Chapter 13

Multiple Integrals

13.1 Double and Triple Integrals

Maple evaluates double and triple integrals as iterated integrals. This is the way that we would expect, and we will not spend much time with that situation. So to evaluate $\int_2^4 \int_1^2 x^2 y\, dydx$ we would use

```
>  Int(Int(x^2*y,y=1..2),x=2..4) = int(int(x^2*y,y=1..2),x=2..4);
```

$$\int_2^4 \int_1^2 x^2\, y\, dy\, dx = 28$$

Notice that you really need to think of this integration working from the inside out. You have an inner integral and an outer integral. The inner integral needs to be done first.

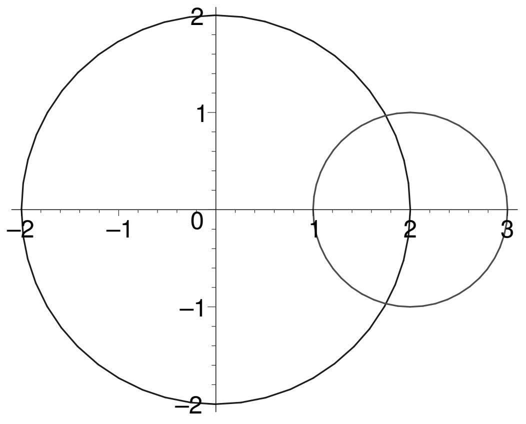
Of course, rectangular regions are not where the trouble lies. It is with more complicated regions that most of us have problems. The ability to graph the region, look at it, and then dissect it into usable pieces is extremely important.

Problem 1: Calculate $\int\int_R xy^2\, dA$, where R is the region between the circle of radius 1, centered at $(2, 0)$ and the circle of radius 2 centered at the origin.

Solution:

First, plot the curves to look at the region.

```
>  plot({[2+cos(t),sin(t),t=0..2*Pi],[2*cos(t),2*sin(t),t=0..2*Pi]
>  },scaling=constrained, color=[red,blue],thickness=3);
```



The easiest way to describe this region is to look at x as a function of y, then integrate. The integral is given by

$$\int_{-b}^{b}\int_{2-\sqrt{1-y^2}}^{\sqrt{4-y^2}} xy^2\,dxdy$$

where b is the point of intersection of the two circles.

```
> solve(2-sqrt(1-y^2)=sqrt(4-y^2),y);
```

$$-\tfrac{1}{4}\sqrt{15},\ \tfrac{1}{4}\sqrt{15}$$

```
> b := %[2];
```

$$b := \tfrac{1}{4}\sqrt{15}$$

```
> J := Int(Int(x*y^2,x = 2-sqrt(1-y^2) .. sqrt(4-y^2)),y = -'b' ..
> 'b')=int(int(x*y^2,x = 2-sqrt(1-y^2) .. sqrt(4-y^2)),y = -b .. b);
```

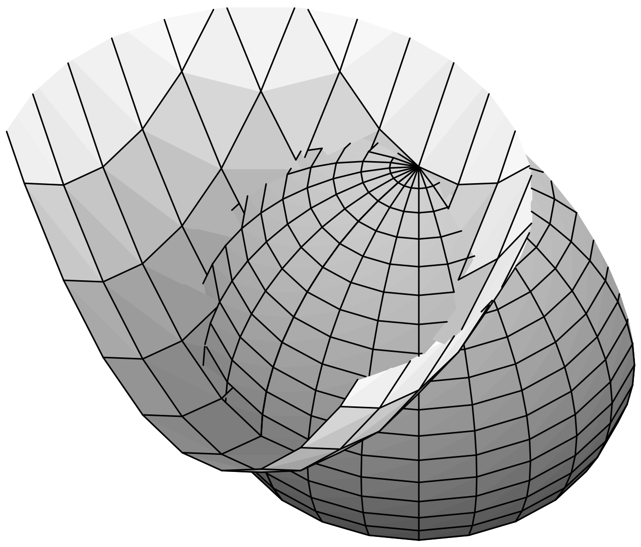
$$J := \int_{-b}^{b}\int_{2-\sqrt{1-y^2}}^{\sqrt{4-y^2}} xy^2\,dx\,dy = -\frac{13}{256}\sqrt{15}+\frac{1}{2}\arcsin(\frac{1}{4}\sqrt{15})$$

```
> evalf(rhs(J));
```

$$.4623831000$$

Problem 2: Find $\int\int\int_R z\,dV$ where R is the region inside the sphere $x^2+y^2+z^2=1$ and above the paraboloid $z=(x-1)^2+y^2$.
Solution:
Let's look at the region.

```
> with(plots):
> sphere:=plot3d([sin(phi)*cos(theta),sin(phi)*sin(theta),cos(phi)],phi
> =0..Pi,theta=0..2*Pi,scaling=constrained,shading=zgreyscale):
> paraboloid :=
> plot3d((x-1)^2+y^2,x=-3..3,y=-3..3,shading=zgreyscale):
> display3d({sphere,paraboloid},scaling=constrained,
> view=[-.5..1.5,-1.5..1.5,-.5..1.5]);
```



First, when we slice through this region, we see that below we are bounded by the paraboloid, and above we are bounded by the sphere, so $(x-1)^2+y^2 \leq z$ and $z \leq \sqrt{1-x^2-y^2}$. Now we need to know where these two surfaces meet. This is found by solving

```
>  S:= solve(x^2+y^2+((x-1)^2+y^2)^2=1,y);
```

$$S := \frac{1}{2}\sqrt{-6+2\sqrt{9-8x}-4x^2+8x},\ -\frac{1}{2}\sqrt{-6+2\sqrt{9-8x}-4x^2+8x},$$
$$\frac{1}{2}\sqrt{-6-2\sqrt{9-8x}-4x^2+8x},\ -\frac{1}{2}\sqrt{-6-2\sqrt{9-8x}-4x^2+8x}$$

