CAD using Termux-Arch Linux

Dinesh

July 2019

1 Introduction

OpenSCAD is not an interactive modeller. Instead it is something like a 3D-compiler that reads in a script file that describes the object and renders the 3D model from this script file. This gives you (the designer) full control over the modelling process and enables you to easily change any step in the modelling process or make designs that are defined by configurable parameters.

OpenSCAD provides two main modelling techniques: First there is constructive solid geometry (aka CSG) and second there is extrusion of 2D outlines. Autocad DXF files can be used as the data exchange format for such 2D outlines. In addition to 2D paths for extrusion it is also possible to read design parameters from DXF files. Besides DXF files OpenSCAD can read and create 3D models in the STL and OFF file formats.

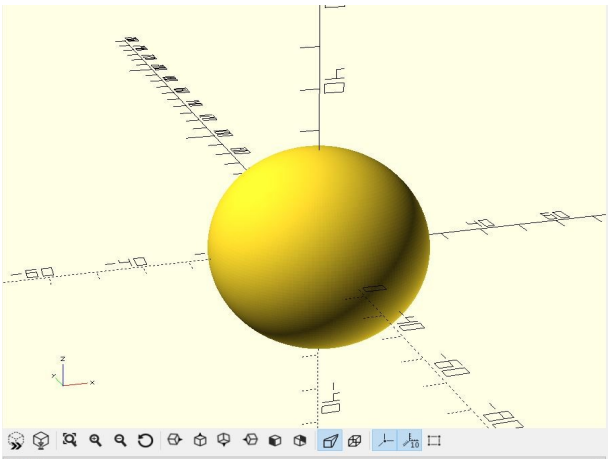
OpenSCAD builds on top of a number of free software libraries; is uses Qt for user interface, CGAL for CSG evaluation, OpenCSG and OpenGL for CSG previews, as well as boost, eigen and glew.

2 Contents

3 Conic Sections

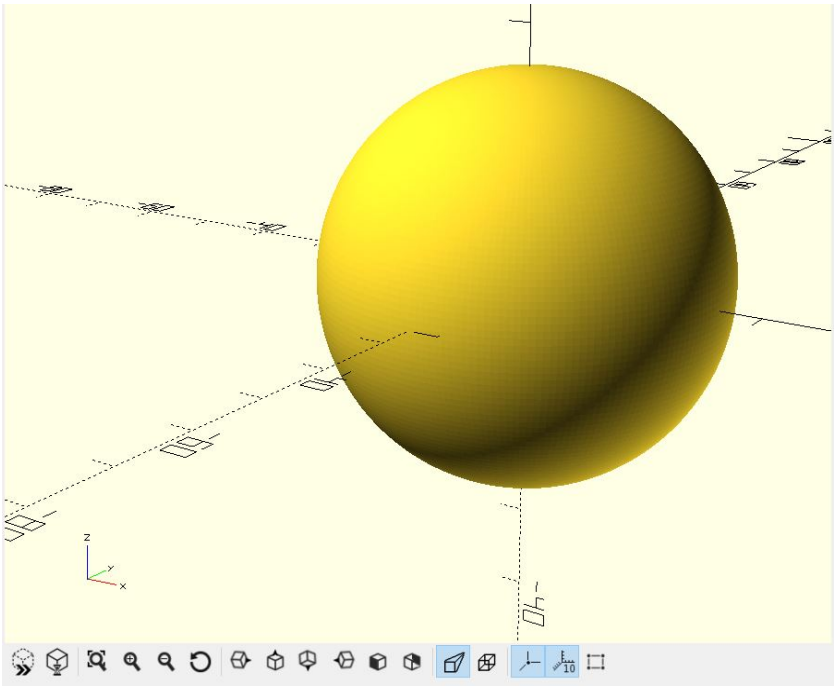
3.1 Sphere

```
translate([0,0,0],$fn=200) {
sphere(r=25);
}
```



