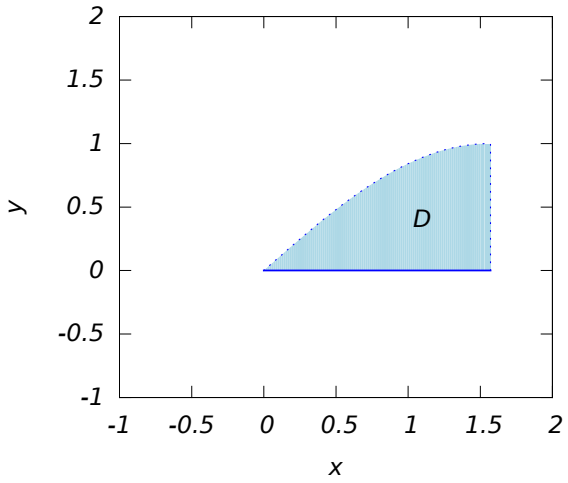
Määramispiirkonna kujutamine

Olgu $D = \left\{(x,y)\colon 0 \leqslant x < \frac{\pi}{2}, 0 \leqslant y < \sin x\right\}$.

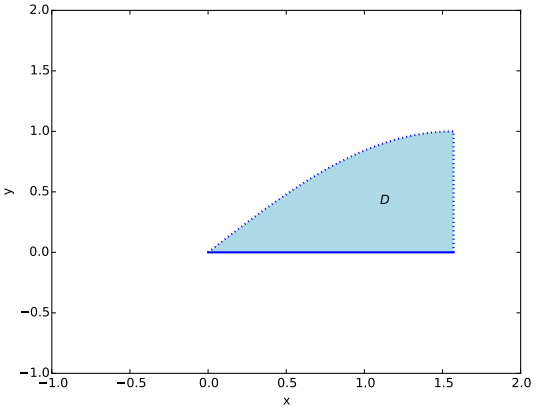
Hulga D kujutamine Maxima abil.

```
load("draw");
draw2d(
  font="Times-Italic",
  xlabel="x", ylabel="y",
  fill_color=light-blue,
  filled_func=0,
  explicit(sin(x), x, 0, %pi/2),
  filled_func=false,
  line_width=2,
  line_type=dots,
  explicit(sin(x), x, 0, %pi/2),
  parametric(%pi/2, y, y, 0, 1),
  line_type=solid,
  parametric(x, 0, x, 0, %pi/2),
  color=black,
  label(["D", 1.1, 0.4]),
  xrange=[-1,2], yrange=[-1,2],
  terminal=pdf,
  file_name="h:\\ma2\\15k\\gr1"
);
```



Hulga D kujutamine Python+Matplotlib abil.

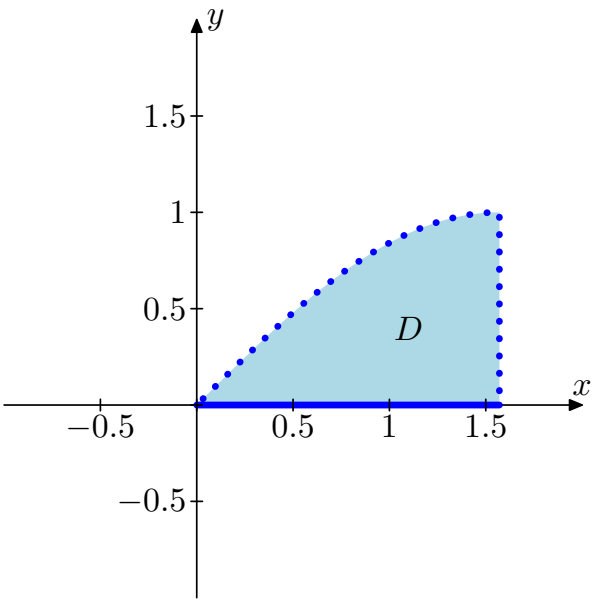
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon
fig, ax = plt.subplots()
ax.set_xlabel('x')
ax.set_ylabel('y')
x = np.linspace(0, np.pi/2, 100)
y = np.sin(x)
poly = Polygon(list(zip(x, y))+[(np.pi/2, 0)], facecolor='#ADD8E6', edgecolor='none')
ax.add_patch(poly)
ax.text(1.1, 0.4, 'D', style='italic')
plt.plot(x, y, 'b', linewidth=2, linestyle=':')
plt.plot(x, np.zeros(len(x)), 'b', linewidth=2, linestyle='-')
y = np.linspace(0, 1, 100)
plt.plot(np.pi/2 * np.ones(len(y)), y, 'b', linewidth=2, linestyle=':')
plt.xlim([-1, 2])
plt.ylim([-1, 2])
plt.savefig("h:\\ma2\\15k\\gr2.pdf")
```



Hulga D kujutamine MetaPosti abil.