Note that the last two solutions have the potential for complex evaluation in our range. Check it yourself to see, but we may ignore these solutions. Thus, let y_0 be the positive solution.

```
>  y0 := S[1];
```

$$y0 := \frac{1}{2}\sqrt{-6+2\sqrt{9-8x}-4x^2+8x}$$

The outer integral goes from 0 to 1, so we now have that the integral is given by:

$$\int_0^1 \int_{-y_0}^{y_0} \int_{(x-1)^2+y^2}^{\sqrt{1-x^2-y^2}} z\, dz dy dx$$

```
>  Int(Int(int(z,z = (x-1)^2+y^2 ..  sqrt(1-x^2-y^2)),y = -'y0' ..
>  'y0'),x = 0 ..  1)=int(int(int(z,z = (x-1)^2+y^2 ..  sqrt(1-x^2-y^2)),y
>  = -y0 ..  y0),x = 0 ..  1);
```

$$\int_0^1 \int_{-y0}^{y0} \frac{1}{2} - \frac{1}{2}x^2 - \frac{1}{2}y^2 - \frac{1}{2}((x-1)^2+y^2)^2 \, dy \, dx =$$

$$\int_0^1 -\frac{1}{160}\%1^{(5/2)} + \frac{1}{12}(-\frac{1}{2}-(x-1)^2)\%1^{(3/2)} + \frac{1}{2}\sqrt{\%1} - \frac{1}{2}x^2\sqrt{\%1} - \frac{1}{2}(x-1)^4\sqrt{\%1}\,dx$$

$$\%1 := -6 + 2\sqrt{9-8x} - 4x^2 + 8x$$

```
>  evalf(rhs(%));
```

$$.1722182285$$

13.2 The Jacobian and Change of Variables

Many, many times we need to use a coordinate system other than the tried-and-true rectangular coordinate system. It could be polar coordinates, cylindrical coordinates or spherical coordinates, or others. Type `?coords` to see what other coordinate systems ***Maple*** has in store for you.

When you change from one coordinate system to another, we have to take into account the relative change of the coordinate change. When we replace x by $2u$, when need to change dx to $2du$ - following along the change. When this is done in several variables, we have to use our old friend the Jacobian.

When making a change of variables in a double integral, we express the old variables , say x and y, in terms of the new variables u and v. The area element, $dxdy$ is replaced by where is the absolute value of the Jacobian determinant

$$J(u,v) = \left|\frac{\partial(x,y)}{\partial(u,v)}\right| = \frac{\partial x}{\partial u}\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial y}{\partial u}.$$

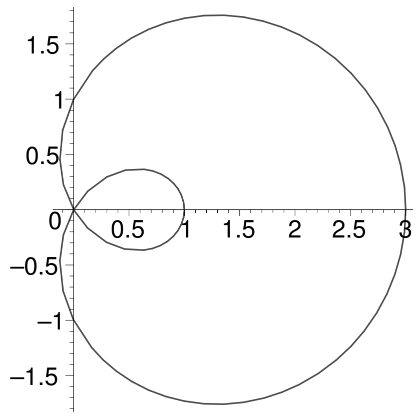
The process is similar for three-dimensions. The problem usually occurs in determining the new limits of integration.

Problem 1: Integrate xy^2 over the region between the two loops of the limaçon of Pascal, the curve given by $r = 1 + 2\cos(\theta)$ in polar coordinates.

Solution

We have to load the Jacobian and the determinant with the `linalg` package.

```
>  with(linalg):

>  plot([1+2*cos(theta),theta,theta=0..2*Pi],coords=polar,scaling=constr
>  ained,thickness=3);
```



To find the area of the region between the two loops using rectangular coordinates is a mess. If we look at this limaçon, the outer loop is determined by $-\frac{2\pi}{3} \le \theta \le \frac{2\pi}{3}$. The smaller loop is determined by $\frac{2\pi}{3} \le \theta \le \frac{4\pi}{3}$, but there $r < 0$.

```
> xpol := r*cos(theta):  ypol := r*sin(theta):
> J1 :=
> Int(Int(xpol*ypol^2*r,r=0..1+2*cos(theta)),theta=-2*Pi/3..2*Pi/3)=int(
> int(xpol*ypol^2*r,r=0..1+2*cos(theta)),theta=-2*Pi/3..2*Pi/3);
```

$$J1 := \int_{-2/3\,\pi}^{2/3\,\pi}\int_0^{1+2\cos(\theta)} r^4\cos(\theta)\sin(\theta)^2\,dr\,d\theta = \frac{405}{224}\sqrt{3} + 2\,\pi$$

```
> J2 :=
> Int(Int(xpol*ypol^2*r,r=0..1+2*cos(theta)),theta=2*Pi/3..4*Pi/3)=int(i
> nt(xpol*ypol^2*r,r=0..1+2*cos(theta)),theta=2*Pi/3..4*Pi/3);
```

$$J2 := \int_{2/3\,\pi}^{4/3\,\pi}\int_0^{1+2\cos(\theta)} r^4\cos(\theta)\sin(\theta)^2\,dr\,d\theta = -\frac{405}{224}\sqrt{3} + \pi$$

```
> rhs(J1)-rhs(J2);
```

$$\frac{405}{112}\sqrt{3} + \pi$$

Cylindrical Coordinates

In cylindrical coordinates the change is $x = r\cos(\theta)$, $y = r\sin(\theta)$, and $z = z$. The volume element then is found from the Jacobian

```
> det(jacobian([r*cos(theta),r*sin(theta),z],[r,theta,z]));
```

$$\cos(\theta)^2\,r + \sin(\theta)^2\,r$$

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

$$r$$

Thus the volume element is $dV = r\,dr\,d\theta\,dz$.