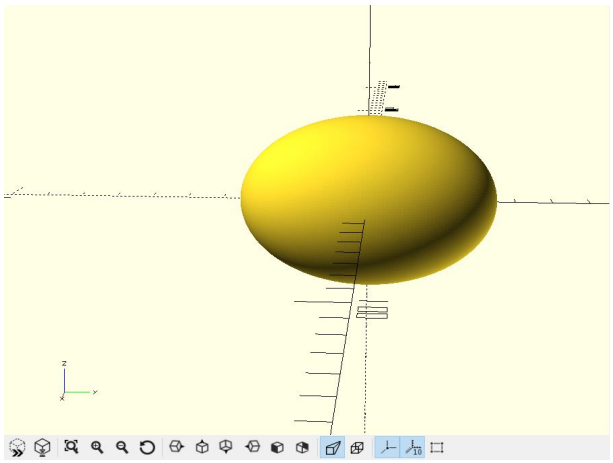
3.2 Hollow Sphere

```
translate([0,0,0],$fn=200) {
difference() { sphere(r=25);
sphere(r=20);
}
}
```



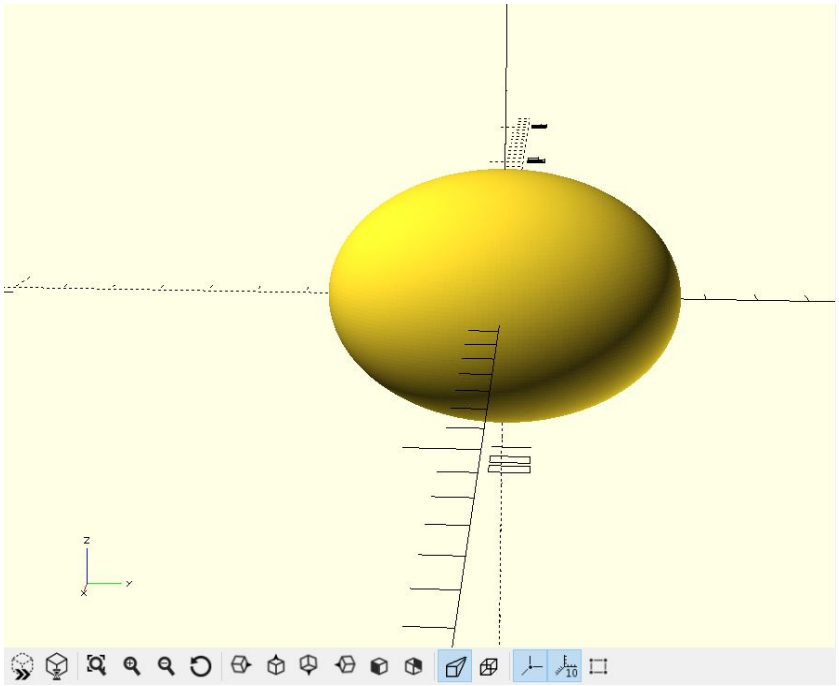
3.3 Ellipsoid

```
translate([0,0,0],$fn=200)
{
resize([50,70,50])
sphere(r=25);
}
```



