Grafische Möglichkeiten in Maple - Lösung der Übung 3

(Praktische Arbeit von Romy Krämer, WS 2000)

1. Man zeichne den Graph der Funktion f$(x) = \frac{e^{(-2\,x)}\,(1+x)}{2}$ sowie Tangente und Normale im Punkt x=-1 in ein Bild und beschrifte dieses.

Definition der Funktion

```
> restart;
> with(plots):
```

```
Warning, the name changecoords has been redefined
```

$f := x \rightarrow \frac{e^{(-2\,x)}\,(1+x)}{2}$

$$f := x \rightarrow \frac{1}{2}\,e^{(-2\,x)}\,(1+x)$$

```
> text1:=textplot([-1.2,f(-1.2),`f(x)`],
> color=red,align={ABOVE,LEFT}):
> df:=D(f);evalf(df(-1));
```

$$df := x \rightarrow -e^{(-2\,x)}\,(1+x) + \frac{1}{2}\,e^{(-2\,x)}$$

$$3.694528050$$

Ermittlung der Tangenten und Normalengleichungen

$x0 := 1\,;\; y0 := \mathrm{f}(x0)\,;\; m := \mathrm{df}(x0)$

$t := x \rightarrow y0 + m\,(x - x0)$

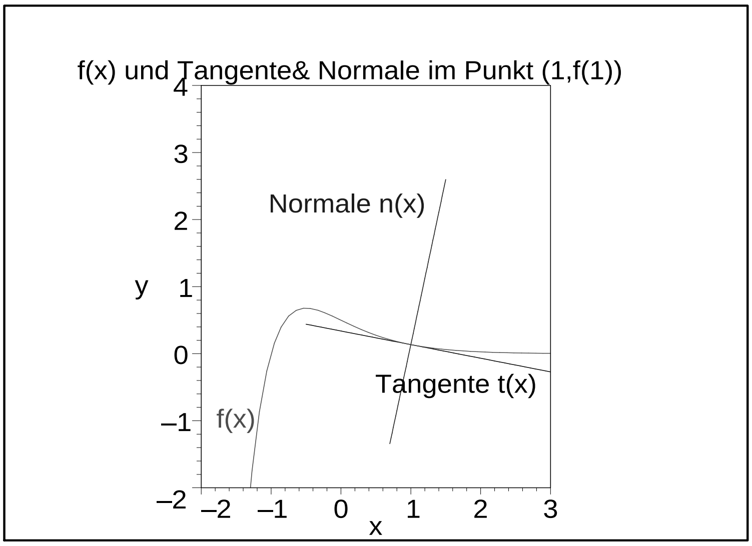
$$t := x \rightarrow y0 + m\,(x - x0)$$

$n := x \rightarrow y0 - \frac{1\,(x-x0)}{m}$

$$n := x \rightarrow y0 - \frac{x - x0}{m}$$

```
> graph1:=plot(f(x), x=-2..3, y=-2..4,color=red,title=`f(x) und
> Tangente& Normale im Punkt (1,f(1))`):

> graph2:=plot(t(x), x=-0.5..3, color=black):
> text2:=textplot([3,t(3),`Tangente t(x)`], color=black,
> align={BELOW,LEFT}):
```

```
> graph3:=plot(n(x), x=0.7..1.5, color=blue):
> text3:=textplot([1.4,n(1.4),`Normale
> n(x)`],color=blue,align={ABOVE,LEFT}):
> display([seq(text||k,k=1..3),seq(graph||k,k=1..3)],axes=boxed,scaling
> =constrained); k:='k':  l:='l':
```



2. Man veranschauliche für obige Funktion den Grenzübergang vom Differenzenquotienten zur Ableitung, indem man eine Reihe von Geraden durch einen Punkt (-1,f'(-1)) und einen variablen Punkt rechts davon legt, Sekanten und Tangenten zeichnet und daraus eine Animation herstellt.

Definition von f, Ermittlung der Ableitung df und der Tangente im Punkt (-1,0):

```
> restart;f:=x->exp(-2*x)*(1+x)/2; x0:=-1:
> df:=D(f);y0:=f(x0):with(plots,animate):t:=y0+df(-1)*(x+1);
```

