

- Introduction

This was intended to be a transcription of the demonstration on November 21, 2001. Unfortunately, WebDrive didn't work properly and most of the worksheet was lost. An attempt will be made to reconstruct examples from memory. Some examples will be lost, but others will be enhanced by this effort.

+ Background

- Some surfaces

- The helicoid

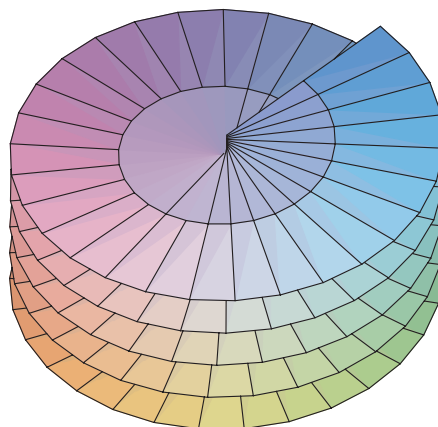
```
> Helicoid:=[r*cos(u),r*sin(u),u/30];
```

$$\text{Helicoid} := \left[r \cos(u), r \sin(u), \frac{1}{30} u \right]$$

```
> Hplot:=plot3d(Helicoid,r=0..1,u=-5*Pi..5*Pi,grid=[3,150]):
```

```
> display(Hplot,title="The real RUscrew");
```

The real RUscrew



The "grid" option adjusts the number of values of each parameter. More points

give a smoother plot, but require more computation. The help file does not reveal what the default setting is, but examination of a plot without this option reveals that it is 25 for each variable. For this surface, a much coarser grid in the radial direction r combined with a much finer grid in the angular direction u gives a fairly smooth image with a useful grid shown on the surface. The grid curves in r for fixed u are straight lines, so there is little need to subdivide them. However, the curves in u for fixed r are helices in which both curvature and torsion need to be shown.

– The torus

```
[ > a:=5:b:=2:
```

```
[ These give fixed parameters for the radii of the circles used to construct the  
[ torus.
```

```
[ > Torus:=[(a+b*cos(v))*cos(u),(a+b*cos(v))*sin(u),b*si  
[ n(v)];
```

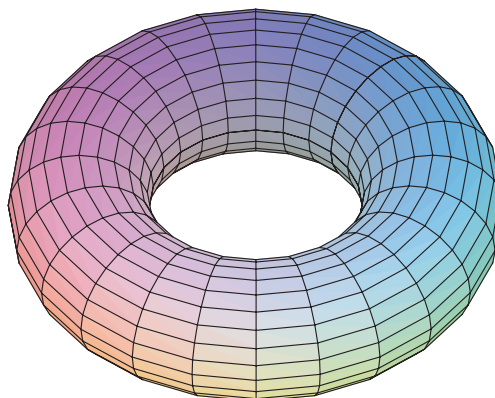
```
[ Torus := [(5 + 2 cos(v)) cos(u), (5 + 2 cos(v)) sin(u), 2 sin(v)]
```

```
[ To cover the whole surface, each parameter should go through an interval of  
[ length 2Pi.
```

```
[ > Tplot:=plot3d(Torus,u=-Pi..Pi,v=-Pi..Pi):
```

```
[ > display(Tplot,title="A perfect donut");
```

A perfect donut



Surfaces from Lab 4

Lab 4 will not be assigned in this section. Excerpts from it will be presented here. You can find the lab description and seed file from the semester web page if you are interested in more operations with these surfaces.

```
> y1:=8-z^2-2*x^2;  
> y2:=z^2;
```

$$y1 := 8 - z^2 - 2x^2$$
$$y2 := z^2$$

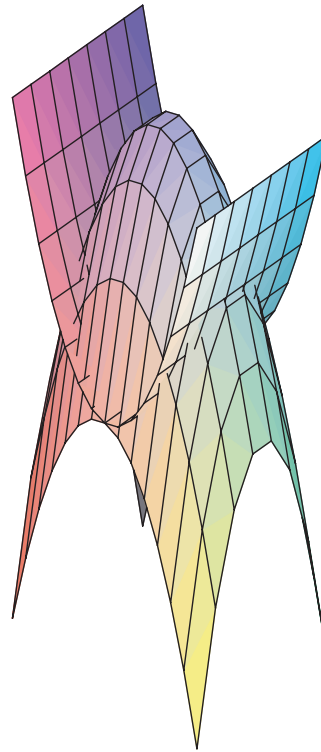
Here x and z are taken as independent variables and y is the dependent variable just to show that it can be done. The names of the variables cause no difficulty in the `plot3d` command. The intervals are chosen to give a rectangle that just contains the part of the paraboloid $y1$ that is on the positive side of the xz plane. A number c is introduced as an abbreviation for one of these dimensions. A coarser grid is used to allow more surface and less grid to be seen. If you aren't sure that $y1 > y2$ for the bounded region, you can construct separate plots in solid colors and combine them in a `display` command with an option that includes marked axes.

```
> c:=sqrt(8);
```

$$c := 2\sqrt{2}$$

```
> plot3d({y1,y2},x=-2..2,z=-c..c,grid=[8,15],title="The surfaces in a box");
```

