Nanyang Technological University
School of Computer Science and Engineering

Laboratory Report

CZ2003
Computer Graphics and Visualization

Lab 3
Parametric Surfaces & Solids

By
Teo Wei Jie (U1822263C), SS2

18 April 2020

Contents

1 3D Plane

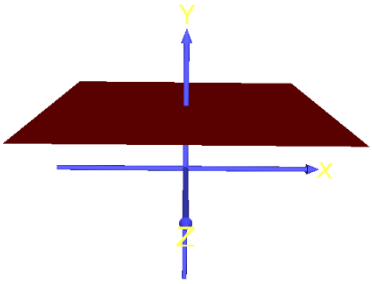

Figure 1

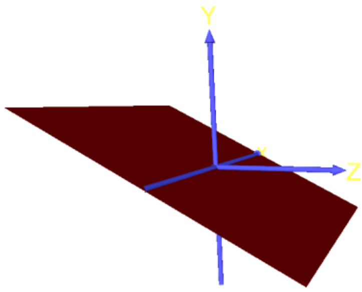

Figure 2

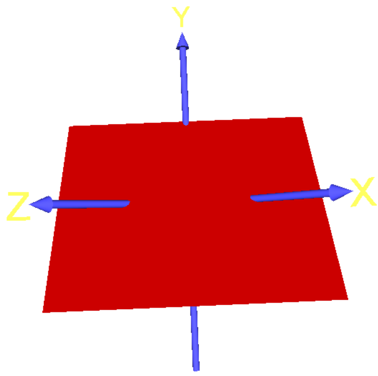

Figure 3

Figure 1 is defined in "3DPlane_1.wrl", Figure 2 is defined in "3DPlane_2.wrl" and Figure 3 is defined in "3DPlane_3.wrl".

The planes are defined parametrically using three points: P1, P2 and P3 with reference to the provided "Plane parametrically by three points.wrl" file. The parameters u and v are in the domain of [0, 1].

Figure 1 depicts a plane lying flat on the X and Z axes, elevated to Y = 0.5. It is defined using the points P1 = (-1, 0.5, -1), P2 = (1, 0.5, -1) and P3 = (-1, 0.5, 1). It is an attempt to create a basic plane to start out the lab without blatantly using the reference code provided.

Figure 2 depicts the plane in Figure 1 with a tilt of 45 degrees around the X axis. It is defined using the points P1 = (-1, 0.5, -1), P2 = (1, 0.5, -1) and P3 = (-1, -0.5, 1). It is an attempt to tilt a plane around an axis by modifying P1, P2 and P3.

Figure 3 depicts a plane tilted at 45 degrees around the X and Y axes, with the tip of the plane lying on the Y axis. It is defined using the points P1 = (-0.5, 0.5, 0.5), P2 = (0.5, 0.5, -0.5) and P3 = (0, -0.5, 1). It is an attempt to experiment with different values of P1, P2 and P3 to create a rectangle with varying tilt and elevation.

All three planes are defined with a resolution of 1 for both parameters u and v as the vertices require only one sampling point for a plane. Increasing the resolution to values like 100 or 1000 yield no difference in the surface rendered, except that rendering with 1000 sampling points resulted in a very low framerate when attempting to rotate the plane in Figures 1 and 2, and the browser crashing after rendering the plane in Figure 3.

2 3D Triangle

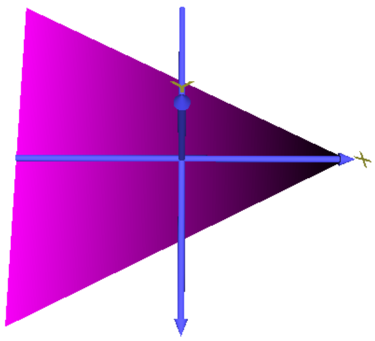

Figure 4

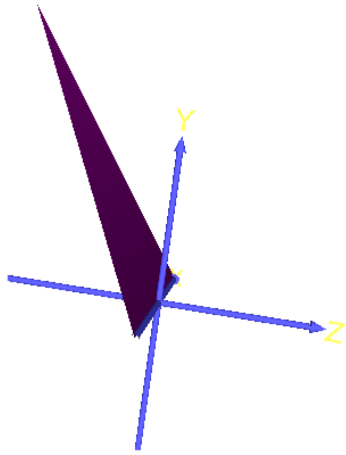

Figure 5

Figure 4 is defined in “3DTriangle_1.wrl” and Figure 5 is defined in “3DTriangle_2.wrl”.

