

Parametric Equations

David Arnold
Multivariable Calculus

January 12, 1998

Abstract

Matlab is used to draw the graphs of parametric equations. **Prerequisites.** None

1 Vectors in Matlab

You enter a *row* vector in Matlab as follows:¹

```
>> x=[1,2,3,4,5]
```

x =

```
    1    2    3    4    5
```

The commas are optional.

```
>> x=[1 2 3 4 5];
```

Note that terminating a command with a semicolon suppresses output to the display. Use semicolons to separate rows and create a *column* vector.

```
>> x=[1;2;3;4;5]
```

x =

```
    1  
    2  
    3  
    4  
    5
```

Matlab's transpose operator ² is the single apostrophe (`'`).

¹The `>>` symbol is Matlab's prompt. You must enter the command that appears after the prompt. You must hit the Enter key on your keyboard to execute the command.

²The transpose of a row vector is a column vector, and vice-versa.

```
>> x=x=[1,2,3,4,5]'
```

```
x =
```

```
    1
```

```
    2
```

```
    3
```

```
    4
```

```
    5
```

```
>> x'
```

```
ans =
```

```
    1    2    3    4    5
```

1.1 Incremental Notation

The Matlab construct `start:increment:finish` is particularly useful for entering regularly spaced data.

```
>> x=1:3:13
```

```
x =
```

```
    1    4    7   10   13
```

You can also use a negative increment.

```
>> x=20:-2:10
```

```
x =
```

```
   20   18   16   14   12   10
```

And you can get quite fancy.

```
>> x=0:pi/2:2*pi
```

```
x =
```

```
    0    1.5708    3.1416    4.7124    6.2832
```

1.2 Vector Arithmetic

Vector addition and subtraction of equal length vectors occurs in a natural way.

```
>> x=1:5,y=6:10
```

```
x =  
    1    2    3    4    5
```

```
y =  
    6    7    8    9   10
```

```
>> x+y
```

```
ans =  
    7    9   11   13   15
```

```
>> x-y
```

```
ans =  
   -5   -5   -5   -5   -5
```

You can multiply a vector by a scalar.

```
>> x=(1:5)'
```

```
x =  
    1  
    2  
    3  
    4  
    5
```

```
>> y=2*x
```

```
y =  
    2  
    4  
    6  
    8  
   10
```

1.3 Element-wise Operations

Matlab is a matrix based environment. Therefore, you cannot multiply two vectors of equal size.

```
>> x=1:5,y=6:10
```

```
x =
```

```
    1    2    3    4    5
```

```
y =
```

```
    6    7    8    9   10
```

```
>> x*y
```

```
??? Error using ==> *
```

```
Inner matrix dimensions must agree.
```

You can, however, multiply vectors of equal length on an element-wise basis.

```
>> z=x.*y
```

```
z =
```

```
    6   14   24   36   50
```

Make sure that you note that each element in z is the product of the corresponding entries in x and y . In a similar manner, you can raise each element of x to the second power.

```
>> z=x.^2
```

```
z =
```

```
    1    4    9   16   25
```

In a similar manner, you can divide each element of x by the corresponding element of y .

```
>> z=x./y
```

```
z =
```

```
    0.1667    0.2857    0.3750    0.4444    0.5000
```

Many of Matlab's built-in functions perform on a vector in an element-wise manner.

```
>> x=0:pi/2:2*pi
```

```
x =
```

```

0    1.5708    3.1416    4.7124    6.2832

>> y=sin(x)

y =

0    1.0000    0.0000   -1.0000    0.0000

```

2 Plotting Parametric Equations

Here is a set of parametric equations, the parameter being t .

$$x = \cos t \quad ((1a))$$

$$y = \sin t \quad ((1b))$$

If you were plotting these equations by hand, you would begin by making a table of values. First, select some arbitrary t -values.

t	x	y
0		
$\pi/2$		
π		
$3\pi/2$		
2π		

Substitute each t -value in the table into equations (1a) and (1b).

t	x	y
0	1	0
$\pi/2$	0	1
π	-1	0
$3\pi/2$	0	-1
2π	1	0

There are now a number of different plots that you can make. You can plot x versus t , y versus t , or y versus x . And, of course, to obtain a smooth plot, we'll need to plot more than the 5 points we have in our table. Matlab makes this task effortless. The following code should produce an image similar to that in Figure 1.

```

>> t=0:.1:2*pi;
>> x=cos(t);
>> y=sin(t);
>> plot(x,y)

