

Math 4530

Solutions to hw set 4 - due February 11

I and II from page 7 of February 4 notes

Chapter 2: 1.12, 1.20, 1.21, 1.22, 2.7, 2.9, 5.6, 5.7, 5.8

This file contains some of the solutions to the chapter 2 problems. Problems 1.22, I, II, 2.7, 2.9 are done separately, by hand.

[> with(plots):

1.12: The catenoid is the surface of revolution obtained by revolving $y=\cosh(x)$ about the x-axis. So a patch is given by

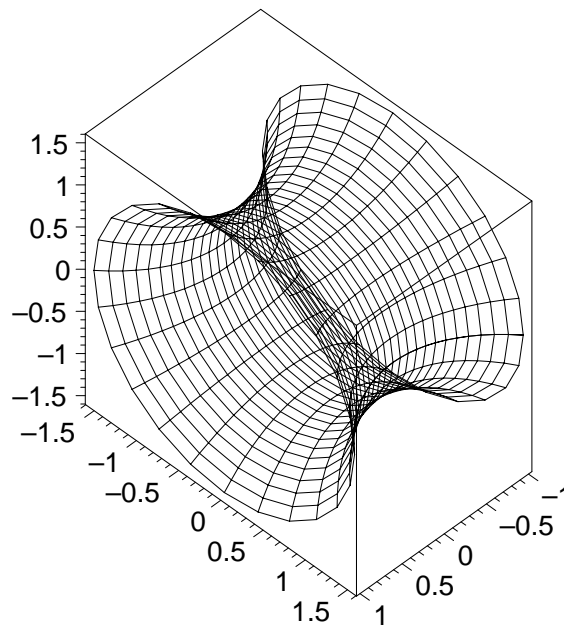
[> X:=(u,v)->[u,cosh(u)*cos(v),cosh(u)*sin(v)];

$$X := (u, v) \rightarrow [u, \cosh(u) \cos(v), \cosh(u) \sin(v)]$$

the patch will be 1-1 as long as $0 < v < 2\pi$. The general discussion of what makes a surface of revolution patch in general is contained on the final page of the hand-written solutions. Here's a picture:

[> with(plots):

[> plot3d(X(u,v),u=-1..1,v=0..2*Pi,axes=boxed,color=black, style=wireframe,scaling=constrained);



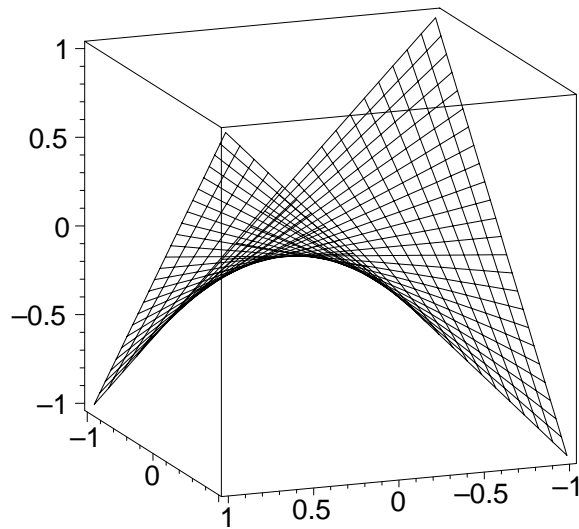
1.20) The Monge parameterization yields a double ruling:

[> Z:=(u,v)->[u,v,u*v];

$$Z := (u, v) \rightarrow [u, v, v u]$$

We can write $Z(u,v)$ as $u \cdot \langle 1,0,v \rangle + \langle 0,v,0 \rangle$ or as $v \cdot \langle 0,1,u \rangle + \langle u,0,0 \rangle$. These are different rulings because the direction vectors are linearly independent (look at the first two entries.) Since Maple shows the coordinate lines when we plot a parametric surface, you can see the double ruling by using the parameterization above!