The surfaces in a box



This lab concentrates on triple integrals over the region between these surfaces. Since each is given with y a function of x and z , we can find the intersection by identifying where $y_1=y_2$. This is very easy to do by hand, but Maple can be forced to give something useful.

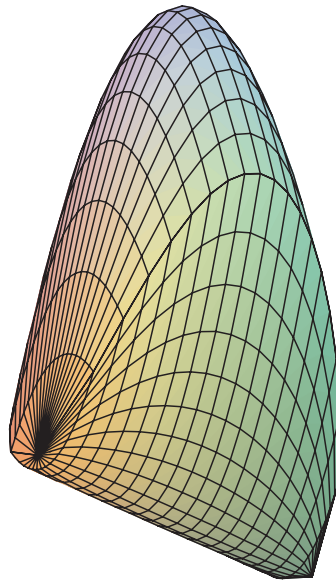
```
> zbd:=solve(y1=y2,z);
```

$$zbd := \sqrt{4-x^2}, -\sqrt{4-x^2}$$

This gives the two ends of the z interval for each fixed x . Note that the larger end is given first. Examining this expression shows that this description is valid for x between -2 and 2 . We can now plot the surfaces restricted to the part that actually bound the solid body of interest.

```
> plot3d({y1,y2},x=-2..2,z=zbd[2]..zbd[1],grid=[20,20],title="The football");
```

The football



– A tangent plane to the helicoid

[We select a point in the parameter space and use it to give a point on the surface.

[> `ptP := {r=1/2, u=Pi/3};`

$$ptP := \left\{ r = \frac{1}{2}, u = \frac{1}{3} \pi \right\}$$

[> `ptH := subs(ptP, Helicoid);`

$$ptH := \left[\frac{1}{2} \cos\left(\frac{1}{3} \pi\right), \frac{1}{2} \sin\left(\frac{1}{3} \pi\right), \frac{1}{90} \pi \right]$$

[> `eval(%); #The algebraic form of the point.`

$$\left[\frac{1}{4}, \frac{1}{4} \sqrt{3}, \frac{1}{90} \pi \right]$$

[We find the normal using the cross product of the tangents to the grid curves. Then

a simple construction gives the equation of the tangent plane.

```
> Hr:=diff(Helicoid,r);
```

```
> Hu:=diff(Helicoid,u);
```

```
> HN:=crossP(Hr,Hu);
```

$$Hr := [\cos(u), \sin(u), 0]$$

$$Hu := \left[-r \sin(u), r \cos(u), \frac{1}{30} \right]$$

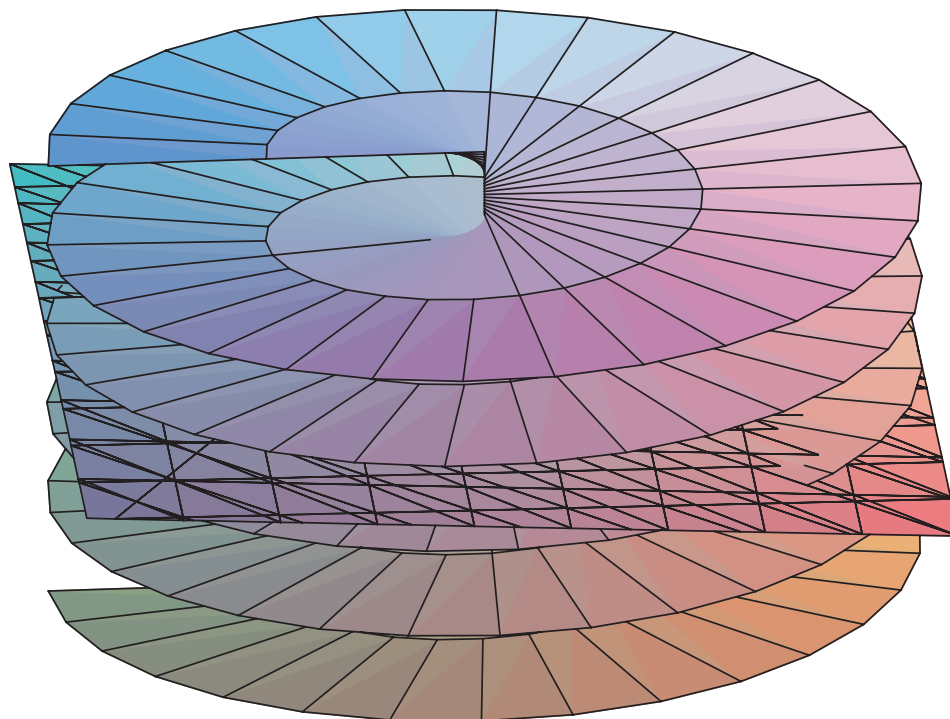
$$HN := \left[\frac{1}{30} \sin(u), -\frac{1}{30} \cos(u), \cos(u)^2 r + \sin(u)^2 r \right]$$

```
> tanP:=dotP(subs(ptP,HN),[x-ptH[1],y-ptH[2],z-ptH[3]])=0;
```

