Report CADQuery Bottle Design

Introduction:

The CADQuery Bottle is made from two parts the top part which is the curvy part with the cap at the top, and the bottom part which is the cylindrical. In the following, I will add the code to this report and explain how the two parts of the parameterized bottle are made and define each parameter and how its variation changes the form of the bottle.

Code:

```
import cadquery as cq
from math import pi

#Bottom part parameters
THeight, MHeight, BHeight, radius = 2, 0.5 ,3 ,2
RDiff=1/4
ROffset=1/4
NRings = 10
RRadius = radius - RDiff

#Volume Calculation
BPart_Volume = pi*(radius**2)*BHeight
TPart_Volume = pi*(radius**2)*THeight
MPart_Height = NRings*((pi/3)*ROffset*((radius**2)+(RRadius**2) + radius*RRadius) + pi*(radius**2)*MHeight)
Bottle_Volume = BPart_Volume + TPart_Volume + MPart_Height

#Construction of the Bottom Part
Bresult = (cq.Workplane("front").
          circle(radius).workplane(offset=BHeight).circle(radius).loft(combine = True)
          .edges('%CIRCLE and <Z').fillet(1))
          
for i in range(NRings):
    Bresult = (Bresult.faces(">Z").workplane(centerOption="CenterOfMass")
              .circle(radius).workplane(offset=ROffset).circle(RRadius).loft(combine = True)
              
              .faces(">Z").workplane(centerOption="CenterOfMass")
              .circle(RRadius).workplane(offset=ROffset).circle(radius).loft(combine = True)
    
              .faces(">Z").workplane(centerOption="CenterOfMass")
              .circle(radius).workplane(offset=MHeight).circle(radius).loft(combine = True))
```

```
Tresult = (Bresult.faces(">Z").workplane(centerOption="CenterOfMass")
          .circle(radius).workplane(offset=THeight).circle(radius).loft(combine = True)
          
          .shell(.15)
          )


#Construction of the Top Part

#Top part parameters
distZ, distX = 0.8, radius/3

sPnts = [
(distX,distZ*4),
(2*distX+0.1,distZ*2),
(distX*3+0.15,0)
]

topPart = cq.Workplane("XZ").lineTo(0,distZ*5).lineTo(distX,distZ*5).lineTo(distX,distZ*4+0.01)
topPart = topPart.spline(sPnts,includeCurrent=True).close()

topPart = topPart.revolve(axisStart=(0,0), axisEnd=(0,1),clean=True)

#Assembly of both parts
topPart=topPart.translate((0,0,THeight + BHeight + 2*ROffset*NRings + (NRings)*MHeight))
```

<u>The top part:</u>

Parameters: distZ: controls the height of the top part
distX: controls the radius of the circle

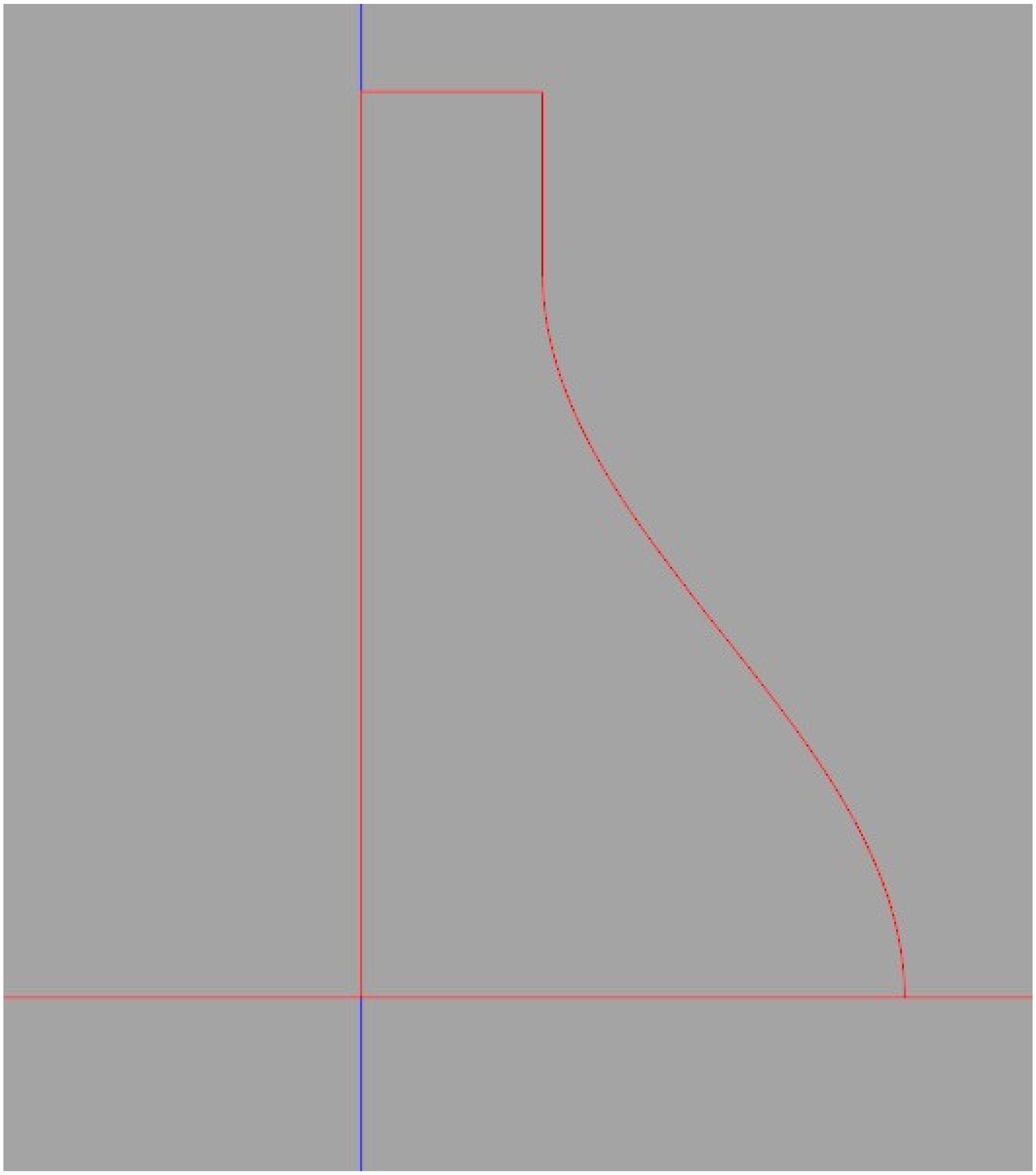
As a first step, I've set a workplane that should be following 2 axes placed in order: Z and X, and then the plan is named s as we see in line 6. This plan will serve as a basis for building blocks one on top of the other until the construction of the entire bottle, part by part which will be explained with more details after.