The triangles are defined as planes parametrically using four points: P1, P2, P3 and P4, with P3 and P4 being the same point to form a triangle. The parameters u and v are in the domain of [0, 1].

Similar to the planes defined in Figures 1, 2 and 3, these triangles are defined through attempts at getting a hang of defining 3D surfaces in VRML.

Figure 4 depicts a triangle lying flat on the X and Z axes with a base and height of 2, defined using the points P1 = (-1, 0, -1), P2 = (-1, 0, 1) and P3 = P4 = (1, 0, 0).

Figure 5 depicts the same triangle “standing” on the X axis and tilted about the X axis at 60 degrees, defined using the points P1 = (-1, 0, 0), P2 = (1, 0, 0) and P3 = P4 = (0, 1.732, -1).

As the triangles are defined as planes, they are defined with a resolution of 1 for both parameters u and v. Similarly, increasing the resolution yield no difference in the surface rendered.

3 Bilinear Surface

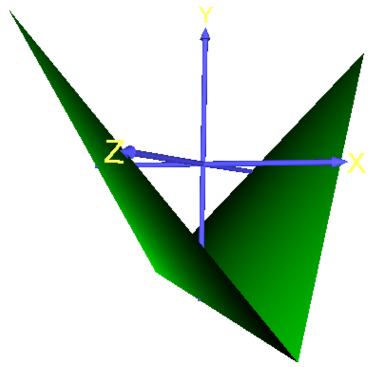

Figure 6 – Resolution 1

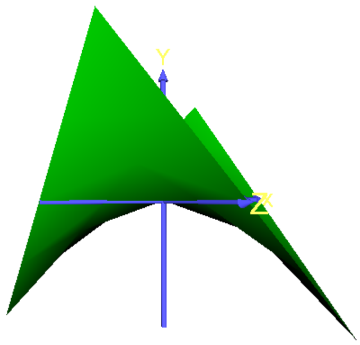

Figure 7 – Resolution 4

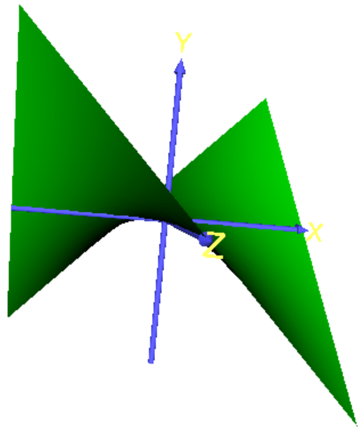

Figure 8 – Resolution 10

Figure 6 is defined in “Bilinear_1_1.wrl”, Figure 7 is defined in “Bilinear_1_4.wrl” and Figure 8 is defined in “Bilinear_1_10.wrl”.

The surfaces are defined parametrically using four points: P1 = (-1, -1, -1), P2 = (1, 1, -1), P3 = (-1, 1, 1) and P4 = (1, -1, 1). All three surfaces are defined using the same points but with varying resolution for parameters u and v. Both parameters are in the domain of [0, 1].

Compared to planes defined in the previous sections, the resolution affects the rendering of the bilinear surface. With only one sampling point, the surface in Figure 6 is simply a plane that is bent in half. Increasing the resolution to 4, the surface in Figure 7 starts bending in curves albeit with rough edges seen on the bent surfaces. Finally, with enough sampling points, the surface in Figure 10 bends smoothly as the resolution is increased to 10 to form a smooth surface.