Problem 2: Find the volume inside both the sphere of radius 3 centered at the origin and the cylinder $(x-2)^2 + y^2 = 4$.

Solution:
The sphere can be rewritten in cylindrical coordinates as $r^2 + z^2 = 9$. The cylinder:

```
> solve(subs(x=r*cos(theta),y=r*sin(theta),(x-2)^2+y^2=4),r);
```

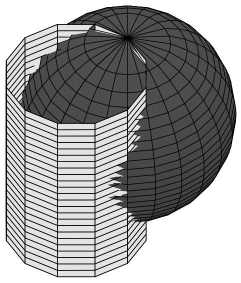
$$0,\ 4\,\frac{\cos(\theta)}{\cos(\theta)^2 + \sin(\theta)^2}$$

```
> simplify(%[2]);
```

$$4\,\cos(\theta)$$

So this is the equation for the cylinder in cylindrical coordinates.

```
> with(plots):
> sphere := cylinderplot(sqrt(9-z^2),theta=0..2*Pi,z=-3..3):
> cylinder :=
> cylinderplot(4*cos(theta),theta=-Pi..Pi,z=-3..3,color=yellow):
> display({sphere,cylinder},scaling=constrained,shading=z);
```



To set up the region of integration, we will use the variable r. The values of z will clearly be trapped in the sphere.

```
> V := Int(Int(Int(r,z=-sqrt(9-r^2)..sqrt(9-r^2)),
> theta=-arccos(r/4)..arccos(r/4)), r=0..3) =
> int(int(int(r,z=-sqrt(9-r^2)..sqrt(9-r^2)),
> theta=-arccos(r/4)..arccos(r/4)), r=0..3);
```

$$V := \int_0^3 \int_{-\arccos(1/4\,r)}^{\arccos(1/4\,r)} \int_{-\sqrt{9-r^2}}^{\sqrt{9-r^2}} r\,dz\,d\theta\,dr = \int_0^3 4\,r\,\sqrt{9-r^2}\,\arccos(\frac{1}{4}\,r)\,dr$$

```
> evalf(rhs(V));
```

$$39.74123918$$

13.3 Surfaces and Curves

When we get into 3-dimensional geometry and calculus, we know that we really need to be able to see the regions about which we are talking. We should be able to use ***Maple*** as a visualization tool. It will do so much more, though.

Load the definitions of the dot product, length, and cross product from earlier.

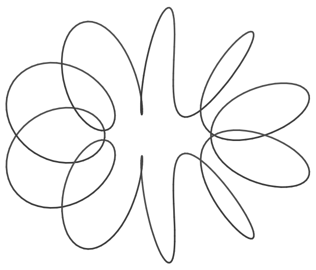
```
>  with(plots):
```

Let's say that we need to plot the curve

```
>  R := t-> [(2+sin(10*t))*cos(t),(2+sin(10*t))*sin(t),cos(10*t)];
```

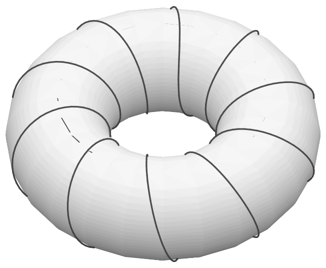
$$R := t \to [(2 + \sin(10\,t))\cos(t),\ (2 + \sin(10\,t))\sin(t),\ \cos(10\,t)]$$

```
>  spacecurve(R(t),t=0..2*Pi,numpoints=501,scaling=constrained,shading=z
>  ,thickness=3);
```



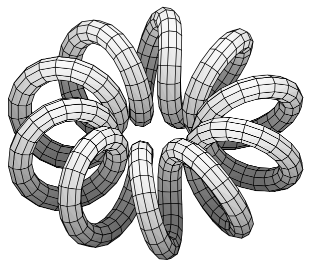
This looks like it wraps around something, doesn't it?

```
>  curve := %:

>  torus :=
>  cylinderplot({sqrt(.95-z^2)+2,2-sqrt(.95-z^2)
>  },t=0..2*Pi,z=-sqrt(.95)..sqrt(.95),style=PATCHNOGRID,shading=zgreysc
>  ale):

>  display({torus,curve});
```



Another useful command is `tubeplot`. This puts a tube around a curve. The default radius is 1, but that can be changed.

```
>  tubeplot(R(t),t=0..2*Pi,numpoints=200,scaling=constrained,radius=1/4,
>  shading=zgreyscale);
```



13.4 Velocity and Acceleration

Let t represent time, then the derivative of position with respect to t is velocity. Thus, if $r(t)$ represents position, it's derivative $r'(t)$ is the velocity, $v(t)$. We have an example, how do we want to differentiate it?

```
>  D(R)(t);
```

$$\mathrm{D}(R)(t)$$

We don't get what we would expect.

```
> diff(R(t),t);
```

$$[10\cos(10\,t)\cos(t)-(2+\sin(10\,t))\sin(t),\ 10\cos(10\,t)\sin(t)+(2+\sin(10\,t))\cos(t),\ -10\sin(10\,t)]$$

but this is not a procedure. We will have to use `unapply` here.

```
> V := unapply(diff(R(t),t),t);
```

$$V := t\rightarrow[10\cos(10\,t)\cos(t)-(2+\sin(10\,t))\sin(t),\ 10\cos(10\,t)\sin(t)+(2+\sin(10\,t))\cos(t),\ -10\sin(10\,t)]$$

Just as the derivative of position is velocity, so is the derivative of velocity is acceleration.

```
> A := unapply(diff(V(t),t),t);
```

$$A := t\rightarrow[-100\sin(10\,t)\cos(t)-20\cos(10\,t)\sin(t)-(2+\sin(10\,t))\cos(t),\ -100\sin(10\,t)\sin(t)+20\cos(10\,t)\cos(t)-(2+\sin(10\,t))\sin(t),\ -100\cos(10\,t)]$$

We would like to draw the arrows that represent these three vectors. To make the arrows we will need the following definitions.

```
> aperp := V -> unit(VP &x V);
```

$$aperp := V\rightarrow \mathrm{unit}(\mathit{VP}\ \&\mathrm{x}\ V)$$

Remember that this will give a vector that is perpendicular to both V and V_p, as long as V and V_p are not parallel. If we happen to have a case when V and V_p are parallel, `unit` will return an error because it will try to divide by 0. In that case, simply choose a different V_p.