As a second step, sPnts contains all the coordinates of 3 points in the workplane (the 2D plan ZX), the first coordinate is following the Z axis and the second is following the X axis (that's why the order of the axes while defining the workplane is important): the parameters which intervene here are distZ and distX multiplied afterwards by coefficients like 4,2 and 3. distZ represents a distance along the Z axis and distX a distance along the X axis the coefficients only allow to have a larger shape of the upper part:

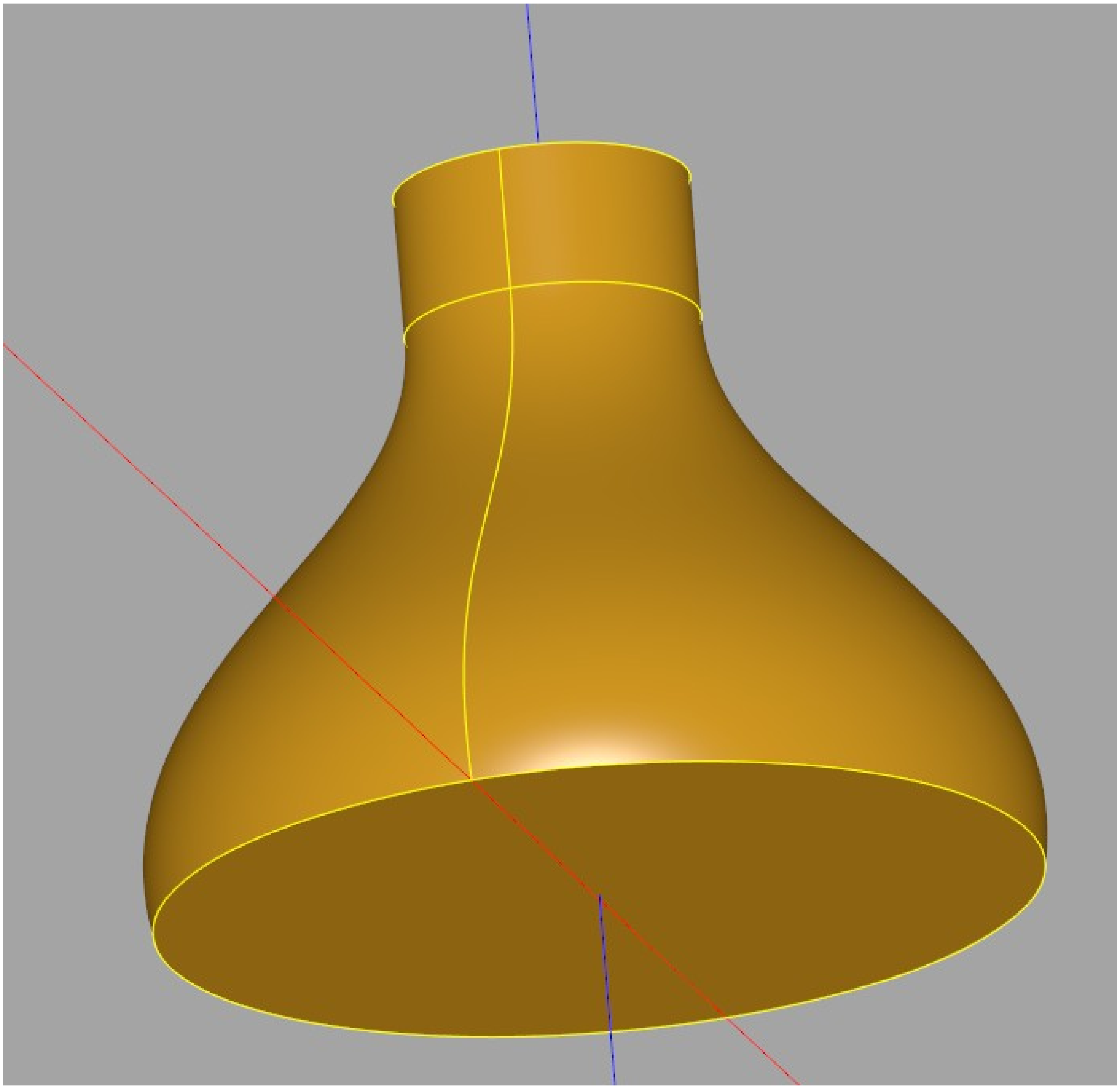
The multiplication by coefficients is linear and doesn’t affect the shape, they could be even deleted since distZ and distX are local variables.

The reconstruction of the topPart after using the spline function gives the result below.