[> plot3d(Z(u,v),u=-1..1,v=-1..1,axes=boxed,scaling=constrained, style=wireframe,color=black);



1.21) The helicoid parameterization already shows it is a ruled surface

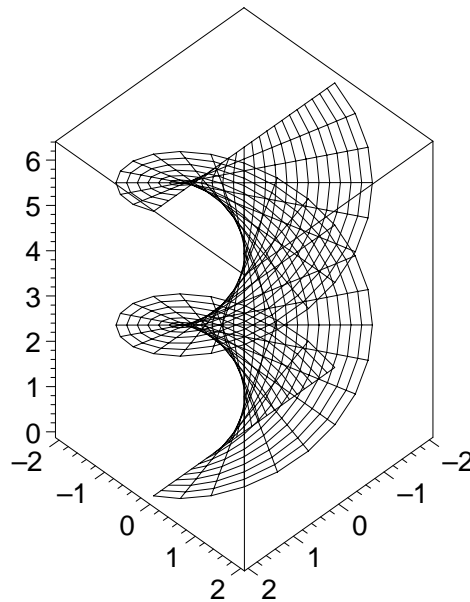
```
> H:=(u,v)->[a*u*cos(v),a*u*sin(v),b*v];
```

$$H := (u, v) \rightarrow [a u \cos(v), a u \sin(v), b v]$$

which we can write as $H(u,v) = u \cdot \langle a \cos(v), a \sin(v), 0 \rangle + \langle 0, 0, v \cdot b \rangle$. Here is a picture:

```
> a:=1:b:=1:
```

```
plot3d(H(u,v),u=-2..2,v=0..2*Pi,scaling=constrained,color=black,
axes=boxed,style=wireframe);
```

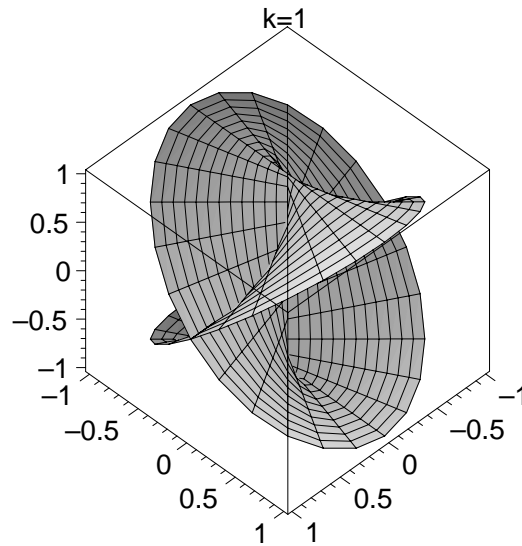


1.22, 2.1.6, 2.1.7: These are done by hand, see the hand work file. Related pictures are shown below.

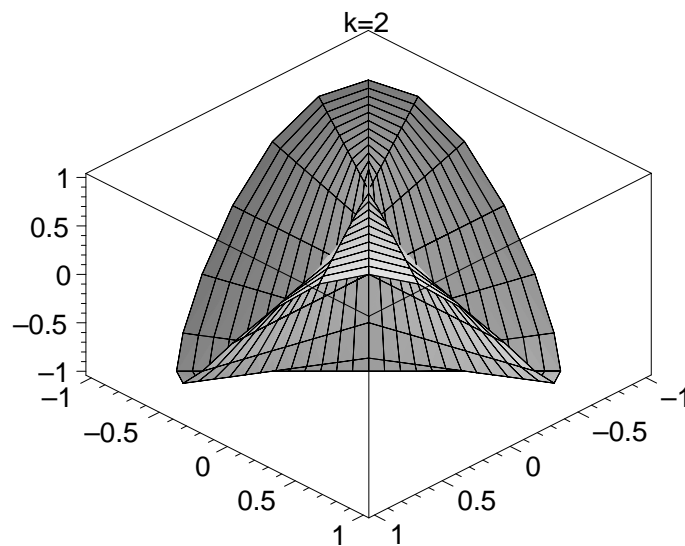
5.6) For fixed u the ruling below traces out a horizontal line with angle u (and $u+\pi$) from the x -direction. As u varies the line moves up and down, with vertical coordinate $\sin(ku)$. Thus, the larger

the k value, the more quickly, or often, the ruling line oscillates. As long as k is an integer the surface will close up as u varies from 0 to 2π .