$$f := x \to \frac{1}{2}\, e^{(-2\,x)}\,(1+x)$$

$$df := x \to -e^{(-2\,x)}\,(1+x) + \frac{1}{2}\, e^{(-2\,x)}$$

$$t := \frac{1}{2}\, e^{2}\,(x+1)$$

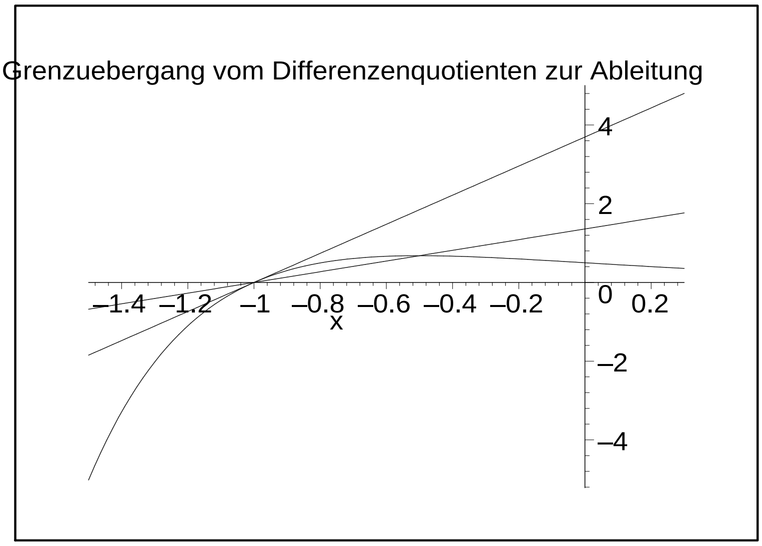
Ermittlung der Sekanten sek, welche durch die Punkte (x0,f(x0)) und (xn,f(xn)) mit xn=x0+2ˆ(-n) geht :

```
> sek:=(x,n)->y0+(f(x0+2^(-n))-y0)/(x0+2^(-n)-x0)*(x-x0);
```

$$sek := (x, n) \to y0 + \frac{(\mathrm{f}(x0 + 2^{(-n)}) - y0)\,(x - x0)}{2^{(-n)}}$$

Animation (mit Graph von f(x), Tangente im Punkt (-1,0) und den Sekanten):

```
> animate({f(x),t,sek(x,n)}, x=-1.5..0.3,
> n=1..20,frames=20,numpoints=100,
> color=blue,title=‘Grenzuebergang vom Differenzenquotienten zur
> Ableitung‘);
```



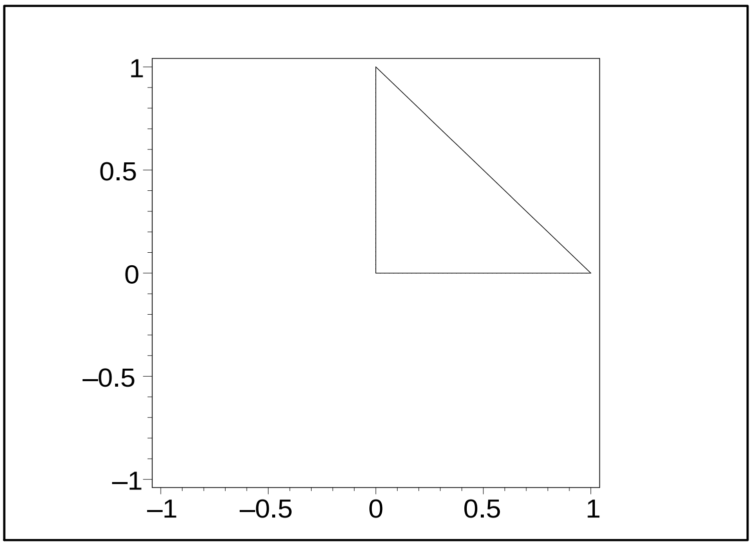
3. Man zeiche ein Dreieck in der Ebene mit einer Ecke im Ursprung. Man lasse dieses Dreieck in einer Animation um den Ursprung rotieren

Erste Variante: Die Seitenlinien werden als Funktion eines Parameters t (laeuft von 0 bis 1) und des Winkels alpha (Drehwinkel, laeuft von 0 bis 2*Pi) dargestellt. Dann zeigt die Animation die Drehung des Dreiecks um (0,0).

```
> restart;
> with(plots):
```

```
Warning, the name changecoords has been redefined
```

```
> animate( {[t*cos(alpha),t*sin(alpha),t=0..1],
> [t*cos(alpha+Pi/2),t*sin(alpha+Pi/2),t=0..1],
> [t*cos(alpha)+(1-t)*cos(alpha+Pi/2),t*sin(alpha)+(1-t)*sin(alpha+Pi/2)
> ,t=0..1]},
> alpha=0..2*Pi,frames=100, scaling=constrained,
> color=blue,axes=boxed);
```



Zweite Variante:

Das Dreieck wird mit Hilfe von polygon gezeichnet. Dann wird das Dreieck durch den Befehl rotate gedreht und diese Graphiken werden in der Liste data gespeichert und schliesslich durch display angezeigt.

```
> restart;with(plots):with(plottools):
> l:=polygon([[0,0], [3,4], [3,1]], color=red):
> data:=[]:
> for i from 0 to 20 do
> phi:=2*Pi/20*i:
> data:=[op(data),rotate(l,phi) ]:
> od:
> display(data,insequence=true, scaling=constrained);
```

4. Man zeichne die folgenden zwei- oder dreidimensionalen Bilder folgender Objekte und beschrifte diese

a) den Rotationskörpers, der entsteht, wenn die von

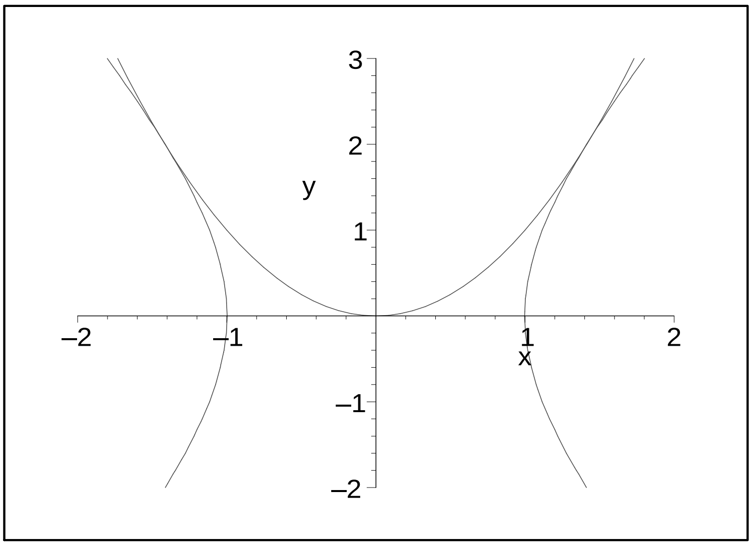
```
> g1:=4*x^2-y^2=4;g2:=y=x^2;
```

$$g1 := 4\,x^2 - y^2 = 4$$
$$g2 := y = x^2$$

und der x-Achse begrenzte Fläche um die x-Achse rotiert.

Veranschaulichung der zwei gegebenen Graphen, Berechnung der Schnittpunkte und zeichnen der zu rotierenden Flaeche:

```
> graph1:=plot(x^2,x=-2..2,y=-2..3):
> graph2:=implicitplot(4*x^2-y^2=4,x=-2..2,y=-2..3):
> display({graph1,graph2});
```

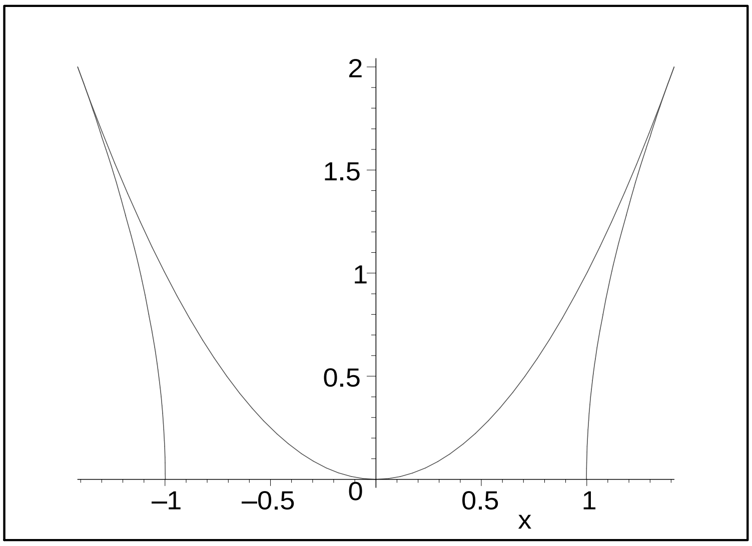



```
> solve(subs(g2,g1),x);
```

$$\sqrt{2},\ -\sqrt{2},\ \sqrt{2},\ -\sqrt{2}$$

Als Schnittpunkte der beiden Funktionen ergeben sich also (sqrt(2),2) und (-sqrt(2),2). Die zu rotierende Fläche sieht so aus:

```
> graph1:=plot(x^2,x=-2^0.5..2^0.5):
> graph2:=implicitplot(g1, x=-sqrt(2)..sqrt(2), y=0..2):
> display({graph1,graph2});
```