```
input latexmp;
beginfig(1);
  numeric u;
  path p;
  u := 2cm;
  drawarrow (-u,0)--(2*u,0);
  drawarrow (0,-u)--(0,2*u);
  p := (0,0);
  for i = 0 upto 100:
```

```
    p := p--(u*i/100*1.5708, u* sind(i/100*90));
  endfor
  fill p--(1.5708*u, 0)--cycle withcolor (173/255, 216/255, 230/255);
  draw p--(1.5708*u, 0) withpen pencircle scaled 2pt withcolor blue dashed withdots;
  draw (0,0)--(1.5708*u, 0) withpen pencircle scaled 2pt withcolor blue;
  label(btex $D$ etex, (1.1*u, 0.4*u));
  label.top(btex $x$ etex, (2*u,0));
  label.rt(btex $y$ etex, (0, 2*u));
  for i = -1 upto 3:
    draw (-0.03*u, i*u/2)--(0.03*u, i*u/2);
        draw (i*u/2, -0.03*u)--(i*u/2, 0.03*u);
        if i <> 0 : label.bot(textext("$"&decimal(i/2)&"$"), (i*u/2, 0)); fi
        if i <> 0 : label.lft(textext("$"&decimal(i/2)&"$"), (0, i*u/2)); fi
  endfor
endfig;
end
```



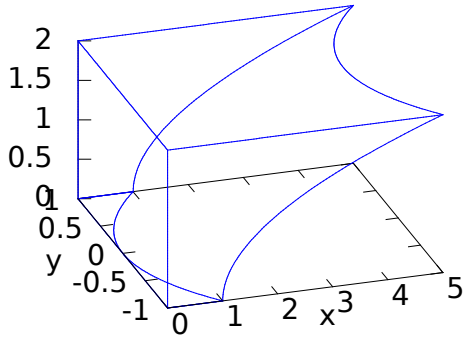
Kolmemõõtmeline keha

Olgu keha E piiratud pindadega $x = y^2 + z^2$, $x = 0$, $y = -1$, $y = 1$, $z = 0$, $z = 2$.

Hulga E kujutamine Maxima abil (sissejuhatus).

```
load("draw");
draw3d(
```

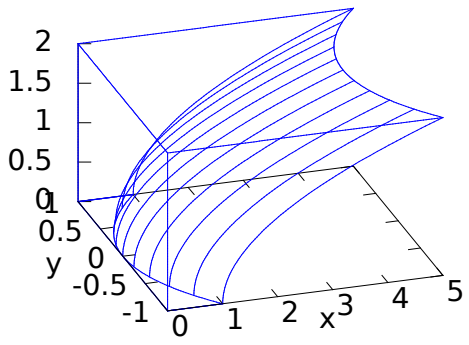
```
  nticks=100,
  parametric(y**2, y, 0, y, -1, 1),
  parametric(y**2+4, y, 2, y, -1, 1),
  parametric(z**2+1, 1, z, z, 0, 2),
  parametric(z**2+1, -1, z, z, 0, 2),
  parametric(x, -1, 0, x, 0, 1),
  parametric(x, 1, 0, x, 0, 1),
  parametric(x, -1, 2, x, 0, 5),
  parametric(x, 1, 2, x, 0, 5),
  parametric(0, y, 2, y, -1, 1),
  parametric(0, y, 0, y, -1, 1),
  parametric(0, -1, z, z, 0, 2),
  parametric(0, 1, z, z, 0, 2),
  xyplane=0, xlabel="x", ylabel="y", zlabel="z",
  view=[54,342],
  terminal=pdf, file_name="h:\\ma2\\15k\\gr4-1"
);
```



Sellisel viisil ei saa hästi aru, millised on kumerad pinnad. Seetõttu peaks eeskätt kumeraid pindu nt. viirutama.