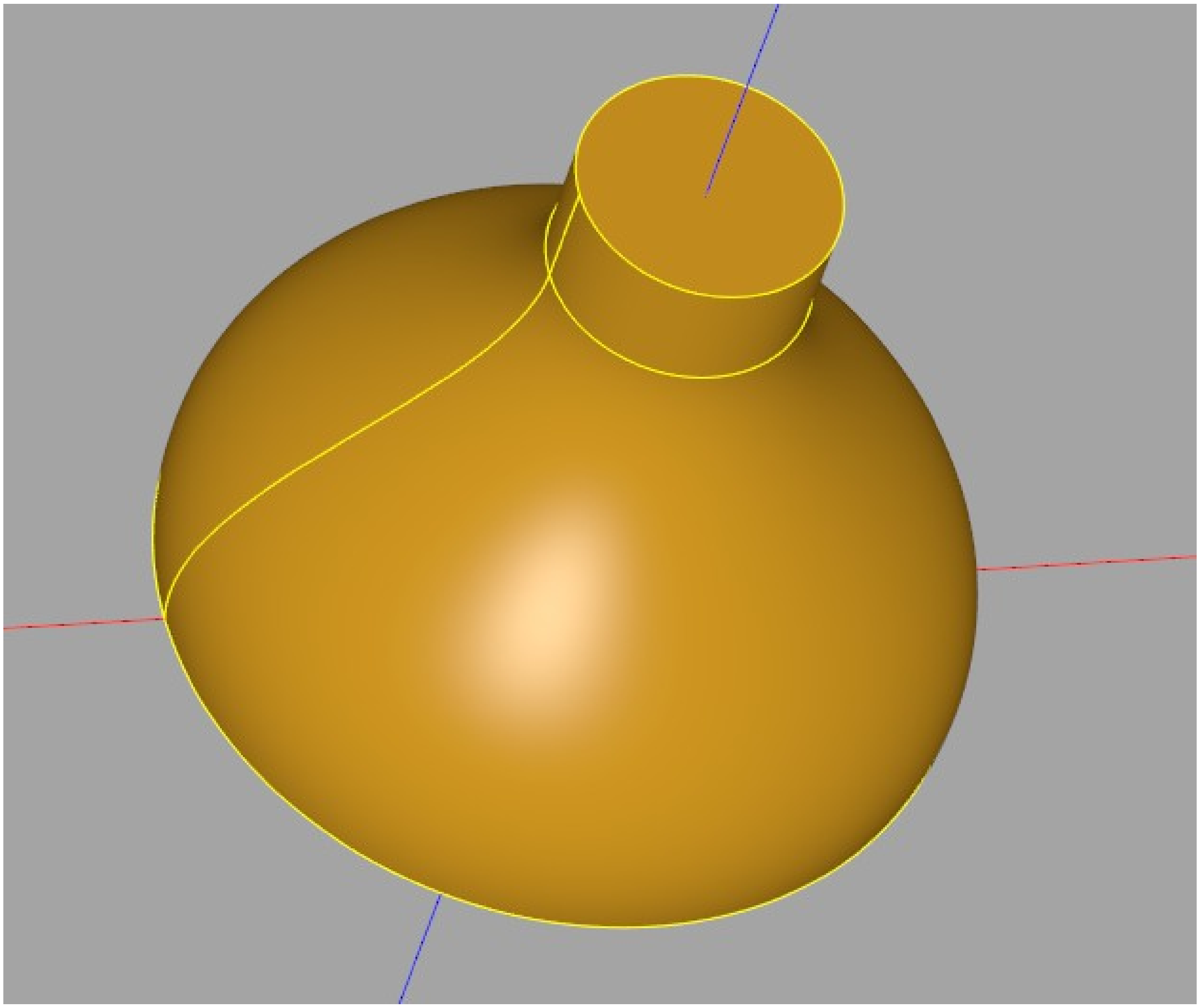
The spline function interpolate the 3 points, and the close function closes the spline. In the next result the parameters distZ=distX=0.5.



The revolve function is used after that on this 2D sketch, and takes as variables axisstart and axisend, to make sure that the rotation will be around the Z axis, and takes also a boolean variable: clean set to True, to clean the 2D sketch. As a result we get the following.

