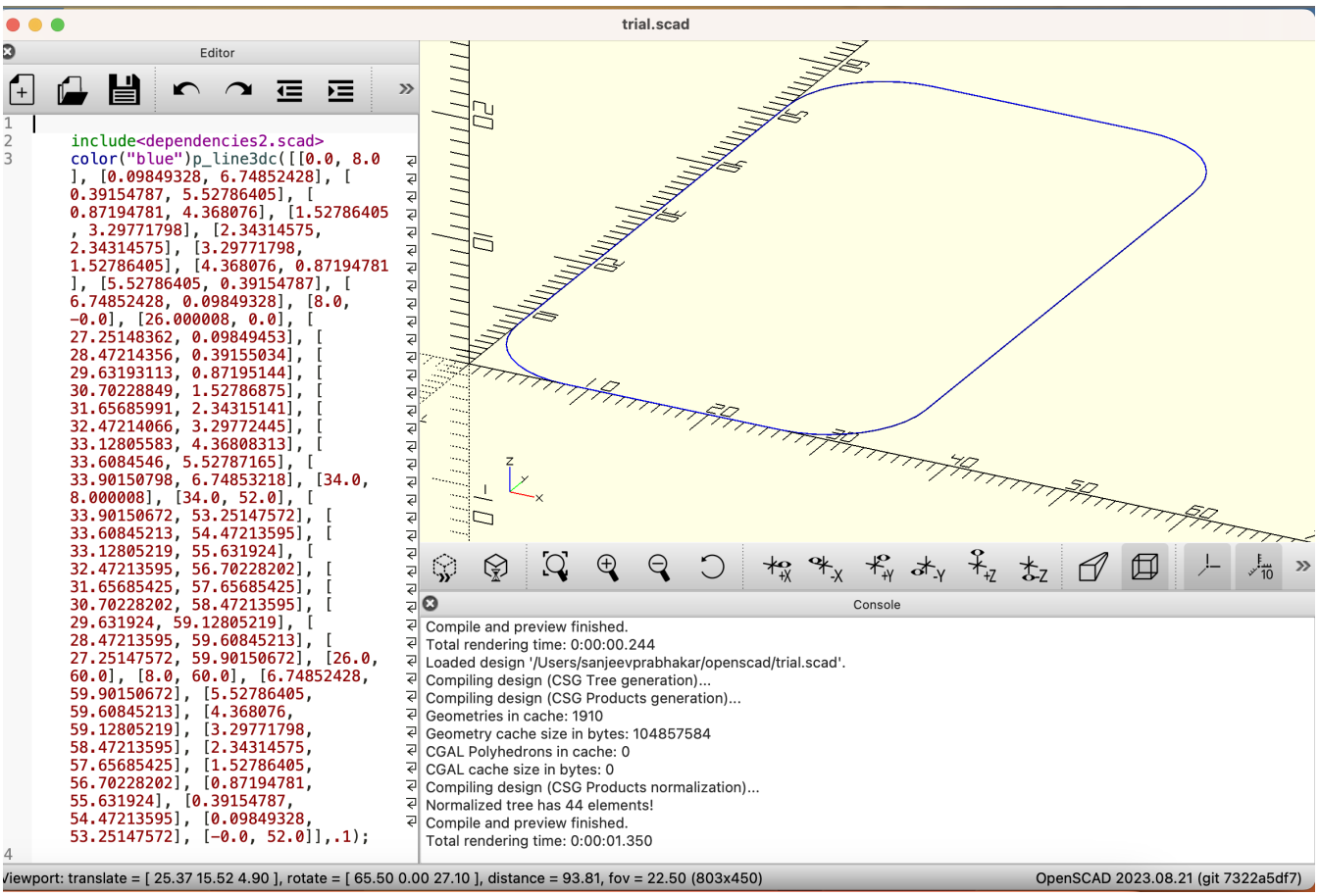
In [2]:
```
from openscad1 import *
```

Approach 2 (offset method)