We will choose the following vector as our default V_p.

```
> VP := [1/2,1/2,1/sqrt(2)]:
```

The following command will draw an arrow representing the vector V, starting at the point R. The line goes from the tip of the arrow to one barb, the other barb, back to the tip and finally to the base of the arrow.

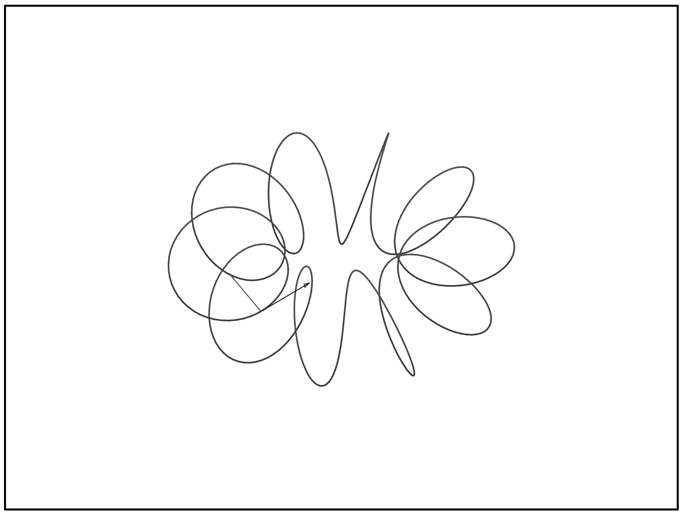
```
> arrow1 := (R,V) ->
> polygonplot3d([evl(R+V),evl(R+V-.1*unit(V)+.05*aperp(V)),evl(R+V-.1*un
> it(V)-.05*aperp(V)),evl(R+V),evl(R)],style=LINE,args[3..nargs]):
> arrow := (R,V) -> arrow1(evalf(R),evalf(V),args[3..nargs]):
```

The following procedure then plots the curve together with the two arrows starting at $R(t)$: a blue one for the velocity $\mathbf{V}(t)$ and a red arrow for the acceleration $\mathbf{A}(t)$. The scaling factors of 1/5 and 1/25 are chosen to make things look nice.

```
> plotav := t ->
> display(
> {curve,arrow(R(t),unit(V(t)),color=blue),arrow(R(t),unit(A(t)),color=
> red,orientation=[130,52])},scaling=constrained);
```

$$plotav := t\rightarrow \mathrm{display}(\{curve,\ \mathrm{arrow}(\mathrm{R}(t),\ \mathrm{unit}(\mathrm{V}(t)),\ color=blue),\ \mathrm{arrow}(\mathrm{R}(t),\ \mathrm{unit}(\mathrm{A}(t)),\ color=red,\ orientation=[130,\ 52])\},\ scaling=constrained)$$

```
> plotav(1);
```



This can also be animated as $\mathbf{R}(t)$ traverses part of the curve.

```
> display([seq(plotav(kk*Pi/20),kk=0..50)],insequence=true,scaling=cons
> trained);
```

The Downhill Skier

We can describe part of a hill with the equation:

$$z = -(x+y)\left(.13 + .1\,\sin\left(\frac{x+y}{8}\right)\right) - .15\,x\,\sin\left(\frac{x}{8} - \frac{3\,y}{40}\right)$$

where x, y, and z are measured in meters. A downhill skier starts from rest at the origin and follows the curve $y = .7\,x - 4\sin(\frac{x}{7})$ for x in the interval (0,40). Study the forces felt by the skier. In particular, does he or she ever leave the surface of the snow?

Begin by defining the hill and the skier's path. We will use the variable x as the time variable.

```
> f := (x,y) ->-(x+y)*(.13+.1*sin((x+y)/8))-.15*x*sin(x/8-3*y/40);
```

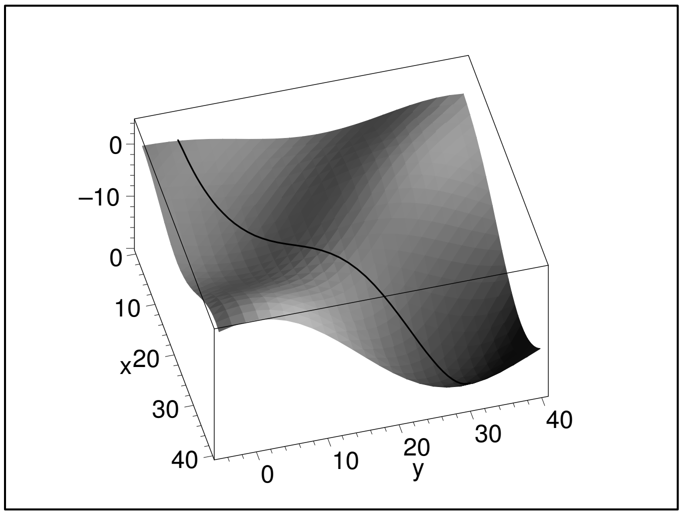
$$f := (x,\,y) \to -(x+y)\,(.13 + .1\sin(\frac{1}{8}\,x + \frac{1}{8}\,y)) - .15\,x\sin(\frac{1}{8}\,x - \frac{3}{40}\,y)$$

```
> yp := .7*x-4*sin(x/7); R:=[x,yp,f(x,yp)]:
```