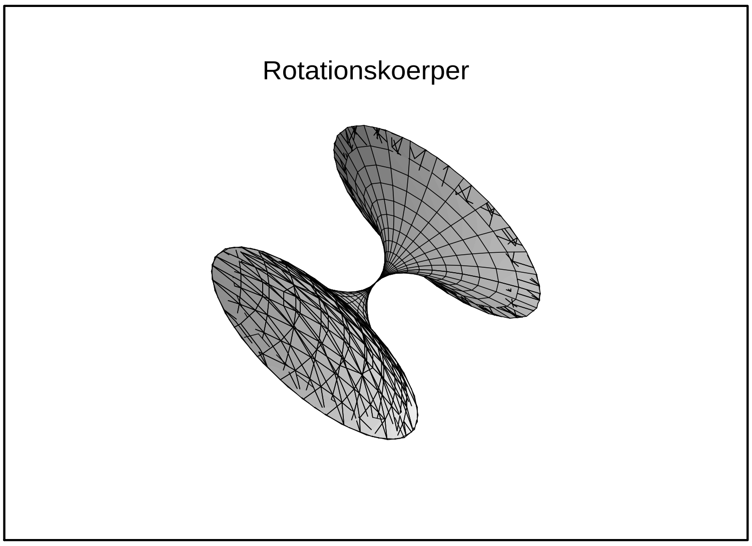

```
> g:=x->x^2;
> rr:=(x,t)->[x,g(x)*cos(t),g(x)*sin(t)];
> graph1:=plot3d(rr(x,t),x=-sqrt(2)..sqrt(2),t=0..2*Pi):
```

Da die Gleichung g1 in impliziter Funktion gegeben war, zeichnen wir hier auch die Rotation um die x-Achse mit einer impliziten Funktion. Es ergibt sich die Gleichung 4*x^2-r^2=4. Dabei ist r der Radius des Kreises, der als Schnittfigur zwischen dem Rotationskoerper und der zur x-z Ebene parallelen Ebene, welche den Punkt (x,0,0) enthaelt. Es gilt also r^2=y^2 + z^2. Setzen dies in die erste Gleichung ein und erhalten 4*x^2-x^2-y^2=4. Dann kann die Flaeche mit implicitplot3d gezeichnet werden.

```
> graph2:=implicitplot3d(4*x^2-y^2-z^2=4, x=-sqrt(2)..sqrt(2),
> y=-2..2,z=-2..2, scaling=constrained):
> display({graph1,graph2},scaling=constrained,
> orientation=[40,30],title='Rotationskoerper');
```

$$g := x \to x^2$$

$$rr := (x,\, t) \to [x,\, \mathrm{g}(x)\cos(t),\, \mathrm{g}(x)\sin(t)]$$



b) die Fläche des Blattes das durch die Kurve

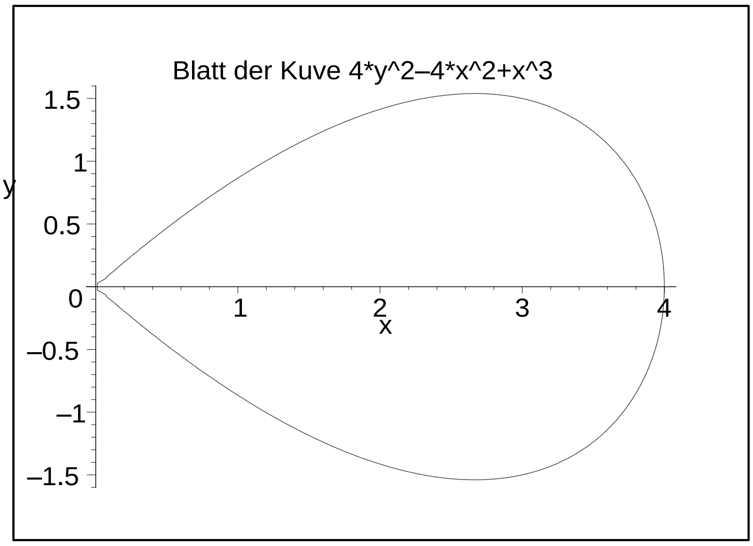
```
> restart; with(plots):gl:=4*y^2-4*x^2+x^3=0;
```

```
Warning, the name changecoords has been redefined
```