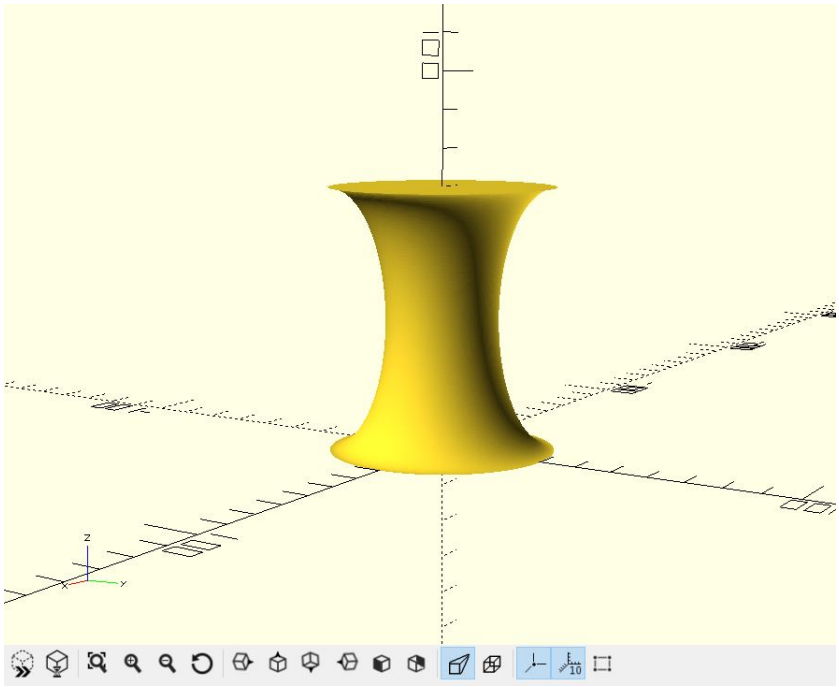
3.4 Hollow Ellipsoid

```
translate([0,0,0],$fn=200)
{
difference(){
resize([50,70,50]) sphere(r=25);
resize([45,65,45]) sphere(r=20);
} }
```



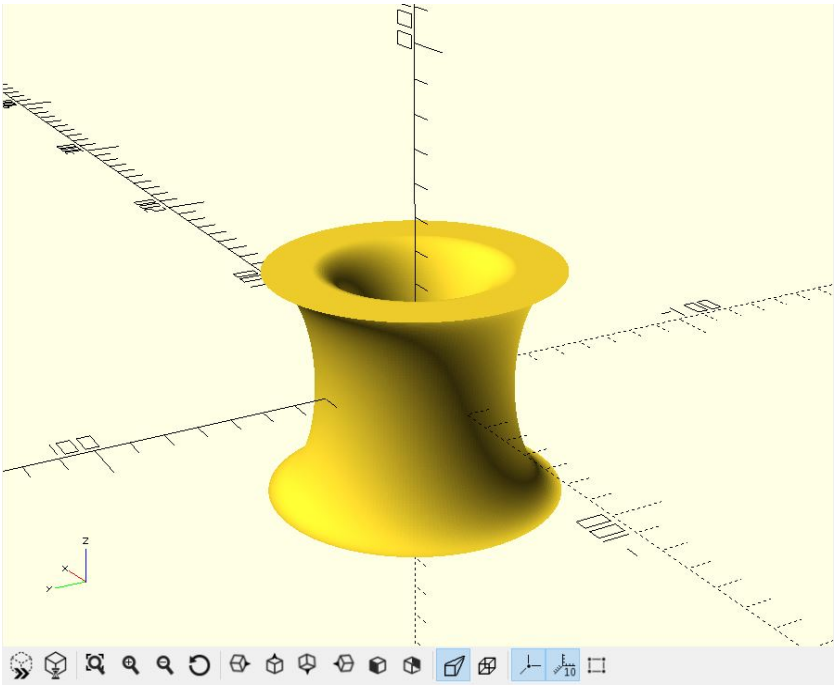
3.5 Hyperboloid

```
translate([0,0,35]){
rotate_extrude($fn = 200){
difference(){
translate([15, 0])
square([30, 70], true);
translate([30, 0])
resize([30, 70])circle(d = 20);
}
}
}
```