$$yp := .7\,x - 4\sin(\frac{1}{7}\,x)$$

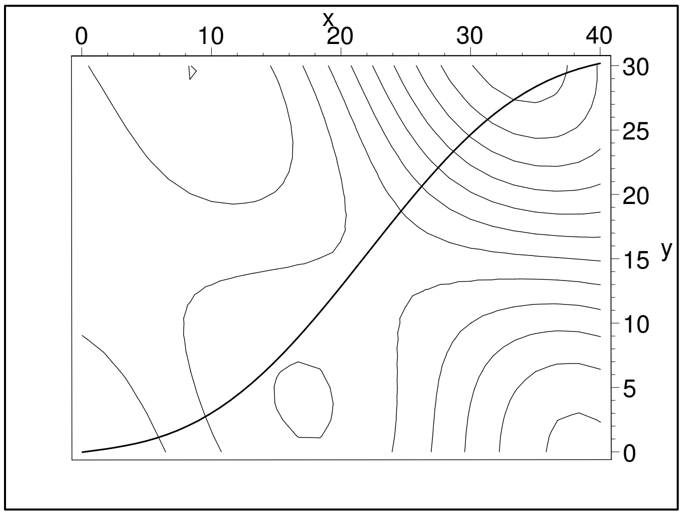
Now to plot the surface and the path, we could use color for the path, or we can move the surface down, very slightly in order to see the curve.

```
> surface :=
> plot3d(f(x,y)-0.2,x=0..40,y=-5..40,style=PATCHNOGRID,shading=zgreyscal
> e,ambientlight=[.1,.1,.4],light=[69,60,.7,.7,.7]):
> curve := spacecurve(R,x=0..40,color=black,thickness=3):
> display({surface,curve
> },orientation=[-15,45],scaling=constrained,axes=BOXED);
```



Consider the contour plot for this surface and curve.

```
> cplot := contourplot3d(f(x,y),x=0..40,y=0..30):
> display({cplot,curve
> },orientation=[-90,0],axes=BOXED,shading=none,style=CONTOUR);
```



So note that the skier starts with a gentle decline, up to about $x = 15$. From that point to about $x = 20$, the skier is going slightly uphill. From $x = 20$ and beyond, there is a steep downhill towards a pit.

We have to make some simplifying assumptions.

1. We assume that the only force acting on the skier that has a component in the direction of the velocity vector is gravity.

2. We assume that the skier does not raise or lower the center of gravity, by bending or straightening the knees or hips.

With these assumptions, the skier's total energy (potential + kinetic) is constant. This energy is given by the equation $\frac{mv^2}{2} + mgz$. Here
$m =$ skier's mass; $v =$ skier's speed; $g = 9.8m/\text{sec}^2$, the acceleration due to gravity.

At the beginning of the run, the velocity and height are both 0, $v = 0$ and $z = 0$. Thus, this constant has to be 0.

```
>  g := 9.8; v := sqrt(-2*g*R[3]):
```

$$g := 9.8$$

Now we want to find the velocity vector **V**. To do so we will differentiate **R**. The unit tangent vector, **T**, is the unit vector in the direction of the velocity vector. The velocity is then v times this unit vector.

```
>  T := evl(unit(diff(R,x))):
```

```
>  V := evl(v*T):
```

Now to find the acceleration, we need the Chain Rule. $A = \frac{dV}{dt} = \frac{dx}{dt}\frac{dV}{dx} = V_1\frac{dV}{dx}$.

```
>  A := evl(V[1]*diff(V,x)):
```

The forces acting on the skier are the gravitational force, $[0, 0, -mg]$, and the force F_s exerted by the skis on the snow. So, $F_s = mA + [0, 0, mg]$.

```
>  Fs := evl(m*A+[0,0,m*g]):
```

We assumed that the only force on the skier that has a component in the direction of the velocity vector is gravity, therefore, the component of the above vector in the direction of the tangent vector, **T**, should be zero. Most of the force between the skis and the snow surface should be perpendicular to the surface, though there can be some sideways force, which is necessary to keep the skier on the right path. The perpendicular to the surface is the normal vector, and that is given by the vector $N_s = [-f_x, -f_y, 1]$.

Note that there is another normal vector **to the curve - the principal normal vector** and these are different. Let F_n denote the component of force in this direction.

```
>  Ns := (x,y) -> evl(unit([-D[1](f)(x,y),-D[2](f)(x,y),1]));
```

$$Ns := (x, y) \to \mathrm{evl}(\mathrm{unit}([-\mathrm{D}_1(f)(x, y), -\mathrm{D}_2(f)(x, y), 1]))$$

```
>  Fn := Fs &.  Ns(x,yp):
```

(No, you really don't want to see that expanded!)

The *sideways* force will be in a direction perpendicular to both the tangent vector to the curve and the normal to the surface. Use the cross product to compute this. Of course by the definition of the cross product and the two vectors that we are using, this sideways force will always point to the skier's right, so we will call the component of the force in that direction, F_r.