create a 2d section

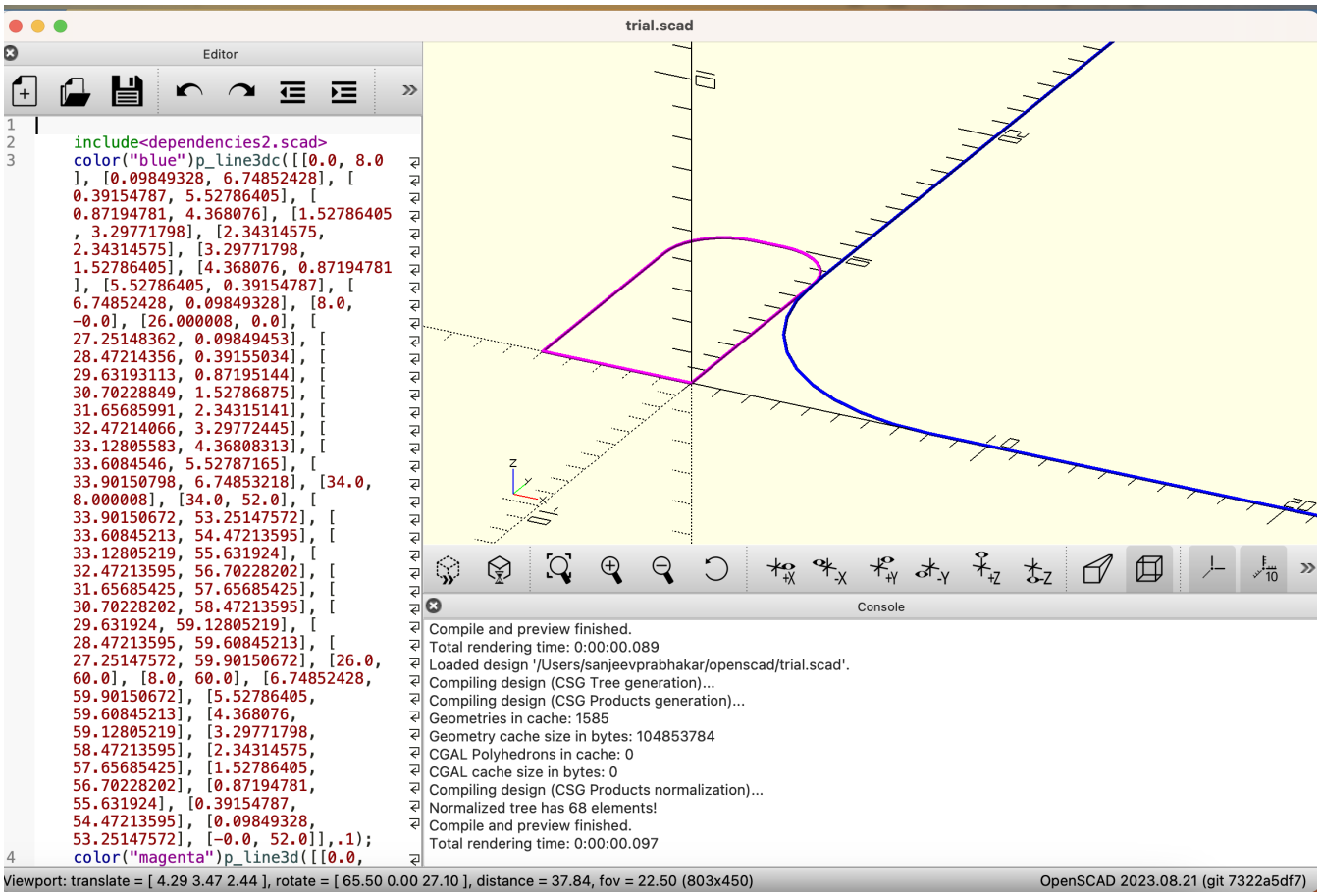
In [3]:
```
sec=cr(pts1([[0,0,8],[34,0,8],[0,60,8],[-34,0,8]]),10)

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

    ''')
```



draw an offset pattern where x-coordinate represent value to offset the section and y-coordinate represent location of section in z-coordinate.