4 Sphere

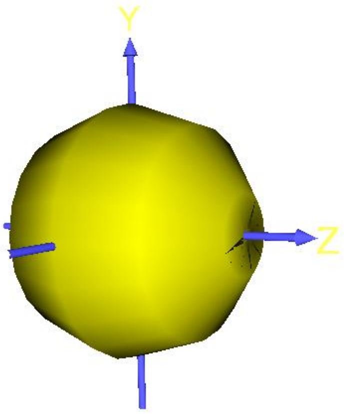

Figure 9 – Resolution 10

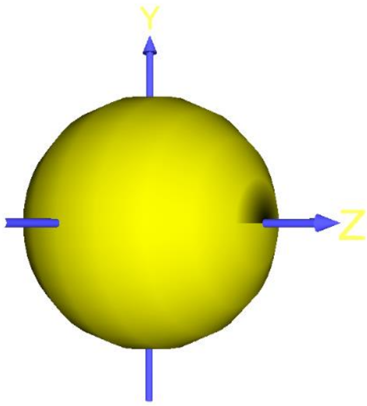

Figure 10 – Resolution 20

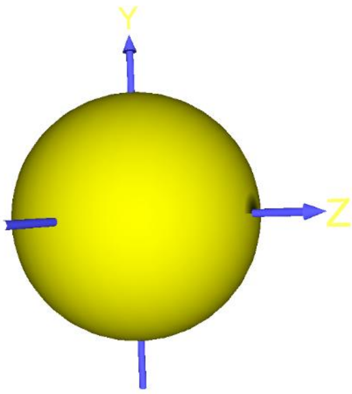

Figure 11 – Resolution 40

Figure 9 is defined in “Sphere_10.wrl”, Figure 10 is defined in “Sphere_20.wrl” and Figure 11 is defined in “Sphere_40.wrl”.

The spheres have a radius of 0.7 and are defined parametrically using the following formulas with parameters u = [0, 2] and v = [0, 1]:

- x = 0.7 * cos(pi * u) * cos(pi * v)
- y = 0.7 * cos(pi * u) * sin(pi * v)
- z = 0.7 * sin(pi * u)

Comparing Figures 9, 10 and 11, Figure 9 illustrates a sphere rendered with 10 sampling points for both parameters u and v. Due to having low amount of sampling points, when the sampled vertices are connected, the individual surfaces are large and hence the sphere rendered is jagged and not smooth.

When the resolution is increased, as illustrated in Figures 10 and 11, it smoothens the edge of the sphere. Increasing the resolution to 20 results in a sphere rendered with lesser jagged edges and increasing the resolution to 40 results in a smooth sphere rendered.

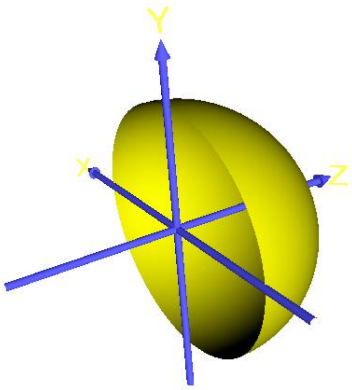

Figure 12

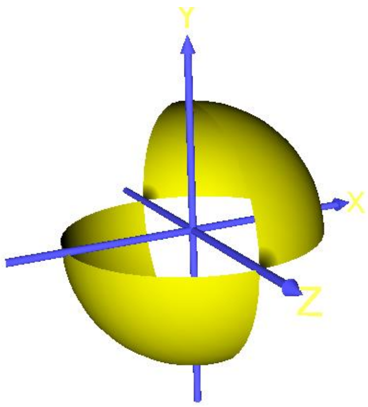

Figure 13

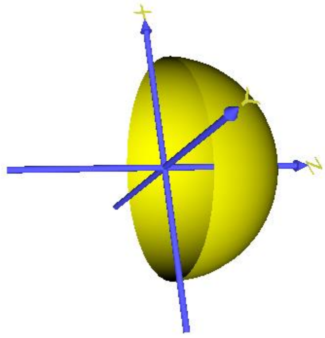

Figure 14

Figure 12 is defined in “Sphere2.wrl”, Figure 13 is defined in “Sphere3.wrl” and Figure 14 is defined in “Sphere4.wrl”.

The spheres are defined parametrically using the same formulas used for Figures 9, 10 and 11. These spheres are defined as an experiment on the effect of varying the parameter domain for u and v.

Figure 12 illustrates a sphere defined with the parameter domain u = [0, 1] and v = [0, 1], while Figure 13 illustrates a sphere defined with the parameter domain u = [0, 2] and v = [0, 2]. Both parameters determine the “completeness” of the spheres, but each parameter determines how the sphere is rendered differently. However, it can be observed that cutting either of the parameters by half reduces the surface rendered by half as well.