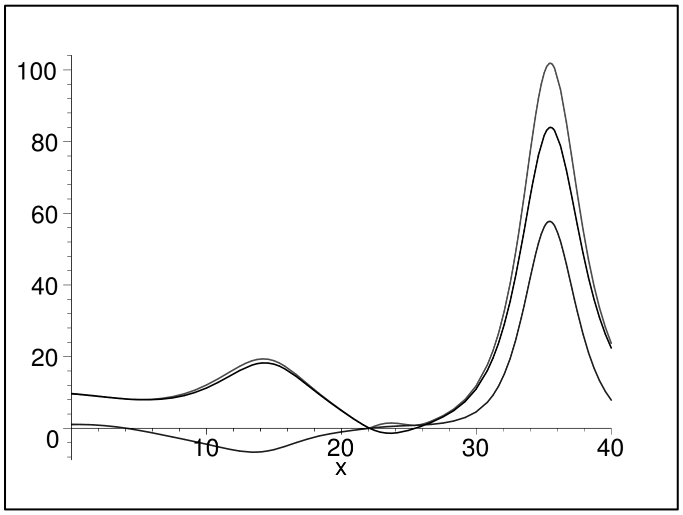
```
>  U := evl(T &x Ns(x,yp)):
```

```
>  Fr := Fs &.  U:
```

Now, we will plot these forces F_r, F_n, and $|F_s|$ as functions of x. We need an estimate for the mass of the skier, or we can normalize everything and set the mass to be 1.

```
>  m := 1:
```

```
> Fs &. Fs:
> lFs := sqrt(Fs &. Fs):
> plot({Fn,Fr,lFs},x=0..40,color=[black,blue,red],thickness=3);
```



So we have that the normal force is the blue curve, the sideways force is the black curve, and the magnitude of the force of the skis on the surface is the red curve. Note that the red curve is always the uppermost - and this should be so, since the other two are components of F_s in certain directions. Note that the sideways force seems to be small in comparison to the normal force. This actually follows nicely from the physical considerations. In general, the forces are a result of friction between two surfaces pressed together, the sideways force is no greater than some fraction (the *coefficient of friction*) of the normal force. Thus $|F_r| \leq cF_n$ for some constant c. Any attempt to violate this would result in the skier slipping off the path.

The most obvious feature of the graph is the peak that all three take together near $x = 35$ where $|F_s|$ rises to over 100 times the mass of the skier. That means that at that point the skier's legs would have to support more than 10 times his/her normal weight - which is $10(mg) = 98m$. Not many people can do that, even the heavy weight lifters. The actual result there might be a crash! Due to the forces acting on the skier there, it could result in serious injury.

Before that happens, note that at about $x = 22$ something else of interest occurs. At that point both F_n and F_r pass through 0. Recall that $|F_r| \leq c\,F_n$, so that as F_n goes to 0, F_r must also go to 0. What is happening here? Skis can push on the snow, but they cannot pull it, so F_n can never be negative. Instead, what happens when F_n hits 0 is that the skis leave the snow surface and the skier flies through the air. After that point, our formula is no longer valid, and the possible crash later on may actually be avoided - we don't know!

Now let's look in detail at what is happening at about $x = 10$. Where is the skier and what are the forces acting on him? The easiest way to do this is to set x equal to 10, evaluate all of the forces, then redefine x as a variable.

```
> dvdx := diff(v,x):
```

```
> x := 10:  R0 := evalf(R); v0 := evalf(v); T0 := evalf(T);
```

$$R0 := [10., 3.040387694, -4.276707539]$$

$$v0 := 9.155515702$$

$$T0 := [.7712439854, .4774018353, -.4210346804]$$

The skier has dropped 4.28 meters in altitude from the start and is travelling at about 9.16 m/s (almost 33 km/h, or almost 20 mph). We can express the direction of motion given by the tangent vector T in terms of angles in spherical coordinates, centered at the point where $x = 10$.

```
> phi0 := evalf(arccos(T0[3])*180/Pi);
```

$$\phi 0 := 114.8999284$$

```
> theta0 := evalf(arctan(T0[2],T0[1])*180/Pi);
```

$$\theta 0 := 31.75760468$$

So at this moment, the skier's direction is about 24 degrees below horizontal ($\phi = 90$) and about 32 degrees counterclockwise from the positive x-axis.

```
> A0 := evalf(A);
```

$$A0 := [2.342335426, 5.478990179, .7031539735]$$

Now, the component of the skier's acceleration in the direction of the tangent vector is the rate of change of speed, which is calculated by the chain rule as $\frac{dv}{dt} = \frac{dv}{dx}\frac{dx}{dt}$. This should also be the component of the gravitational acceleration $[0, 0, -g]$ because we assumed that the non-gravitational forces have no component in this direction. Thus, we need to compute $\mathbf{A} \cdot \mathbf{T}$ at this point, $\frac{dv}{dt}$ at this point and the gravitational component in this direction at this point.

```
> [A0 &.  T0, evalf(dvdx*V[1]),[0,0,-g]&.T0];
```

$$[4.126139868, 4.126139870, 4.126139868]$$

These are pretty close! Let's look at the force F_s of the snow on the skis, its magnitude and the components normal to the snow surface and sideways along the surface.

```
> Fs0 := evalf(Fs); len(Fs0); Fn0 := evalf(Fn); Fr0:=evalf(Fr); Ns0 :=
> evalf(Ns(x,yp));
```

