In [ ]:

```
from openscad1 import *
```

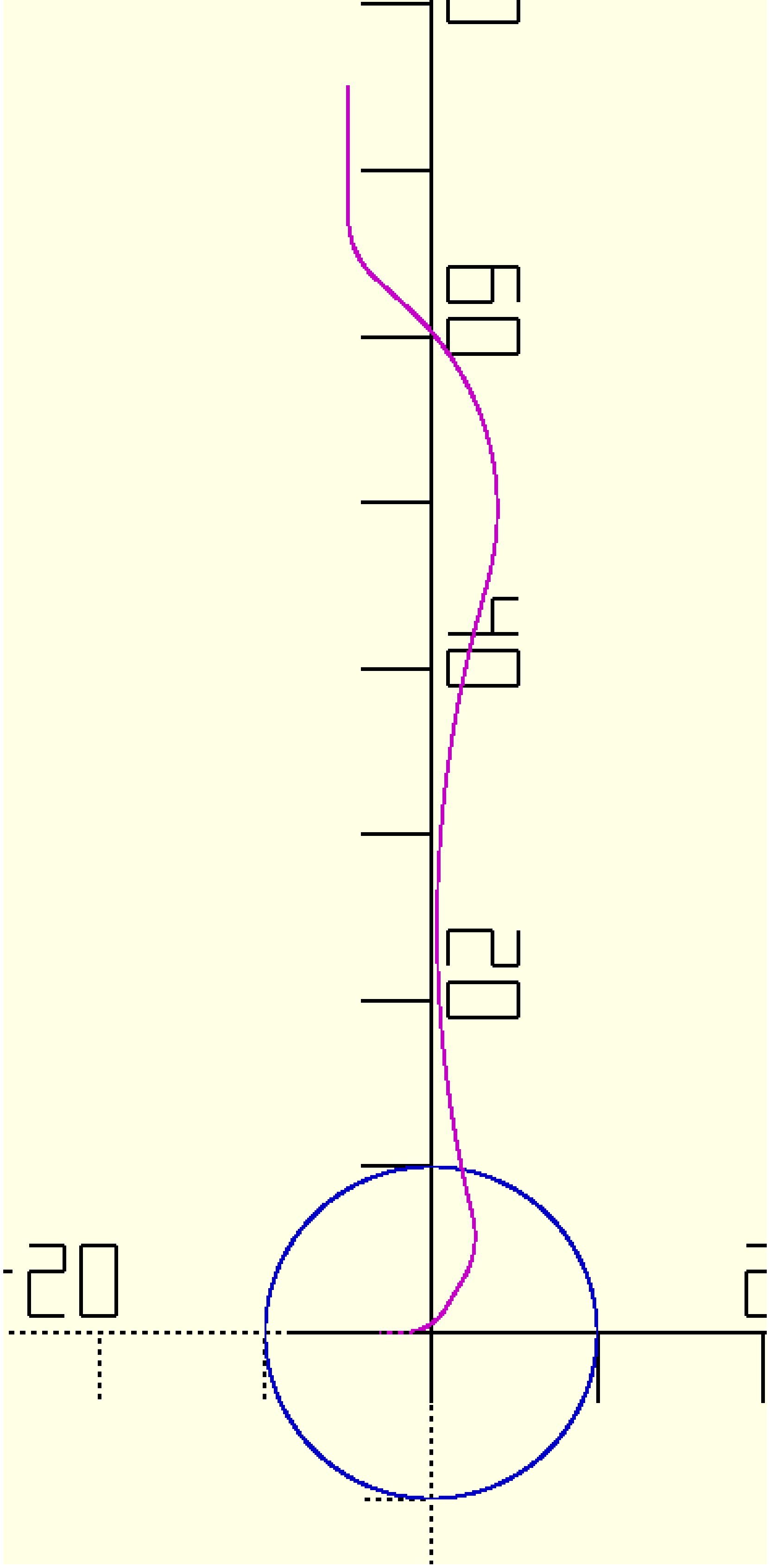
draw section and path to create bottle

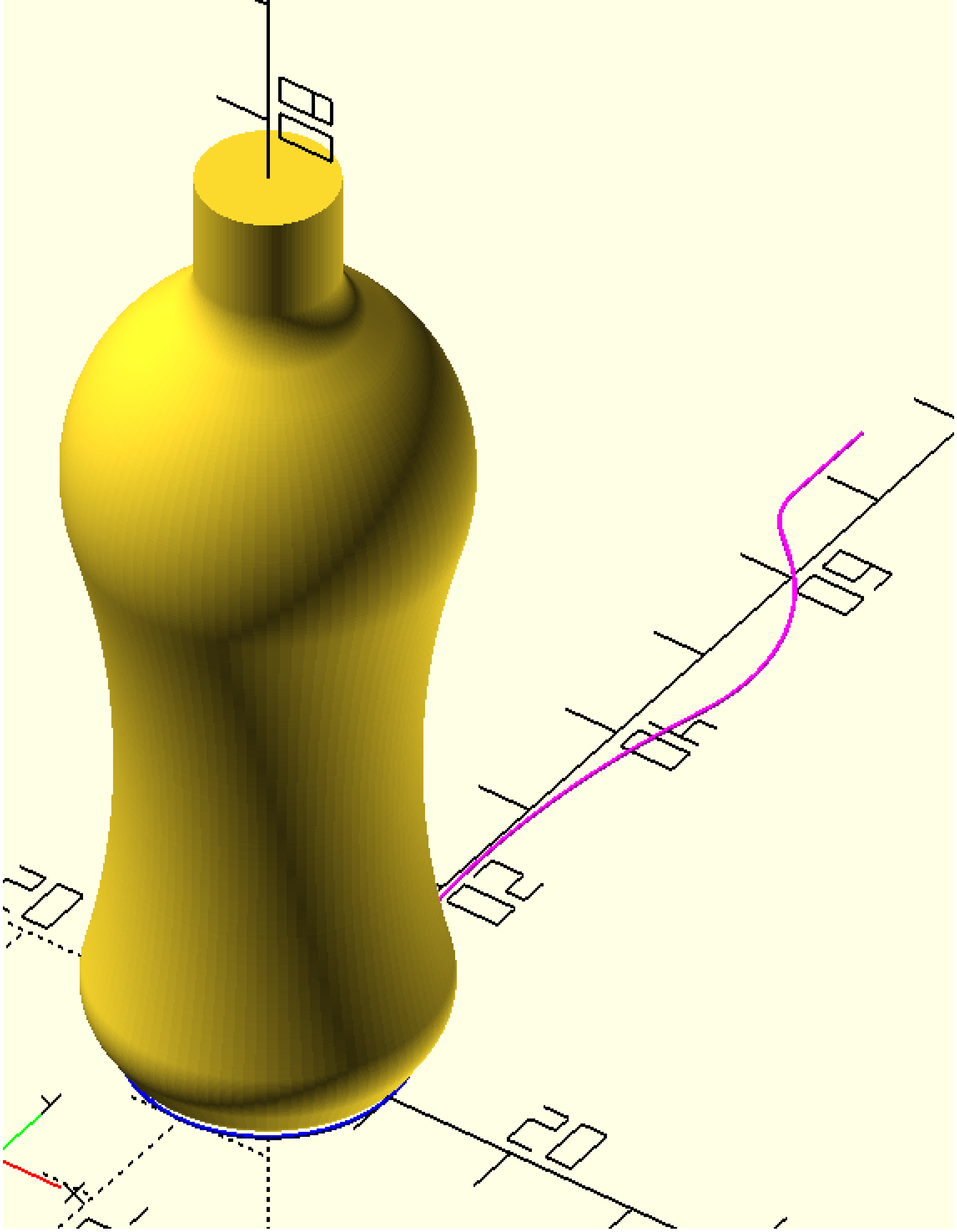
In [48]:

```
t0=time.time()
sec=circle(10,s=100)
path=cr(pts1([[-3,0,0],[3,0,3],[3,5,7],[-5,20,100],[8,30,20],[-11,10,5],[0,10,0]]),50)
path=equidistant_path(path,500)
sol=prism(sec,path)

with open('/users/sanjeevprabhakar/openscad/trial.scad','w+')as f:
    f.write(f'''
include<dependencies2.scad>
color("blue") p_line3dc({sec},.2,1);
color("magenta") p_line3d({path},.2,1);

{swp(sol)}

    ''')
t1=time.time()
t1-t0
```