```

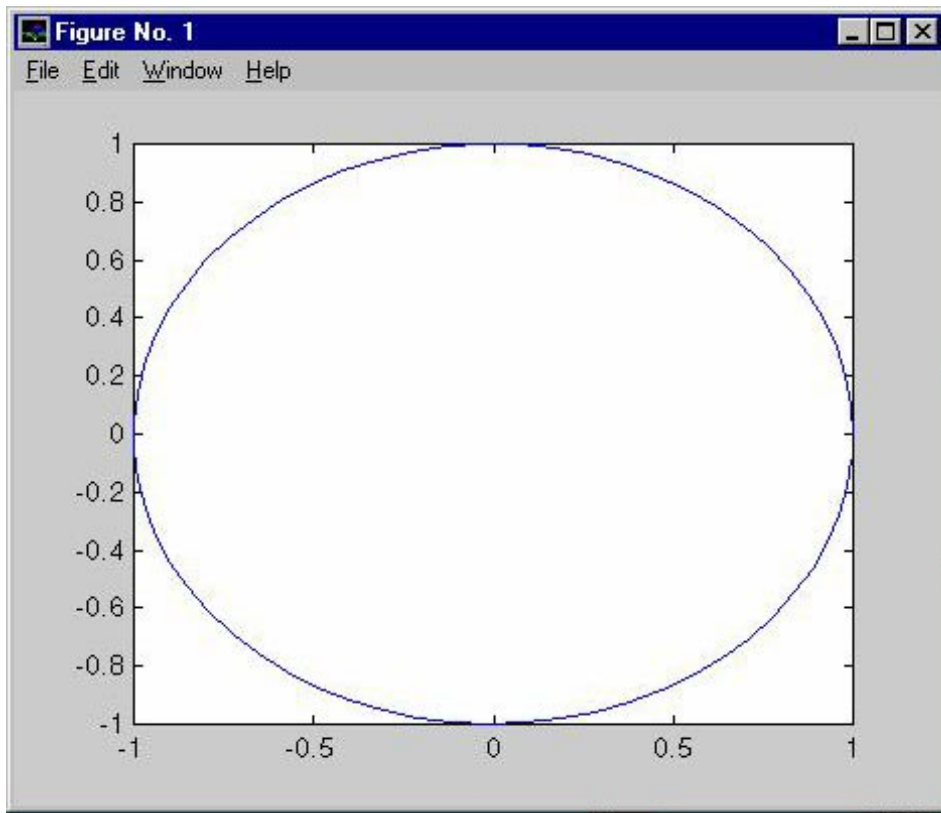


Figure 1:

2.1 Lissajous Curves

Let's plot the graph of the following set of parametric equations.

$$x = 2 \cos 3t \quad ((2a))$$

$$y = 3 \sin 5t \quad ((2b))$$

The following sequence of commands should give a plot similar to that in Figure 2.³

```
>> t=linspace(0,2*pi,500);
>> x=2*cos(3*t);
>> y=3*sin(5*t);
>> plot(x,y)
>> xlabel('x-axis')
>> ylabel('y-axis')
>> title('x=2*cos(3*t), y=3*sin(5*t)')
```

2.2 Comets

For a bit of fun with this “Lissajous Curve,” enter the following command at the Matlab prompt.

```
>> close all, comet(x,y)
```

Think of the parametric equations (2a) and (2b) as describing the position (x, y) of a particle at time t . Matlab's `comet` command gives you an excellent feel for the motion of the particle with respect to time.

2.3 A Parabola

Plot the image of the following set of parametric equations over the time interval $[-2, 2]$.

$$x = 2t + 2 \quad ((3a))$$

$$y = t^2 - 1 \quad ((3b))$$

The first two steps are straightforward.

```
>> t=-2:.01:2;
>> x=2*t+3;
```

The following produces an error.

³`linspace(0,2*pi,500)` produces 500 equally spaced points starting with 0 and ending with 500. Type `help linspace` at the Matlab prompt to learn more about the `linspace` command.