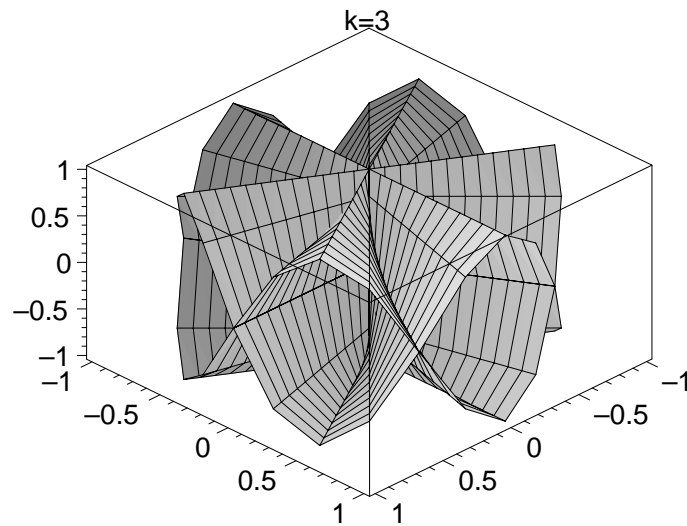
```
[ > Rul:=[v*cos(u),v*sin(u),sin(k*u)]:  
[ > Rul1:=subs(k=1,Rul);  
      Rul1 := [v cos(u), v sin(u), sin(u)]  
[ > plot3d(Rul1,u=-Pi..Pi,v=-1..1,axes=boxed,scaling=constrained,  
  title='k=1');
```



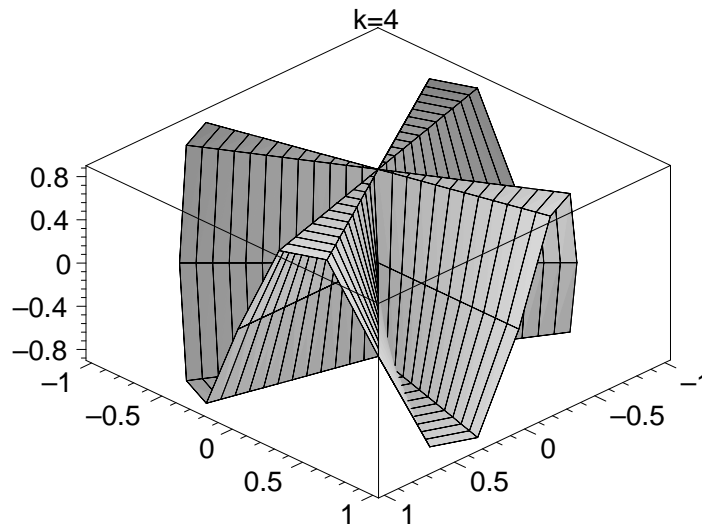
```
[ > Rul2:=subs(k=2,Rul):  
[ > plot3d(Rul2,u=-Pi..Pi,v=-1..1,axes=boxed,title='k=2');
```



```
[ > Rul3:=subs(k=3,Rul):  
  plot3d(Rul3,u=-Pi..Pi,v=-1..1,axes=boxed,title='k=3');
```



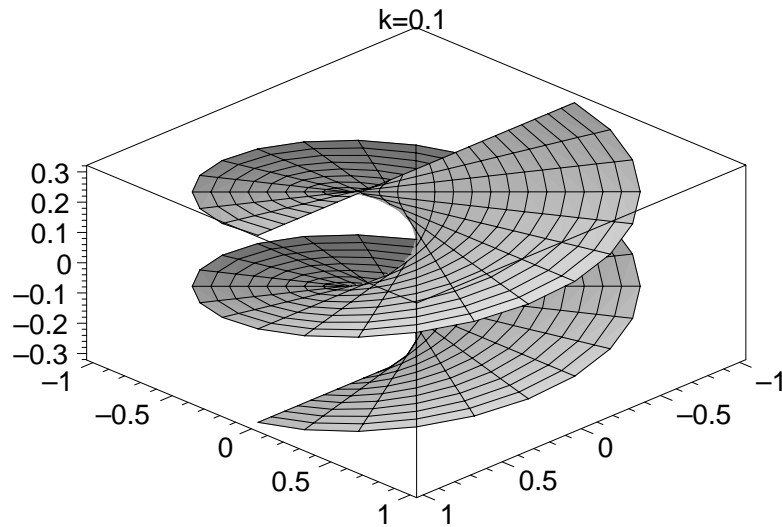
```
> Rul4:=subs(k=4,Rul):
  plot3d(Rul4,u=-Pi..Pi,v=-1..1,axes=boxed,title='k=4');
```