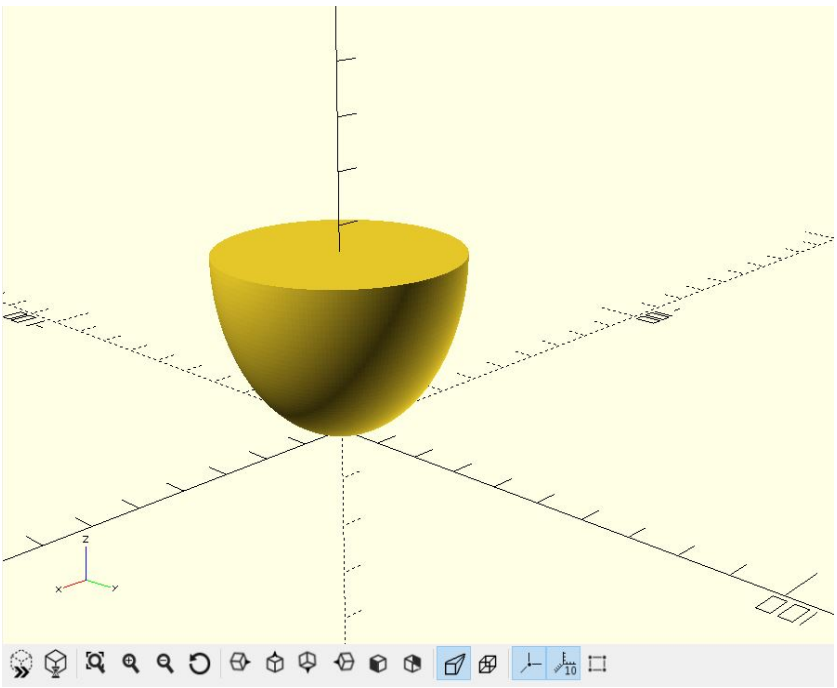
3.6 Hollow Hyperboloid

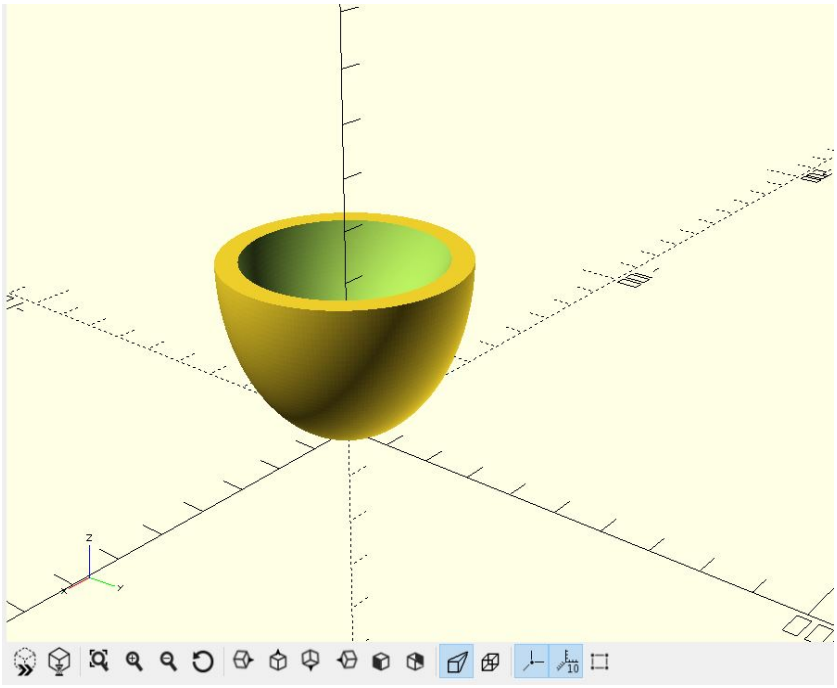
```
rotate_extrude($fn = 200){
difference(){
difference(){
translate([30, 0])
square([30, 70], true);
translate([45, 0])
resize([30, 70])circle(d = 20);
}
difference(){
translate([15, 0])
square([30, 70], true);
translate([30, 0])
resize([30, 70])circle(d = 20);
}
}
}
```



3.7 Paraboloid

```
translate([0,0,35],$fn=200){
rotate_extrude(){
difference(){
difference(){
translate([0, 0, 0])
{
resize([50, 70])
circle(r = 25);
}
translate([−25, 0, 0])
{
resize([50, 35])
square(r = 25);
}
}

translate([-25,-35,0])
{
resize([25,35])
square(r=25);
}
}
}
}
}
```