```
load("draw");
n : 10;
loend : makelist(-1 + k/n*2, k, 0, n);
vekt : makelist(parametric(z**2+p**2, p, z, z, 0, 2), p, loend);
apply(draw3d, append(
[
  nticks=100,
  parametric(y**2, y, 0, y, -1, 1)
```

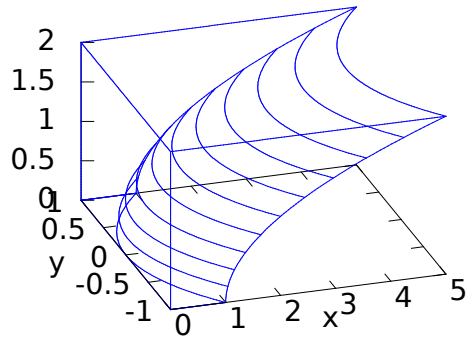
```
], vekt,
[
  parametric(y**2+4, y, 2, y, -1, 1),
  parametric(x, -1, 0, x, 0, 1),
  parametric(x, 1, 0, x, 0, 1),
  parametric(x, -1, 2, x, 0, 5),
  parametric(x, 1, 2, x, 0, 5),
  parametric(0, y, 2, y, -1, 1),
  parametric(0, y, 0, y, -1, 1),
  parametric(0, -1, z, z, 0, 2),
  parametric(0, 1, z, z, 0, 2),
  xyplane=0, xlabel="x", ylabel="y", zlabel="z",
  view=[54,342],
  terminal=pdf, file_name="h:\\ma2\\15k\\gr4-2"
] )
);
```



Teine võimalus viirutamiseks.

```
load("draw");
n : 10;
loend : makelist(k/n*2, k, 0, n);
vekt : makelist(parametric(p**2+y**2, y, p, y, -1, 1), p, loend);
apply(draw3d, append(
[
  nticks=100
], vekt,
[
  parametric(z**2+1, 1, z, z, 0, 2),
```

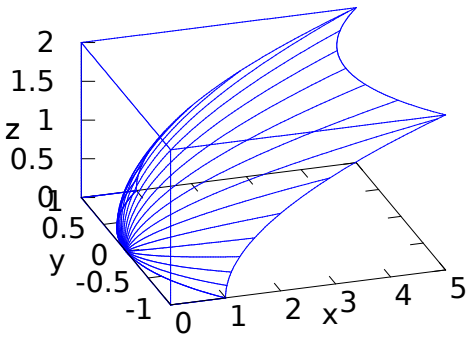
```
  parametric(z**2+1, -1, z, z, 0, 2),
  parametric(x, -1, 0, x, 0, 1),
  parametric(x, 1, 0, x, 0, 1),
  parametric(x, -1, 2, x, 0, 5),
  parametric(x, 1, 2, x, 0, 5),
  parametric(0, y, 2, y, -1, 1),
  parametric(0, y, 0, y, -1, 1),
  parametric(0, -1, z, z, 0, 2),
  parametric(0, 1, z, z, 0, 2),
  xyplane=0, xlabel="x", ylabel="y", zlabel="z",
  view=[54,342],
  terminal=pdf, file_name="h:\\ma2\\15k\\gr4-3"
] )
);
```



Kolmas võimalus viirutamiseks (polaarkoordinaadid).

