Mould your imagination

6 Sweep Representations

Department of Computer Science and Software Engineering
University of Canterbury, New Zealand.

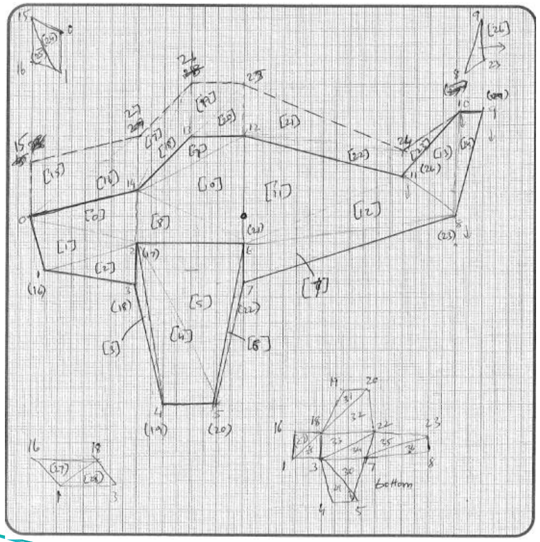
R. Mukundan (mukundan@canterbury.ac.nz)

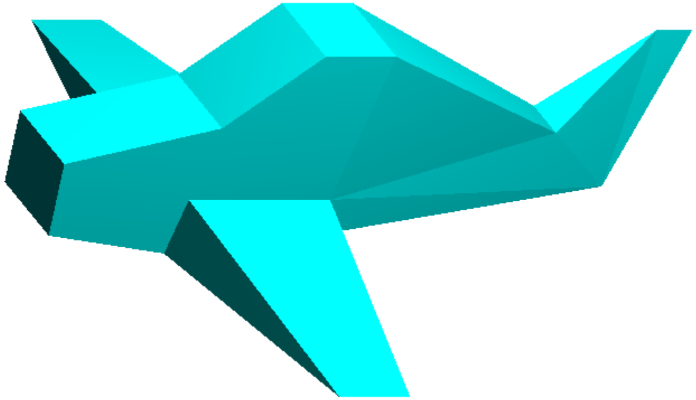
Lecture Outline

- Modelling of simple polygonal meshes using vertices and polygonal definitions
  - Eg. Lab02: Octahedron
- Modelling sweep surfaces
  - Extruded shapes
  - Surfaces of revolution
- Quad strips and triangle strips
- Parametric surfaces

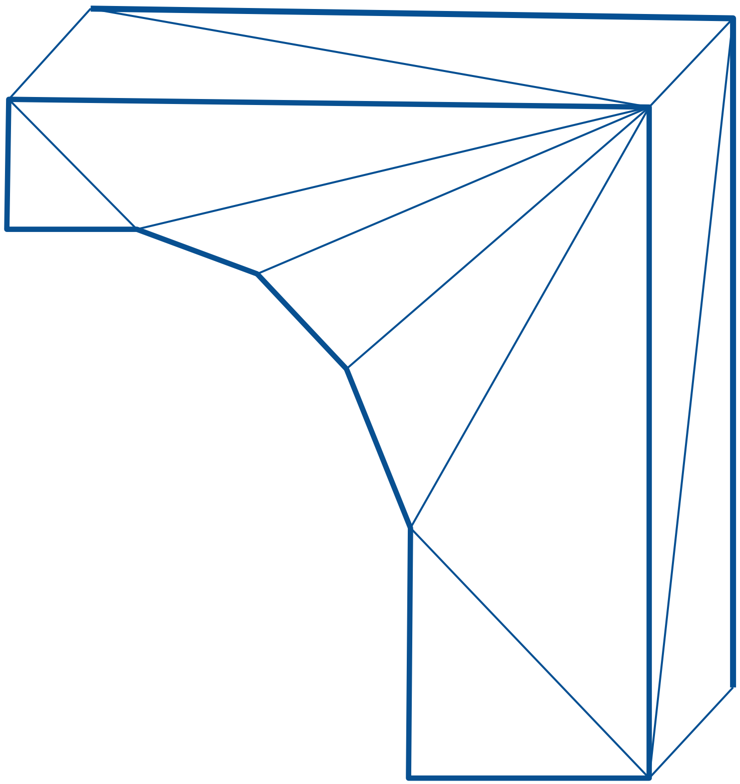
Modelling of Simple Objects

- Create a simple sketch on a graph paper.
- With the centre point as the origin, get the *x*, *y* coordinates.
- The *z* coordinate is the out-of-plane distance (depth)
- Tessellate the object into a set of triangles
- Create the triangle definitions using an anti-clockwise ordering of vertex indices.