$$Fs0 := [2.342335426, 5.478990179, 10.50315397]$$

$$12.07568267$$

$$Fn0 := 11.23306395$$

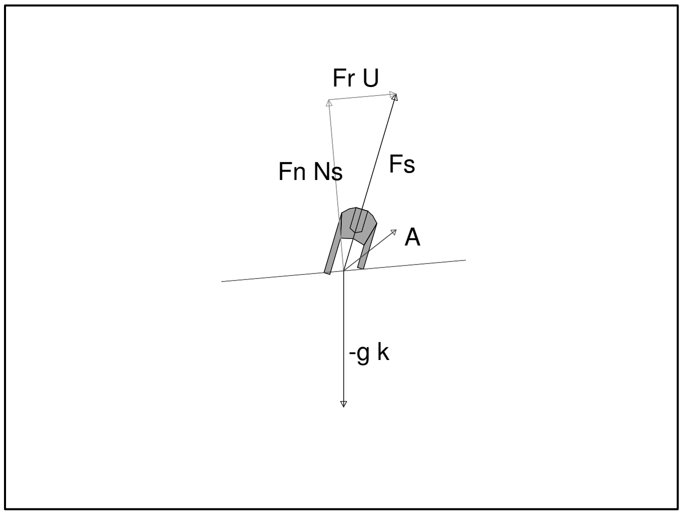
$$Fr0 := -4.431747544$$

$$Ns0 := [.4029343439, .1459030579, .9035243286]$$

Now let's look at the situation:

```
> VP := T0:  origin := [0,0,0]:  U0:= evalf(U):
> B := evl(unit(Fs0)):  C := evl(VP &x B): p:= 1/(C &.  U0):
> rleg := polygonplot3d([evl(-.3*p*U0),evl(-.3*C+.9*B),evl(-.2*C+.5*B),
> evl(-.3*p*U0+.1*C)],color=tan):
> lleg := polygonplot3d([evl(+.3*p*U0),evl(+.3*C+.9*B),evl(+.2*C+.5*B),
> evl(+.3*p*U0-.1*C)],color=tan):
```

```
> body:= polygonplot3d([evl(-.3*C+.9*B), evl(-.2*C+B),
> evl(-.1*C+1.05*B), evl(+.1*C+1.05*B), evl(+.1*C+.7*B),
> evl(.65*B),evl(-.1*C+.7*B), evl(-.1*C+1.05*B), evl(.1*C+1.05*B),
> evl(.2*C+B), evl(.3*C+.9*B), evl(.2*C+.5*B), evl(.55*B),
> evl(-.2*C+.5*B)], color=tan):
> labels := textplot3d({[op(evl(.15*Fn0*Ns0+.4*U0)),‘ Fn Ns
> ‘,align=RIGHT], [op(evl(.28*Fn0*Ns0-.45*U0)),‘ Fr U ‘],
> [op(evl(.15*Fs0-.4*U0)),‘ Fs ‘], [op(evl(.2*A0-.4*U0)),‘ A ‘],
> [op(evl([0,0,-.15*g]-.4*U0)),‘ -g k ‘]}, color=black):
> display({arrow(origin, A0/4,color=red), arrow(origin, Fs0/4,
> color=blue), arrow(origin,[0,0,-g/4], color=blue),
> arrow(origin,Fn0*Ns0/4,color=green),arrow(Fn0*Ns0/4,Fr0*U0/4,color=gre
> en),
> polygonplot3d([evl(-2*U0),evl(2*U0)],style=wireframe,color=magenta),
> rleg,
> body,lleg,labels},orientation=[theta0,phi0],scaling=constrained);
```



13.4.1 Curvature, Torsion and the Frenet Frame

When you studied curves in 3-space, we found that there were three special vectors for each curve, and they carried a lot of information about the curve. They were the unit tangent vector, the unit normal vector (or principal normal), and the binormal vector. You start with the unit tangent vector, **T**. You have to re-parameterize the curve in terms of the arclength parameter, s. Then $\frac{dT}{ds}$ points in the normal direction, and the principal normal vector is the unit vector in that direction. We define the curvature to be $\kappa = \left|\frac{dT}{ds}\right|$. This makes the principal normal vector $N = \frac{1}{\kappa}\frac{dT}{ds}$. The binormal is defined to be the cross product of the unit tangent and the principal normal vector. Then we can show that the derivative of the binormal is parallel to the principal normal and $\frac{dB}{ds} = -\tau\, N$, and the quantity τ is called the *torsion.* The curvature, κ, measures how much the path curves at a point, while the torsion, τ, measures how much it twists. Normally, we do not have curves parameterized by arclength - it is hard to do so, usually. We usually have the curve parameterized by another variable t. Then by the Chain Rule if we let $v = \frac{ds}{dt}$ denote the speed, then $\frac{d}{ds} = \frac{1}{v}\frac{d}{dt}$. This leads to an alternative set of formulas for the quantities above which, while they may look more complicated, lead to simpler, less complicated calculations.

$$V \times A = v^3\,\kappa\,B$$
$$N = B \times T$$
$$\tau = \frac{V \times A}{|V \times A|^2}\frac{dA}{dt}$$

Let's look at the curve $x = \cos(t)$, $y = \sin(t)$, and $z = \frac{1}{2}\sin(2t)$. Find the curvature and torsion for this curve.

```
>  x := 'x':
>  R := [cos(t),sin(t),sin(2*t)/2]:
>  V := diff(R,t); A := diff(V,t); dAdt := diff(A,t);
```

$$V := [-\sin(t),\ \cos(t),\ \cos(2\,t)]$$
$$A := [-\cos(t),\ -\sin(t),\ -2\sin(2\,t)]$$
$$dAdt := [\sin(t),\ -\cos(t),\ -4\cos(2\,t)]$$

```
>  v := combine(len(V),trig);
```

$$v := \frac{1}{2}\sqrt{6 + 2\cos(4\,t)}$$

Note that `unit(V)` or `evl(V/v)` will both give the unit tangent vector, **T**. However, `unit(V)` will need to be simplified.

```
>  T := evl(V/v);
```

$$T := [-2\,\frac{\sin(t)}{\sqrt{6+2\cos(4\,t)}},\ 2\,\frac{\cos(t)}{\sqrt{6+2\cos(4\,t)}},\ 2\,\frac{\cos(2\,t)}{\sqrt{6+2\cos(4\,t)}}]$$

We will need $V \times A$ several times, so let's go ahead and calculate it.

```
>  VxA := combine(V &x A,trig);
```

$$VxA := [-\frac{1}{2}\sin(3\,t) - \frac{3}{2}\sin(t),\ -\frac{3}{2}\cos(t) + \frac{1}{2}\cos(3\,t),\ 1]$$