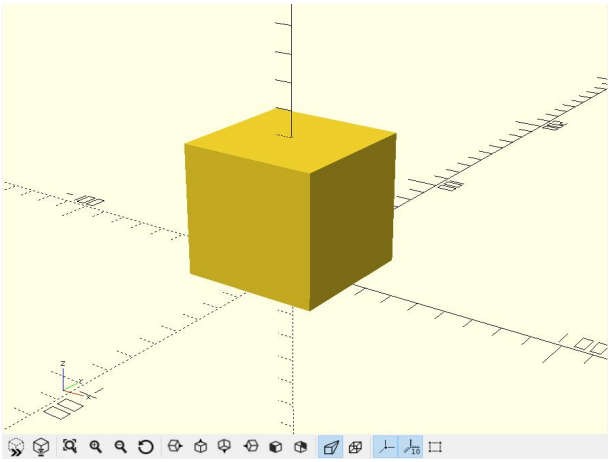
3.8 Hollow Paraboloid

```
difference(){
translate([0,0,35],$fn=200){
rotate_extrude(){
difference(){
difference(){
translate([0, 0, 0])
{
resize([50, 70])
circle(r = 25);
}
translate([−25, 0, 0])
{
resize([50, 35])
square(r = 25);
}
}

translate([-25,-35,0])
{
resize([25,35])
square(r=25);
}
}
}
}

translate([0,0,55],$fn=200){
rotate_extrude(){
difference(){
difference(){
translate([0, 0, 10])
{
resize([50, 70])
circle(r = 25);
}
translate([−25, 0, 10])
{
resize([50, 35])
square(r = 25);
}
}

translate([-25,-35,10])
{
resize([25,35])
square(r=25);
}
}
}
}
}
```

4 Prisms and Pyramids

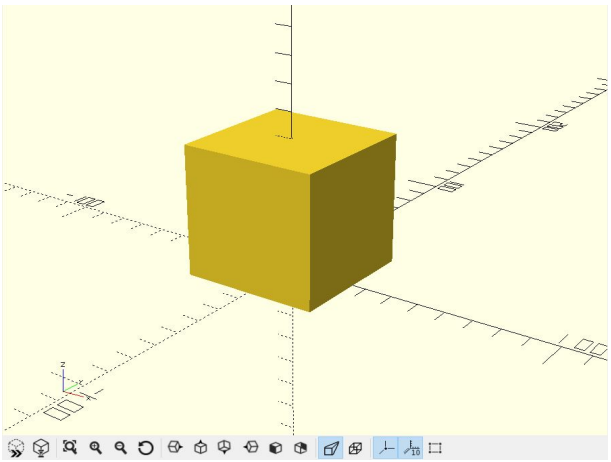
4.1 Cube

```
linear_extrude(height=50){
translate([-25,-25,0]){
square(50);
}
}
```



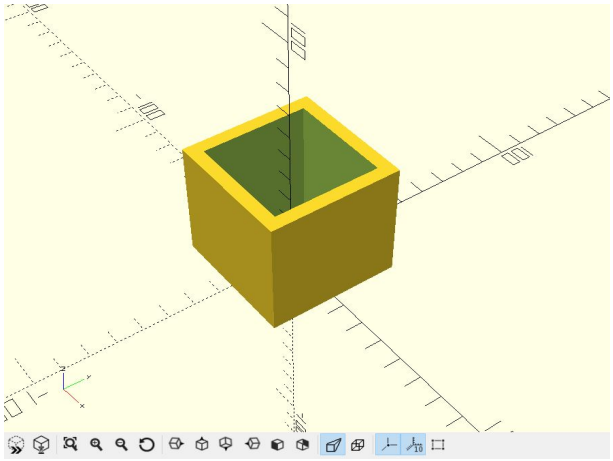
4.2 Hollow Cube

```
difference(){
linear_extrude(height = 50){
translate([-25, -25, 0]){
square(50);
}
}
linear_extrude(height = 40){
translate([-20, -20, 0]){
square(40);
}
}
}
```