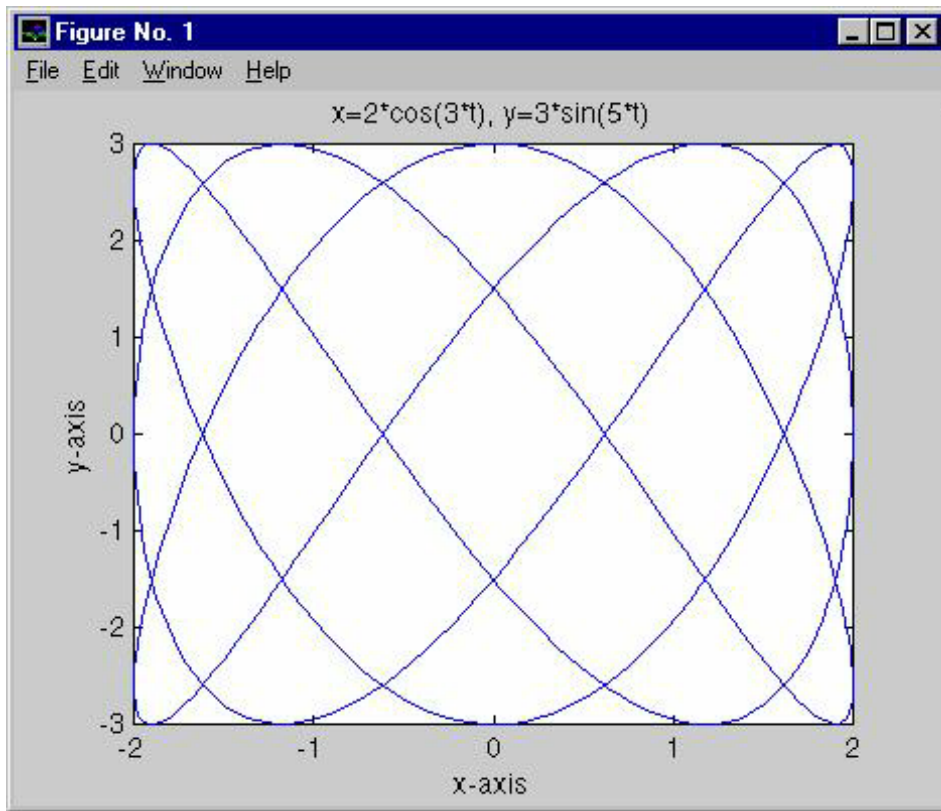


Figure 2:

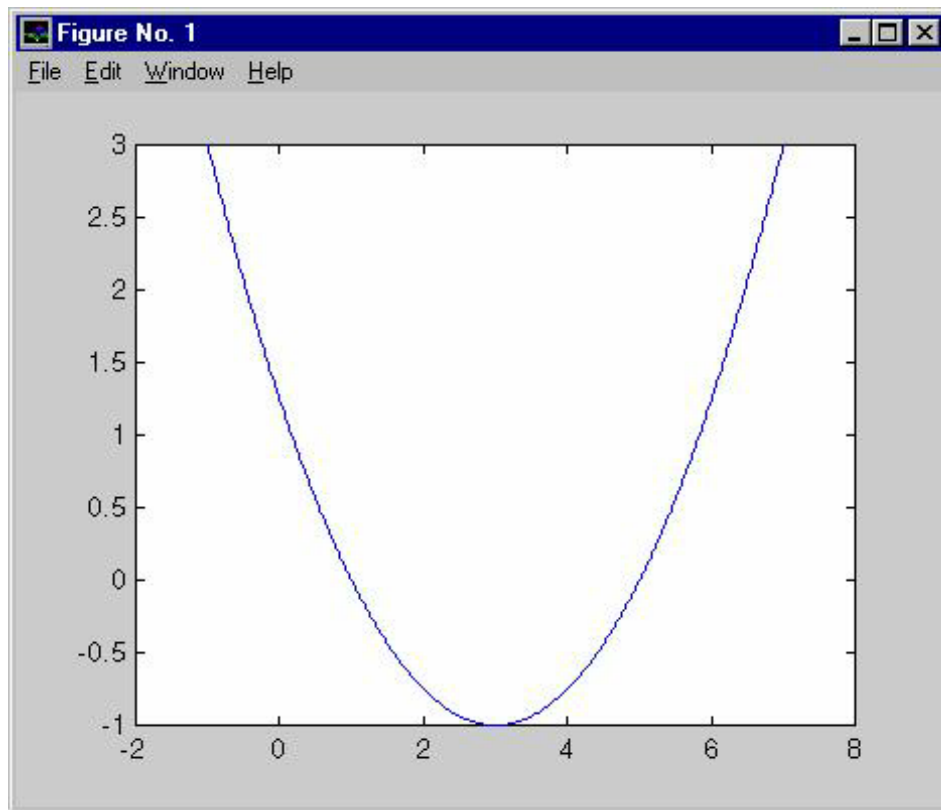


Figure 3:

```
>> y=t^2-1
??? Error using ==> ^
Matrix must be square.
```

This error is expected, particularly in light of the discussion in the section “Element-wise Operations.” What you actually want to do is square each *element* of t , then subtract 1.

```
>> y=t.^2-1;
```

The following command should now produce an image similar to that in Figure 3.