Conclusions: if we take integer k 's, then the even- k pictures cover a surface "doubly" and are graphs above the x - y plane (except at the origin.) The odd- k surfaces are "double-sheeted graphs" with respect to the x - y plane. The reason for this is the interplay between the vertical component $\sin(ku)$ and the ruling lines: If k is even then the image of $(u+\pi, -v)$ is the same as the image of (u, v) . But if k is odd these image points have the same horizontal coordinates, but the opposite vertical ones. The even k surfaces look like graphs one studies in Math 3220, when one discusses continuity vs. continuity along lines.

Some of you also discovered that it is interesting to take k near zero - in this case, if you let Maple rescale, pieces of the surface starts to look like pieces of a helicoid, because $\sin(kv)$ is about kv , for kv small:

```
> Rultiny:=subs(k=.1,Rul):
  plot3d(Rultiny,u=-Pi..Pi,v=-1..1,axes=boxed,title='k=0.1');
```

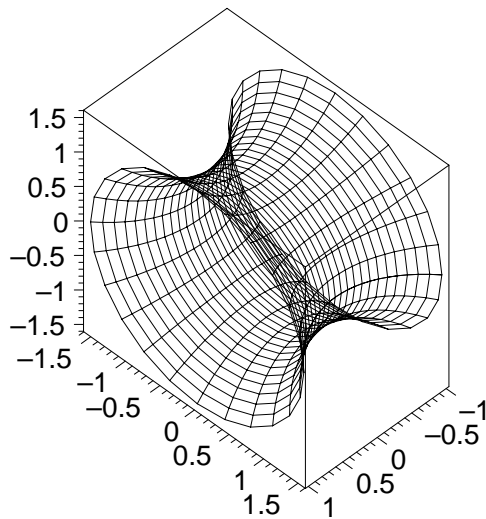


5.7) It's a very short procedure, but I'm hoping you are getting better at writing these! (If you're still pained by Maple, perhaps the "New Users Tour", under the help icon, can help. For example, it has a little blurb about procedures under the subject of programming. The procedure below isn't the only possible solution. In fact, Oprea may just have wanted a procedure to write the parameterization function, like he did in his ruled-surface procedure. Then to get the plot you would compose the procedure with a plot3d command.

```

> surfrev:=proc(f,g,a,b) #rotate about x-axis
    #(f,g) is the profile curve, [a,b] is parameter interval
    plot3d([f(u),g(u)*cos(v),g(u)*sin(v)],u=a..b,v=0..2*Pi,
    axes=boxed, scaling=constrained,color=black, style=wireframe);
end:
> f1:=t->t:
g1:=t->cosh(t): #for catenoid
surfrev(f1,g1,-1,1);

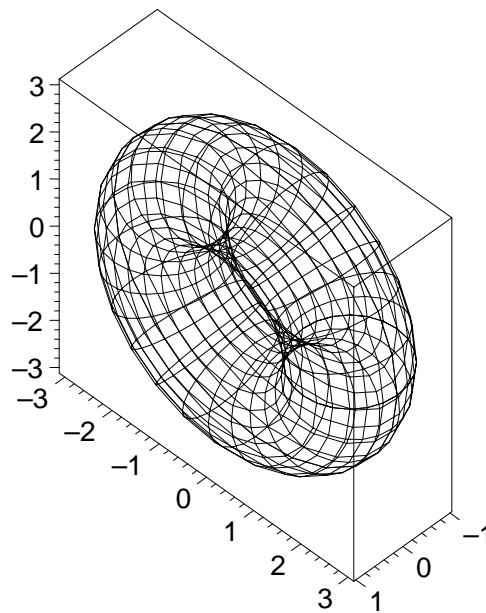
```



```

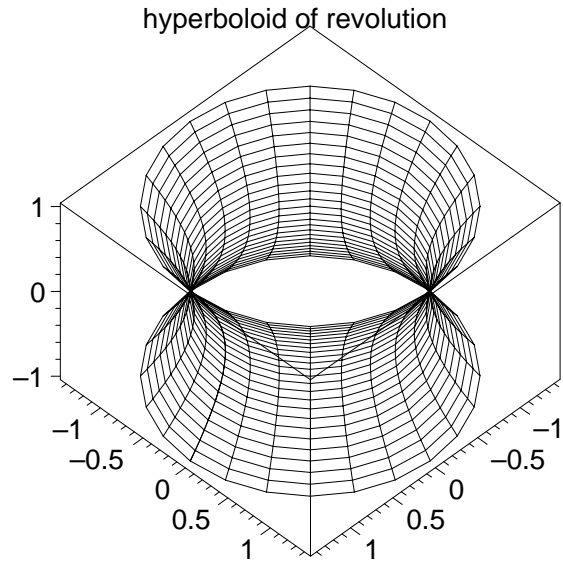
> f2:=t->sin(t);
g2:=t->2+cos(t); #for torus, with center at (0,2), profile curve
#of radius 1.
surfrev(f2,g2,0,2*Pi); #I rotate picture with mouse
      f2 := sin
      g2 := t → 2 + cos(t)

