Second 340 Matlab assignment

Vector operations with Matlab

You can take the dot and cross product as follows:

```
>>P0 = [ 1, 3, 6]
>>P1 = [0 -2 8]
>>a = dot(P0,P1)
>>B = cross(P0,P1)
```

Then a will be 42, the dot product of P0 and P1, and B will be the vector $(36, -8, -2)$ which is the cross product.

```
>>dot(P0,B)
```

gives the answer 0 as it should.

You can find the length of a vector by

```
>>sqrt(dot(P0,P0))
```

or you can use the built-in function norm

```
>>norm(P0)
```

Besides these vector functions, Matlab has many functions which work on scalar quantities. Some useful functions are `abs`, `cos`, `sin`, `tan`, `sqrt`, `log`. `exp`, `acos`, `asin`, `atan`. Type `help exp` for example to find out about `exp`. Note that in Matlab, `log(x)` computes the natural logarithm $\ln(x)$.

.

Example: The length of the projection of b to a is $\frac{|a \cdot b|}{\|a\|}$ so to find the length of the projection of $(0, 1, 3)$ to $(1, 1, 5)$ you could type the following (since `abs` is the absolute value.).

```
>> b = [0 1 3]
>> a = [1 1 5]
>> abs(dot(a,b))/norm(a)
```

Example: Let P0 and P1 be as above and let $P2 = (0, 1, 2)$. Find an equation of the plane through P0, P1, and P2.

```
>>P2 = [0 1 2]
>>N=cross(P1-P0,P2-P0)
```

Matlab gives the answer $N = (24, -6, -3)$ so you know the equation is $24(x - 1) - 6(y - 3) - 3(z - 6) = 0$.
Example: Find the distance from P1 to the line through P0 with direction $(1, -1, 3)$

```
>>L=[1 -1 3]
>>dist = norm(cross(P1-P0,L))/norm(L)
```

I should point out that Matlab does not understand the $\mathbf{i}, \mathbf{j}, \mathbf{k}$ vector notation. Try typing in `3*i+2*j` and see what happens. In Matlab, $\mathbf{i}$ and $\mathbf{j}$ both stand for the complex number $\sqrt{-1}$.

Symbolic math

To use matlab for symbolic math such as differentiation you need to tell it what variables you want to be symbolic with a `syms` command. For example

```
>>syms t x y
```

says that t, x, and y will all be symbolic variables. All the examples below presume you have first typed in a `syms t x y` command. We can now define functions in these symbolic variables. For example to define $r(t) = t\mathbf{i} + e^t\mathbf{j} + \sin t\mathbf{k}$ and $f(x, y) = \cos(x \ln y)$ type in

```
>> f = cos(x*log(y))
>> r = [t,  exp(t), sin(t)]
```

You can differentiate such symbolic functions with the `diff` command. In the following, $g = \partial^2 f/\partial y \partial x$.

```
>> dfdx = diff(f,x)
>> dfdy = diff(f,y)
>> g = diff(dfdx,y)
```

The vector operations `dot` and `cross` work for symbolic functions. Unfortunately the `norm` command does not work for symbolic vectors. Also, while the `dot` command gives the correct answer, its result looks fairly complicated since it is meant to work with with complex valued functions as well as real valued functions. Consequently its result is peppered with `conj()` (which stands for complex conjugate). To avoid this you can use the `sum( .* )` operation which performs the real dot product. So to calculate $r'(t) \cdot r''(t)$ and $\mid r'(t) \mid$ you could type

```
>> v = diff(r,t)
>> a = diff(v,t)
>> sum(v.*a)
>>speed = sqrt(sum(v.*v))
```

(For those who wonder why this works, `v.*a` multiplies the corresponding entries of v and a, and `sum` adds up the resulting entries.)

Evaluating symbolic functions

Matlab has a `subs` command which substitutes something in for a variable. Let us find $r(.5)$ and $f(1,2)$.

```
>> subs(r,{t},{.5})
>> subs(f,{x,y},{1,2})
```

This substitutes .5 for t in r and substitutes $x = 1$ and $y = 2$ in f.

You can also use the subs command to compose functions. For example, if $s(t) = t^2\mathbf{i} - t^3\mathbf{j}$ then $f(s(t))$ can be calculated as

```
>>subs(f,{x,y},{t^2,-t^3})
```

It may also be calculated as follows.

```
>>s = [t^2 -t^3]
>>subs(f,{x,y},s)
```

If you are evaluating a function many times, it may be worthwhile converting it to an inline function or anonymous function which can be evaluated using fewer keystrokes and more natural syntax. This is most easily done with scalar valued functions. Below, the scalar valued function `f` is converted to an inline function `inf` and then evaluated at $(1, 2)$ and $(3, 4)$.

```
>> inf = inline(vectorize(f),'x','y')
>> inf(1,2)
>> inf(3,4)
```

Inline functions are relatively slow and recent versions (7.0 and greater) of matlab introduced an alternative, anonymous functions. To convert `f` to an anonymous function and evaluate:

```
>> anf = @(x,y) cos(x*log(y))
>> anf(1,2)
>> anf(3,4)
```

Drawing graphs and contour plots.