$$gl := 4\,y^2 - 4\,x^2 + x^3 = 0$$

gebildet wird in kartesischen und in Polarkoordinaten. (Anmerkung: durch numpoints wird bei den kartesischen Koordinaten die Anzahl der berechneten Punkte erhöht, so dass eine einigermassen schöne Graphik entsteht)

```
> implicitplot(gl, x=0..5, y=-2..2,numpoints=5000,title=`Blatt
der Kuve
> 4*y^2-4*x^2+x^3`); #kartesische Koordinaten
```




```
>  implicitplot3d(-4*x^2+x^3+4*(y^2+z^2),
>  x=0..5,y=-2..3,z=-2..3,numpoints=500,title=‘Rotation des Blattes
g1 um
>  die x-Achse‘);
>  gl_p:=subs(x=r*cos(phi), y=r*sin(phi), gl); #Polarkoordinaten
>  [solve(gl_p,r)]:R:=op(select(has,%,phi));
>  plot([R,phi,phi=-Pi/4..Pi/4], coords=polar,title=‘Blatt der Kurve
>  4*y^2-4*x^2+x^3‘);
>  polarplot([R,phi, phi=-7*Pi/24..7*Pi/24],
>  scaling=constrained,title=‘Blatt der Kurve 4*y^2-4*x^2+x^3‘);
```

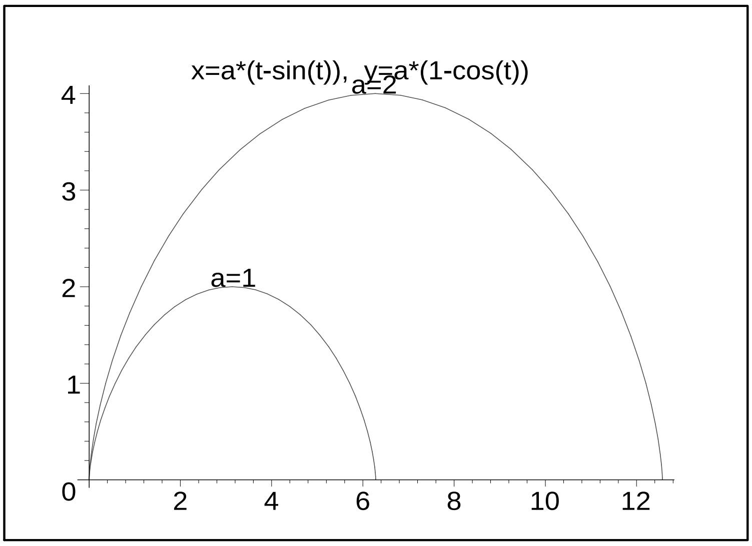
c) einen Bogen der Zykloide

```
>  restart;with(plots):x:=a*(t-sin(t));
>  y:=a*(1-cos(t));graph1:=plot([subs(a=1,x),y ,t=0..2*Pi]):
>  x1:=subs({a=1,t=Pi},x):y1:=subs({a=1,t=Pi},y):
>  graph1:=plot([subs(a=1,x),subs(a=1,y) ,t=0..2*Pi]):
>  text1:=textplot([x1,y1,‘a=1‘],align=ABOVE):
>  graph2:=plot([subs(a=2,x),subs(a=2,y) ,t=0..2*Pi]):
>  x2:=subs({a=2,t=Pi},x):y2:=subs({a=2,t=Pi},y):
>  text2:=textplot([x2,y2,‘a=2‘],align=ABOVE):
>  display({graph1,graph2,text1,text2}, title=‘x=a*(t-sin(t)),
>  y=a*(1-cos(t))‘);
```

```
Warning, the name changecoords has been redefined
```

$$x := a\,(t - \sin(t))$$

$$y := a\,(1 - \cos(t))$$



und den Rotationskörper, der bei Rotation um die x-Achse entsteht.

Der Rotationskoerper wird wieder auf die gleiche Art und Weise wie oben gezeichnet.

```
>  x1:=s->subs({a=1,t=s},x);x1(t);y1:=s->subs({a=1,t=s},y);

>  rr:=(s,t)->[x1(t),y1(t)*cos(s),y1(t)*sin(s)];
```

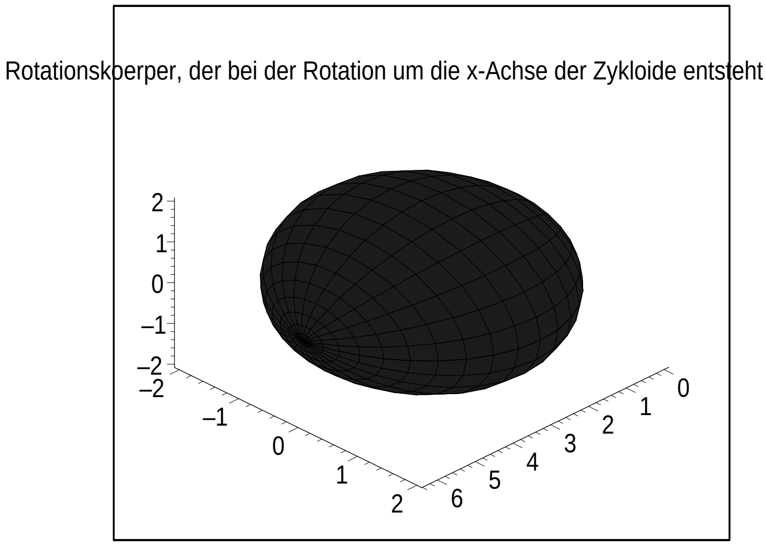
$$x1 := s \to \mathrm{subs}(\{a = 1,\, t = s\},\, x)$$

