Contents

MATH241 Guide/Project 2

The basic format of this guide is similar to the first except what you will submit will be different. Your second project (and third) will be submitted as a published M-file. This is pretty and far less com-plicated that it might sound. Basically an M-file is just a text file which contains a series of Matlab commands. You can create and edit one by simply typing

```
edit project.m
```

in the Matlab command window. When you do this an editor will open up and you can edit the file. Inside an M-file there are various ways to make things pretty but all you need to do for this project is make sure that each of your tasks is preceded by `%%` with a title for that task. This will ensure that your output is what you would expect. Basically the `%%` divide the document into cells and each cell will contain the command and its output, separate from all the rest. In a nutshell if you had two tasks, one was `1+1` and the next was `sin(3)` then your m-file would look like this:

```
%% Here is Task 1

1+1

%% Here is Task 2, it's fun!

sin(3)
```

and so on...

Publishing an M-file basically takes the file, runs the Matlab commands inside it and then creates an HTML document which contains not only the commands but also the output of the commands including pictures. This guide is actually a published M-file. You can publish your M-file by selecting the publish option in Matlab. For your project simply publish your M-file and print the resulting HTML or PDF document, depending which you choose.

Partial Derivatives

Since we can tell Matlab which variable to take the derivative with respect to, finding partial derivatives is as easy as regular derivatives. Here's an example of a partial derivative with respect to y:

```
syms x y;
diff(x^2*y+y^2/x,y)

ans =

(2*y)/x + x^2
```

Task 1: Find $\frac{d}{dy}\left[\frac{\sin(x^2y)}{y^2}\right]$

Here's an example of a second partial derivative: First finding the derivative with respect to y and then x:

```
diff(diff(x^2*exp(x*y^2),y),x)

ans =

2*x^3*y^3*exp(x*y^2) + 6*x^2*y*exp(x*y^2)
```

Task 2: Find $\frac{\partial^2}{\partial x \partial y}\left[\frac{x^3+y^2}{x-y}\right]$

Matlab does have a gradient command but it gives numerical approximations, not what we want. Instead we want a specific manifestation of the Matlab jacobian command. Specifically for a function of two variables we do the following. What this really does is take a bunch of derivatives with respect to the letters given and product the output we desire. This is how we find ∇f for $f(x,y) = x^2y^3$:

```
jacobian(x^2*y^3,[x y])

ans =

[ 2*x*y^3, 3*x^2*y^2]
```

Task 3: Find ∇f for $f(x,y) = \frac{xy}{x^2y+y}$.

Here's the same gradient with values plugged in for x and y. Note the use of the familiar subs command but with `{x,y}` to substitute for both values:

```
subs(jacobian(x^2*y^3,[x y]),{x,y},{-1,3})
```

```
ans =

[ -54, 27]
```

Task 4: Find $\nabla f(-1,2)$ for $f(x,y) = 5x^3y^2 - \frac{x}{y}$.

Here's one with three variables, I only `syms z` because `x` and `y` are still active from earlier in my work.

```
syms z;
jacobian(x*exp(x*y)/z,[x y z])
```

```
ans =

[ exp(x*y)/z + (x*y*exp(x*y))/z, (x^2*exp(x*y))/z, -(x*exp(x*y))/z^2]
```

Directional Derivatives

We learned in class that the directional derivative is the dot product of the unit vector with the gradient. Consider the following compound command. Really think about what it all does:

```
a=[1 2];
dot((a/norm(a)),subs(jacobian(x*y^2-1/y,[x y]),{x,y},{-2,3}))
```

```
ans =

-(133*5^(1/2))/45
```

You (hopefully!) guessed it! This is the directional derivative of $f(x,y) = xy^2 - 1/y$ in the direction of $\mathbf{a} = 1\mathbf{i} + 2\mathbf{j}$ at the point $(-2,3)$. We make the vector a a unit vector and dot it with the gradient.

Task 5: Find the directional derivative of $g(x,y) = x^2 + y^3$ at $(1,-2)$ in the direction of $a = 3\mathbf{i} - 7\mathbf{j}$.

Critical Points

We know how to solve a single equation equal to zero in Matlab:

```
syms x
solve(x^2+2*x==0)
```

```
ans =

 -2
  0
```

We can solve a system of equations by simply stuffing the two equations into a vector. Since the output is a vector we need to assign it to make it legible. If that confuses you don't worry, just do it. Here is how to simultaneously solve $xy - 3 = 0$ and $x - y - 2 = 0$:

```
[xsoln,ysoln]=solve([x*y-3==0,x-y-2==0])
```

```
xsoln =

 -1
  3


ysoln =

 -3
  1
```

Be aware that the first vector xsoln is giving you the two possible x values. These pair up with the two possible y values. Thus in friendlier terms the two solutions are $(3, 1)$ and $(-1, -3)$. Also note that the solve command returns the solution in alphabetical order, meaning it returns x and then y. This is why we assign the solution to `[xsoln,ysoln]=`.

Now then when we look for critical points we're setting both partial derivatives equal to 0. This is the same as setting the gradient equal to the 0 vector. Here's example 4 from page 878 in the book:

```
f(x)=x^2-2*x*y+1/3*y^3-3*y;
[xsoln,ysoln]=solve(jacobian(f,[x y]))
```

```
xsoln =

 -1
  3


ysoln =

 -1
  3
```

Task 6: Find all critical points for $f(x,y) = (y-2)\ln(xy)$. Remember that ln in Matlab is `log`. On your printout write the points as coordinate pairs next to the output.

Task 7: Find all critical points for $f(x,y) = x^3 + y^3 - 6xy$. On your printout write the points as coordinate pairs next to the output.

Lagrange Multipliers

When we solve a problem using Lagrange multipliers what we're doing is solving $\nabla f = L\nabla g$ along with the constraint. The first of these is the same as solving $\nabla f - L\nabla g = 0$ and the constraint can be rewritten to equal 0 too. The tricky thing is solving them all at once. This is how it's done for finding the extreme values for $f(x,y) = 3x^2 + 2y^2 - 4y + 1$ subject to $x^2 + y^2 = 16$. Well first here we'll just solve the equations:

```
clear all;
syms x y L;
f(x,y)=3*x^2+2*y^2-4*y+1;
g(x,y)=x^2+y^2-16;
firstpart=jacobian(f,[x y])-L*jacobian(g,[x y]);
[Lsoln,xsoln,ysoln]=solve([firstpart,g])
```

```
Lsoln =

   3
   3
 3/2
 5/2


xsoln =
 -2*3^(1/2)
  2*3^(1/2)
          0
          0


ysoln =

 -2
 -2
  4
 -4
```

Again note that the values group together. For example $L = 3/2$ corresponds to the point $(0, 4)$ and so on through all four. Also again note the order `[Lsoln,xsoln,ysoln]` because solve returns the solutions to `L`, `x` and `y` in alphabetical order. Even better we can plug `xsoln` and `ysoln` into `f` at the end:

```
subs(f(x,y),{x,y},{xsoln,ysoln})


ans =

 53
 53
 17
 49
```

and simply pick off the largest and smallest. In this case the maximum is 53 occuring at both $(2 * 3^(1/2), -2)$ and $(-2 * 3^(1/2), -2)$ and the minimum is 17 occuring at $(0, 4)$.

Task 8: Use Lagrange multipliers to find the maximum and minimum values of $f(x, y) = xy^2$ subject to the constraint $x^2 + y^2 = 16$. On your printout write a neat summary next to the output.