$$\tan P := \frac{1}{60} \sqrt{3} \left(x - \frac{1}{4} \right) - \frac{1}{60} y + \frac{1}{240} \sqrt{3} + \frac{1}{2} z - \frac{1}{180} \pi = 0$$

```
> TPplot:=implicitplot3d(tanP,x=-1..1,y=-1..1,z=-1/2..1/2):
```

```
> display({TPplot,Hplot},title="helicoid with a tangent plane");
```



– Surface area of a slice of the torus.

```
> p:=6;
```

```
p:=6
```

We will cut off the portion of the torus to the right of $x=p$ and find its area. A symbolic name is used to make it easy to modify this method to consider other sections of this type. Here the variable x on the torus is the first entry of the list Torus. The inequality leads to a statement in parameter space that we attempt to solve.

```
> EqT:=Torus[1]=6;
```

$$EqT := (5 + 2 \cos(v)) \cos(u) = 6$$

```
> uBd:=solve(EqT,u);
```

$$uBd := \arccos\left(6 \frac{1}{5 + 2 \cos(v)}\right)$$

This gives one value of u for each acceptable u on the curve. Maple needs our help to do more. We are actually interested in the values of u for which the cosine is larger than it is here. From what we know about the behavior of the cosine function, this is a symmetric interval around 0 with this as its right endpoint. We also need to know the values of v for which this quantity can be defined as a real number.

Again, large values of cos(v) are acceptable, and this corresponds to v=0, but small values will lead us to try to find the arccos of a number greater than 1.

```
> vBd:=solve(uBd,v);
```

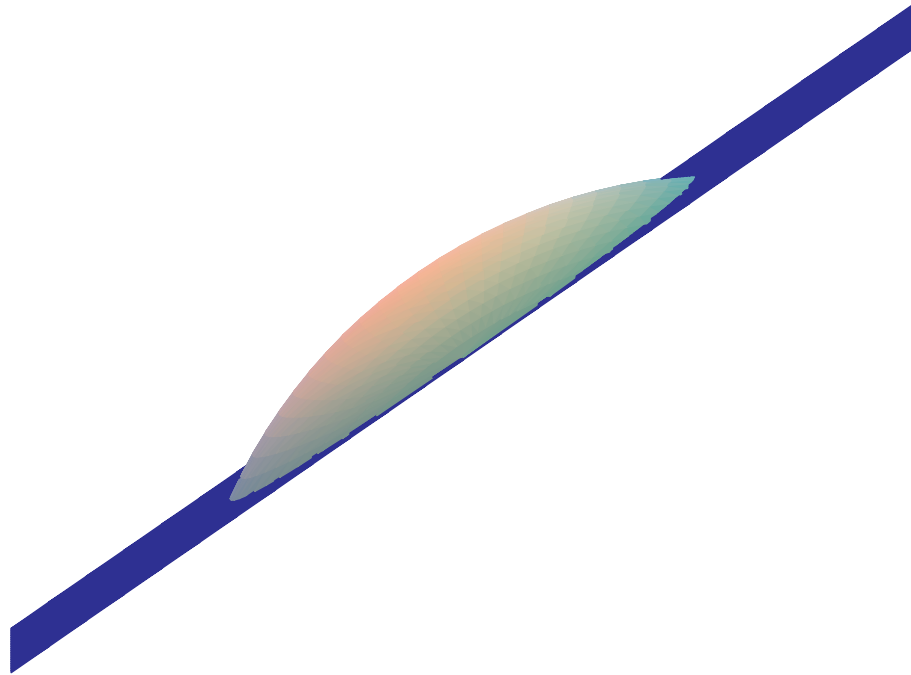
$$vBd := \frac{1}{3} \pi$$

To check, we construct the portion of the torus defined by this piece of parameter space.

```
> PlotTorusPart:=plot3d(Torus,v=-vBd..vBd,u=-uBd..uBd,style=PATCHNOGRID):
```

```
> PlaneCut:=plot3d([p,y,z],y=-(a+b)..(a+b),z=-b..b,style=PATCHNOGRID,color=BLUE):
```

```
> display({PlotTorusPart,PlaneCut},title="Sliced Donut");
```