$$t - \sin(t)$$

$$y1 := s \to \mathrm{subs}(\{a = 1,\, t = s\},\, y)$$

$$rr := (s,\, t) \to [\mathrm{x1}(t),\, \mathrm{y1}(t)\cos(s),\, \mathrm{y1}(t)\sin(s)]$$

```
>  plot3d(rr(s,t),s=0..2*Pi,t=0..2*Pi,color=blue,axes=framed,
>  title=`Rotationskoerper, der bei der Rotation um die x-Achse
der
>  Zykloide entsteht`);
```

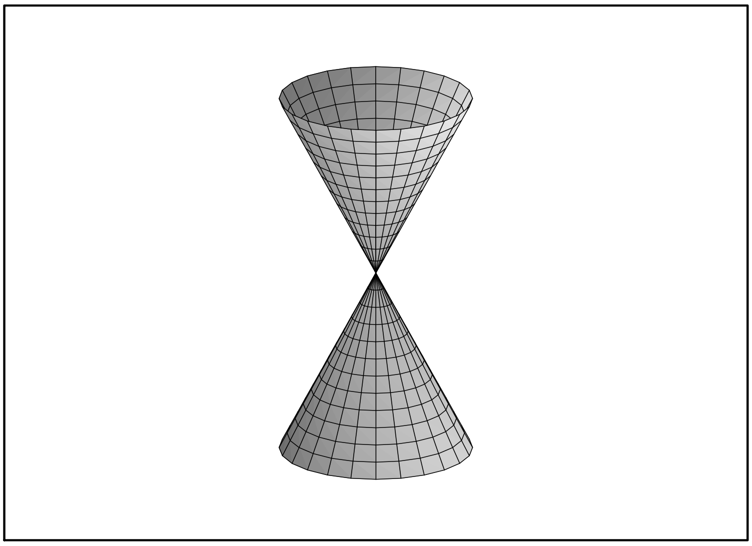


5. Man zeichne einen Doppelkegel in verschiedenen Koordinatensystemen sowie die Schnittfigur des Kegels mit einer Ebene.

zuerst in Zylinderkoordinaten:

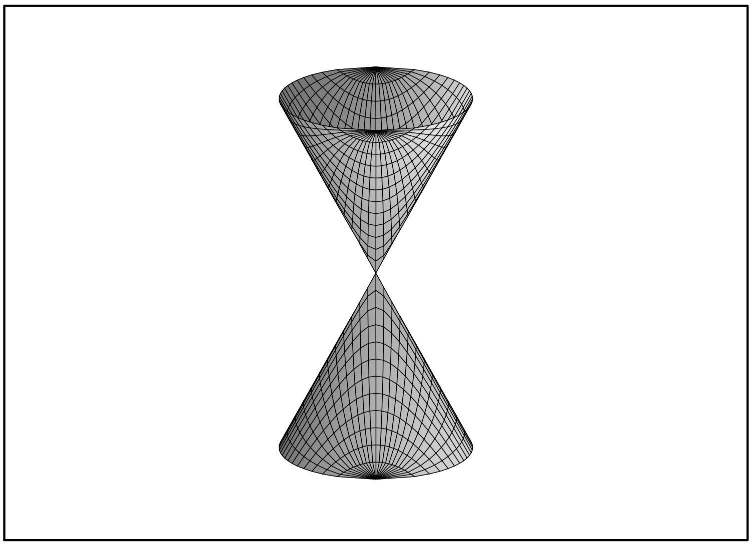
```
> restart; with(plots):plot3d(0.5*z,
> theta=0..2*Pi,z=-3..3,coords=cylindrical,scaling=constrained,orientati
> on=[0,70]);
```

```
Warning, the name changecoords has been redefined
```



nun in kartesischen Koordinaten

```
> plot3d({2*sqrt(x^2+y^2),-2*sqrt(x^2+y^2)},x=-3..3,
> y=-sqrt(9-x^2)..sqrt(9-x^2),orientation=[0,70],scaling=constrained);
```



Iin Kugelkoordinaten kann kein solcher Doppelkegel dargestellt werden.