```
load("draw");
n : 20;
loend : makelist(k/n*(%pi), k, 0, n);
vekt : makelist(parametric(t**2, t*cos(p), t*sin(p), t, 0,
          if 2*abs(cos(p)) > sin(p) then 1/abs(cos(p)) else 2/sin(p)), p, loend);
apply(draw3d, append(
[
  nticks=100,
  parametric(y**2, y, 0, y, -1, 1)
], vekt,
[
  parametric(y**2+4, y, 2, y, -1, 1),
```

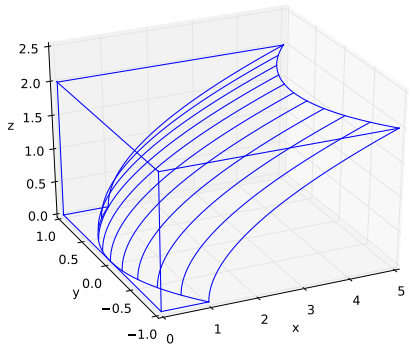
```
  parametric(z**2+1, 1, z, z, 0, 2),
  parametric(z**2+1, -1, z, z, 0, 2),
  parametric(x, -1, 0, x, 0, 1),
  parametric(x, 1, 0, x, 0, 1),
  parametric(x, -1, 2, x, 0, 5),
  parametric(x, 1, 2, x, 0, 5),
  parametric(0, y, 2, y, -1, 1),
  parametric(0, y, 0, y, -1, 1),
  parametric(0, -1, z, z, 0, 2),
  parametric(0, 1, z, z, 0, 2),
  xyplane=0, xlabel="x", ylabel="y", zlabel="z",
  view=[54,342],
  terminal=pdf, file_name="/home/zolki/oppetoo/ma2/15k/graafikud/gr4-4"
] )
);
```



Teeme Pythonis läbi esimest tüüpi viirutuse. (Ülejäänud variandid on võimalik kergesti tuletada eelnevast.)

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.view_init(elev=30, azim=245)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
y = np.linspace(-1, 1.01, num=100)
```

```
plt.plot(y**2, y, 0, 'b')
plt.plot(y**2+4, y, 2, 'b')
plt.plot(np.zeros(len(y)), y, 0, 'b')
plt.plot(np.zeros(len(y)), y, 2, 'b')
n = 10
for k in range(0, n+1):
  p = -1 + k/n*2
  z = np.linspace(0, 2, num=100)
  plt.plot(z**2+p**2, np.ones(len(z))*p, z, 'b')
z = np.linspace(0, 2.01, num=100)
plt.plot(np.zeros(len(z)), -np.ones(len(z)), z, 'b')
plt.plot(np.zeros(len(z)), np.ones(len(z)), z, 'b')
x = np.linspace(0, 1, num=100)
plt.plot(x, -np.ones(len(x)), 0, 'b')
plt.plot(x, np.ones(len(x)), 0, 'b')
x = np.linspace(0, 5, num=100)
plt.plot(x, -np.ones(len(x)), 2, 'b')
plt.plot(x, np.ones(len(x)), 2, 'b')
plt.xlim([0, 5])
plt.ylim([-1, 1])
plt.savefig("gr5.pdf")
```



Silinder sfääri sees

Ülesanne. Koostage arvuti abil kolmemõõtmeline joonis silindri $x^2+y^2=2x$ sellest osast, mis asub sfääri $x^2+y^2+z^2=4$ sees.

Lahendus. Silindri võrrand polaarkoordinaatides on $r^2=2r\cos\varphi$, millest $r=2\cos\varphi$, seega

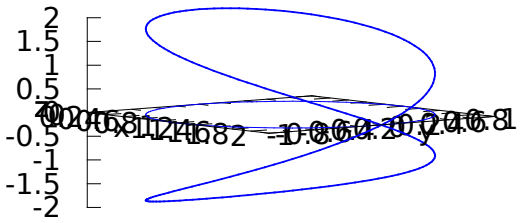
silindri parameetrilised võrrandid xy-tasandil on: $x = 2\cos t \cdot \cos t$, $y = 2\cos t \cdot \sin t$, kus $t = -\frac{\pi}{2} \ldots \frac{\pi}{2}$.

Sfääri ja silindri lõikejoone z-koordinaadi väärtused konkreetse t korral on $z^2 = 4 - (2\cos t)^2 = 4(\sin t)^2$, seega $z = \pm 2\sin t$.

Kujutame joonisel sfääri ja silindri lõikejoone, silindri lõikejoone xy-tasandiga ning viirutame silindri kirjeldatud osa püstiste joontega.