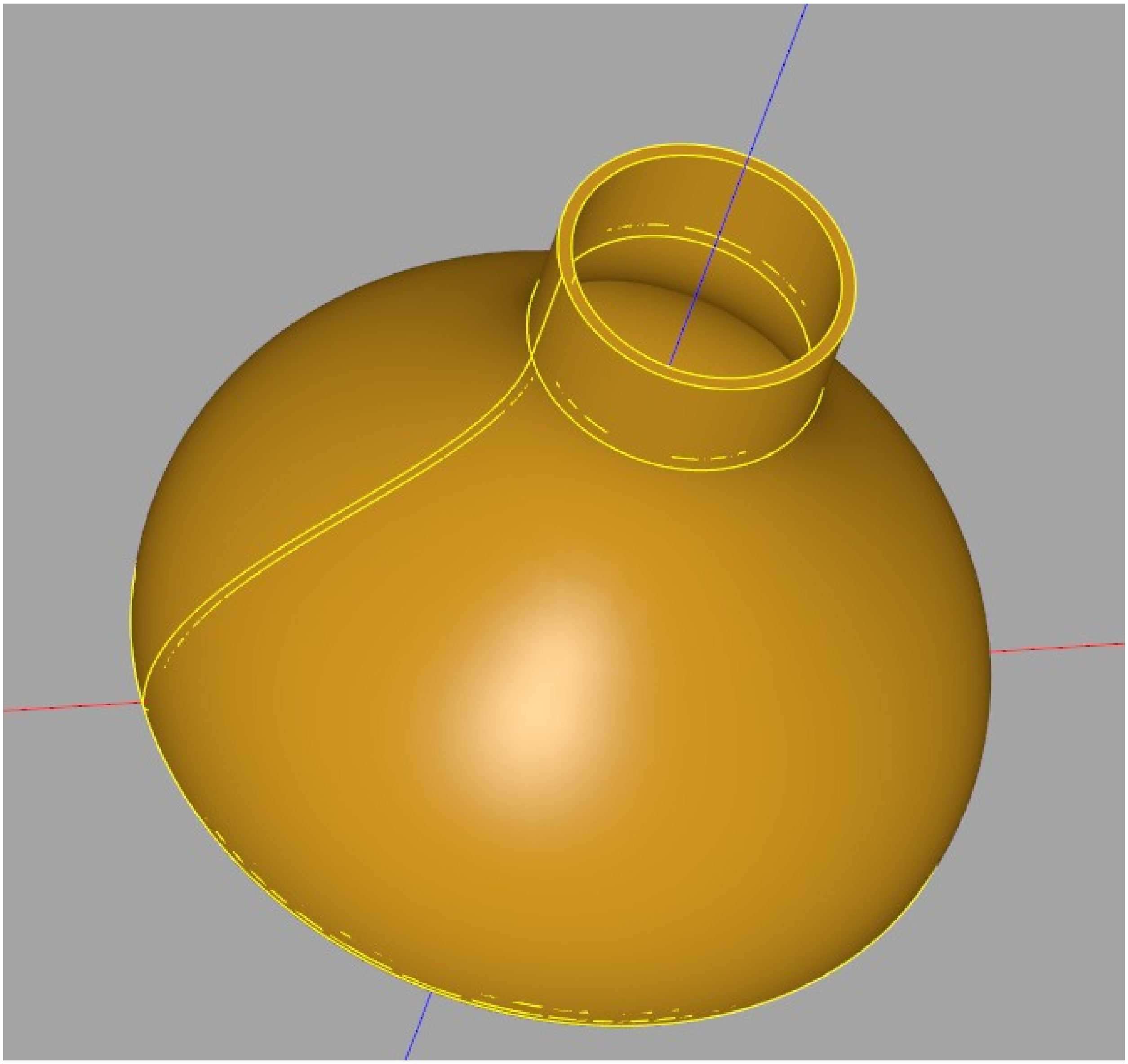


distZ=0.5 & distX=0.5



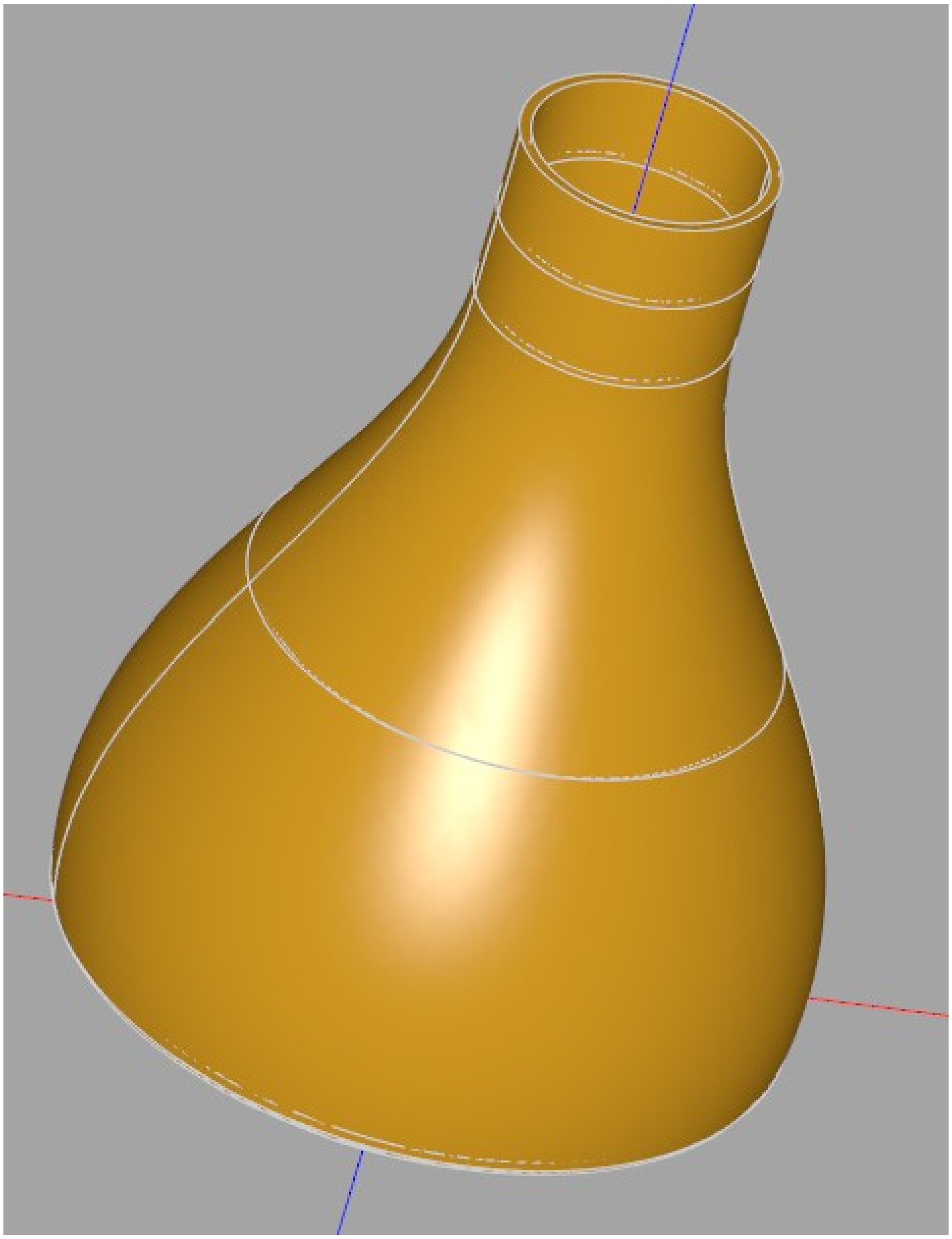
distZ=0.5 & distX=0.5

I applied a shell function to this part with a thickness = 0.05, starting from the upper face (faces (“>Z”)) to the bottom face, creating the bottle’s wall by emptying the inside of the shape above. And gives the next result