Figure 14 illustrates a sphere defined with the parameter domain u = [0, 1] and v = [0, 2]. Overflowing the parameter v does not seem to have any effect on the rendering of the sphere as compared to Figure 12 where v is the only variable between the two domains defined.

Note 1: The formula is generalised, and the parameters are varied such that the surface is modified as a whole when the parameter changes. For example, changing the domain of u from [0, 2] to [0, 1], the resulting surface is still a sphere as opposed to changing 2 * pi * u to pi * u **for just x**, where u is now = [0, 1] and the resulting surface would be a funny shape. This also saves time having to modify three variables across each VRML files for experiments on the same surface. The same justification is applied to every other solid and surface in this report.

5 Ellipsoid

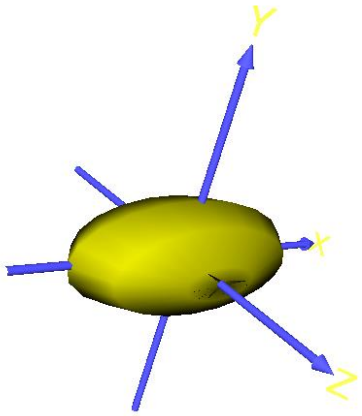

Figure 15 – Resolution 10

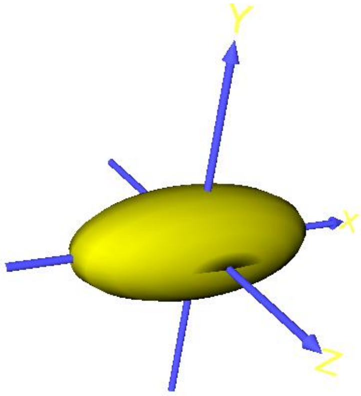

Figure 16– Resolution 20

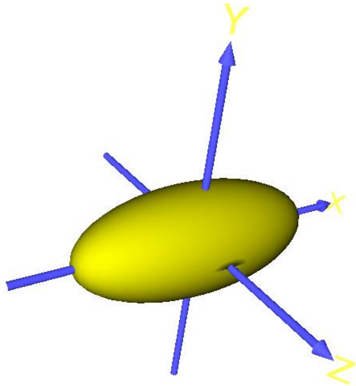

Figure 17 – Resolution 40

Figure 15 is defined in “Ellipsoid_10.wrl”, Figure 16 is defined in “Ellipsoid_20.wrl” and Figure 17 is defined in “Ellipsoid_40.wrl”.

The ellipses have a radius of 0.7 along the X axis and a radius of 0.3 along the Y and Z axes. The surfaces are defined parametrically using the following formulas similar to that used for spheres, where the Y and Z axes are squished with parameters u = [0, 2] and v = [0, 1]:

- x = 0.7 * cos(pi * u) * cos(pi * v)
- y = 0.3 * cos(pi * u) * sin(pi * v)
- z = 0.3 * sin(pi * u)

The observations made when comparing the surfaces in Figures 15, 16 and 17 are similar to that in Figures 9, 10 and 11 with respect to the resolution used for each surface.

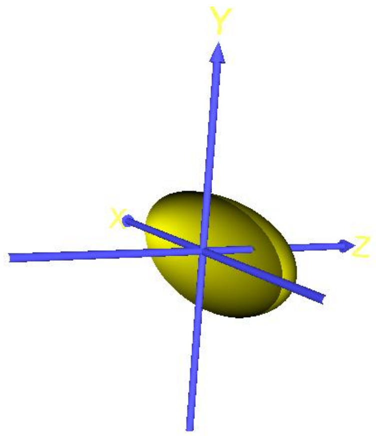

Figure 18

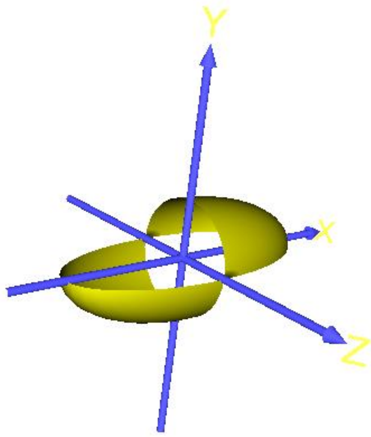

Figure 19

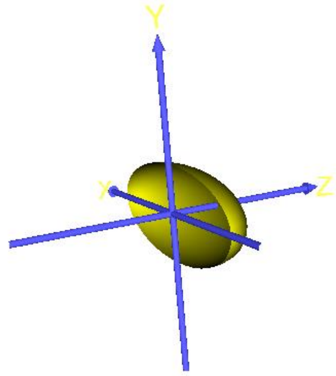

Figure 20

Figure 18 is defined in “Ellipsoid2.wrl”, Figure 19 is defined in “Ellipsoid3.wrl” and Figure 20 is defined in “Ellipsoid4.wrl”.