Out[48]:

```
4.188665151596069
```


-20
2
20
40
60

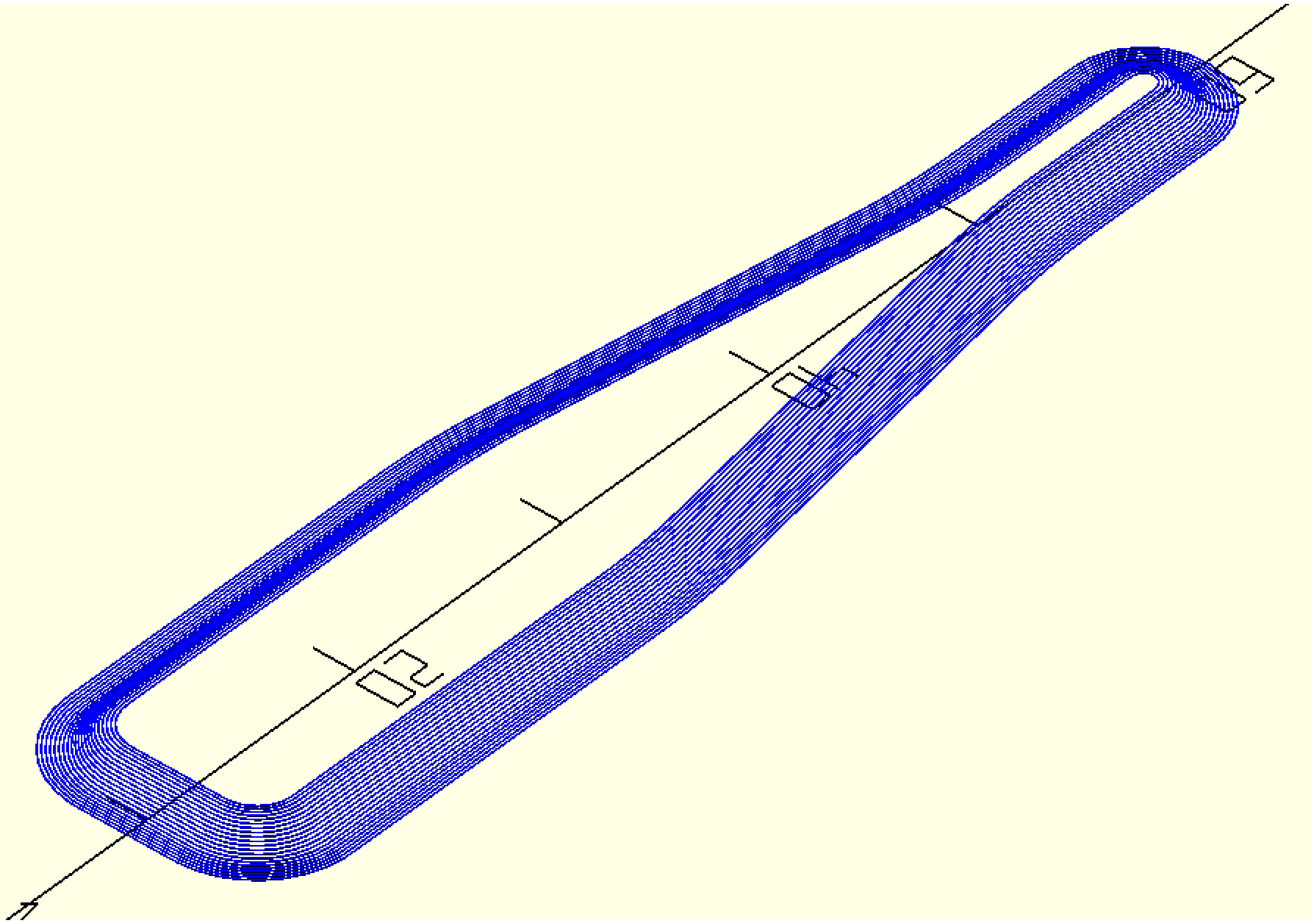


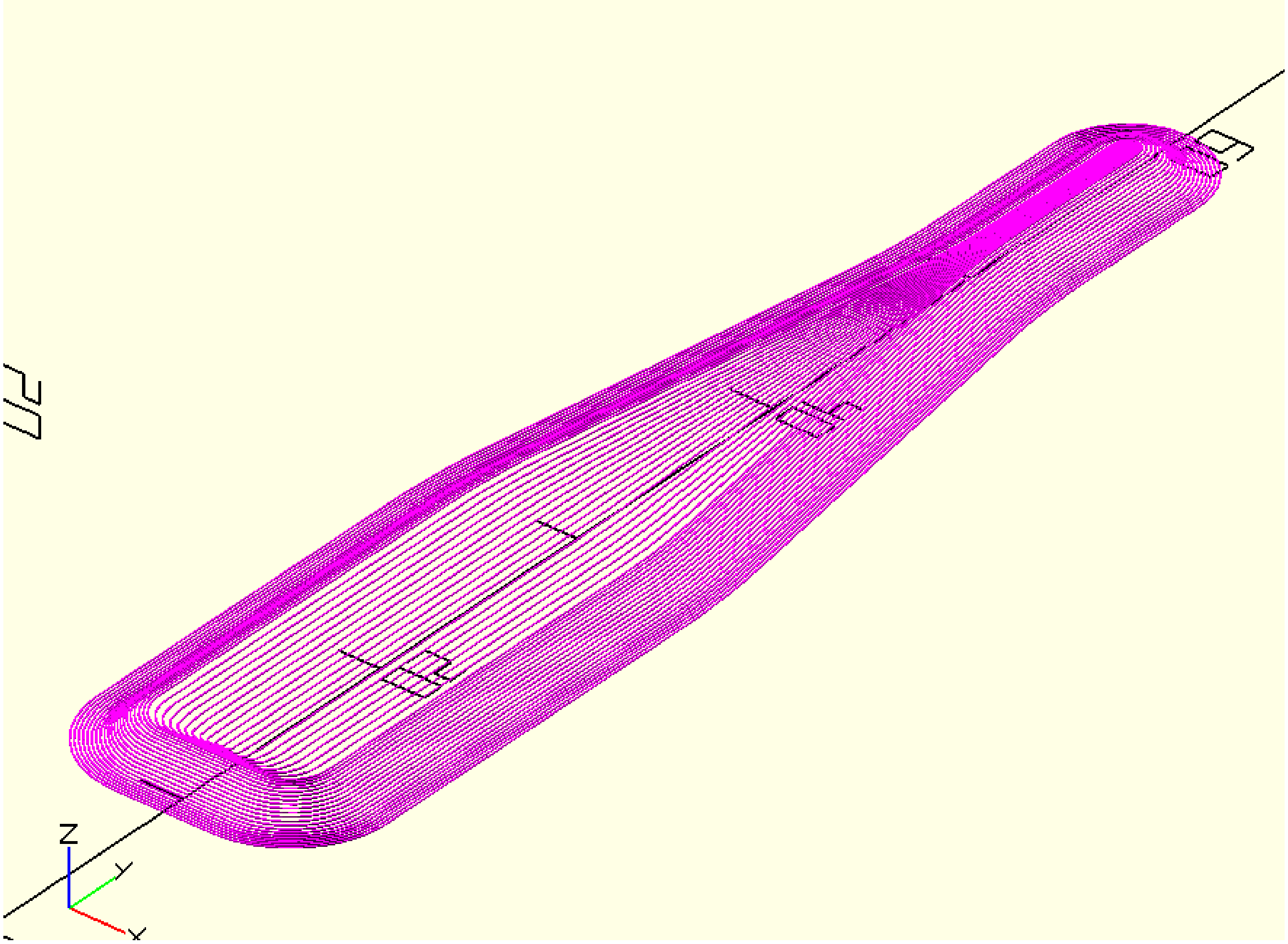
Draw another solid for cut-out

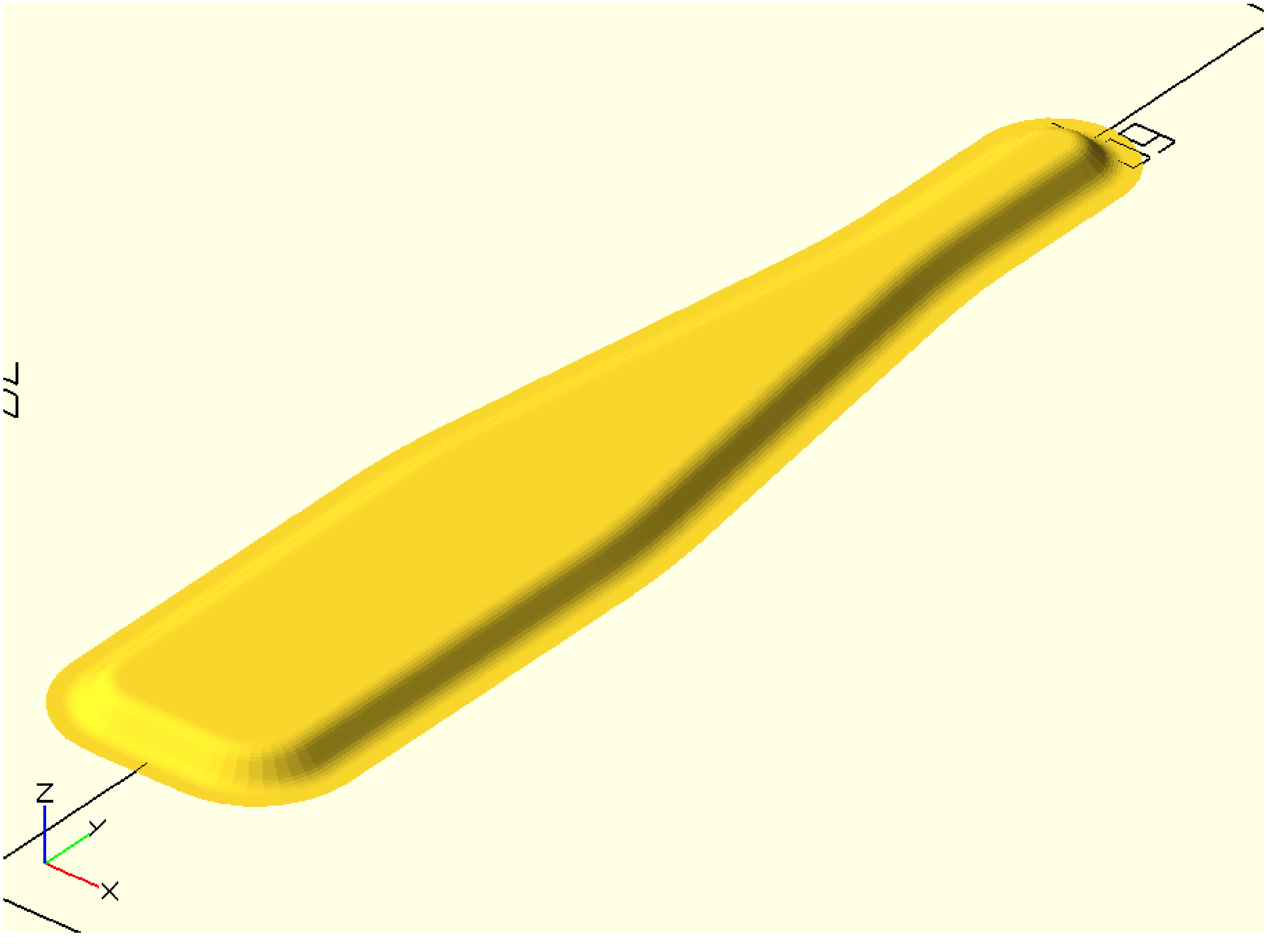
In [54]:

```
sec1=cr(pts1([[-5,10,3],[10,0,3],[0,20,30],[-3,20,30],[0,10,2],[-4,0,2],[0,-10,30],[-3,-
sec1=equidistant_pathc(sec1,200)
path1=cr(pts1([[1,0],[-1,0,1],[-1,1,1],[-.5,0]]),10)
path1=equidistant_path(path1,20)
sol1=prism(sec1,path1)
# Solid above has some open space at the center which might create artifacts on bending
#  so to avoid that the empty space needs to be filled with more points
l1=equidistant_path([[0,12,1],[0,58,1]],100)
l1=sort_points(sol1[-1],l1)
sol1_1=sol1[:-1]
sol1_2=slice_sol([sol1[-1]]+[l1],10)
sol1=sol1_1+sol1_2
```

```
with open('/users/sanjeevprabhakar/openscad/trial.scad','w+')as f:
    f.write(f'''
include<dependencies2.scad>

{swp(sol1)}

    ''')
```