By our formula above this is $v^3\kappa B$, so this vector has length $v^3\kappa$ and the unit vector in this direction is the binormal, **B**. So we can calculate **B** and κ.

```
>  B := combine(unit(VxA),trig);
```

$$B := [\frac{-\sin(3\,t) - 3\sin(t)}{\sqrt{14 - 6\cos(4\,t)}},\ \frac{-3\cos(t) + \cos(3\,t)}{\sqrt{14 - 6\cos(4\,t)}},\ 2\,\frac{1}{\sqrt{14 - 6\cos(4\,t)}}]$$

```
>  kappa := combine(len(VxA)/v^3,trig);
```

$$\kappa := 4\,\frac{\sqrt{14 - 6\cos(4\,t)}}{(6 + 2\cos(4\,t))^{(3/2)}}$$

Now, **N** is the cross product $\mathbf{B} \times \mathbf{T}$, and τ is given by the formula above.

```
>  N := combine(B &x T,trig);
```

$$N := [\frac{-6\cos(t) - 3\cos(3\,t) + \cos(5\,t)}{\sqrt{14 - 6\cos(4\,t)}\,\sqrt{6 + 2\cos(4\,t)}}, \frac{-6\sin(t) + \sin(5\,t) + 3\sin(3\,t)}{\sqrt{14 - 6\cos(4\,t)}\,\sqrt{6 + 2\cos(4\,t)}},$$
$$-8\,\frac{\sin(2\,t)}{\sqrt{14 - 6\cos(4\,t)}\,\sqrt{6 + 2\cos(4\,t)}}]$$

```
>  tau := combine(VxA &.  dAdt/len(VxA)^2,trig);
```

$$\tau := 6\,\frac{\cos(2\,t)}{-7 + 3\cos(4\,t)}$$

Now, we want to see the curve and these three vectors, $(\mathbf{T}, \mathbf{B}, \mathbf{N})$ - called the Frenet frame - at different points. First we will make them into procedures.

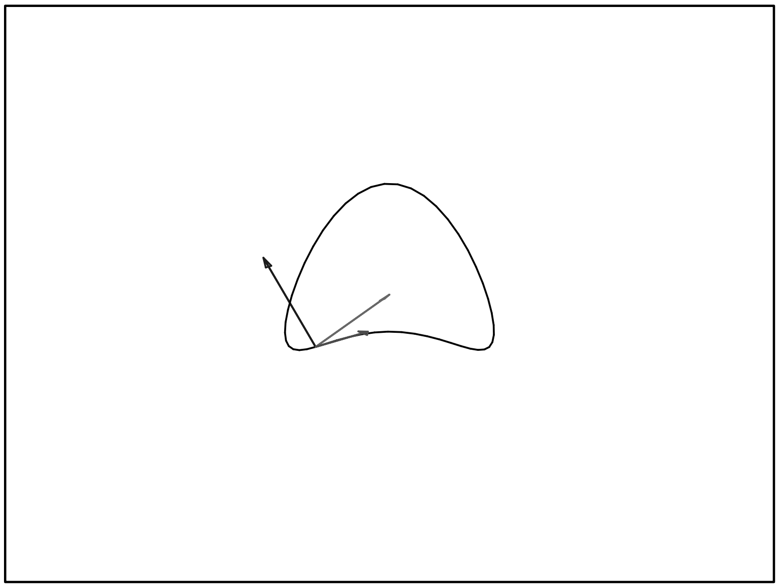
```
>  R := unapply(R,t):
>  T := unapply(T,t):
>  N := unapply(N,t):  B := unapply(B,t):
>  kappa := unapply(kappa,t):  tau := unapply(tau,t):
>  curve := spacecurve(R(t),t=0..2*Pi,color=black):
```

Now we want to draw the vectors, $\mathbf{T}(t)$, $\mathbf{B}(t)$, and $\mathbf{N}(t)$ at each point $\mathbf{R}(t)$ along the curve. Color code them so that $\mathbf{T}$ is red, $\mathbf{B}$ is blue, and $\mathbf{N}$ is magenta.

```
>  frenet := t->
>  display(
>  {curve,arrow(R(t),T(t),color=red),arrow(R(t),N(t),color=magenta),arro
>  w(R(t),B(t),color=blue)},thickness=3);
```

$frenet := t \to \mathrm{display}(\{curve, \mathrm{arrow}(\mathrm{R}(t), \mathrm{B}(t), color = blue),$
$\mathrm{arrow}(\mathrm{R}(t), \mathrm{N}(t), color = magenta), \mathrm{arrow}(\mathrm{R}(t), \mathrm{T}(t), color = red)\}, thickness = 3)$

```
>  display([seq(frenet(kk*Pi/8),
>  kk=0..15)],insequence=true,scaling=constrained);
```



Let's take a look at our inductor running around the torus:

```
> R := [(2+sin(10*t))*cos(t),(2+sin(10*t))*sin(t),cos(10*t)];
```

$$R := [(2+\sin(10\,t))\cos(t),\,(2+\sin(10\,t))\sin(t),\,\cos(10\,t)]$$

```
> V := diff(R,t):  A := diff(V,t):  dAdt := diff(A,t):
> v := combine(len(V),trig):
> T := evl(V/v):
> VxA := combine(V &x A,trig):
> B := combine(unit(VxA),trig):
> kappa := combine(len(VxA)/v^3,trig):
> N := combine(B &x T,trig):
> tau := combine(VxA &.  dAdt/len(VxA)^2,trig):
> R := unapply(R,t):
> T := unapply(T,t):
> N := unapply(N,t):  B := unapply(B,t):
> kappa := unapply(kappa,t):  tau := unapply(tau,t):
> curve := spacecurve(R(t),t=0..2*Pi,numpoints=501,color=black):
> display([seq(frenet(kk*Pi/20),
> kk=0..41)],insequence=true,scaling=constrained);
```