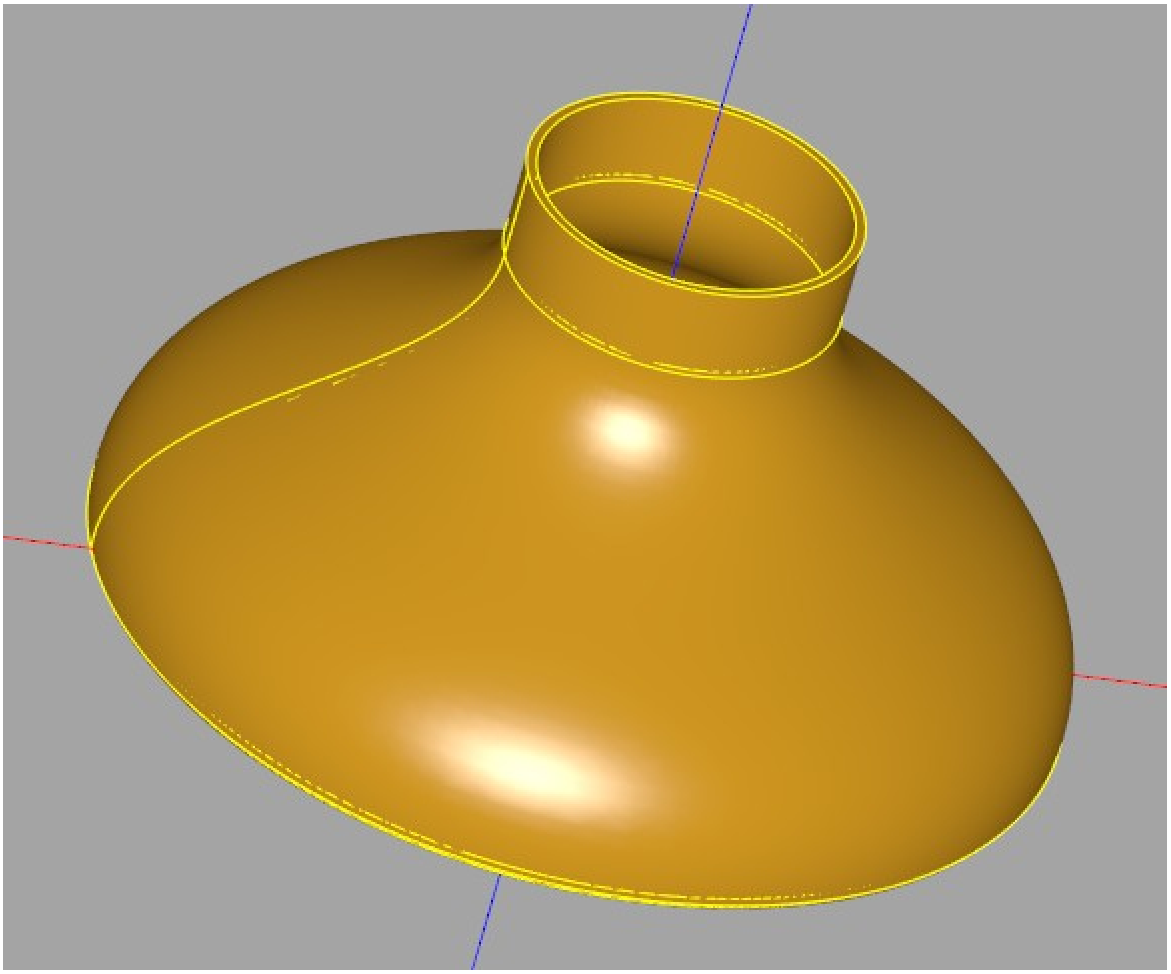


distZ=0.5 & distX=0.5

Below, we find the results after variation of the parameters distZ and distX.