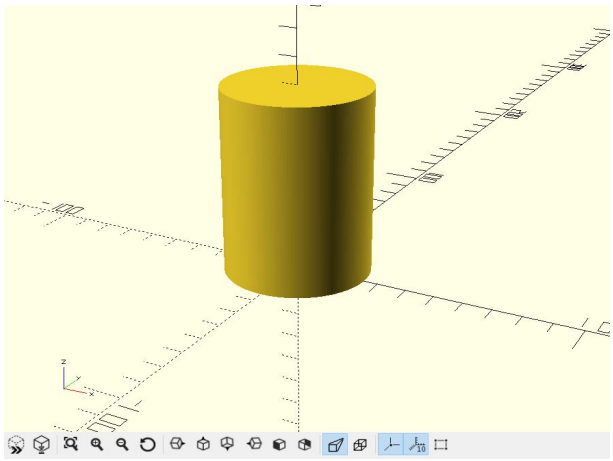
4.3 Box

```
difference(){
linear_extrude(height = 50){
translate([-25, -25, 0]){
square(50);
}
}
linear_extrude(height = 60){
translate([-20, -20, 0]){
square(40);
}
}
}
```



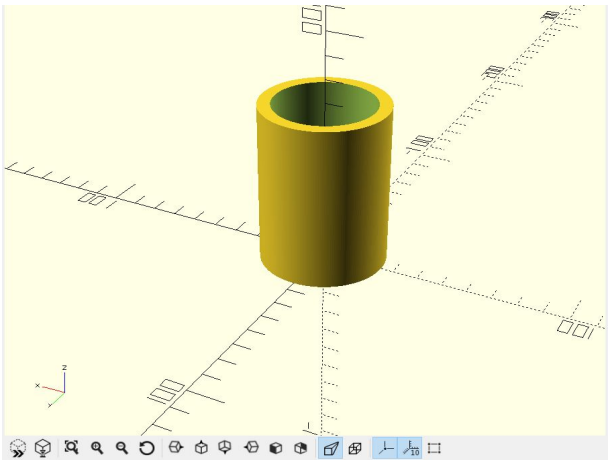
4.4 Cylinder

```
$fn=200;
linear_extrude(height = 70){
translate([0, 0, 0]){
circle(r = 25);
}
}
```



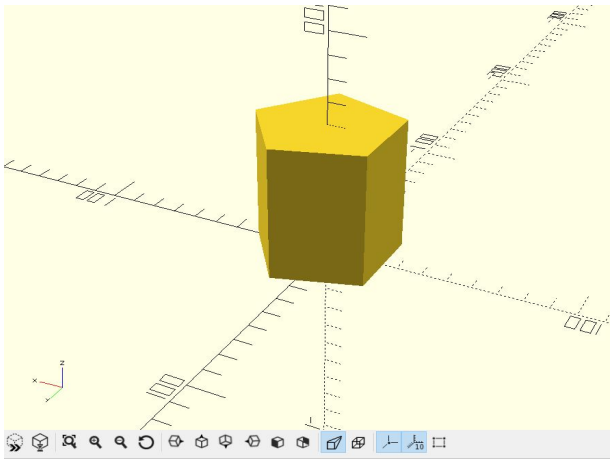
4.5 Hollow Cylinder

```
$fn=200;
difference(){
linear_extrude(height=70){
translate([0,0,0]){
circle(r=25);
}
}
translate([0,0,-10]){
linear_extrude(height = 90){
translate([0, 0, 0]){
circle(r = 20);
}
}
}
}
```