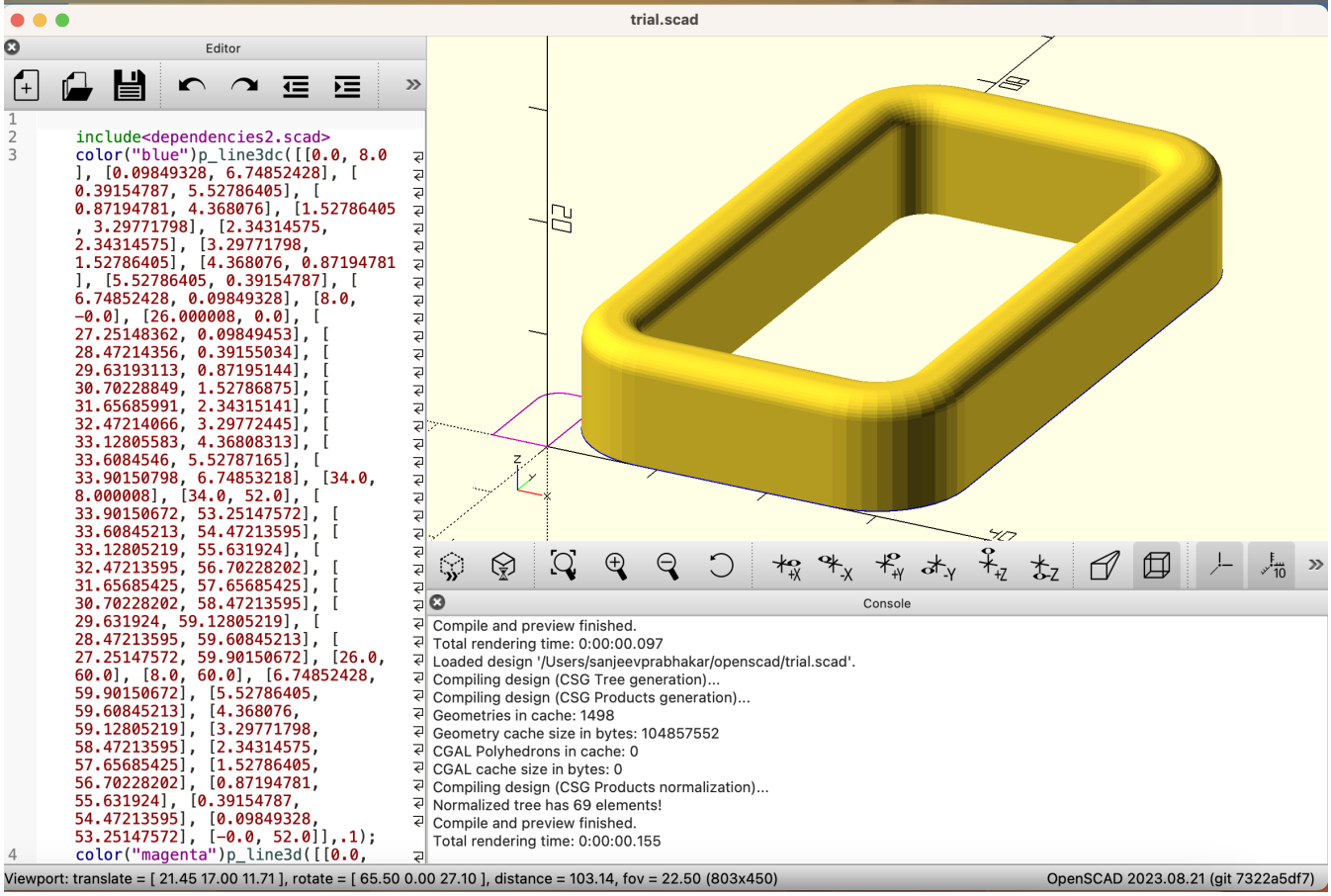
example: cylinder can be drawn by drawing 2 circles, one in the xy plane and other by translating the circle to a required height

In [7]:
```
path=cr(pts1([[0,0],[0,10,2],[-5,0,2],[0,-10]]),10)
path=path+[path[0]]

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

    ''')

# please note that 'magenta' line x-coordinates are offset values of the bigger section
# y-coordinates are the height values
```