The options were chosen to keep grid lines from interfering with the view while confirming that we have exactly the portion on one side of the plane. More than one view might be needed to be convincing.

Now that we have the exact region defined in parameter space, an integral for the surface area can be constructed. A normal to a parameterized surface is found as it was for the Helicoid.

```

> TorDu:=diff(Torus,u);
> TorDv:=diff(Torus,v);
      TorDu := [-(5 + 2 cos(v)) sin(u), (5 + 2 cos(v)) cos(u), 0]
      TorDv := [-2 sin(v) cos(u), -2 sin(v) sin(u), 2 cos(v)]
> TorN:=crossP(TorDu,TorDv);
TorN := [2 (5 + 2 cos(v)) cos(u) cos(v), 2 (5 + 2 cos(v)) sin(u) cos(v),
        2 (5 + 2 cos(v)) sin(u)2 sin(v) + 2 (5 + 2 cos(v)) cos(u)2 sin(v)]

```

```
> TorDS:=simplify(lenV(TorN));
```

$$TorDS := 2 \operatorname{csgn}(5 + 2 \cos(v)) (5 + 2 \cos(v))$$

The "simplify" part was added when the length was found not to be simple enough. This still isn't simple because of the "csgn" factor. Help identifies this as a sign of the real part of a complex number, but we are confining attention to a domain on which the argument of that function is real and positive, so this is just a messy version of the number 1. The length is quite manageable, so we try to integrate it over our region in parameter space. We first do this step by step, then the pieces will be cut and pasted to give a single expression for the double integral. Normally, only the latter need be given, and numerical methods used if the symbolic approach fails.

```
> SIu:=int(TorDS,u=-uBd..uBd);
```

$$SIu := 4 \operatorname{csgn}(5 + 2 \cos(v)) (5 + 2 \cos(v)) \arccos\left(6 \frac{1}{5 + 2 \cos(v)}\right)$$

```
> SI:=int(SIu,v=-vBd..vBd);
```

$$SI := \int_{-1/3\pi}^{1/3\pi} 4 \operatorname{csgn}(5 + 2 \cos(v)) (5 + 2 \cos(v)) \arccos\left(6 \frac{1}{5 + 2 \cos(v)}\right) dv$$

```
> evalf(SI);
```

$$24.08894671$$

```
> SIcheck:=evalf(int(int(TorDS,u=-uBd..uBd),v=-vBd..vBd),20);
```

$$SIcheck := 24.088946712500310435$$

The numerical computation to additional accuracy took a long time. I initially tried even greater accuracy, but it took an unacceptable amount of time.

– Some properties of the football of Lab 4

We know that we have an exact description of the solid studied in Lab 4, so we can set up triple integrals for physical properties of the object. We will step through the evaluation of the integrals, but then construct a single formula giving the whole computation as a check.

First the volume. We need to integrate 1 over the region.

```
> FVII:=int(1,y=y2..y1);
```

$$FVII := 8 - 2z^2 - 2x^2$$

```
> FVI2:=int(FVII,z=zbd[2]..zbd[1]);
```

$$FVI2 := 16\sqrt{4-x^2} - \frac{4}{3}(4-x^2)^{(3/2)} - 4x^2\sqrt{4-x^2}$$

```
> FVI:=int(FVI2,x=-2..2);
```

$$FVI := 16\pi$$

```
> int(int(int(1,y=y2..y1),z=zbd[2]..zbd[1]),x=-2..2);
```

$$16\pi$$

This illustrates a single formula for the volume. Now try the single formula to find all first moments.

```
> FMx:=int(int(int(x,y=y2..y1),z=zbd[2]..zbd[1]),x=-2..2);
```

$$FMx := 0$$

```
> FMy:=int(int(int(y,y=y2..y1),z=zbd[2]..zbd[1]),x=-2..2);
```

$$FMy := \frac{160}{3}\pi$$

```
> FMz:=int(int(int(z,y=y2..y1),z=zbd[2]..zbd[1]),x=-2..2);
```

$$FMz := 0$$

The integrals that are zero are zero for reasons of symmetry.