```



5.8) I won't bother with the procedure for ruled surfaces, it doesn't seem worth the trouble. We first draw the hyperboloid as a surface of revolution, but I won't use the procedure above because we want to rotate about the z-axis. I could write a separate procedure to rotate about the z-axis if I wanted to do a lot of these. Note that for fixed z the circle of revolution has radius = $\sqrt{1+z^2}$. We could also have used the cosh-sinh parameterization suggested by the text.

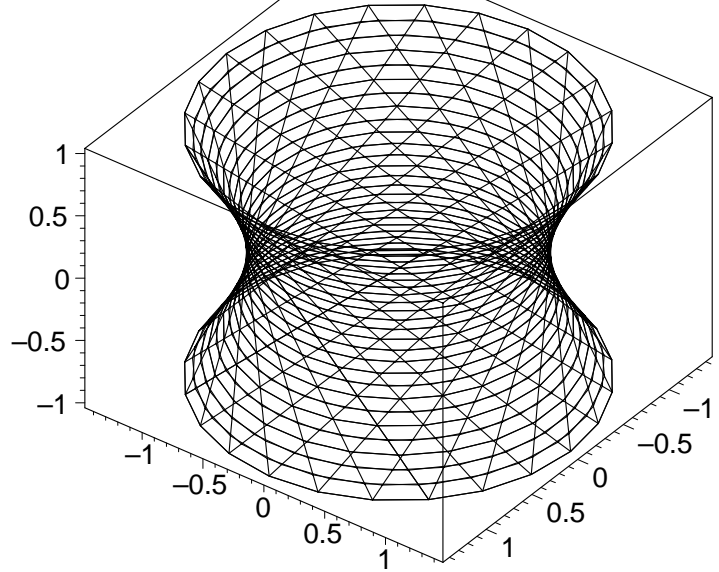
```
> plot3d([sqrt(1+z^2)*cos(v),sqrt(1+z^2)*sin(v),z],z=-1..1,
v=0..2*Pi, scaling=constrained, axes=boxed, style=wireframe,
color=black,title='hyperboloid of revolution');
```



I'll use the parameterizations suggested in problem 2.1.22:

```
> Hyp:=[a*(cos(u)-v*sin(u)),b*(sin(u)+v*cos(u)),c*v];
      Hyp:= [cos(u)-v sin(u), sin(u)+v cos(u), c v]
> Hyp1:=subs({a=1,b=1,c=1},Hyp);
Hyp2:=subs({a=1,b=1,c=-1},Hyp);
      Hyp1:= [cos(u)-v sin(u), sin(u)+v cos(u), v]
      Hyp2:= [cos(u)-v sin(u), sin(u)+v cos(u), -v]
> plot3d({Hyp1,Hyp2},u=0..2*Pi,v=-1..1, axes=boxed,color=black,
style=wireframe,scaling=constrained, title=
'doubly-ruled hyperboloid');
```

doubly-ruled hyperboloid



[>