Basic Computations in Multivariable Calculus Made Easy

I. Vector Geometry

Single Variable Functions

For single variable functions we use the notation f[x_]:=.

```
f[x_] := x^2
f[2]
```

```
4
```

Multi-Variable Functions

For functions of two or more variables we just use a comma in between each variable and make sure to use the underscore i.e for two variables it becomes f[x_,y_]:= and for three it becomes f[x_,y_,z_]:= and so forth.

```
f[x_, y_] := x^2 + y^2
f[1, 2]
```

```
5
```

```
g[x_, y_, z_] := 2 x + y + z
g[1, 1, 1]
```

```
4
```

Dot Product

The notation for dot product is given by {a,b,c} [dot] {x,y,z} or {a,b,c}.{x,y,z} i.e just put a period in between the arrays.

```
{1, 0, 0}.{0, 1, 0}
```

```
0
```

```
{1, 2, 3}.{1, 1, 1}
```

```
0
```

```
6
```

Orthogonal Projection

The notation for computing the orthogonal projection is given by Projection[u,v].

```
u := {0, 1}
v := {1, 0}
Projection[u, v]
```

```
{0, 0}
```

Cross Product

The notation for the cross product is given by Cross[u,v] where u,v are vectors entered in like above i.e u={a,b,c} and v={x,y,z}.

```
Cross[{1, 0, 0}, {0, 1, 0}]
```

```
{0, 0, 1}
```

II. Vector Calculus

Length (Norm) of Vectors

To compute the length or norm of a vector we use Norm[{x,y,z}].

```
a := {1, 1}
Norm[a]
```

$\sqrt{2}$

Differentiating Vector-Valued Functions

To differentiate a vector valued function, we think of it as a n x 1 matrix (or similarly 1 x n) and use D[{x_1,...,x_n},{t}}].

```
D[{2 t, t}, {t}]
```

{2, 1}

```
a := {2 t, t}
D[a, {t}]
```

{2, 1}

```
f[t_] := {t, t}
f'[t_]
```

{1, 1}

Integrating Vector-Valued Functions

To integrate a vector valued function we use Integrate[{x_1,...,x_n},{t,0,t}].

```
a = {2 t, t}
Integrate[a, {t, 0, t}]
```

{2 t, t}

$\left\{t^2, \frac{t^2}{2}\right\}$

Arc-Length Function

We’ve already talked about how to define functions and now with the above we know how to integrate them. It follows that s[t_]:=Integrate[Norm[D[{x_1,...,x_n},{t}]],{t,0,t}] is one way to compute the arc-length function.

```
s[t_] := Integrate[Norm[D[{t, t}, {t}]], {t, 0, t}]
s[t]
```

$\sqrt{2}\ t$

Curvature for Plane and Space Curves

Again, we will just be combining previous definitions and so it follows that one way to denote the curvature is k[t_]:=Norm[Cross[r’[t] ,r’’[t]]]/Norm[r’[t]]^3. Below we compute the curvature of a line, which should be zero.

```
r[t_] := {t, 0, 0}
k[t_] := Norm[Cross[r'[t], r''[t]]] / Norm[r'[t]]^3
k[t]
```

0

Unit Tangent and Normal Vectors

Arguably two of the most annoying things to compute in multivariable calculus is now simplified! We just use the command Normalize[v]. Below is an example with the standard helix and lastly we show that T,N are perpendicular for all t. There are letters which are previously defined in Mathematica and for that reason g(t) = T(t) and h(t) = N(t). The notation is given by g(t) = Normalize[r’[t]] and N(t) = Normalize[g’[t]].

```
r[t_] := {Cos[t], Sin[t], t}
Normalize[r'[t]]
```

$$\left\{-\frac{\text{Sin}[t]}{\sqrt{1+\text{Abs}[\text{Cos}[t]]^2+\text{Abs}[\text{Sin}[t]]^2}},\ \frac{\text{Cos}[t]}{\sqrt{1+\text{Abs}[\text{Cos}[t]]^2+\text{Abs}[\text{Sin}[t]]^2}},\ \frac{1}{\sqrt{1+\text{Abs}[\text{Cos}[t]]^2+\text{Abs}[\text{Sin}[t]]^2}}\right\}$$

```
g[t_] := {-Sin[t] / Sqrt[2], Cos[t] / Sqrt[2], 1 / Sqrt[2]}
Normalize[g'[t]]
```