distZ=0.8 & distX=0.5



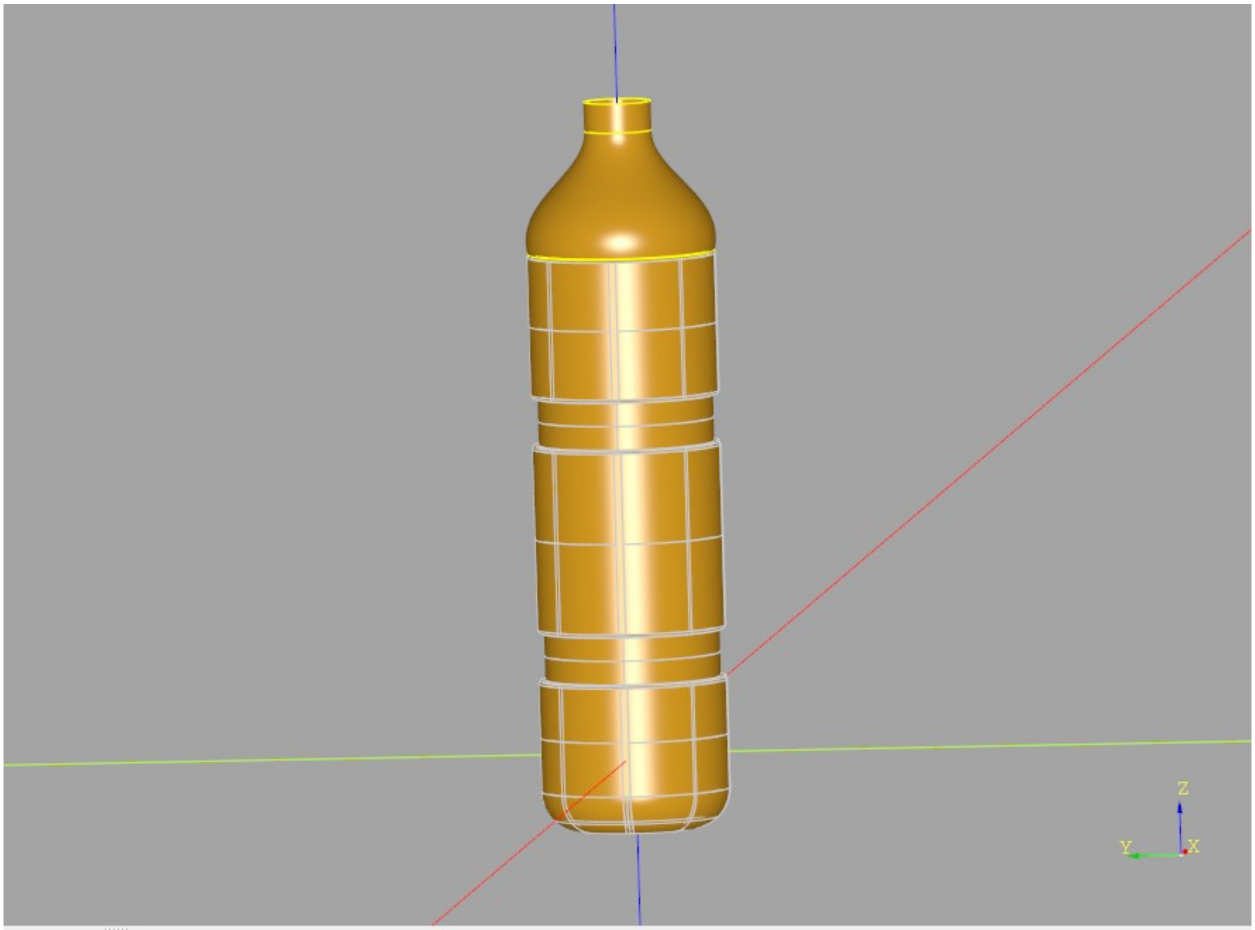
distZ=0.5 & distX=0.8

The bottom part:

Parameters: THeight: Top cylinder height
MHeight: Middle cylinder height
BHeight: Bottom cylinder height
radius: Radius of Top, Middle and Bottom cylinders
RRradius = radius – Rdiff: Radius of one ring (increasing Rdiff reduces the radius of the ring)
Nrings= Number of rings

Before

As a first result using the method developed in the prvious report, we had the following result, where the number of rings is just two and isn't a parameter.

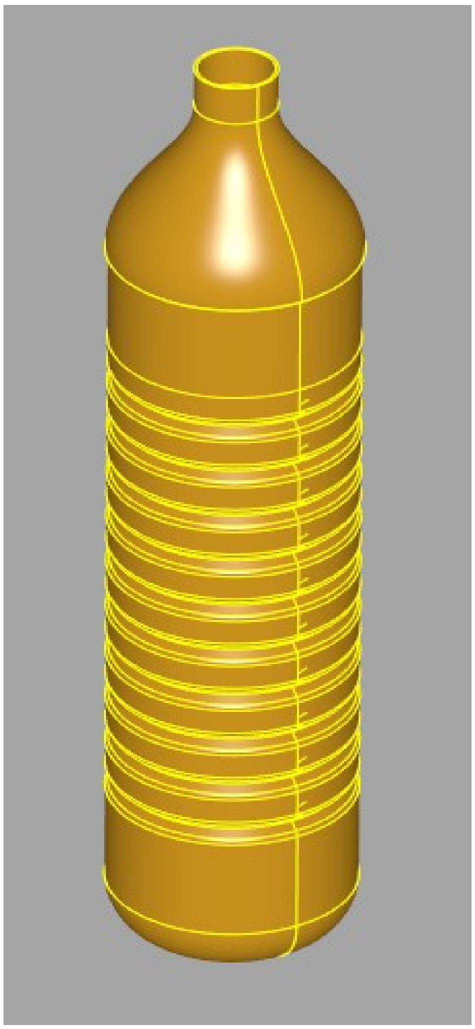


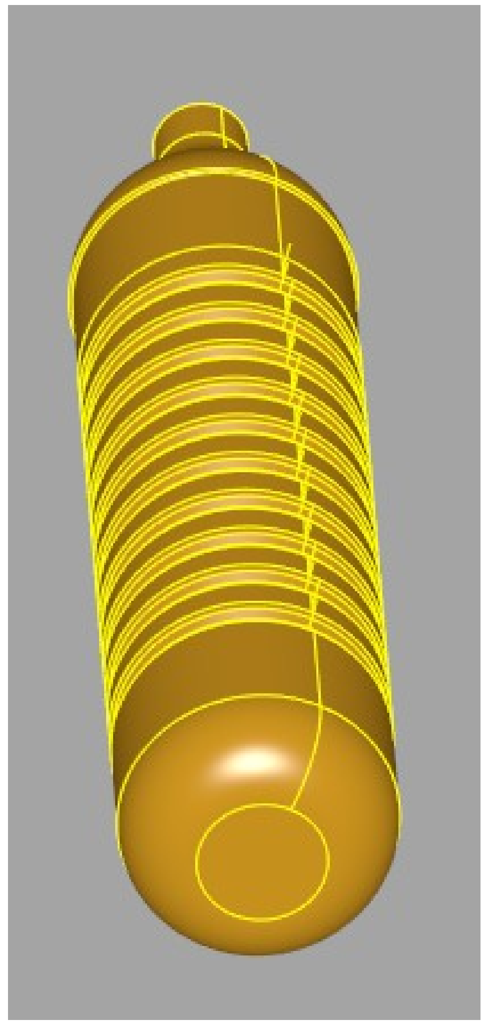
<u>After</u>

The new script is more structured and introduces more parameters, it computes the volume of each part and the total volume of the bottle.