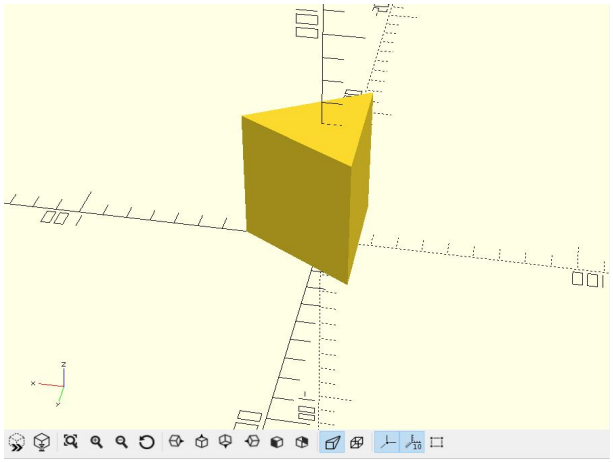
4.6 Pentagonal Prism

```
linear_extrude(height=60){
translate([ 0, 0])
circle(30,$fn=5);
}
```



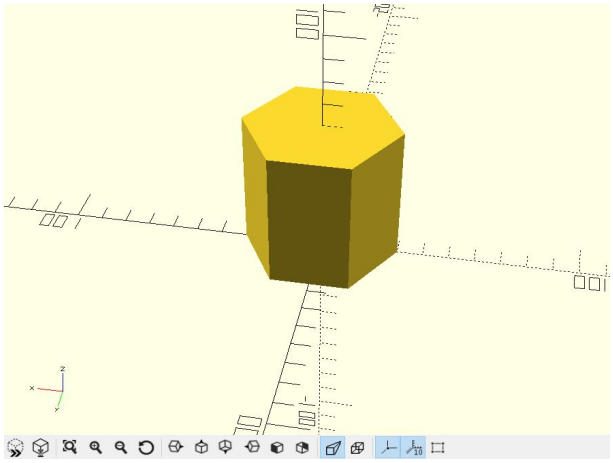
4.7 Triangular Prism

```
linear_extrude(height=60){
translate([ 0, 0])
circle(30,$fn=6);
}
```



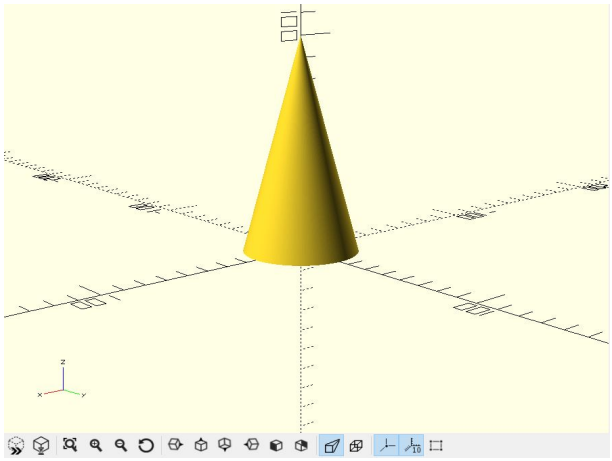
4.8 Hexagonal Prism

```
linear_extrude(height=60){
translate([ 0, 0])
circle(30,$fn=6);
}
```



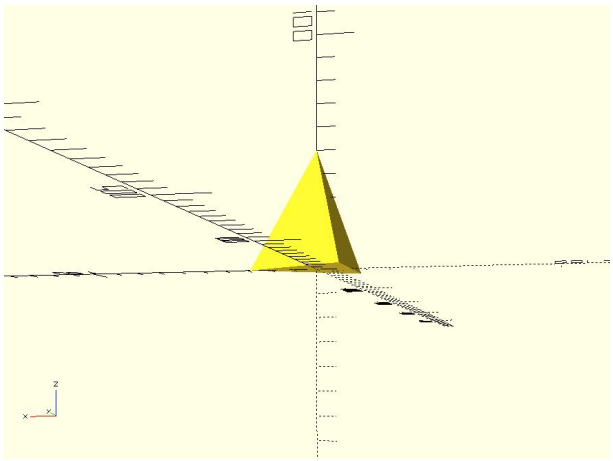
4.9 Cone

```
$fn=200;
cone(0,0,50);
module cone(x=0,y=0,z=0){
translate([x,y,z])
cylinder( r1=25, r2=0, h=100, center=true );
}
```