The ellipsoids are defined parametrically using the same formulas used for Figures 15, 16 and 17. Similarly, these ellipsoids are defined as an experiment on the effect of varying the parameter domain for u and v.

The observations made for each surface is similar to that in Figures 12, 13 and 14 with respect to the parameter domains defined for each surface.

6 Cone

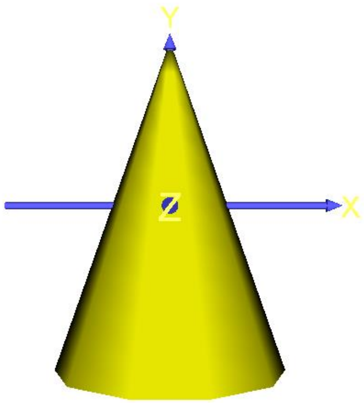

Figure 21 – Resolution 10

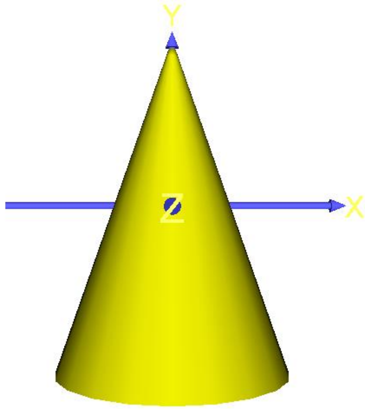

Figure 22 – Resolution 20

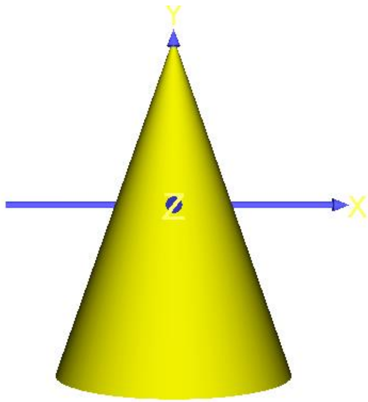

Figure 23 – Resolution 40

Figure 21 is defined in "Cone_10.wrl", Figure 22 is defined in "Cone_20.wrl" and Figure 23 is defined in "Cone_40.wrl".

The cones have a radius of 0.7 at the base and a height of 2. The surfaces are defined parametrically using the following formulas with parameters u = [0, 2] and v = [0, 2]:

- x = (0.7 - 0.7 * u / 2) * cos(pi * v)
- y = u - 1
- z = (0.7 - 0.7 * u / 2) * sin(pi * v)

The observations made when comparing the surfaces in Figures 21, 22 and 23 are similar to that in Figures 9, 10 and 11 with respect to the resolution used for each surface.

7 Solid Box

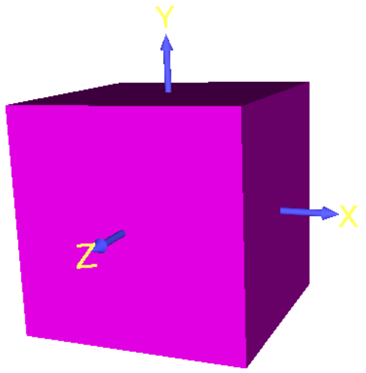

Figure 24

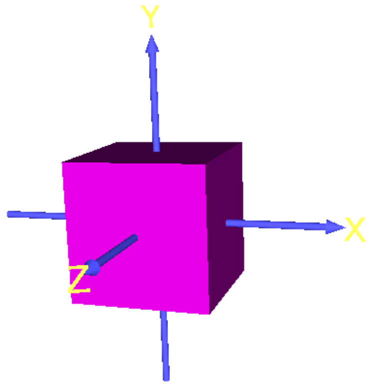

Figure 25

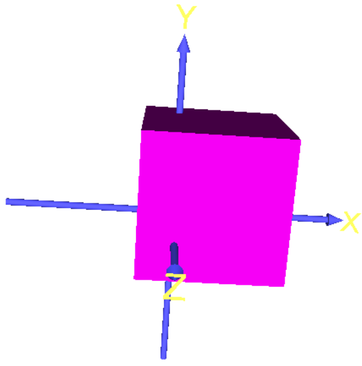

Figure 26

Figure 24 is defined in "Solid_Box_1.wrl", Figure 25 is defined in "Solid_Box_2.wrl" and Figure 26 is defined in "Solid_Box_3.wrl".