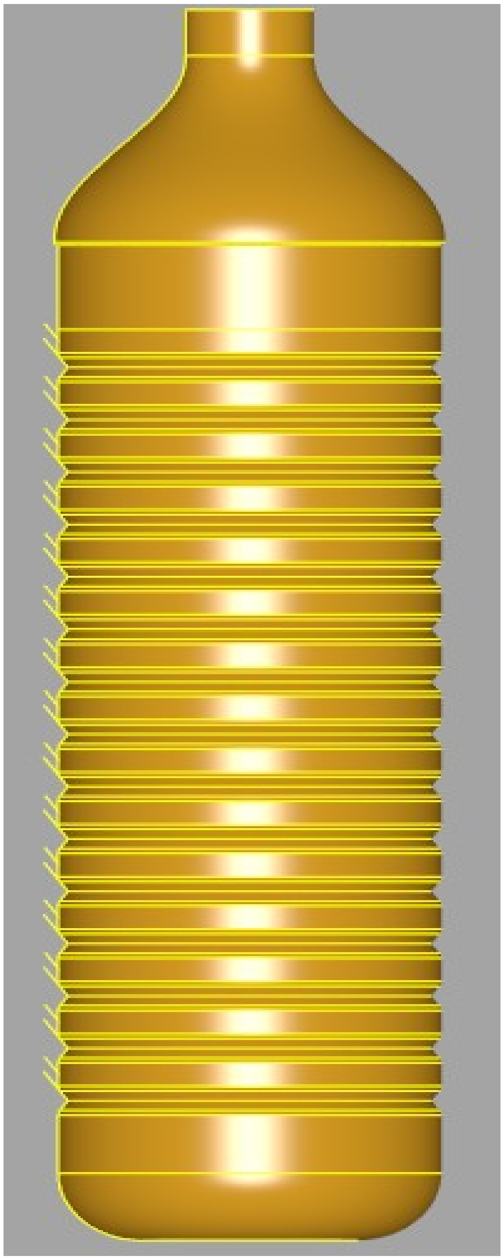
The bottom part of the bottle is now made from 3 parts: Bottom part (a cinlinder) + Middle part (were the rings are located) and the Top part (a cylinder)

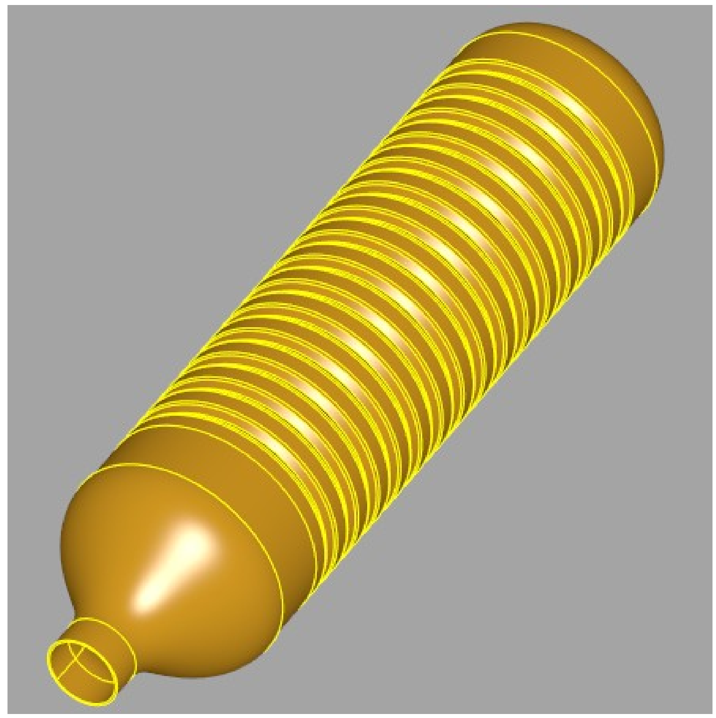
The number of rings is controled by the parameter NRings used in the ‘for’ loop (see the code above) , as a result we get the following shapes after changing the parameters