$$\left\{-\frac{\text{Cos}[t]}{\sqrt{2}\sqrt{\frac{1}{2}\text{Abs}[\text{Cos}[t]]^2+\frac{1}{2}\text{Abs}[\text{Sin}[t]]^2}},\ -\frac{\text{Sin}[t]}{\sqrt{2}\sqrt{\frac{1}{2}\text{Abs}[\text{Cos}[t]]^2+\frac{1}{2}\text{Abs}[\text{Sin}[t]]^2}},\ 0\right\}$$

```
h[t_] := {-Cos[t], -Sin[t], 0}
Dot[h[t], g[t]]
```

0

III. Differentiation in Several Variables

Partial Derivatives

Given a function f(x_1,...,x_n) the command D[f,x_j] gives the partial with respect to the variable x_j. The commands D[f,{x_j,n}], D[f,x_1,x_2,...] give the x_j derivative n-times and mixed partial derivative respectively. You can also play with the D[f, {x_1,n},{x_2,m},...]. This will gives the x_j derivative computed n-times first, then the x_2 derivative computes m-times and so forth.

```
f[x_, y_] := x^2 + y^2
D[f[x, y], x]

2 x

D[f[x, y], {x, 3}]

2 x

0

g := Cos[x + y]
D[g, x]

-Sin[x + y]
```

Gradient (and general derivative)

Given a real-valued function f= f(x_1,...,x_n), the command D[f,{{x_1,...,x_n}}] gives its derivative (or gradient). The command D[{f^1,...,f^m},{{x_1,...,x_n}}] gives the Jacobian i.e derivative of f, an m x n matrix.

```
f := x^2 + y^2
D[f, {{x, y}}]

{2 x, 2 y}

g := {x, y}
D[g, {{x, y}}]

{2 x, 2 y}
```

```
{{1, 0}, {0, 1}}
```

Hessian Matrix (second derivative test)

The Hessian matrix of a real valued function f is given by the command D[f,{{x_1,...,x_n},2}]. To check that the example below makes sense, just remember that the Hessian is computed by taking the gradient of the gradient (transpose).

```
f := x^2 + y^2
D[f, {{x, y}, 2}]
{{2, 0}, {0, 2}}
```

IV. Integration with Multiple Variables

.

Single Variable Integration

To integration a real-valued single variable function f(x), use the command Integrate[f,x] to return the indefinite integral and Integrate[f,{x,a,b}] to return the definite integral.

```
f := x
Integrate[f, {x, 0, 1}]
```

$\frac{1}{2}$

Several Variable Integration

To integration a real-valued function f=f(x_1,...,x_n) we use also have two commands. The first is Integrate[f,{x_1,-,-},{x_2,-,-},...,{x_n,-,-}]. The dashes after the variable x_j are its lower and upper bounds respectively. The second command is Integrate[f,{x_1,...,x_n} \in reg]. The term reg refers to region in which Mathematica has a few previously defined such as the sphere, disk, circle, etc.

```
f := 1
Integrate[f, {x, y} ∈ Disk[]]
```

π

```
g := 2 - x - y
Integrate[g, {x, 0, 2}, {y, 0, 2 - x}]
```

$\frac{4}{3}$

V. Fundamental Theorems of Vector Analysis

Divergence

Given a vectors-valued function f=(f^1,...,f^n), its divergence is given by the command Div[{f^1,...,f^n},{x_1,...,x_n}]. Moreover, one can use Div[{f^1,...,f^n},{x_1,...,x_n}, chart] to give the divergence in specific coordinate variables. We won't illustrate this example since it is a bit fancy. A short cut is to use "esc + del+esc, ctrl +" which produces $\nabla_{\{x,y,z\}}$.

```
f := {x, y, z}
∇{x,y,z} . {y, -x, z}
1
```

Curl

Given a vectors-valued function f=(f^1,...,f^n), its divergence is given by the command Curl[f,{x_1,...,x_m}]. Additionally we can compute the curl in a specific coordinate chart by using Curl[f,{x_1,...,x_m}, chart].

```
Curl[{x * y, Exp[x], y + z}, {x, y, z}]
```

$\{1, 0, e^x - x\}$