Chapter 15

Line and Surface Integrals

15.1 Line Integrals

There are two main types of line integrals: the line integral of a scalar field along a curve, $\int_C f\,ds$, and the line integral of the tangential component of a vector field along a curve. ***Maple*** has a `Lineint` command in the `Student` package that does the line integral of a 2-dimensional scalar field, but if we learn how to do it ourselves, we won't be restricted to just 2 dimensional functions.

You will need to load the definitions of `evl`, `&.`, `len`, `unit`, and `&x` from earlier.

We will also define some procedures to help with our vector integrals. First we need to define the velocity vector, which is simply the tangent vector to the curve.

```
> velocity_vector := (C,R) -> diff(map(rhs,C),op(1,R)):
```

Recall the definition of the line integral of a scalar field, $\int_C f\,ds = \int_a^b f(\sigma(t))\|\sigma'(t)\|\,dt$, where $\sigma(t)$ describes the curve for $t \in [a,b]$.

So, we define

```
> slineint := (F,C,R) -> int(subs(C,F)*len(velocity_vector(C,R)),R):
```

We introduce an inert form as well.

```
> Slineint := subs(int=Int,eval(slineint)):
```

Recall the definition of the line integral of a vector field,

$$\int_C \mathbf{F}\cdot d\mathbf{s} = \int_C F_1dx + F_2dy + f_3dz = \int_a^b \mathbf{F}(\sigma(t))\cdot\sigma'(t)\,dt$$

where $\sigma(t)$ describes the curve for $t \in [a,b]$.

We define

```
> vlineint := (F,C,R) -> int(subs(C,evl(F)) &.
> velocity_vector(C,R),R):
```

and introduce an inert form as well.

```
> Vlineint := subs(int=Int,eval(vlineint)):
```

Let's look at `slineint`. We need it to take three inputs: a scalar field, a parametric curve, and a parameter interval. The scalar field will be expressed in the usual coordinate variables, usually x, y and z, if we are in three dimensions. The parametric curve will be written as a list of equations expression each coordinate variable in terms of the parameter. For this we use `subs` to give us the scalar field at a point along the curve. It will have to take the form:

```
slineint(f(x,y),[x=g(t),y=h(t)],t=a..b);
```

The result should be

$$\int_a^b f(g(t),h(t))\,\sqrt{(\frac{dg}{dt})^2+(\frac{dh}{dt})^2}\,dt$$

Let's try it.

```
> slineint(f(x,y),[x=g(t),y=h(t)],t=a..b);
```

$$\int_1^{2/5} 5\,e^{(-(\mathrm{u}(t)-\mathrm{tahn}(u)(t))^2)}\,\sqrt{((\frac{\partial}{\partial t}\,\mathrm{u}(t))-(\frac{\partial}{\partial t}\,\mathrm{tahn}(u)(t)))^2+(\frac{\partial}{\partial t}\,\mathrm{sech}(u)(t))^2}\,dt$$

```
> slineint(x-y*z,[x=sin(t),y=cos(t),z=t],t=0..Pi);
```

$$4\sqrt{2}$$

```
> Slineint(x-y*z,[x=sin(t),y=cos(t),z=t],t=0..Pi);
```

$$\int_0^{\pi}(\sin(t)-\cos(t)\,t)\,\sqrt{\cos(t)^2+\sin(t)^2+1}\,dt$$

```
> simplify(%);
```

