1: Interactive Graphing

Matlab, which is a name contracted from MATrix LAB, is a widely used scientific software package. The goal of these notes is to teach you how to use it to

- construct graphs of curves and surfaces
- perform symbolic calculations, such as differentiation and integration
- write programs to facilitate carrying out the first two tasks
- present your results as a finished report suitable for printing or presentation.

In this section we will focus on interactive graphing. Later, we will consider aspects of Matlab that deal with numerical calculations, done both interactively and via programming.

We will make use of the Symbolic Toolbox, which is installed on the system in the Math department lab and comes bundled as part of the Student Edition of Matlab. Using the Symbolic Toolbox allows us to employ a language and style that are closer to everyday mathematics. This often makes it easier to figure out how to do things.

As we said, this chapter will involve a lot of interactive work with the program. This means "show and tell." We will demonstrate what to do in class, but not write much about how to do it. So coming to the lab will be very useful in figuring out how to complete the assignment described later.

First we will create plots of expressions in one variable.

1.1 Simple Plots

First declare a symbolic variable:

```
syms x
```

Matlab gives no response, but having declared the variable symbolic, we can enter expressions in this variable.

```
y=sin(x)
```

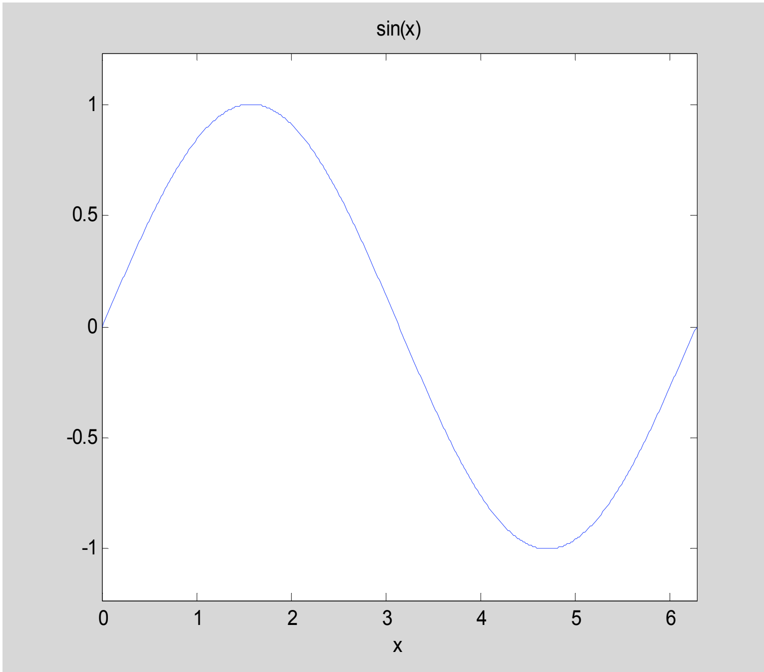
```
y =
sin(x)
```

After a brief delay while Matlab loads the package of symbolic commands, the system returns the definition you proposed.

Example 1.1: Plot $y = \sin(x)$ on the interval $[0, 2\pi]$. Enhance the graph with additional features.

Solution: We enter the following command at the prompt.