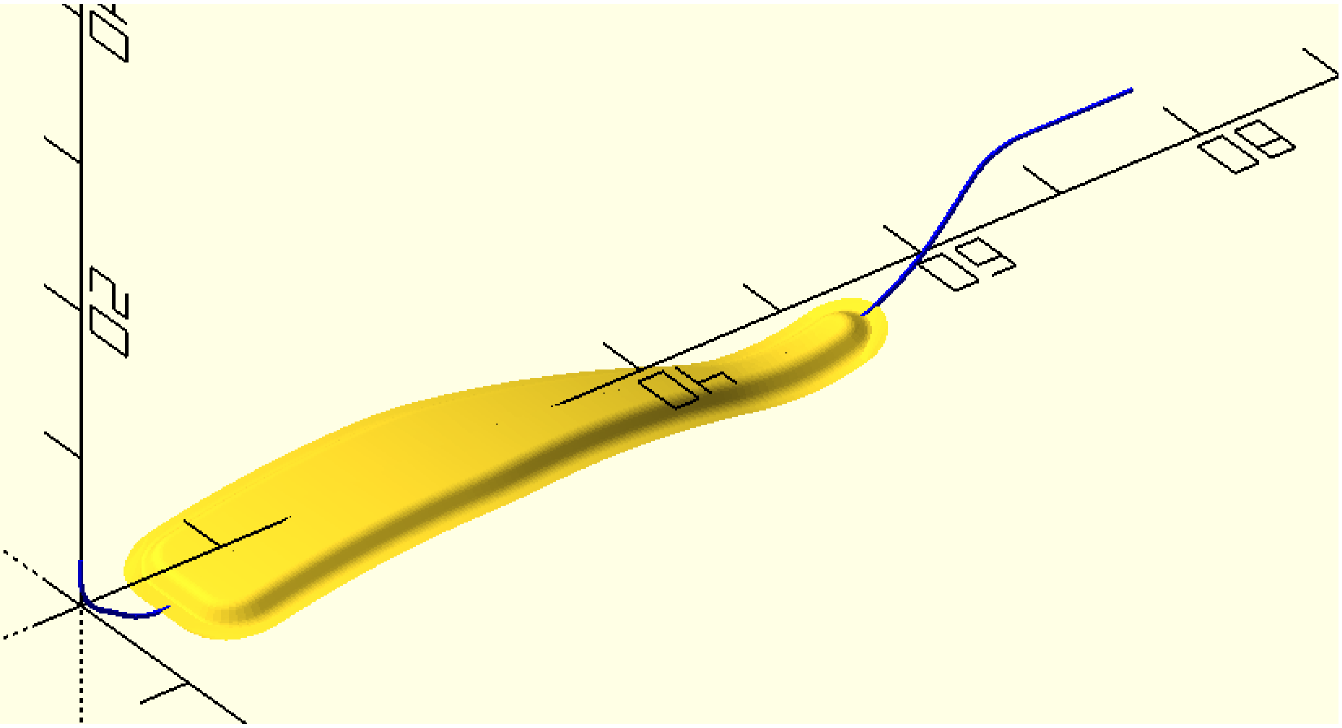

Z
Y
X

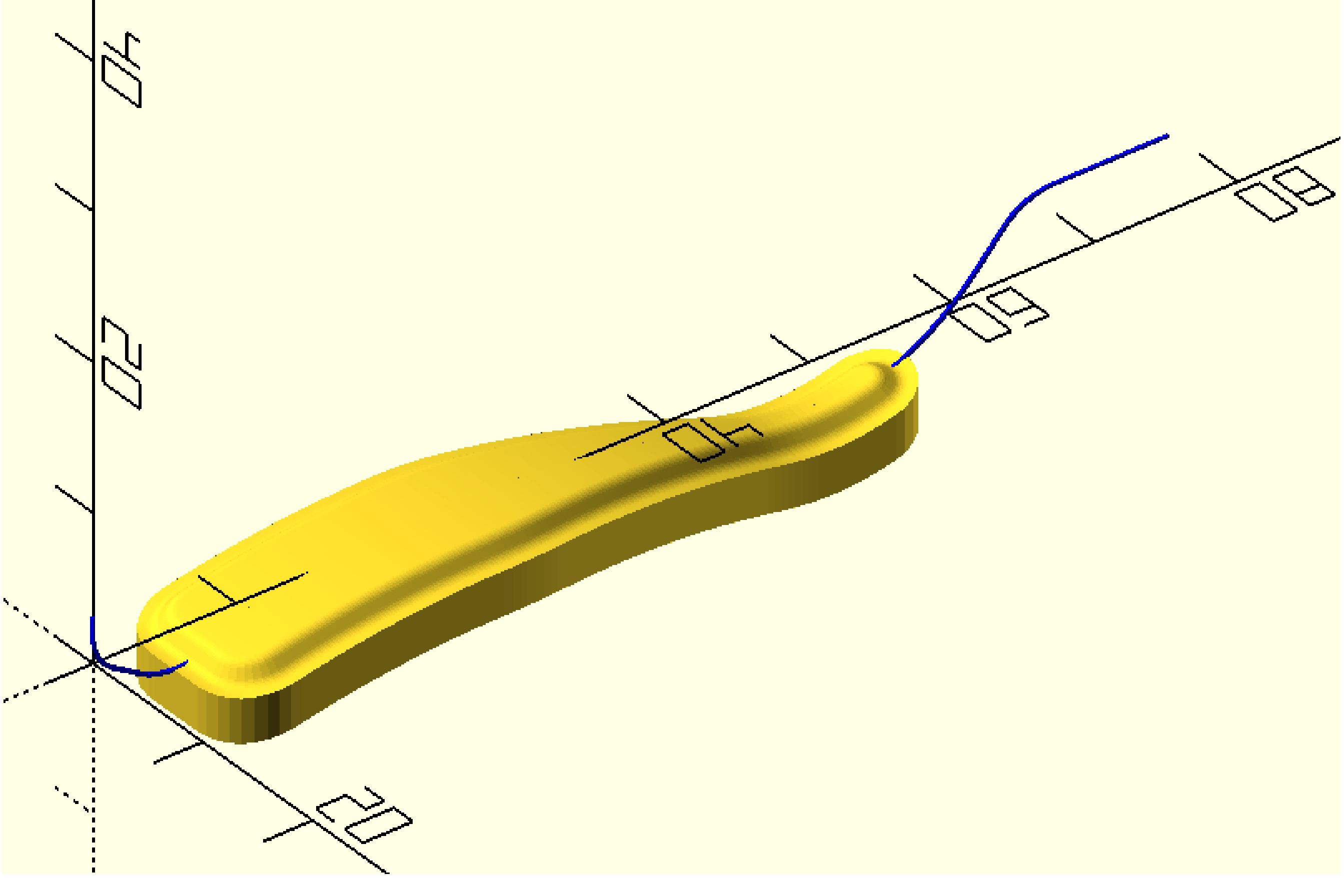

Z
Y
X

use the original path of bottle to fold the cut-out in one direction and make it 3 mm thick

In [56]:

```
path2=q_rot(['y90'],path)

sol2=[wrap_around(p,path2) for p in sol1]

sol2=flip(sol2)+translate([0,0,-3],sol2)

with open('/users/sanjeevprabhakar/openscad/trial.scad','w+')as f:
    f.write(f'''
include<dependencies2.scad>

color("blue")p_line3d({path2},.2,1);
{swp(sol2)}

    ''')
```

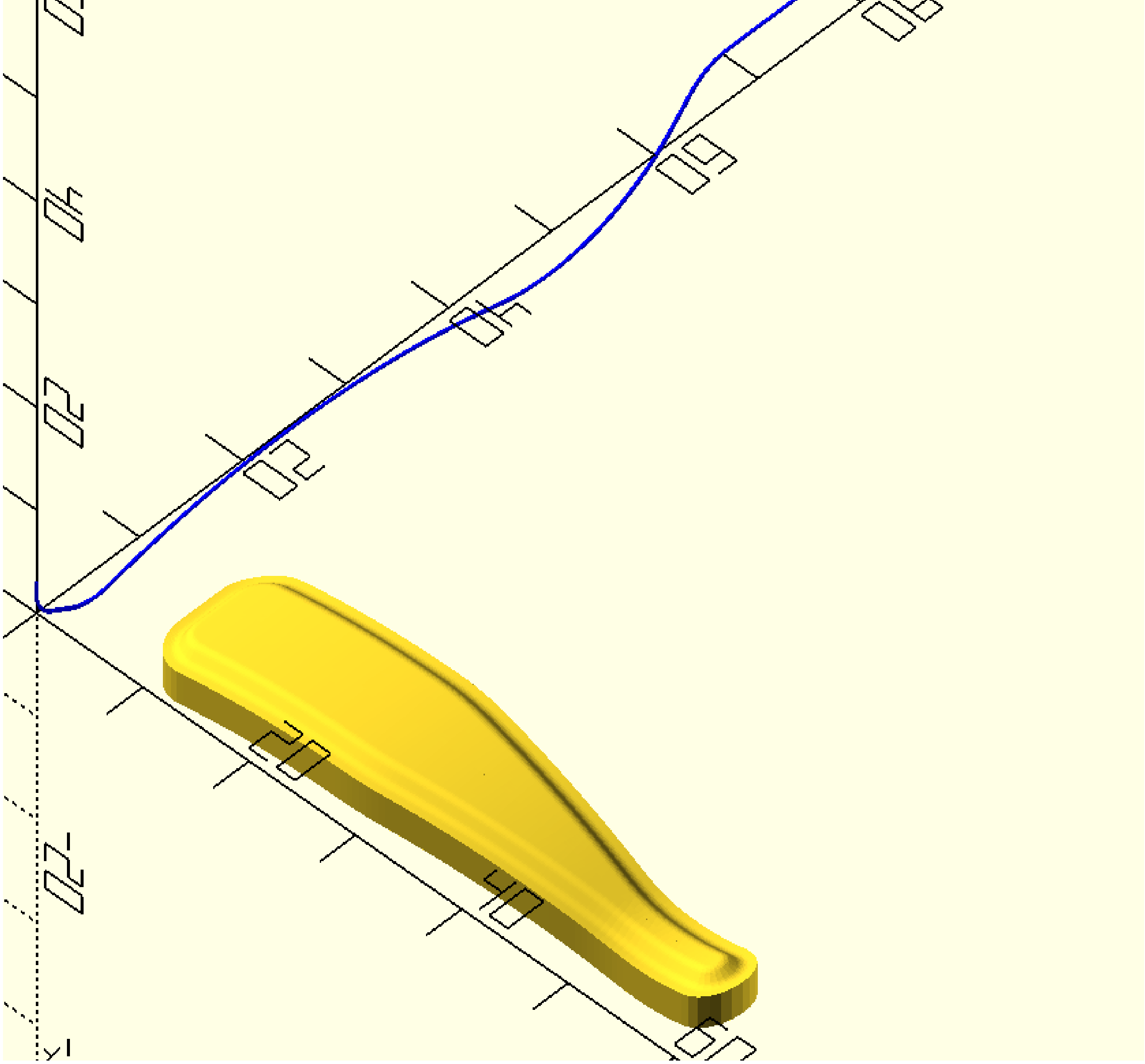




rotate the above folded cut-out to fold in the other direction

In [57]:

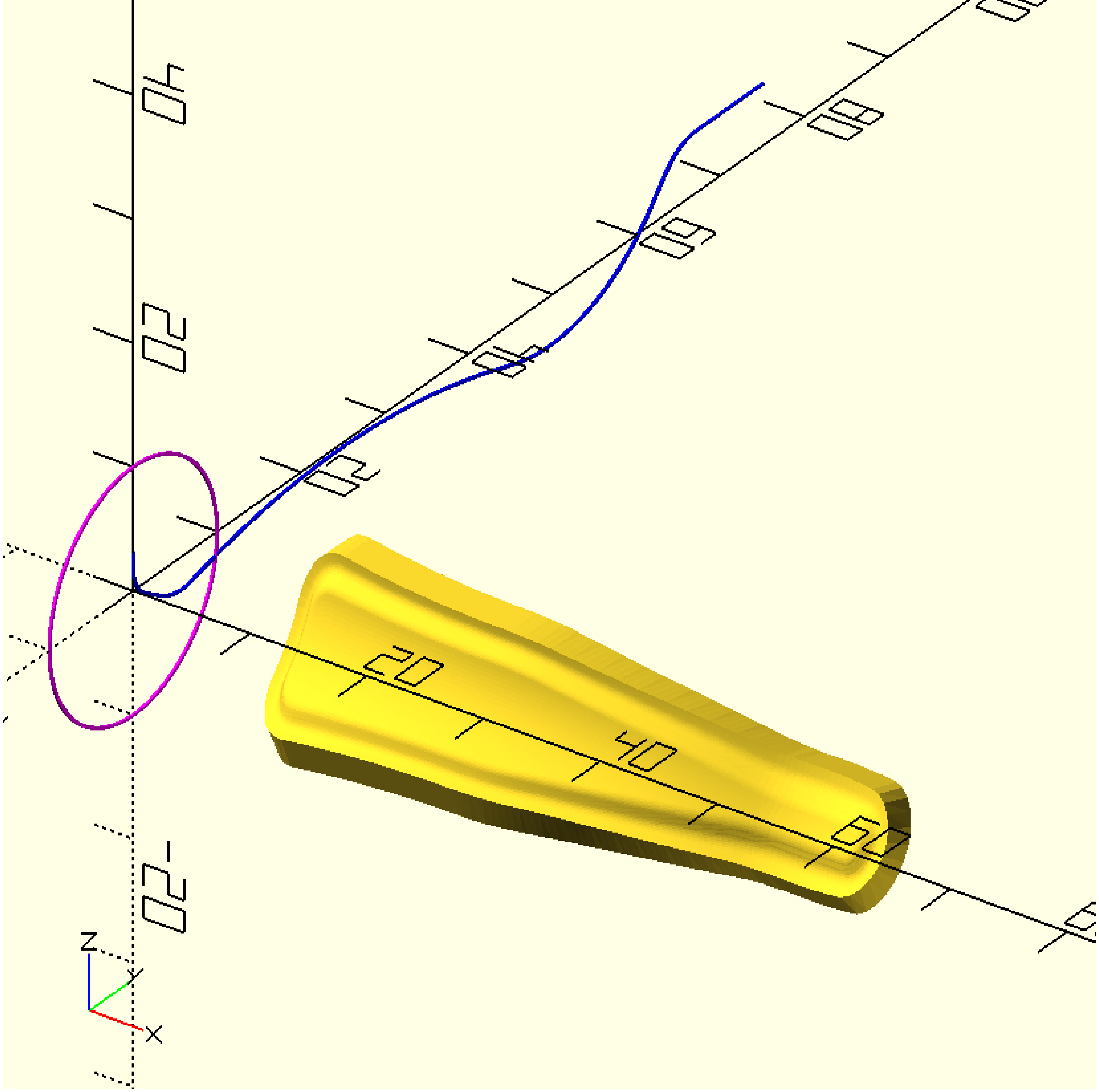
```
sol2=translate([0,10,0],q_rot(['z-90'],sol2))

with open('/users/sanjeevprabhakar/openscad/trial.scad','w+')as f:
    f.write(f'''
include<dependencies2.scad>

color("blue")p_line3d({path2},.2,1);
{swp(sol2)}

    ''')
```



use circle to bend the cut-out in the other direction

In [58]:

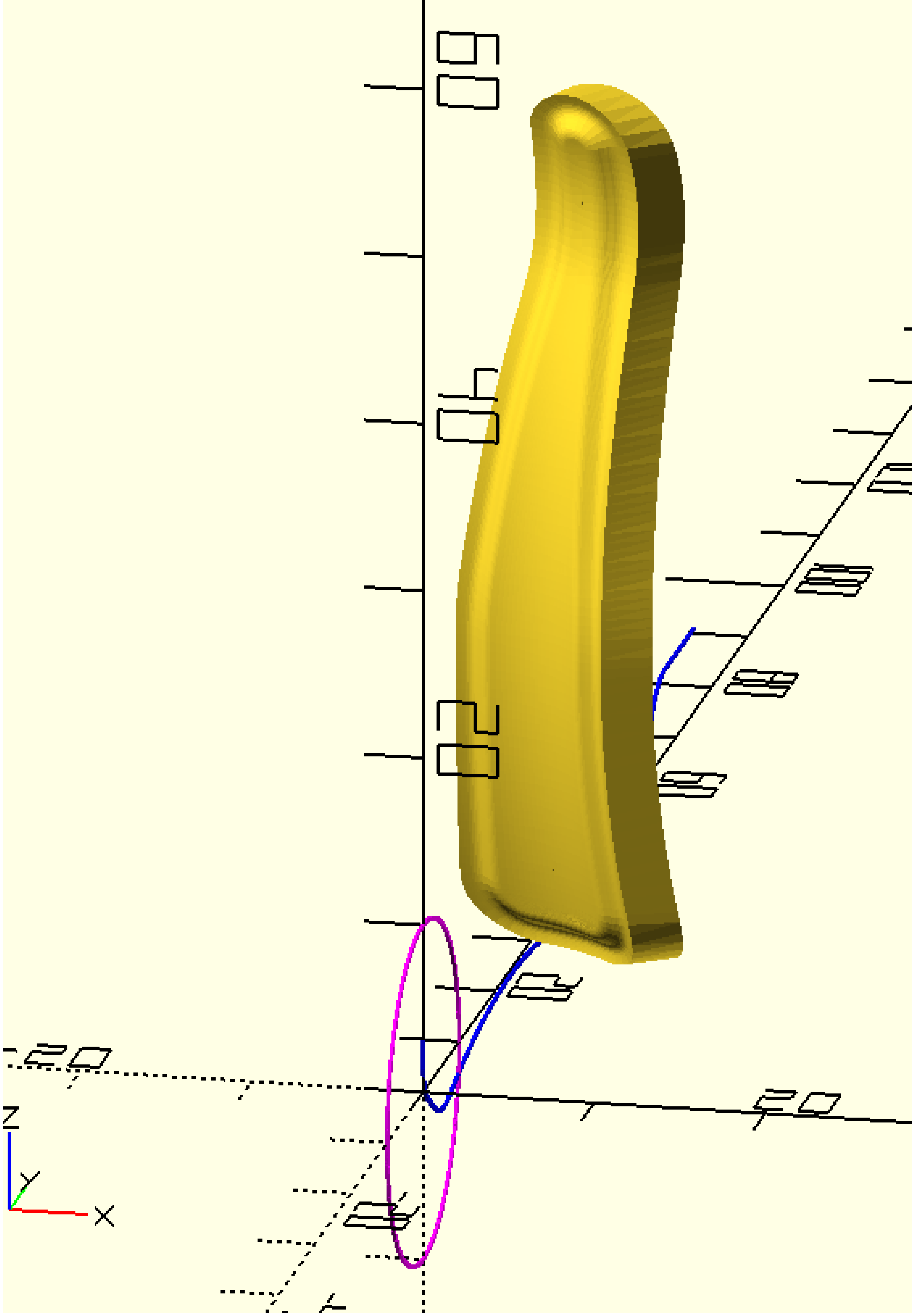
```
path3=q_rot(['y90'],circle(10,s=1000))
sol2=[wrap_around(p,path3) for p in sol2]

with open('/users/sanjeevprabhakar/openscad/trial.scad','w+')as f:
    f.write(f'''
include<dependencies2.scad>

color("blue")p_line3d({path2},.2,1);
color("magenta")p_line3d({path3},.2,1);

{swp(sol2)}

    ''')
```

