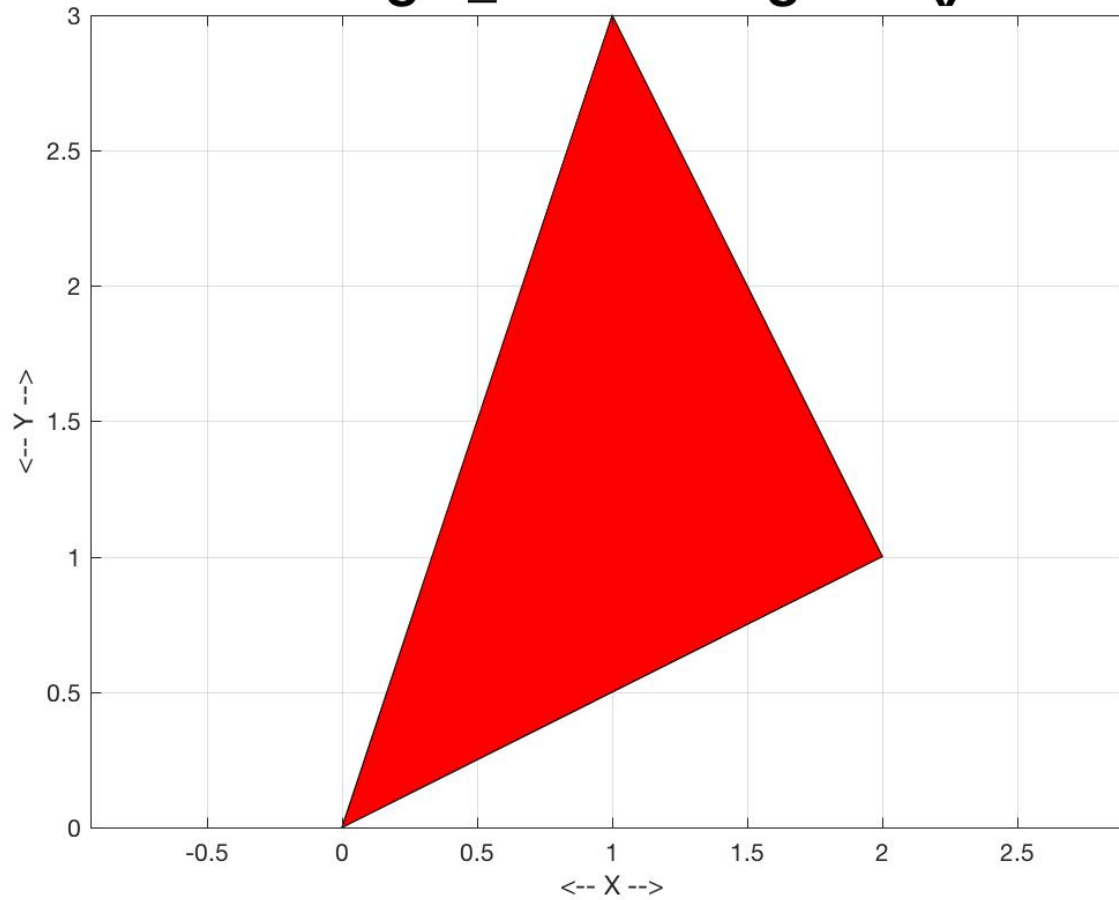
'k', 'w' (black/white)

Drawing a filled triangle: triangle_fill.m

```
txlist = [ 0.0, 2.0, 1.0 ];  
tylist = [ 0.0, 1.0, 3.0 ];  
fill ( txlist, tylist, 'r' );  
grid on  
title ( 'Triangle fill()', 'FontSize', 24 );  
xlabel ( '<-- X -->' );  
ylabel ( '<-- Y -->' );  
axis equal  
print ( '-djpeg', 'triangle_fill1.jpg' );
```

A filled triangle

triangle_fill1: Triangle fill()



Last Trick: Sine, Cosine, Circle

```
alist = linspace ( 0.0, 2*pi, 101 );
```

```
clist = cos ( alist );
```

```
slist = sin ( alist );
```

```
hold on
```

```
plot ( alist, clist );
```

```
plot ( alist, slist );
```

```
title ( 'Cos(x) and Sin(x) from 0 to 2pi' );
```

```
hold off
```

Draw a Circle

```
plot ( clist, slist );
```

```
%
```

```
% That doesn't look right! Try again!
```

```
%
```

```
axis equal;
```

```
plot ( clist, slist ); ← It's a circle now!
```

"axis equal" tells MATLAB to use the same scale in both X and Y directions.