The boxes are defined parametrically using the following formulas with varying domains for parameters u, v, and w:

- x = u
- y = v
- z = w

Figure 24 illustrates a box defined with the parameters u = v = w = [-0.7, 0.7]. The resulting box has a length of 1.4 across each edge, centred at the origin.

Figure 25 illustrates a box defined with the parameters u = v = w = [-0.4, 0.4]. The resulting box has a length of 0.8 across each edge, centred at the origin.

Figure 26 illustrates the same box as Figure 25 but defined with the parameters u = v = w = [-0.2, 0.6]. The resulting box has a length of 0.8 across each edge but is instead centred at x = y = z = 0.2.

The parameters are the same to ensure every edge of the box are of equal length.

8 Solid Sphere

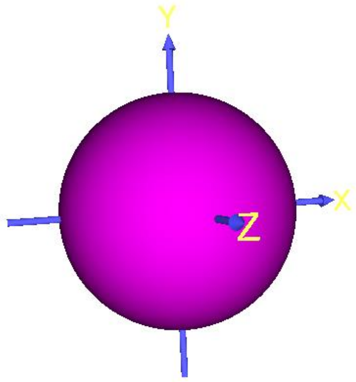

Figure 27

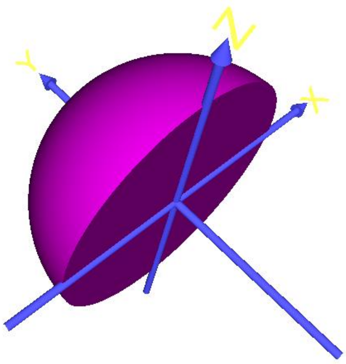

Figure 28

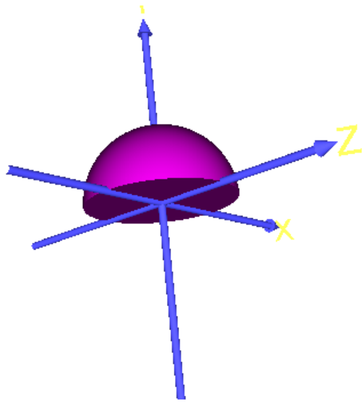

Figure 29

Figure 27 is defined in “Solid_Sphere_1.wrl”, Figure 28 is defined in “Solid_ Sphere _2.wrl” and Figure 29 is defined in “Solid_ Sphere _3.wrl”.

The spheres are defined parametrically using the following formulas with varying domains for parameters u, v, and w:

- x = w * cos(pi * u) * sin(pi * v)
- y = w * sin(pi * u)
- z = w * cos(pi * u) * cos(pi * v)

Figure 27 illustrates a sphere defined with the parameters u = [0, 2], v = [0, 2] and w = [0, 0.7]. The resulting sphere is a complete sphere as the angles scale from 0 to 2pi with radius of 0.7.

Figure 28 illustrates a sphere defined with the parameters u = [0, 1], v = [0, 2] and w = [0, 0.7]. The resulting sphere is a half sphere as the angle with parameter u scales from 0 to pi only with radius of 0.7.

Figure 29 illustrates a sphere defined with the parameters u = [0, 1], v = [0, 1] and w = [0, 0.4]. The resulting sphere is a still a half sphere despite the decrement in domain for parameter v with radius of 0.4.

It can be observed that the parameter u defines the completeness of the sphere and w defines the radius of the sphere.