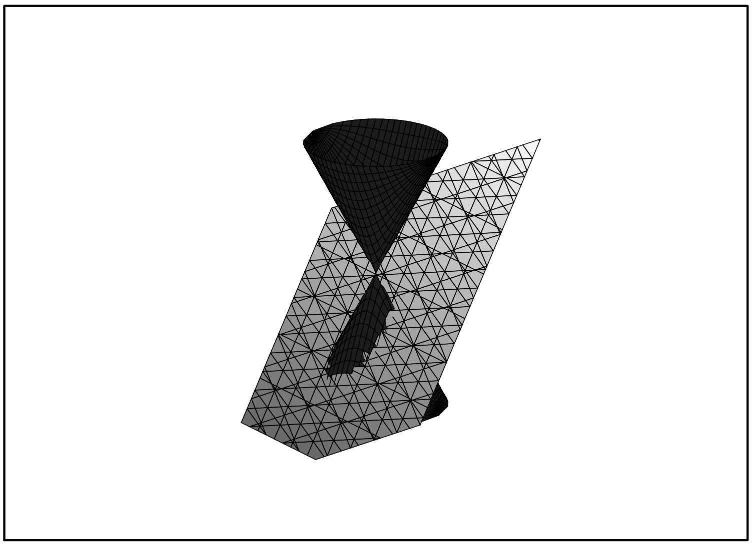
Jetzt soll die Schnittfigur des Kegels mit einer Ebene gezeichnet werden.
Ebenengleichung: 0.5*x+y-z=2

```
> restart;
> with(plots):ebene:=1/2*x+y-z=2;x:='x':graph1:=plot3d(
> {2*sqrt(x^2+y^2),-2*sqrt(x^2+y^2)},x=-3..3,
> y=-sqrt(9-x^2)..sqrt(9-x^2),orientation=[0,70],scaling=constrained,col
> or=blue):
> graph2:=implicitplot3d(ebene,x=-5..5,y=-5..5,z=-7..7):
> display(graph1,graph2,orientation=[-60,70],scaling=constrained);
```

Die Graphik zeigt, dass die Schnittfigur eine Ellipse ist.

```
Warning, the name changecoords has been redefined
```

$$ebene := \frac{x}{2} + y - z = 2$$



```
> gl1:=z=2*sqrt(x^2+y^2);gl2:=z=-2*sqrt(x^2+y^2);
```

$$gl1 := z = 2\sqrt{x^2+y^2}$$

$$gl2 := z = -2\sqrt{x^2+y^2}$$

Aus obiger Graphik ist zu erkennen, dass nur Schnittpunkte zwischen gl2 und ebene gesucht werden muessen.

```
> ebene;gl3_x:=x=solve(ebene,x);
```

$$\frac{x}{2} + y - z = 2$$

$$gl3\_x := x = -2\,y + 2\,z + 4$$

```
> subs(gl3_x,gl2);solve(%,z);z1:=unapply(%[1],y);z2:=unapply(%
> %[2],y);
```

$$z = -2\sqrt{(-2\,y + 2\,z + 4)^2 + y^2}$$

$$-\frac{32}{15} + \frac{16\,y}{15} + \frac{2\sqrt{16 - 16\,y - 11\,y^2}}{15},\ -\frac{32}{15} + \frac{16\,y}{15} - \frac{2\sqrt{16 - 16\,y - 11\,y^2}}{15}$$

$$z1 := y \to -\frac{32}{15} + \frac{16}{15}\,y + \frac{2}{15}\sqrt{16 - 16\,y - 11\,y^2}$$

$$z2 := y \to -\frac{32}{15} + \frac{16}{15}\,y - \frac{2}{15}\sqrt{16 - 16\,y - 11\,y^2}$$

Wenn wir jetzt z=z1(y) bzw. z=z2(y) in gl3_x einsetzen, erhalten wir x in Abhängigkeit von y:

```
> x1:=unapply(subs(z=z1(y),rhs(gl3_x)),y);
```

$$x1 := y \to \frac{2}{15}\,y - \frac{4}{15} + \frac{4}{15}\sqrt{16 - 16\,y - 11\,y^2}$$

```
> x2:=unapply(subs(z=z2(y),rhs(gl3_x)),y);
```

$$x2 := y \to \frac{2}{15}\,y - \frac{4}{15} - \frac{4}{15}\sqrt{16 - 16\,y - 11\,y^2}$$

```
> graph1:=spacecurve([x1(t),t,z1(t)],t=-3..1,numpoints=800):
> graph2:=spacecurve([x2(t),t,z2(t)],t=-3..1,numpoints=800):
> display({graph1,graph2
> },orientation=[-60,70],scaling=constrained,color=blue);
```

Selbst durch Ehoehung von numpoints ist es mir nicht gelungen, die Luecken in der Graphik zu beseitigen.