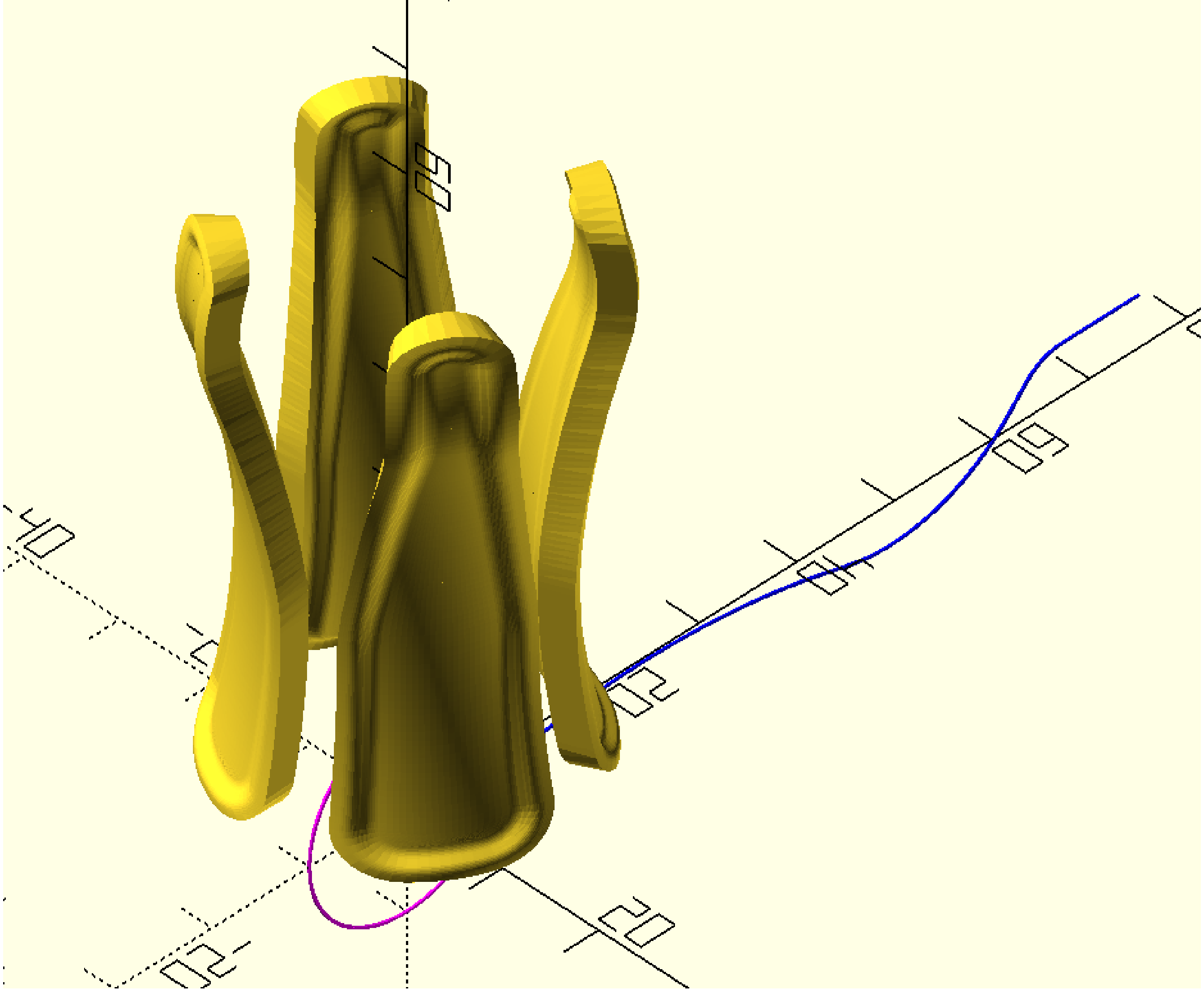


rotate the final cut-out to original position and generate copies (4 numbers)

In [61]:

```
sol3=flip(q_rot(['y-90'],sol2))

with open('/users/sanjeevprabhakar/openscad/trial.scad','w+')as f:
    f.write(f'''
include<dependencies2.scad>

color("blue")p_line3d({path2},.2,1);
color("magenta")p_line3d({path3},.2,1);
for(i=[0,90,180,270])
rotate([0,0,i])
{swp(sol3)}

    ''')
```