To draw the graph of $f(x,y) = x^2 - y\sin(x)$ for $-1 \le x \le 2$ and $0 \le y \le 4$ you can use the following commands:

```
>> syms x y
>> f = x^2 - y*sin(x)
>> ezsurf(f,[-1 2 0 4])
```

2

To get a different look you could give the command `ezmesh(f,[-1 2 0 4])` and to plot the level curves you could use the command `ezcontour(f,[-1 2 0 4])`.

Note that as usual we start our session with a syms command. In the following examples we assume that this syms command has already been typed in, otherwise they will not work.

If you wanted to draw two (or any number) of graphs in the same window you could use the `hold on` command. For example:

```
>> ezsurf(x^2+4*y^2-12,[-4 -2 -2 0])
>> hold on
>> ezsurf((4*x+y^2+19)/8,[-4 -2 -2 0])
>> hold off
```

displays the two surfaces at once.

Using Matlab as a microscope

One way of visualizing the $\lim_{(x,y)\to(a,b)} f(x,y)$ is to look at the graph $z = f(x,y)$ near $x = a$, $y = b$. If we look at the graph with a micorscope using higher and higher magnification, and the graph seems to be getting close to a single z value L, then probably $\lim_{(x,y)\to(a,b)} f(x,y) = L$. But if the graph does not seem to be getting close to any one z value then probably the limit does not exist.

We can also see if f is differentiable at (a,b). If the graph under high magnification seems close to being a plane, then f is probably differentiable at (a,b), and if not, it probably is not differentiable.

For example, consider $f(x,y) = x^4 + y^4$ and let us examine its limit and differentiablility at the origin $(0,0)$.

```
>>f = x^4+y^4
>>a=[-1 1 -1 1]
>>ezsurf(f,a)
>>ezsurf(f,a/10)
>>ezsurf(f,a/100)
>>ezsurf(f,a/1000)
>>ezsurf(f,a/10000)
>> axis equal
```

We start out looking at the graph with $-1 \le x, y \le 1$. Then we look at the graph with $-1/10 \le x, y \le 1/10$, in other words magnified by 10. Then we look at it at 100, 1000, and 10000 magnification. The last command `axis equal` is just to make sure the magnification of the z axis, is the same as that of the x and y axes. Otherwise matlab in this case magnifies the z coordinate much more. We see from the final picture that the limit of f appears to exist and be 0, since the z coordinates near the center of the graph are all close to zero. Also f looks differentiable at 0 since the graph looks like a plane (once we give the `axis equal` command).

For another example,

```
>>f = x^2/(x^2+y^2)
>>a=[-1 1 -1 1]
>>ezsurf(f,a)
>>ezsurf(f,a/10)
>> axis equal
>>ezsurf(f,a/100)
>>ezsurf(f,a/1000)
>> axis equal
```

Here we see that no matter how much we magnify, the z coordinate goes from 0 to 1, thus the limit does not exist. Since the z range is so much bigger than the x and y ranges, the `axis equal` command results in a tall skinny graph.

You should be aware that it is always possible to fool the computer with a well chosen example. For example if $f(x,y) = y^6/(y^6 + (x - y^2)^2)$ then under high magnification matlab's drawing of the graph will look like a plane through the origin. So you might expect that f is differentiable at $(0,0)$ and its limit is 0. However, note that $f(y^2, y) = 1$ so $\lim_{(x,y)\to(0,0)} f(x,y)$ does not exist and f is not differentiable at the

origin. (The limit does not exist because as (x, y) approaches the origin along the parabola $x = y^2$ then f approaches 1, but as (x, y) approaches the origin along any line $y = mx$ then f approaches 0.)

Problems to turn in, Due Monday, Oct. 30

You are encouraged to work on these problems in groups of two or three. Each group should turn in only one copy of the assignment, (but put all your names on the assignment you turn in). Your assignment should include a printout of your matlab session (i.e., the matlab commands you gave and matlab's responses). Answers should be clearly indicated and can be hand written or typed. All the problems below should be solved using Matlab, even if they are easy to do by hand.

Problem 1: Find the unit vector in the direction of $(1, -2, 4)$.

Problem 2: Find the angle (in radians) between the vectors $(1, -2, 3)$ and $(3, 1, 4)$. (The (radian) arccosine function $\cos^{-1} x$ is `acos(x)`.)

Problem 3: Print out a graph and a plot of the level curves for the function

$$f(x, y) = (x^3 - 3x)(2y^3 - y)$$

for $-2 \leq x \leq 2$ and $-1 \leq y \leq 1$. On your plot of the level curves draw (by hand) vectors in the direction of the gradient of f at four points near the four corners.

Problem 4: Examine the graph of $f(x, y) = (x^3 + 8y^3 - 10y^4)/(x^2 - 2xy + 4y^2)$ at several different magnifications. Decide on the basis of your observations whether or not $\lim_{(x,y)\to(0,0)} f(x, y)$ exists and if so, what it is. Decide on the basis of your observations whether or not f is differentiable at $(0, 0)$ if we set $f(0, 0) = \lim_{(x,y)\to(0,0)} f(x, y)$.

Problem 5: Examine the graph of $f(x, y) = (x^3 - 8y^3 - 10y^4)/(x^2 - 2xy + 4y^2)$ at several different magnifications. Decide on the basis of your observations whether or not $\lim_{(x,y)\to(0,0)} f(x, y)$ exists and if so, what it is. Decide on the basis of your observations whether or not f is differentiable at $(0, 0)$ if we set $f(0, 0) = \lim_{(x,y)\to(0,0)} f(x, y)$.