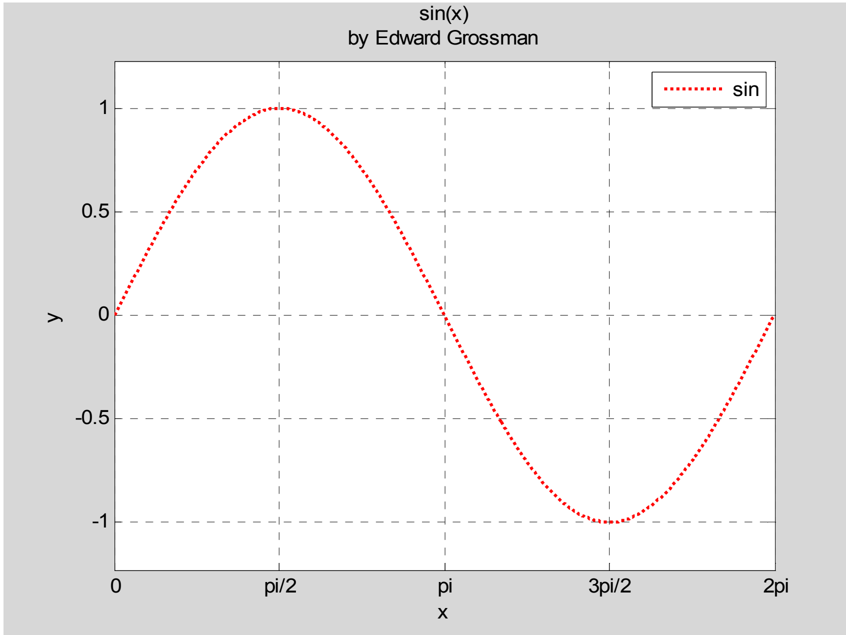
```
ezplot(y,[0,2*pi])
```



This figure appears in a new window, called a Figure window. We can customize many features of the graph interactively using the window. Among the features we will learn to change are

a) adding grid lines
b) adding or changing labels and tick marks on the axes
c) changing the color of the graph
d) changing the plotting style of the graph, for example, dotted, or a heavier line
e) adding a legend (useful when we have more than one graph displayed on the same axes)
f) adding or changing the graph title
g) changing the tick marks on the axes

All of these can be done using the Plot Tools icon at the top of the Figure window. After some experimentation, you should be able to use this tool to produce the following picture:



■

Now suppose we want to add another graph to this one.

Example 1.2: Plot the *sine* function and its derivative on the same plot

Solution: In this case we know the derivative and we can simply define it directly. For more complicated functions, we can have Matlab compute it for us.

```
y2=cos(x)
```

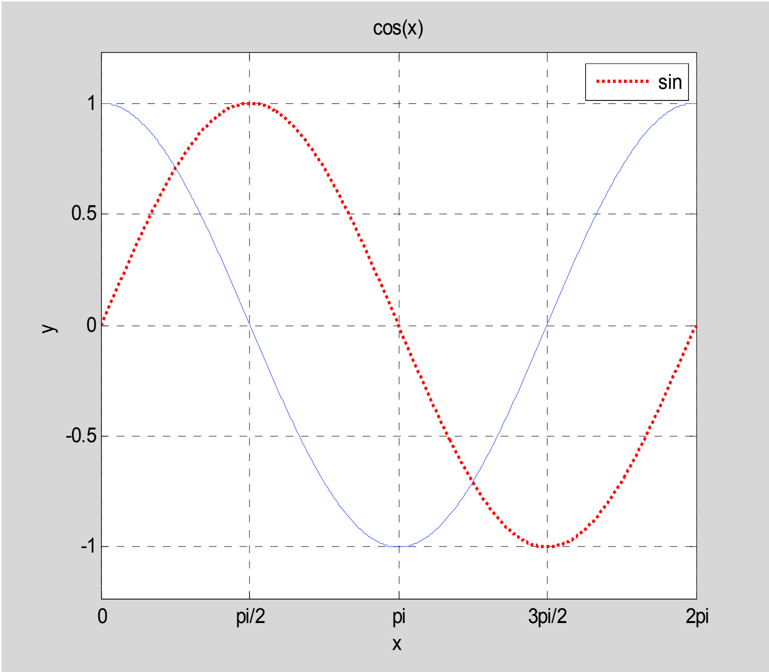
```
y2 =
cos(x)
```

In order to add the graph of $\cos(x)$ to the previously drawn graph, we must tell Matlab not to erase the first graph when it draws the second one. This is done by typing the command

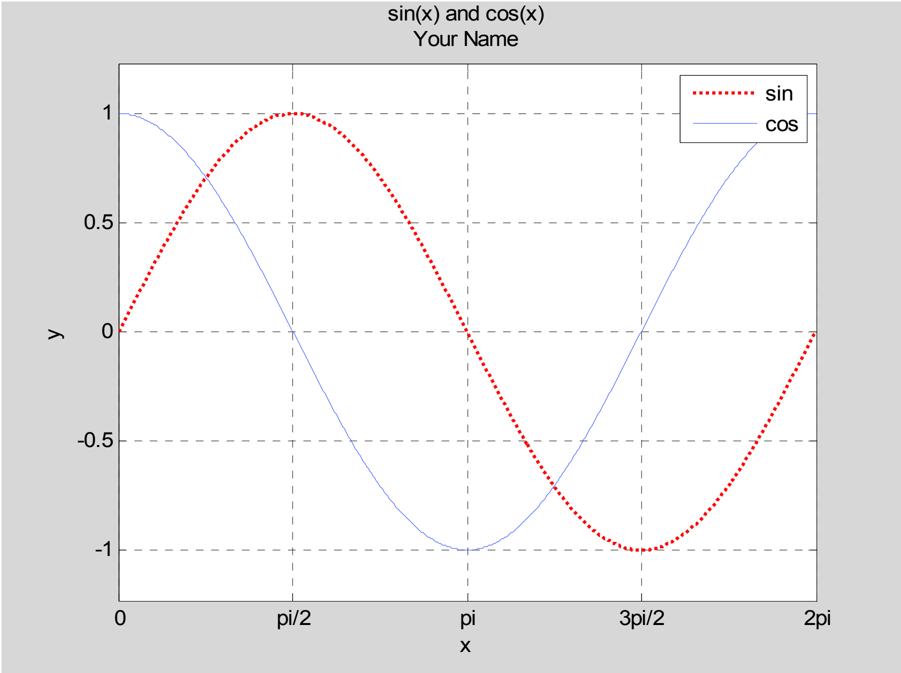
`hold on` .

Then we simply repeat the `ezplot` command with y2 instead of y.

```
ezplot(y2,[0, 2*pi])
```



Besides adding the new graph to the figure window, Matlab changed the title and did not update the legend. We need to manipulate these features manually, using the Plot Tools windows. See if you can produce the following picture:



■

That was pretty easy. But there is one more challenge for this lesson. In Example 1.2 both graphs produced the same range. When the ranges are different, which is usually the case, the `ezplot` command uses the range of the last graph plotted. This may not produce the most useful looking graph.

Example 1.3: Plot the expression $x\sin(4x)$ and its derivative on the interval $[0,2\pi]$.

Solution: This time we will use the Matlab command `diff` to compute the derivative. We can proceed as follows

```
y=x*sin(4*x); dy=diff(y,x)    % the diff command computes the derivative
```

```
dy =
sin(4*x)+4*x*cos(4*x)
```

(The semi-colon used in the input line suppresses the display of the output of the first command).

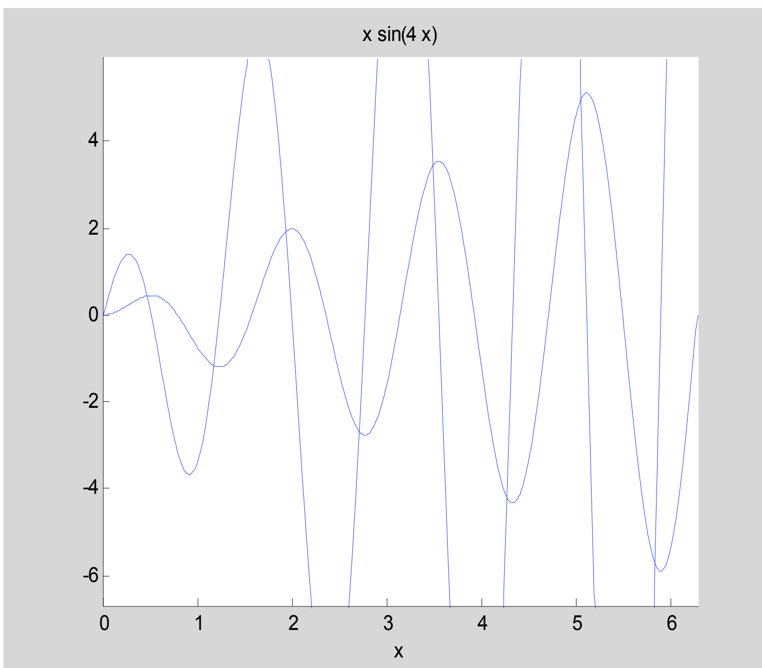
Before we create the plots, we must clear the previous graphs. We do this via

```
clf ,
```

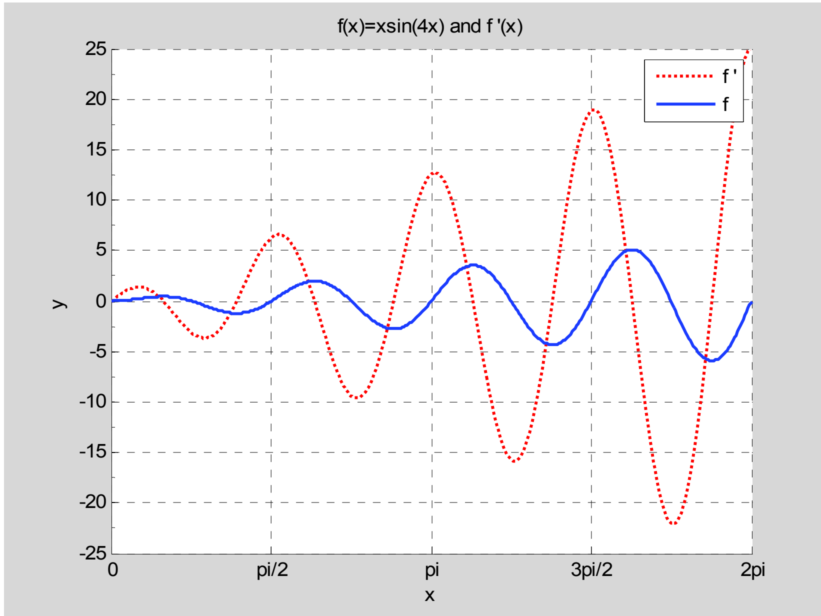
which stands for "clear figure." We must again ensure that "hold on" is in effect.

Now we use our `ezplot` command:

```
hold on;ezplot(dy,[0,2*pi]);ezplot(y,[0,2*pi])
```



Because we plotted the derivative first and the function second, part of the first graph was chopped off. Interactively, we can recover a more useful picture by adjusting the range of the axes. (This can also be done by simply reversing the plotting order, but discovering the best order would be time consuming when we had to deal with more than two expressions.) Use Plot Tools to obtain the picture below by changing the range of the *y*-axis and modifying the labels and tickmarks on the *x*-axis.



■

1.2 Summary

OK, that's it for the first lesson. Here's what we discussed in terms of commands:

`clf` – clear graphics figure

`diff` – compute a symbolic derivative with respect to a variable

`ezplot` – name says it all

`hold on` - draw new graph on top of old one (there is also `hold off` if you want to go back to the overwrite mode)

`syms` – declaring symbolic variables

Your exploration with Plot Tools should have also taught you a lot about the structure of Matlab's graphs. You also became familiar with manipulating commands in the Command Window. In addition to items a) to g) listed earlier, you should also know how to change the range of the axes. Not bad for the first lesson. Later we will see how to control these features in a program, so they do not have to be repeated when you need to change the input to a problem.

1.3 Exercises

In each of the following exercises you should produce the specified plot on the stated interval. Your plot should contain the following features:

a) a legend identifying the graph(s)

b) distinct line styles for the two graphs (one dotted, one solid) and different colors if you are using a color printer

c) both graphs displayed without chopping any values in the range

d) labels and appropriate tick marks on both axes

e) graph should display grid lines

f) A title similar to the one produced in the last graph above. If you will be handing in the output, the title should also contain YOUR NAME.

Exercise 1.1 Let $y = \sin(x^2)\cos(x)$. Produce a plot of y and y' over the interval $[-\pi, \pi]$. The x-axis tick marks should be at the points with coordinates $-\pi, -\pi/2, 0, \pi/2, \pi$.

Exercise 1.2 Let $y = \dfrac{2x}{\sqrt{x^2+1}}$. Produce a plot of y and y' over the interval $[-2, 2]$. (Note: You can enter the square root either using exponents or the Matlab function `sqrt`.)

Exercise 1.3 Plot the expressions $y = \sqrt{1-x^2}$ and $y = -\sqrt{1-x^2}$ on the interval $[-1,1]$. The two graphs should form the circle $x^2 + y^2 = 1$. Look up the `axis` command in Help to see how to get the picture to actually look like a circle, rather than an ellipse.

Exercise 1.4 Let $y = \dfrac{1}{5}\left(\dfrac{\sin 5x}{\sin x}\right)^2$. Produce a plot of y and y' over the interval $[-\pi, \pi]$.

From the graph, what is the value of $\displaystyle\lim_{x\to 0} \frac{1}{5}\left(\frac{\sin 5x}{\sin x}\right)^2$? Confirm the result separately using what you know about limits.

Exercise 1.5 What is the smallest positive value of x (in radians) such that $\tan x = x$? Estimate the answer by plotting the graphs of $y = \tan x$ and $y = x$ on a suitable interval and examining the points of intersection. You can use the Data Cursor icon in the figure window to estimate coordinates of points as well as to leave a data marker that can be printed to show your estimate.