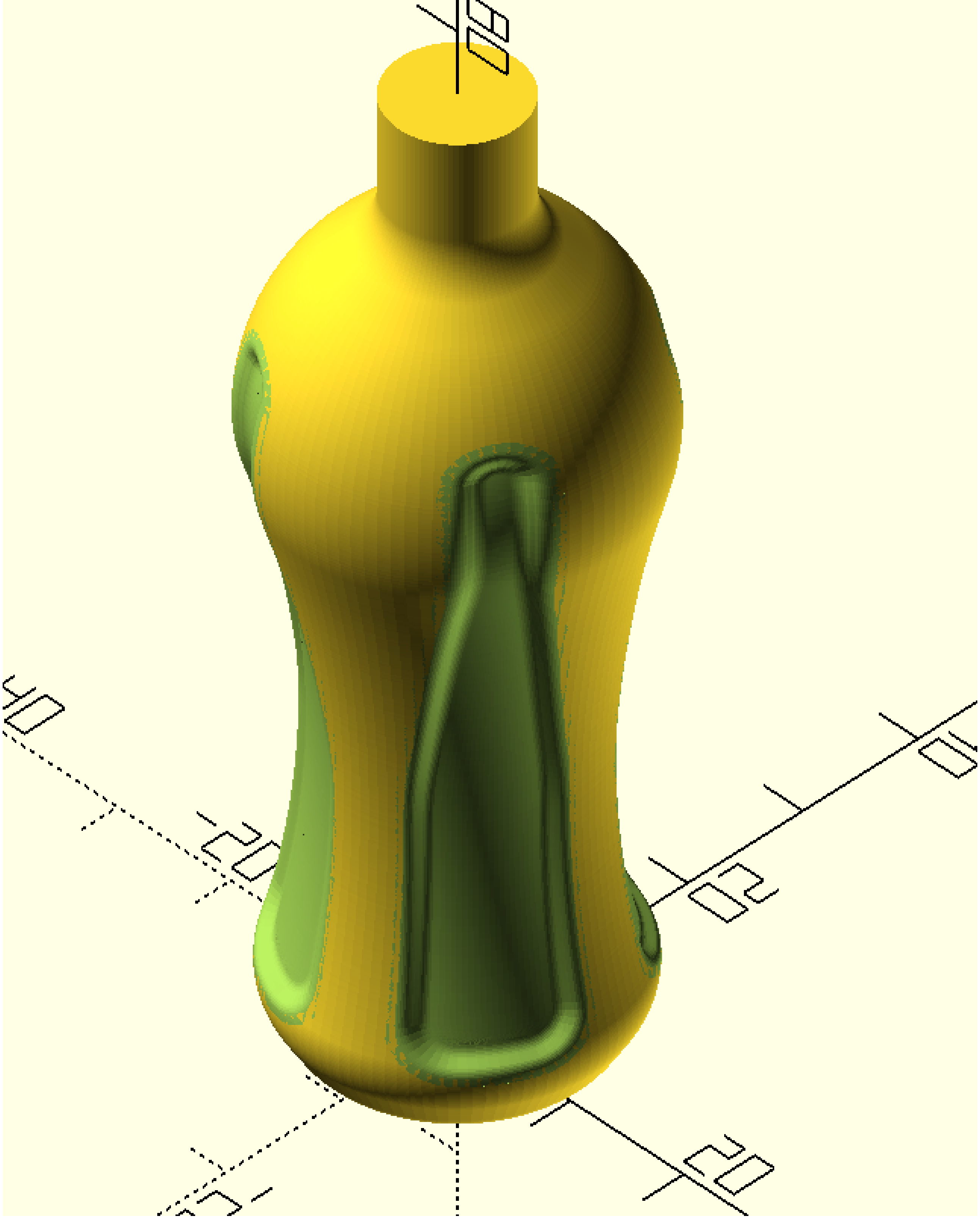

60
40
20
-20
20
Z
Y
X



subtract the above from the bottle

In [62]:

```
with open('/users/sanjeevprabhakar/openscad/trial.scad','w+')as f:
    f.write(f'''
include<dependencies2.scad>
difference(){{
{swp(sol)}
for(i=[0,90,180,270])
rotate([0,0,i])
{swp(sol3)}
}}

    ''')
```

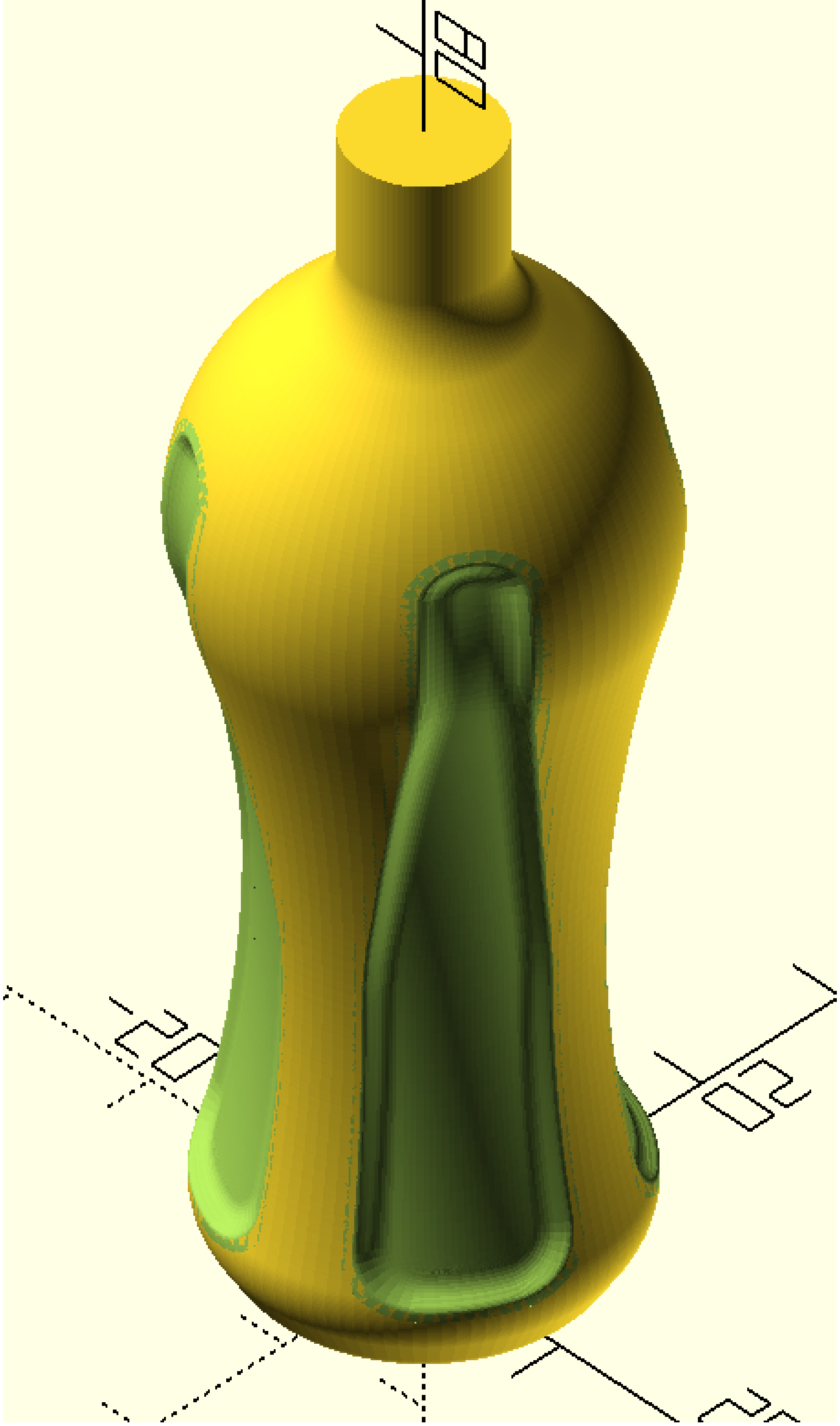


create another copy of this by offsetting the original bottle by - 0.5mm and offsetting the cut-out by +0.5 mm

In [63]:

```
# offset the original bottle by -0.5mm
sol4=prism(sec,path_offset(path,-.5))

sec1=cr(pts1([[-5,10,3],[10,0,3],[0,20,30],[-3,20,30],[0,10,2],[-4,0,2],[0,-10,30],[-3,-
sec1=equidistant_pathc(sec1,200)
```