$$-\sqrt{2}\int_0^{\pi}-\sin(t)+\cos(t)\,t\,dt$$

```
> value(%);
```

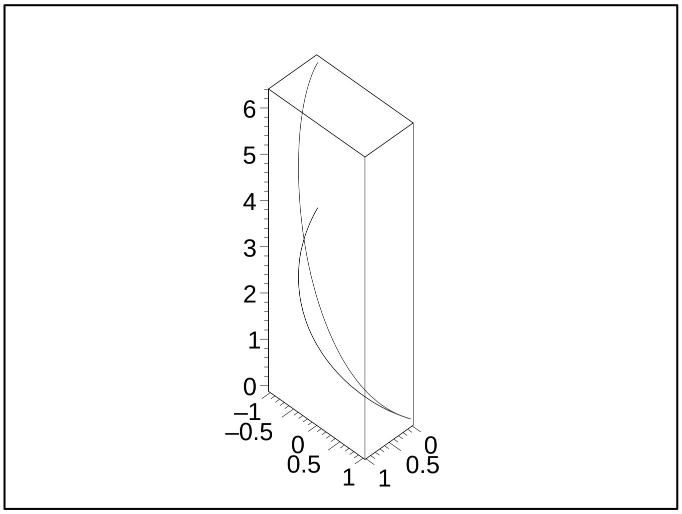
$$4\sqrt{2}$$

The values of the function along the curve can be realized as a *fence*. That is, we can plot the curve and then at each point along the curve, the value of this function at this curve will give us the height of the fence. Then drawing the surface, so defined, the line integral will give the surface area of the fence - a signed area of course.

```
> fence:=spacecurve([sin(t),cos(t),t+(sin(t)-t*cos(t))],t=0..Pi,color=r
> ed):
```

```
> curve:=spacecurve([sin(t),cos(t),t],t=0..Pi,color=blue):
```

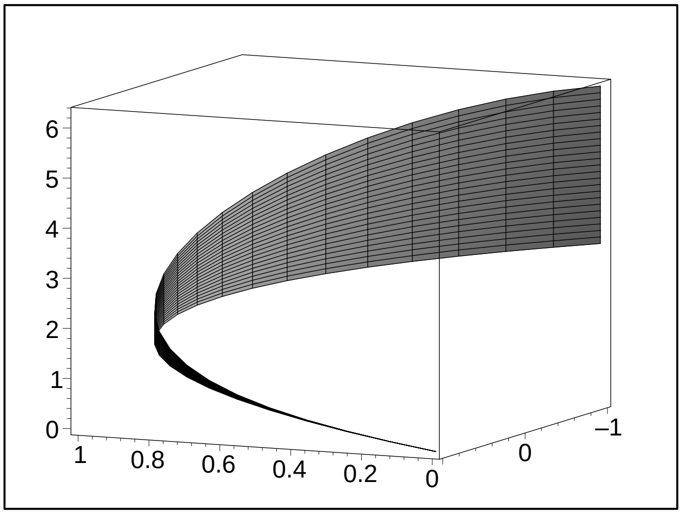
```
> display({curve,fence},axes=boxed,scaling =constrained);
```



```
>  z1 := t+(sin(t)-t*cos(t)):
>  z2 := u*t+(1-u)*z1;
```

$$z2 := u\,t + (1-u)\,(t + \sin(t) - t\cos(t))$$

```
>  plot3d([sin(t),cos(t),z2],t=0..Pi,u=0..1,axes=boxed,orientation=[115,
>  80],scaling=unconstrained);
```



Here you see the curve, the values of the function along the curve and the fence. The line integral gives us the area of the fence.

We deal with the line integrals of vector fields easily. To defer any problems that might arise because of the way that ***Maple*** may deal with a vector, we used `evl` to make everything a list.

Consider the vector field $\mathbf{F}(x, y) = [y^2, xy]$ along the curve $y = x^2$ from $(0, 0)$ to $(1, 1)$. You are probably more used to seeing this line integral in the form $\int y^2\,dx + xy\,dy$.

```
>  Vlineint([y^2,x*y],[x=t,y=t^2],t=0..1);
>  vlineint([y^2,x*y],[x=t,y=t^2],t=0..1);
```

$$\int_0^1 3\,t^4\,dt$$

$$\frac{3}{5}$$

Find the value of $\int z^3\,dx + y^2\,dy + x\,dz$, where C is the helix $x = \cos(t)$, $y = \sin(t)$, $z = t$, $t \in [0, 2\pi]$.

```
> Vlineint([z^3,y^2,x],[x=cos(t),y=sin(t),z=t],t=0..2*Pi);
```

$$\int_0^{2\pi} -\sin(t)\,t^3 + \cos(t)\sin(t)^2 + \cos(t)\,dt$$

```
> vlineint([z^3,y^2,x],[x=cos(t),y=sin(t),z=t],t=0..2*Pi);
```

$$8\,\pi^3 - 12\,\pi$$

15.2 Surface Integrals

Just as in the case of line integrals, there are two types of surface integrals: integrating a scalar field over a surface or integrating a vector field over a surface (the flux of the vector field across the surface).