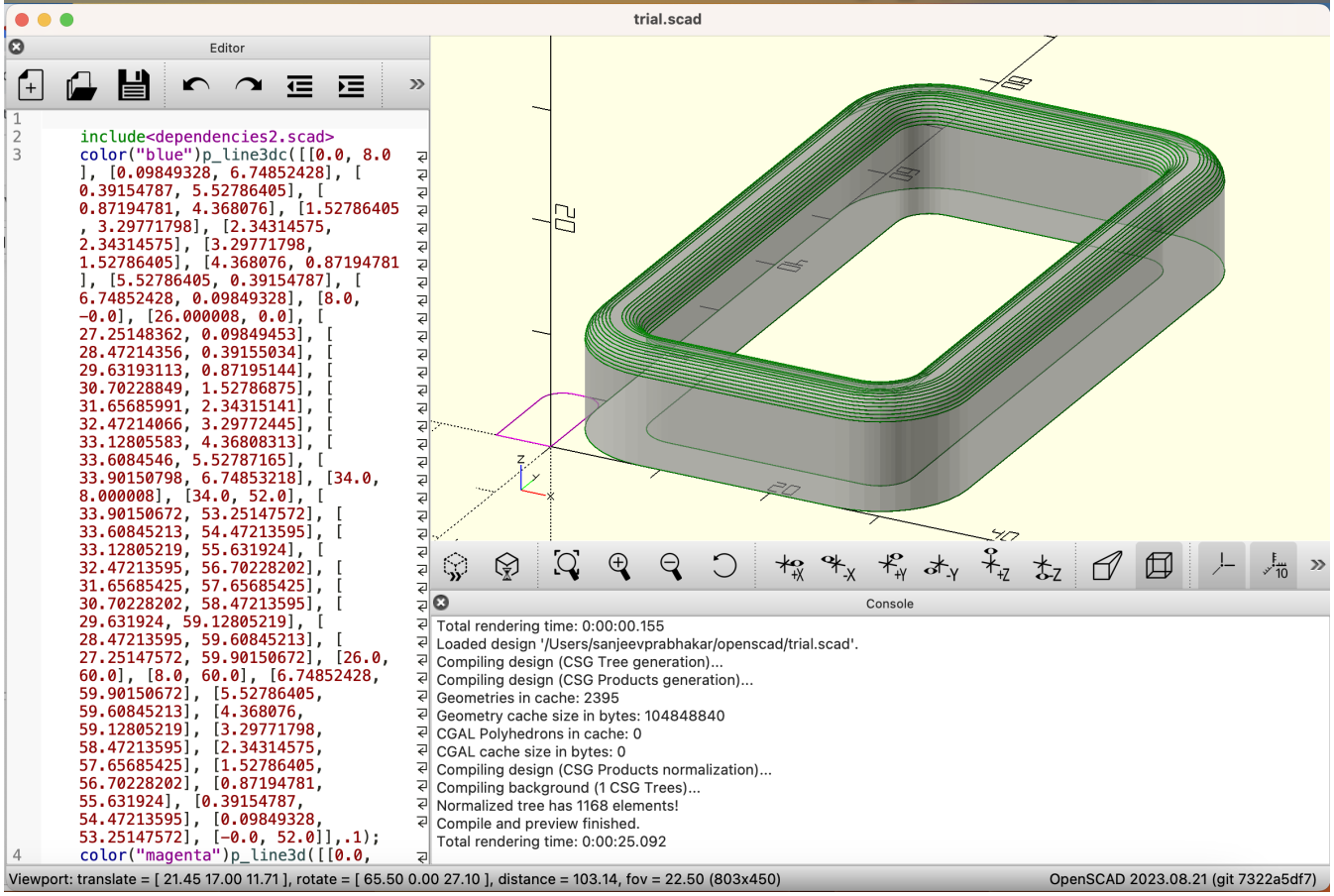


Create a solid by offsetting the section

In [8]:
```
sol=[translate([0,0,y],offset(sec,x)) for (x,y) in path]

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

    ''')
```



In [9]:
```
with open('/users/sanjeevprabhakar/openscad/trial.scad','w+') as f:
    f.write(f'''
    include<dependencies2.scad>
    color("blue")p_line3dc({sec},.1);
    color("magenta")p_line3d({path},.1);
    %{swp_c(sol)}
    color("green")for(p={sol})p_line3dc(p,.1);

    ''')
```


trial.scad
Editor
include<dependencies2.scad>
color("blue")p_line3dc([[0.0, 8.0
], [0.09849328, 6.74852428], [
0.39154787, 5.52786405], [
0.87194781, 4.368076], [1.52786405
, 3.29771798], [2.34314575,
2.34314575], [3.29771798,
1.52786405], [4.368076, 0.87194781
], [5.52786405, 0.39154787], [
6.74852428, 0.09849328], [8.0,
-0.0], [26.000008, 0.0], [
27.25148362, 0.09849453], [
28.47214356, 0.39155034], [
29.63193113, 0.87195144], [
30.70228849, 1.52786875], [
31.65685991, 2.34315141], [
32.47214066, 3.29772445], [
33.12805583, 4.36808313], [
33.6084546, 5.52787165], [
33.90150798, 6.74853218], [34.0,
8.000008], [34.0, 52.0], [
33.90150672, 53.25147572], [
33.60845213, 54.47213595], [
33.12805219, 55.631924], [
32.47213595, 56.70228202], [
31.65685425, 57.65685425], [
30.70228202, 58.47213595], [
29.631924, 59.12805219], [
28.47213595, 59.60845213], [
27.25147572, 59.90150672], [26.0,
60.0], [8.0, 60.0], [6.74852428,
59.90150672], [5.52786405,
59.60845213], [4.368076,
59.12805219], [3.29771798,
58.47213595], [2.34314575,
57.65685425], [1.52786405,
56.70228202], [0.87194781,
55.631924], [0.39154787,
54.47213595], [0.09849328,
53.25147572], [-0.0, 52.0]],.1);
color("magenta")p_line3d([[0.0,
Console
Total rendering time: 0:00:00.155
Loaded design '/Users/sanjeevprabhakar/openscad/trial.scad'.
Compiling design (CSG Tree generation)...
Compiling design (CSG Products generation)...
Geometries in cache: 2395
Geometry cache size in bytes: 104848840
CGAL Polyhedrons in cache: 0
CGAL cache size in bytes: 0
Compiling design (CSG Products normalization)...
Compiling background (1 CSG Trees)...
Normalized tree has 1168 elements!
Compile and preview finished.
Total rendering time: 0:00:25.092
Viewport: translate = [21.45 17.00 11.71], rotate = [65.50 0.00 27.10], distance = 103.14, fov = 22.50 (803x450)
OpenSCAD 2023.08.21 (git 7322a5df7)