9 Solid Cylinder

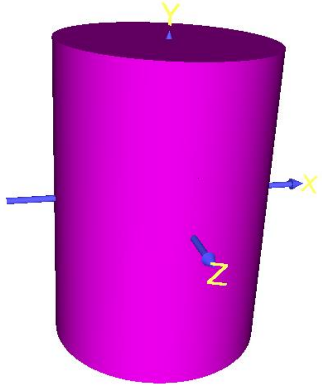

Figure 30

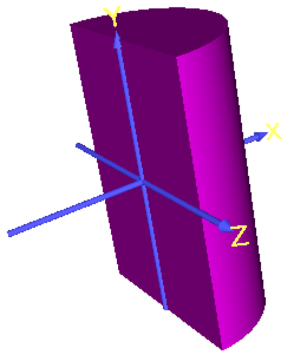

Figure 31

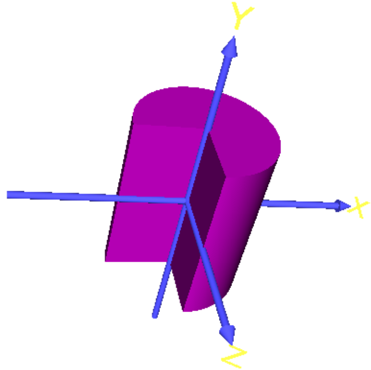

Figure 32

Figure 30 is defined in “Solid_Cylinder_1.wrl”, Figure 31 is defined in “Solid_Cylinder_2.wrl” and Figure 32 is defined in “Solid_Cylinder_3.wrl”.

The cylinders are defined parametrically using the following formulas with varying domains for parameters u, v, and w:

- x = u * sin(pi * v)
- y = w
- z = u * cos(pi * v)

Figure 30 illustrates a cylinder defined with the parameters u = [0, 0.7], v = [0, 2] and w = [-1, 1]. The resulting cylinder is a complete cylinder as the angle scale from 0 to 2pi with radius of 0.7 and height of 2, from y = -1 to y = 1.

Figure 31 illustrates a cylinder defined with the parameters u = [0, 0.7], v = [0, 1] and w = [-1, 1]. The resulting cylinder is a half cylinder as the angle scales from 0 to pi only with radius of 0.7 and height of 2, from y = -1 to y = 1.

Figure 32 illustrates a cylinder defined with the parameters u = [0, 0.4], v = [0, 1.5] and w = [-0.5, 0.5]. The resulting cylinder is a ¾ cylinder as the angle scales from 0 to 1.5pi with radius of 0.4 and height of 1, from -0.5 to 0.5.

It can be observed that the parameter u defines the radius, v defines the completeness of the cylinder and w defines the height of the cylinder.

10 Solid Cone

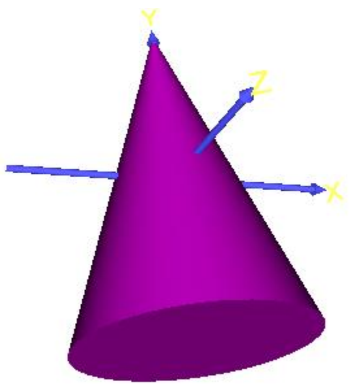

Figure 33

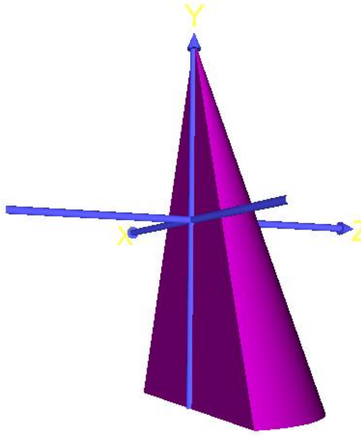

Figure 34

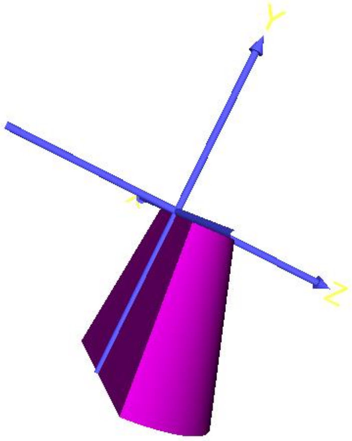

Figure 35

Figure 33 is defined in "Solid_Cone_1.wrl", Figure 34 is defined in "Solid_Cone_2.wrl" and Figure 35 is defined in "Solid_Cone_3.wrl".