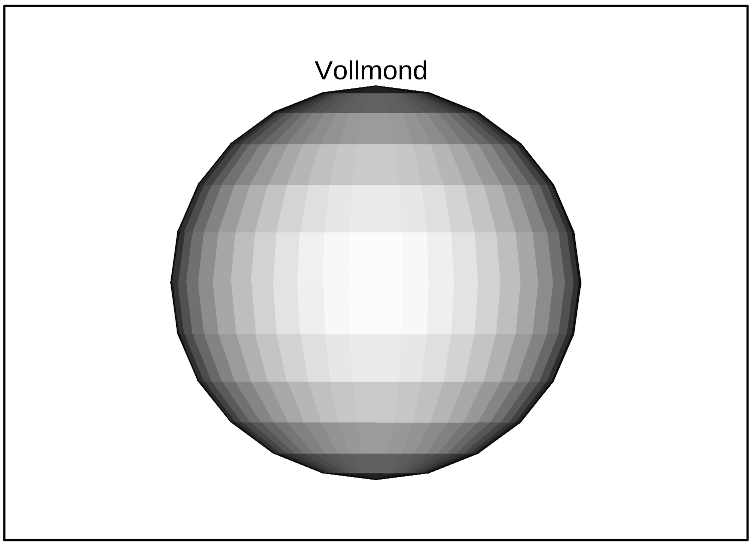
6. Man zeichne eine Kugel und versuche durch Färbung und Beleuchtung die Mondphasen zu simulieren.

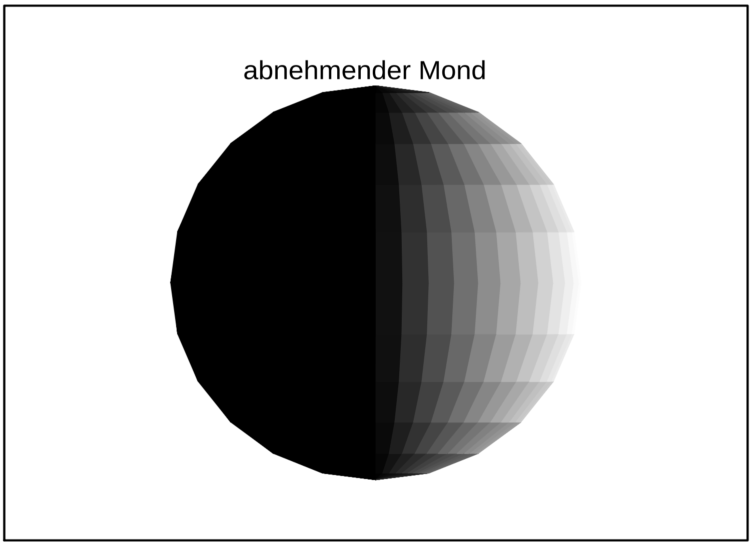
Zuerst wird eine weisse Kugel gezeichnet und diese Graphik in kugel gespeichert:

```
> restart; with(plots):kugel:=plot3d([sin(s)*cos(t),
> cos(s)*cos(t),
> sin(t)],
> s=0..Pi, t=0..2*Pi,
> scaling=constrained,color=white ):

Warning, the name changecoords has been redefined

> ori:=orientation=[0,90]:
```

Nun schauen wir uns die Graphik mit einer Beleuchtung (von vorn, rechts, hinten, links) an.

```
> display(kugel,ori,light=[90,0,1,1,1],title=‘Vollmond‘,style=patchnogr
> id);
> display(kugel,ori,light=[90,90,1,1,1],title=‘abnehmender
> Mond‘,style=patchnogrid);
> display(kugel,ori,light=[90,180,1,1,1],title=‘Neumond‘,style=patchnog
> rid);
> display(kugel,ori,light=[90,270,1,1,1],title=‘zunehmender
> Mond‘,style=patchnogrid);
```





Der Versuch einer Animation gelingt nicht::

```
>  s:=k->plot3d([sin(s)*cos(t),
>  cos(s)*cos(t),
>  sin(t)],
>  s=0..Pi, t=0..2*Pi, orientation=[0,90],style=patchnogrid,
>  scaling=constrained,color=white, light=[90,k*360/12,1,1,1 ]):
```

```
> display3d([seq(s(k),k=0..12)],insequence =true);
> s:=k->plot3d([sin(s)*cos(t),
> cos(s)*cos(t),
> sin(t)],
> s=0..Pi, t=0..2*Pi, orientation=[0,90],style=patchnogrid,
> scaling=constrained,color=white, light=[90,k*360/12,1,1,1 ]):
```