The divergence theorem

The first steps of these integrals echo a proof of the divergence theorem by giving a vector field concentrated in the second component (that is, the y component) whose divergence is the integrand of the triple integral. Here are those vector fields.

```
> VFM:=[0,y,0];
```

```
> VFMx:=[0,x*y,0];
```

```
> VFMy:=[0,y^2/2,0];
```

```
> VFMz:=[0,y*z,0];
```

$$VFM := [0, y, 0]$$

$$VFMx := [0, xy, 0]$$

$$VFMy := \left[0, \frac{1}{2}y^2, 0 \right]$$

$$VFMz := [0, yz, 0]$$

To integrate against the outward pointing vector element of surface area, we use gradients to construct the normal vectors. These will be normalized so that the integrals are to be performed with respect to x and z over the projection of the

figure. These lead to the integrals used in the second and third steps of the detailed integration. The divergence theorem gives a form of the first step. Here are the normal vectors, obtained as gradients, with formulas set up to force the vectors to be outward pointing. Note that the second components have been normalized.

```
> Ny1:=grad(y-y1);
> Ny2:=grad(y2-y);
```

$$Ny1 := [4x, 1, 2z]$$

$$Ny2 := [0, -1, 2z]$$

The integrands are found by taking the dot products of the vector fields with the normal vectors and substituting the expression for y on each surface. This will be constructed as a function to allow it to be used more easily. For these examples, this looks like a complicated way to do the first step of the iterated integral. All of our vector fields are concentrated in the second component which has been normalized in the normal vectors, so the value will simply be the difference of the values of the components on the two surfaces -- exactly the result of the first integral.

```
> Flux:=VF->subs(y=y1,dotP(VF,Ny1))+subs(y=y2,dotP(VF,Ny2));
```

$$Flux := VF \rightarrow \text{subs}(y = y1, \text{dotP}(VF, Ny1)) + \text{subs}(y = y2, \text{dotP}(VF, Ny2))$$

```
> IG1:=Flux(VFM);
> IGx:=Flux(VFMx);
> IGy:=Flux(VFMy);
> IGz:=Flux(VFMz);
```

$$IG1 := 8 - 2z^2 - 2x^2$$

$$IGx := x(8 - z^2 - 2x^2) - xz^2$$

$$IGy := \frac{1}{2}(8 - z^2 - 2x^2)^2 - \frac{1}{2}z^4$$

$$IGz := (8 - z^2 - 2x^2)z - z^3$$

Integrating over the projection has the following description

```
> int(int(IG1,z=zbd[2]..zbd[1]),x=-2..2); #Volume
```

$$16\pi$$

```
> int(int(IGy,z=zbd[2]..zbd[1]),x=-2..2); #y moment
```

$$\frac{160}{3}\pi$$

Here are the inner integrals for the other moments.

```
> int(IGx,z=zbd[2]..zbd[1]);
```

$$-\frac{4}{3}x(4-x^2)^{(3/2)} + 2x(8-2x^2)\sqrt{4-x^2}$$

```
> int(IGz, z=zbd[2]..zbd[1]);
```

0

Other vector fields can be used to find these moments. For example, one can use the vector fields corresponding to integrating first with respect to z. In contrast to a change of order of integration, this convenient description of the surface can continue to be used to evaluate a different flux integral. The divergence theorem guarantees that the integral has the correct physical interpretation even if it looks different. Here are the vector fields that correspond to integrating first with respect to z.

```
> VFMa := [0, 0, z];
```

```
> VFMax := [0, 0, x*z];
```

```
> VFMay := [0, 0, y*z];
```

```
> VFMAz := [0, 0, z^2/2];
```

$$VFMA := [0, 0, z]$$

$$VFMax := [0, 0, xz]$$

$$VFMay := [0, 0, yz]$$

$$VFMAz := \left[0, 0, \frac{1}{2}z^2 \right]$$

```
> IG1a := Flux(VFMA);
```

```
> IGxa := Flux(VFMax);
```

```
> IGya := Flux(VFMay);
```

```
> IGza := Flux(VFMAz);
```

```
>
```

$$IG1a := 4z^2$$

$$IGxa := 4xz^2$$

$$IGya := 2(8 - z^2 - 2x^2)z^2 + 2z^4$$

$$IGza := 2z^3$$

```
> int(int(IG1a, z=zbd[2]..zbd[1]), x=-2..2); #Volume
```

16π

```
> int(int(IGya, z=zbd[2]..zbd[1]), x=-2..2);
```

$\frac{160}{3}\pi$