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

Be sure to also try the `comet` command on this data.

2.4 An Astroid

The curve defined by the following set of parametric equations is called an “astroid.”

$$x = \cos^3 t \quad ((4a))$$

$$y = \sin^3 t \quad ((4b))$$

When you cube, you want to cube *each element*, so use the `.^` notation for this purpose. The following sequence of commands should produce an image similar to that in Figure 4.⁴

```
>> t=linspace(0,2*pi,500);  
>> x=(cos(t)).^3;  
>> y=(sin(t)).^3;  
>> plot(x,y)  
>> axis square
```

Be sure to try the `comet` command on this data.

2.5 Homework

Use Matlab to plot each of the following sets of parametric equations. Obtain a printout of each plot and hand in next class.

1. $x = t - \sin t, y = 1 - \cos t, 0 \leq t \leq 2\pi$
2. $x = 1 + 2t, y = 4 - t^2, 0 \leq t \leq 1$
3. $x = 2 \cos t + \cos 2t, y = 2 \sin t - \sin 2t, -\pi \leq t \leq \pi$
4. $x = 4(1 + \sin t), 4 \cos t(1 + 2 \sin t), -\pi \leq t \leq \pi$

⁴The `axis square` command “squares up” the image. For more help on the `axis` command, type `help axis` at the Matlab prompt.

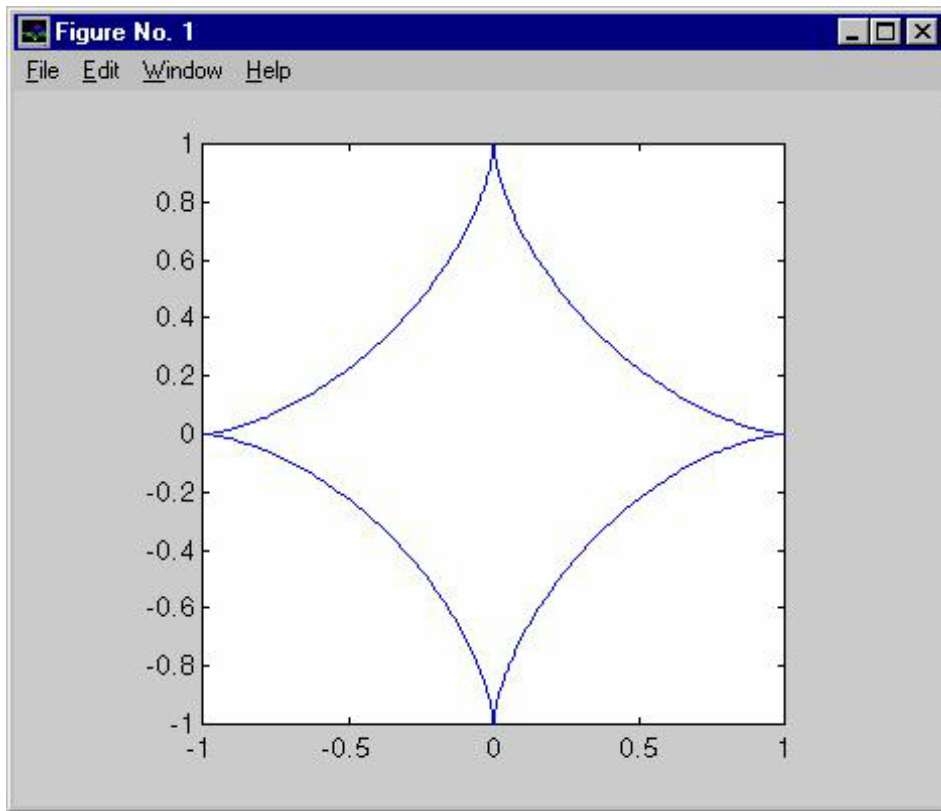


Figure 4: