Functions and Their Graphs

Michael Penna, Indiana University – Purdue University, Indianapolis

Objective

The objective of this project is to illustrate one way a user can define functions in MATLAB, and one way to graph functions in MATLAB. The commands discussed in this project are all "general purpose" MATLAB commands: they can be used whether or not MATLAB has been configured to use its Symbolic Toolbox.

Narrative

Since MATLAB's library of functions cannot include all functions of interest to you, MATLAB provides several ways for you to define your own functions. In this project we illustrate how you can define your own functions using "in-line" functions.

MATLAB also provides several ways to graph functions, not only its own built-in functions, but also "in-line" functions. In this project we illustrate how you can graph functions using the `ezplot` command:

`ezplot(f(x),[a,b])`	graphs $f = f(x)$ over the interval $[a, b]$.
`ezplot(f(x),[a,b,c,d],figure(`n`))`	graphs $f = f(x)$ over the interval $[a, b]$, restricting output values to the interval $[c, d]$, in Figure n.

As you will see, the graphs of functions appear in "figure windows" which (magically!) appear when a command such as `ezplot` is issued.

Task

1. Type the following commands into MATLAB.

```
>> % Your name, today´s date
>> % Functions and Their Graphs
>> % Task 1:  In-line functions
>> clear all
>> sqr(3)
```

MATLAB should respond to this last line of code with "`???  Undefined command/function ´sqr´.`", illustrating that MATLAB has no built-in squaring function `sqr`.

Continue by typing the following the following commands into MATLAB.

```
>> sqr = inline(´x^2´)
>> sqr(3)
```

The first of command defines the function `sqr(x) = x^2`, and in response to the second command MATLAB should respond with an answer of 9, indicating that you have successfully defined your first in-line MATLAB function.

2. Continue by typing the following commands in the left-hand column below into MATLAB.

`>> % Task 2:  Graphs of functions`	
`>> ezplot(´sin(x)´,[-pi,pi])`	Graph $f(x) = \sin x$ over the interval $[-\pi, \pi]$. Observe how the window `Figure 1` magically appears!
`>> ezplot(sqr,[-1.0,1.0])`	Graph $f(x) = x^2$ over the interval $[-1, 1]$.

Note the difference between the way `ezplot` is used to graph a built-in function and an in-line function. Also note how our first graphic was over-written by our second graphic when we drew the second graphic.

We can control which figure window MATLAB draws in by adding the appropriate argument to `ezplot`. To illustrate this, type the following commands into MATLAB. The first graphs $f(x) = \sin x$ over the interval $[-\pi, \pi]$ in `Figure 2`, and the second graphs $f(x) = \sin x$ over the interval $[-\pi, \pi]$, restricting output values to the interval $[-0.5, 0.5]$ in `Figure 3`.

```
>> ezplot(´sin(x)´, [-pi,pi], figure(2))
>> ezplot(´sin(x)´, [-pi,pi,-0.5,0.5], figure(3))
```

If you want to graph more than one function in a figure window, you can do so by specifying the figure window in which you wish to draw, putting a “hold” on to the window while you graph in it, and taking the “hold” off the window and displaying it when you are done graphing. To illustrate this, type the following commands in the left-hand column below into MATLAB.

`>> figure(4)`	
`>> hold on`	Get ready to plot, but don’t quit plotting until you’re told to.
`>> ezplot(´sin(x)´, [-pi,pi])`	Graph $f(x) = \sin x$ over the interval $[-\pi, \pi]$.
`>> ezplot(´sin(2*x)´, [-pi,pi])`	Graph $f(x) = \sin 2x$ over the interval $[-\pi, \pi]$.
`>> ezplot(´sin(3*x)´, [-pi,pi])`	Graph $f(x) = \sin 3x$ over the interval $[-\pi, \pi]$.
`>> ezplot(´sin(4*x)´, [-pi,pi])`	Graph $f(x) = \sin 4x$ over the interval $[-\pi, \pi]$.
`>> hold off`	Stop plotting, and display the figure window.

At this time, make a hard copy of MATLAB’s Command Window and figure windows. Then:

3. On each graphic you produced in this project, draw and label the coordinate axes by hand.

Your hard copies of MATLAB’s Command Window and figure windows will be your lab report for this project. Staple these hard copies together, the Command Window on top, and then your figure windows, in order.

Comments

1. A great deal more can (and later will) be said about graphing functions in MATLAB. The objective of this project was not to be encyclopedic, however: it was just to provide enough background for us to be able to get started using MATLAB. The more you use MATLAB the more you will learn about it. One of the best ways to learn about it is to use the MATLAB helpdesk: it is a valuable tool. Again, to access MATLAB’s helpdesk, simply type `helpdesk` in MATLAB’s Command Window.

2. Just as we begin each project with the command `clear all` (to clear MATLAB’s memory and “start with a clean workspace”) we will henceforth also begin each project with the command `close all` to close any MATLAB figure windows (remaining from previous work).