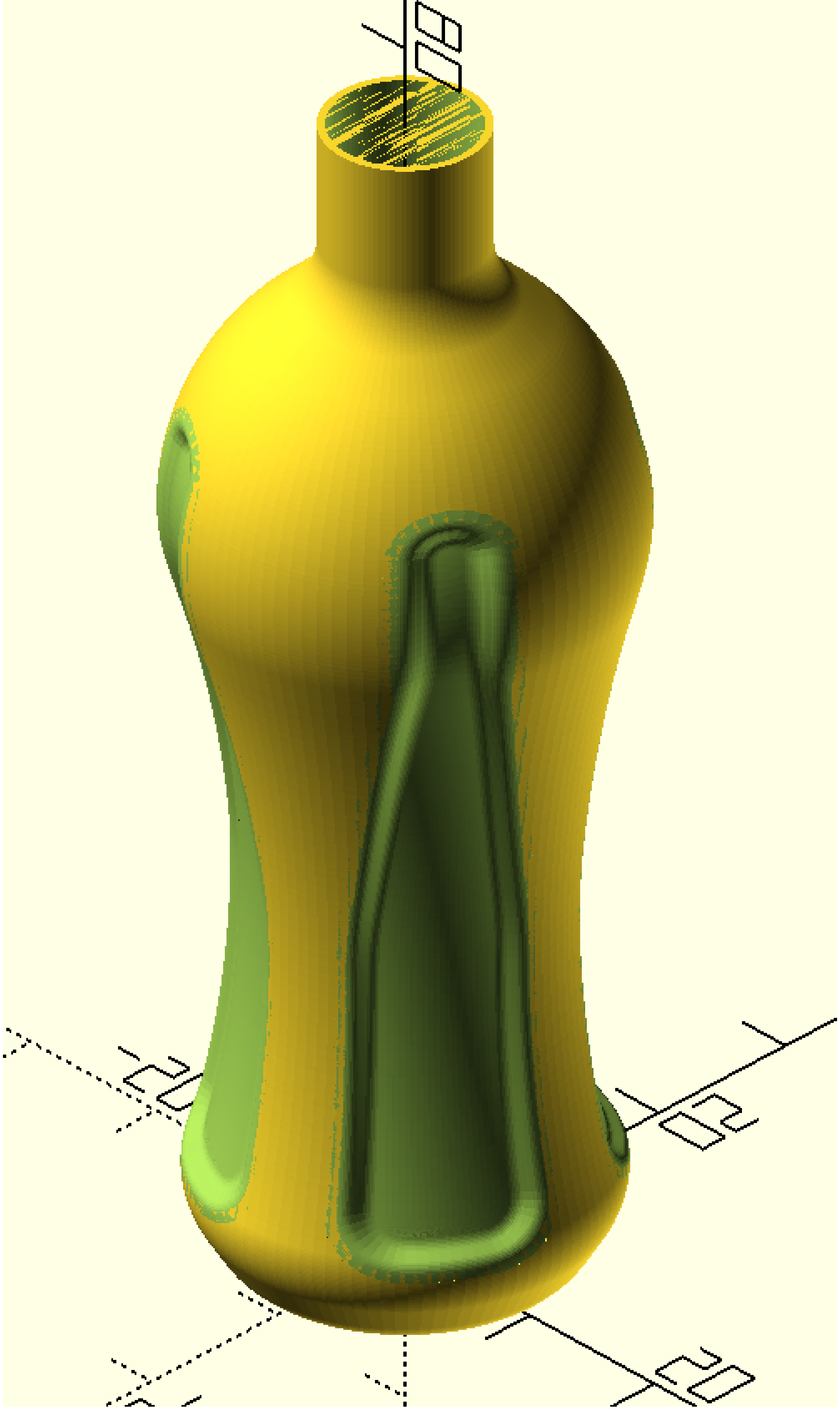
```
# offset the cut-out path by 0.5mm
path1=path_offset(path1,.5)

sol1=prism(sec1,path1)
l1=equidistant_path([[0,12,1.5],[0,58,1.5]],100)
l1=sort_points(sol1[-1],l1)
sol1_1=sol1[:-1]
sol1_2=slice_sol([sol1[-1]]+[l1],10)
sol1=sol1_1+sol1_2
path2=q_rot(['y90'],path)

sol2=[wrap_around(p,path2) for p in sol1]


sol2=flip(sol2)+translate([0,0,-3],sol2)
sol2=translate([0,10,0],q_rot(['z-90'],sol2))
path3=q_rot(['y90'],circle(10,s=500))
sol2=[wrap_around(p,path3) for p in sol2]
sol5=flip(q_rot(['y-90'],sol2))

with open('/users/sanjeevprabhakar/openscad/trial.scad','w+')as f:
    f.write(f'''
include<dependencies2.scad>

difference(){{
{swp(sol4)}
for(i=[0,90,180,270])
rotate([0,0,i])
{swp(sol5)}
}}

    ''')
```

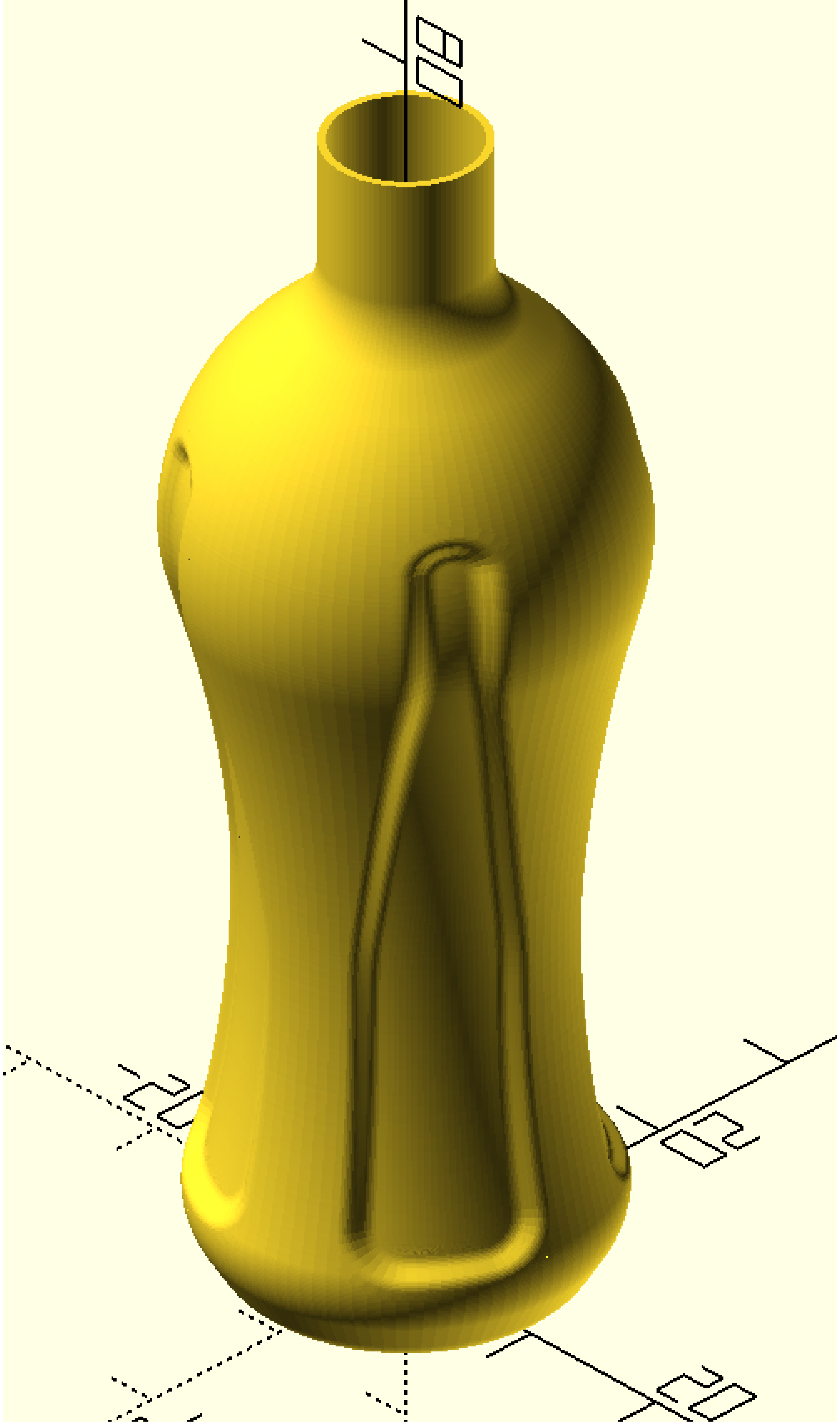


subtract the 2nd solid from the 1st solid

In [64]:

```
with open('/users/sanjeevprabhakar/openscad/trial.scad','w+')as f:
    f.write(f'''
include<dependencies2.scad>

difference(){{
difference(){{
{swp(sol)}
for(i=[0,90,180,270])
rotate([0,0,i])
{swp(sol3)}
}}


difference(){{
{swp(sol4)}
for(i=[0,90,180,270])
rotate([0,0,i])
{swp(sol5)}

}}
}}


    ''')
```


80
20
20
20


80
20
20
20