The cones are defined parametrically using the following formulas with varying domains for parameters u, v, and w:

- x = u * (1 - w) * cos(pi * v)
- y = 2 * w - 1
- z = u * (1 - w) * sin(pi * v)

Figure 33 illustrates a cone defined with the parameters u = [0, 0.7], v = [0, 2] and w = [0, 1]. The resulting cylinder is a complete cylinder as the angle scale from 0 to 2pi with radius of 0.7 and height of 2, as the height is multiplied by 2 in y, from y = -1 to y = 1.

Figure 34 illustrates a cylinder defined with the parameters u = [0, 0.7], v = [0, 1] and w = [0, 1]. The resulting cylinder is a half cylinder as the angle scales from 0 to pi only with radius of 0.7 and height of 2, from y = -1 to y = 1.

Figure 35 illustrates a "cylinder" defined with the parameters u = [0, 0.7], v = [0, 1] and w = [0, 0.5]. The resulting "cylinder" is a ½ cylinder as the angle scales from 0 to pi with radius of 0.7. As w is halved, the resulting "cylinder" is also cut by half along the Y axis, with the pointy part of the cylinder not rendered.

It can be observed that the parameter u defines the radius, and v and w defines the completeness of the cylinder along the X and Y axis respectively.

An exception to the justification mentioned in note 1 is that the height is explicitly specified in the formula instead of modifying the parameter w accordingly as modifying the parameter affects the radius of the cone as well.

11 Converting Closed Surface to Solid

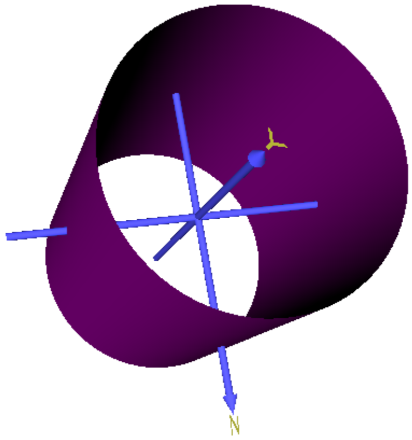

Figure 36 – Cylinder surface

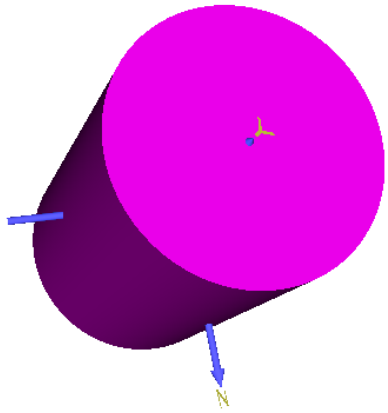

Figure 37 – Cylinder solid derived from Figure 36

Figure 36 is defined in "Cylinder_Surface.wrl" and Figure 37 is defined in "Cylinder_Solid.wrl". The surface is parametrically defined using the following formulas with parameters u = [0, 2] and v = [0, 2]:

- x = 0.7 * sin(pi * v)
- y = u - 1
- z = 0.7 * cos(pi * v)

The surface may be converted into a solid, as illustrated in Figure 37, by introducing a third parameter, w, and the formula is modified to:

- x = w * sin(pi * v)
- y = u - 1
- z = w * cos(pi * v)

where u = [0, 2], v = [0, 2] and w = [0, 0.7].

The surface is now converted into a solid by replacing the radius constant with the new parameter. Note that the definition for the cylinder in Figure 37 is slightly different than that in Figure 30.

12 Translational & Rotational Sweeping

Sweeping Method	**Surface / Solid**	**File**
Rotational	Sphere Surface	Sweep_Sphere.wrl
Rotational	Sphere Solid	Sweep_Sphere_Solid.wrl
Translational	Cylinder Surface	Sweep_Cylinder.wrl
Translational	Cylinder Solid	Sweep_Cylinder_Solid.wrl
Both	Helix Solid	Sweep_Helix.wrl
Both	Sine Curve Solid	Sweep_sinx.wrl (attempted but seems wrong)
Both	Sine Curve Solid	Sweep_sinx_2.wrl (attempted but seems wrong)

Table 1 – Files with defined translational & rotational sweeping

No snapshot is provided in this section as it does not accurately capture the sweeping motion of surfaces and solids. Please run the VRML files to see the sweeping in action.

The formulas used for sweeping are adapted from the surfaces and solids illustrated in the previous sections and labs.