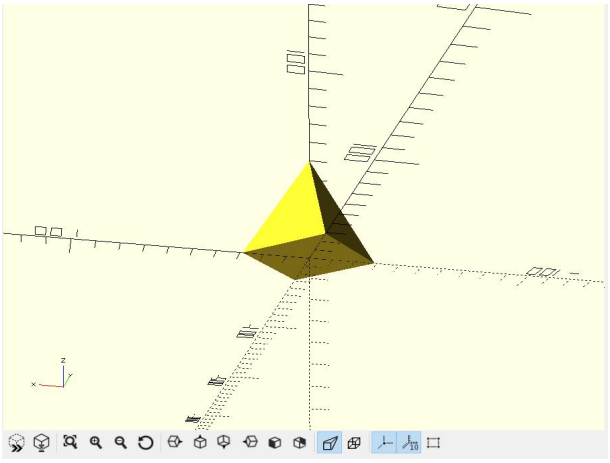
4.10 Tetrahedron

```
r1=50;
h=r1/1.73205;
module tetrahedron(x,y,z){
translate([x,y,z])
cylinder( r1, r2=0, h, center=true );
}
translate([0,0,2.6*h/3]){
$fn=3;
tetrahedron(0,0,0);
}
```



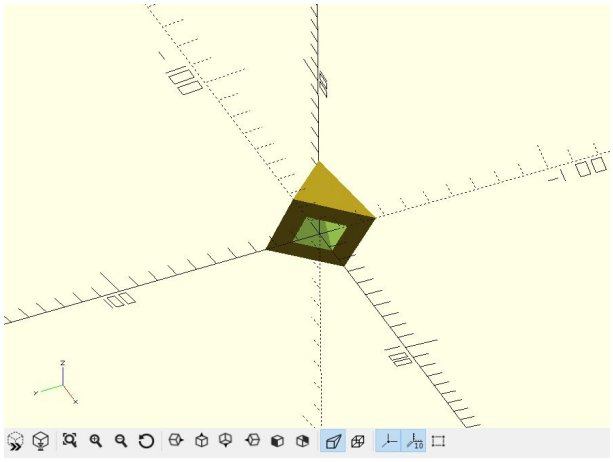
4.11 Square Pyramids

```
r1=50;
h=r1/1.73205;
module tetrahedron(x,y,z){
translate([x,y,z])
cylinder( r1, r2=0, h, center=true );
}
translate([0,0,2.6*h/3]){
$fn=4;
tetrahedron(0,0,0);
}
```



4.12 Hollow Sqaure Pyramids

```
r1=50;
h1=r1/1.73205;
module sq1(x,y,z){
translate([x,y,z])
cylinder( r1, r2=0, h1, center=true );
}

r1=45;
h=r1/1.73205;
module sq2(x,y,z){
translate([x,y,z])
cylinder( r1, r2=0, h, center=true );
}

difference(){
translate([0,0,2.6*h1/3]){
$fn=4;
sq1(0,0,0);
}

translate([0,0,2.6*h/3]){
$fn=4;

sq2(0,0,0);
}

}
```



4.13 Octahedron

```
r1=50;
r2=0;

h1=43.3012;

z1=1.7*h1/3;
module downright(x,y,z){
translate([x,y,z])
cylinder( r1, r2, h1, center=true );
}

r1=50;
h=43.3012;
z=1.7*h/3;
module upright(x,y,z){
translate([x,y,z])
cylinder( r1, r2=0, h, center=true );
}

union(){
$fn=4;
downright(0,0,-z1);

$fn=4;
upright(0,0,z);

}
```