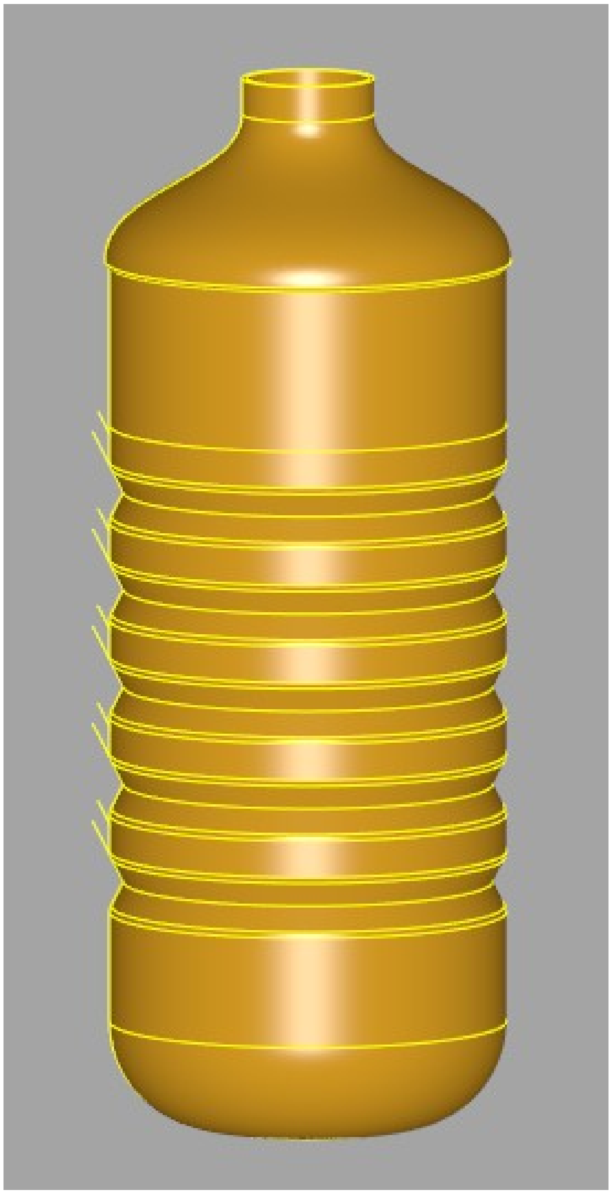


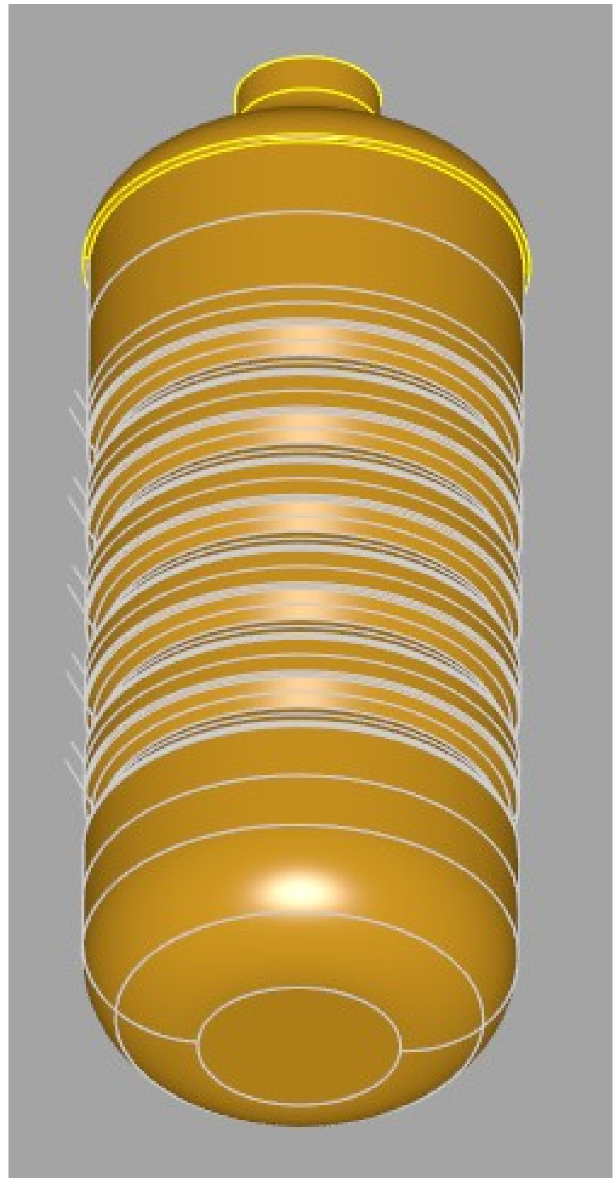
THeight, MHeight, BHeight, radius = 2, 0.5 ,3 ,2 & Rdiff=1/4 & Roffset=1/4 & NRings = 10 & distZ = 0.8 & distX = radius / 3
Bottle_Volume = 153.3162664798769



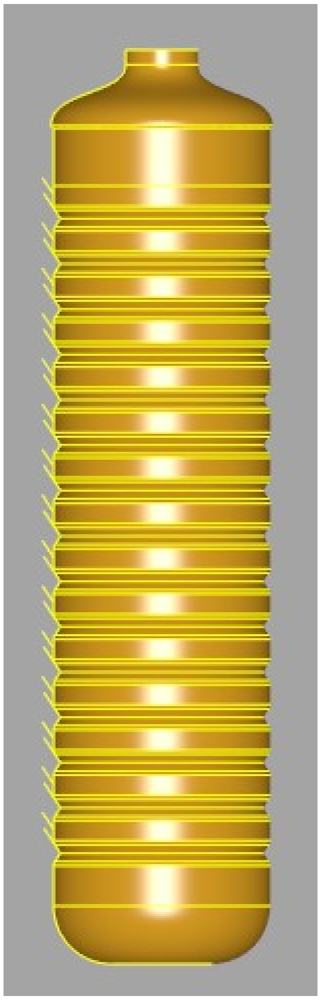


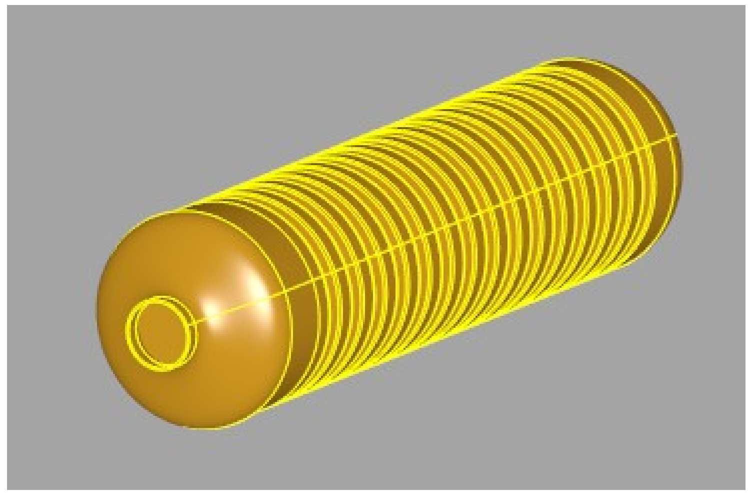
THeight, MHeight, BHeight, radius = 1.5, 0.4 ,2 ,2 & RDiff=1/6 &
Roffset=1/4 & NRings = 15 & distZ = 0.8 & distX = radius / 3
Bottle_Volume = 162.68650290152144





THeight, MHeight, BHeight, radius = 1.8, 0.4 ,2.2 ,2 & RDiff=1/6 &
ROffset=1/3 & NRings = 5 & distZ = 0.4 & distX = radius / 3
Bottle_Volume = 94.6453268262036





THeight, MHeight, BHeight, radius = 1.2, 0.4, 1.8 ,2 & RDiff=1/6 & ROffset=1/4 & NRings = 15 & distZ = 0.3 & distX = radius / 3
Bottle_Volume = 156.40331759434184

Generating Data

Using a for loop we can change the parameters value, as a result we will have a large data set of different bottles, and then export the in STL format using cadquery.export function.

Complexity & Memory

The complexity of the algorithm above is linearly dependent of the number of rings: bottles with a high number of rings take more time to be constructed. It will take around 3 hours to generate 100 bottles

STL files are large in term of memory, due to the high quality and resolution of the mesh