In the case of surface integrals there is a very important vector - the area element: $|D_u(R) \times D_v(R)|$.

```
> aelt := (S,R1,R2) -> len(velocity_vector(S,R1) &x
> velocity_vector(S,R2)):
> surfint := (f,S,R1,R2) -> int(int(subs(S,f)*aelt(S,R1,R2),R2),R1):
> Surfint := subs(int=Int,eval(surfint)):
```

Let's try to integrate x^2 over the torus of our previous lecture.

```
> torus := [x=(2+cos(u))*cos(v),y=(2+cos(u))*sin(v),z=sin(u)]:
> surfint(x^2,torus,u=0..2*Pi,v=0..2*Pi);
```

$$22\,\pi^2$$

```
> Surfint(x^2,torus,u=0..2*Pi,v=0..2*Pi);
```

$$\int_0^{2\pi}\int_0^{2\pi} (2+\cos(u))^2\cos(v)^2 \mathrm{sqrt}(\cos(u)^2\,(2+\cos(u))^2\cos(v)^2 \\ + \cos(u)^2\,(2+\cos(u))^2\sin(v)^2 \\ + (-\sin(u)\cos(v)^2\,(2+\cos(u)) - \sin(u)\sin(v)^2\,(2+\cos(u)))^2) dvdu$$

```
> J := simplify(%);
```

$$J := \int_0^{2\pi}\cos(v)^2\,dv \int_0^{2\pi}(2+\cos(u))^3\,\mathrm{csgn}(2+\cos(u))\,du$$

Now, let's try to define a ***Maple*** procedure to compute the flux.

```
> flux := (F,S,R1,R2) -> int(int(subs(S,evl(F)) &.
> (velocity_vector(S,R1) &x velocity_vector(S,R2)),R2),R1);
```

$$\mathit{flux} := (F,\ S,\ \mathit{R1},\ \mathit{R2}) \to \int$$
$$\int \text{`\&.`}(\mathrm{subs}(S,\ \mathrm{evl}(F)),\ \mathrm{velocity\_vector}(S,\ \mathit{R1})\ \&\mathrm{x}\ \mathrm{velocity\_vector}(S,\ \mathit{R2}))\ d\mathit{R2}\ d\mathit{R1}$$

```
>  Flux := subs(int=Int,eval(flux)):
```

Just be careful of which normal vector you are required to use for your computation. Our definition uses $D_u(R) \times D_v(R)$.

Find the flux of the vector field $F = [x, y, z]$ outward through the torus.

Note that we saw previously that the normal we are using here points inward. Thus we shall need the negative of our previous computation:

```
>  -flux([x,y,z],torus,u=0..2*Pi,v=0..2*Pi);
```

$$12\,\pi^2$$

```
>  -Flux([x,y,z],torus,u=0..2*Pi,v=0..2*Pi);
```

$$-\int_0^{2\,\pi}\int_0^{2\,\pi} -\cos(u)\,(2+\cos(u))^2\cos(v)^2 - \cos(u)\,(2+\cos(u))^2\sin(v)^2$$
$$+\,(-\sin(u)\cos(v)^2\,(2+\cos(u)) - \sin(u)\sin(v)^2\,(2+\cos(u)))\sin(u)dvdu$$

```
>  simplify(%);
```

$$-\int_0^{2\,\pi} -5\int_0^{2\,\pi} 1\,dv\cos(u) - 2\int_0^{2\,\pi} 1\,dv\cos(u)^2 - 2\int_0^{2\,\pi} 1\,dv\,du$$

```
>  value(%);
```

$$12\,\pi^2$$