Ilma viirutuseta:

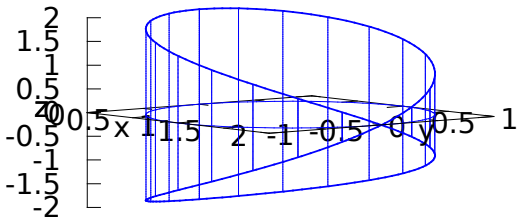
```
load("draw");
draw3d(
  nticks=100,
  line_width=2,
  parametric(2*cos(t)*cos(t), 2*cos(t)*sin(t), 2*sin(t), t, -%pi/2, %pi/2),
  parametric(2*cos(t)*cos(t), 2*cos(t)*sin(t), -2*sin(t), t, -%pi/2, %pi/2),
  line_width=1,
  parametric(2*cos(t)*cos(t), 2*cos(t)*sin(t), 0, t, -%pi/2, %pi/2),
  xyplane=0, xlabel="x", ylabel="y", zlabel="z",
  view=[82,51],
  terminal=pdf, file_name="/home/zolki/oppetoo/ma2/15k/graafikud/gr6-1"
);
```



Koos viirutusega. Peame tegema tsükli, kus loendi muutuja saab väärtused teatud sammuga $-\frac{\pi}{2}$ kuni $\frac{\pi}{2}$.

```
load("draw");
n : 20;
loend : makelist(-%pi/2 + k/n*(%pi), k, 0, n);
```

```
vekt : makelist(parametric(2*cos(p)*cos(p), 2*cos(p)*sin(p), z,
            z, -2*abs(sin(p)), 2*abs(sin(p))), p, loend);
apply(draw3d, append(
[
  nticks=100,
  line_width=2,
  parametric(2*cos(t)*cos(t), 2*cos(t)*sin(t), 2*sin(t), t, -%pi/2, %pi/2),
  parametric(2*cos(t)*cos(t), 2*cos(t)*sin(t), -2*sin(t), t, -%pi/2, %pi/2),
  line_width=1,
  parametric(2*cos(t)*cos(t), 2*cos(t)*sin(t), 0, t, -%pi/2, %pi/2)
], vekt,
[
  xyplane=0, xlabel="x", ylabel="y", zlabel="z",
  view=[82,51],
  terminal=pdf, file_name="/home/zolki/oppetoo/ma2/15k/graafikud/gr6-2"
] )
);
```

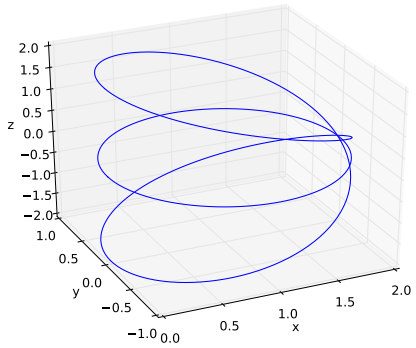


Nüüd kordame sedasama Pythonis.

Ilma viirutuseta:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.view_init(elev=30, azim=245)
ax.set_xlabel('x')
```

```
ax.set_ylabel('y')
ax.set_zlabel('z')
t = np.linspace(-np.pi/2, np.pi/2, num=100)
plt.plot(2*np.cos(t)*np.cos(t), 2*np.cos(t)*np.sin(t), 2*np.sin(t), 'b')
plt.plot(2*np.cos(t)*np.cos(t), 2*np.cos(t)*np.sin(t), -2*np.sin(t), 'b')
plt.plot(2*np.cos(t)*np.cos(t), 2*np.cos(t)*np.sin(t), 0, 'b')
plt.xlim([0, 2])
plt.ylim([-1, 1])
plt.savefig("gr6-3.pdf")
```



Koos viirutusega:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.view_init(elev=20, azim=300)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
t = np.linspace(-np.pi/2, np.pi/2, num=100)
plt.plot(2*np.cos(t)*np.cos(t), 2*np.cos(t)*np.sin(t), 2*np.sin(t), 'b', \
        linewidth=2.0)
plt.plot(2*np.cos(t)*np.cos(t), 2*np.cos(t)*np.sin(t), -2*np.sin(t), 'b', \
        linewidth=2.0)
plt.plot(2*np.cos(t)*np.cos(t), 2*np.cos(t)*np.sin(t), 0, 'b')
n = 20
for k in range(0, n+1):
  p = -np.pi/2 + k/n*np.pi
```

```
  z = np.linspace(-2*abs(np.sin(p)), 2*abs(np.sin(p)), num=3)
  plt.plot(np.ones(len(z))*2*np.cos(p)*np.cos(p), \
           np.ones(len(z))*2*np.cos(p)*np.sin(p), z, 'b')
plt.xlim([0, 2])
plt.ylim([-1, 1])
plt.savefig("gr6-4.pdf")
```