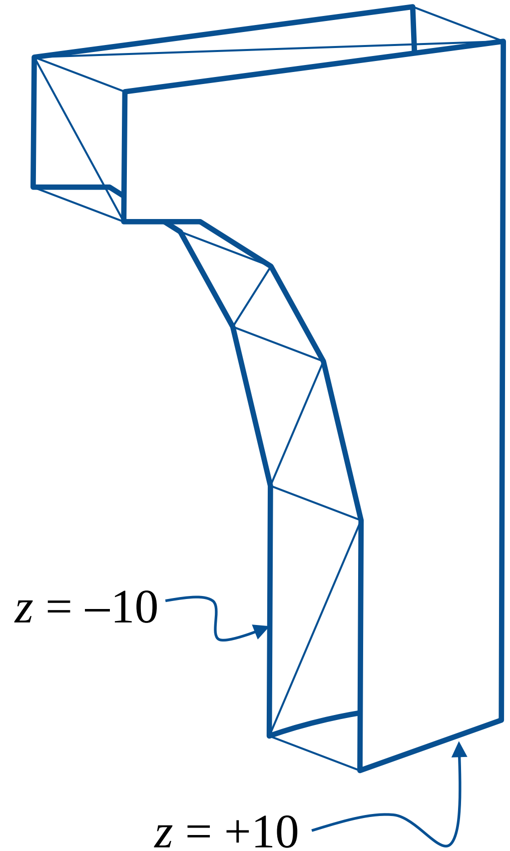




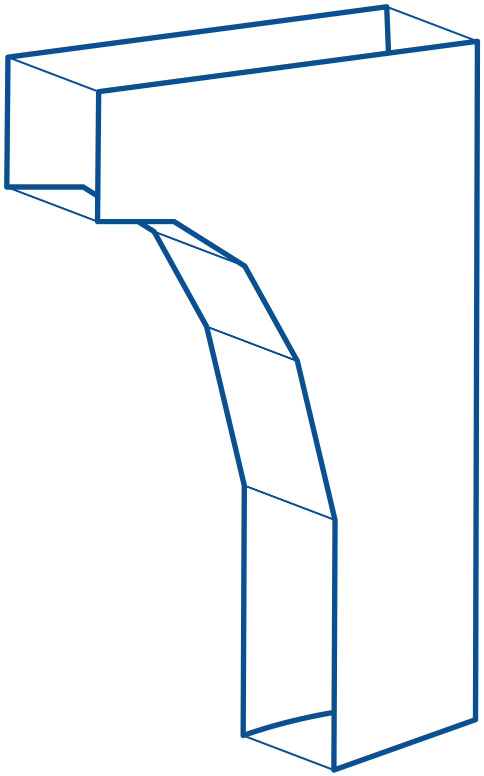
Modelling of Planar Shapes



Tessellation using triangles



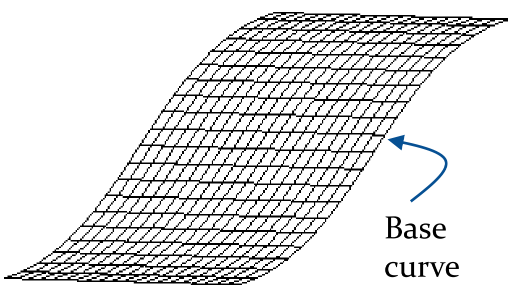
Tessellation using triangle strips



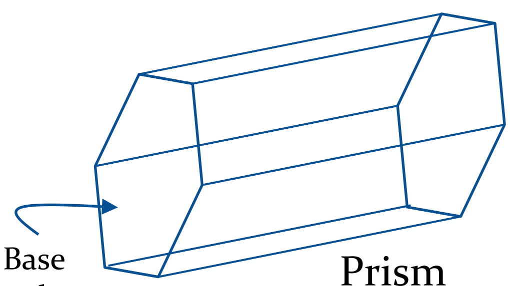
Tessellation using quad strips

Sweep Representations

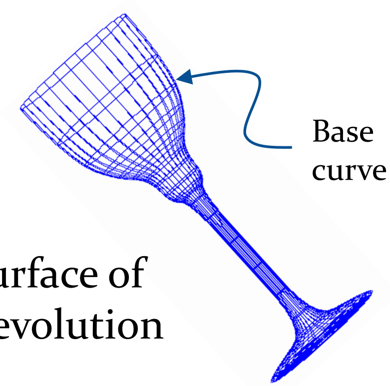
- Sweep representations are useful for both surface modeling and solid modeling.
- A large class of shapes (both surfaces and solid models) can be formed by *sweeping* or *extruding* a 2D shape (**base polygon** or **base curve**) through space.
- A **ruled surface** is generated by moving a straight line in a particular trajectory.
- A polyhedron obtained by sweeping (extruding) a polygon along a straight line is called a **prism**.


Base curve

Ruled Surface


Base polygon

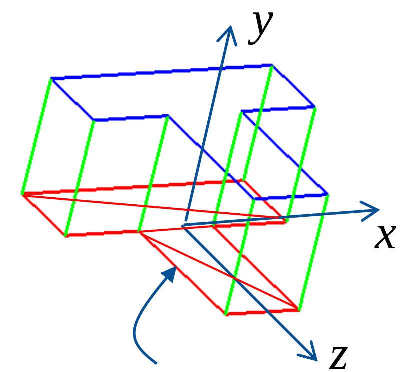
Prism

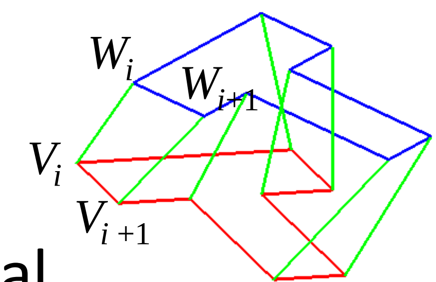

Base curve

Surface of Revolution

Extruded Surfaces

- Create a base polygon, centred at the origin. For proper rendering of this polygon, subdivide it into a set of triangles.
  Vertex List: $V_i = (x_i, y_i, z_i)$, $i = 0..\, n_v - 1$.
- Transform the above vertices using a matrix $\boldsymbol{M}$ to get the vertices of the extruded shape:
  Vertex List: $W_i = \boldsymbol{M}V_i$, $i = 0..\, n_v - 1$.
- Generate the boundary surface using quadrilaterals $V_i\, V_{i+1}\, W_{i+1}\, W_i$
- Compute normal vectors of each quadrilateral using vertex coordinates.





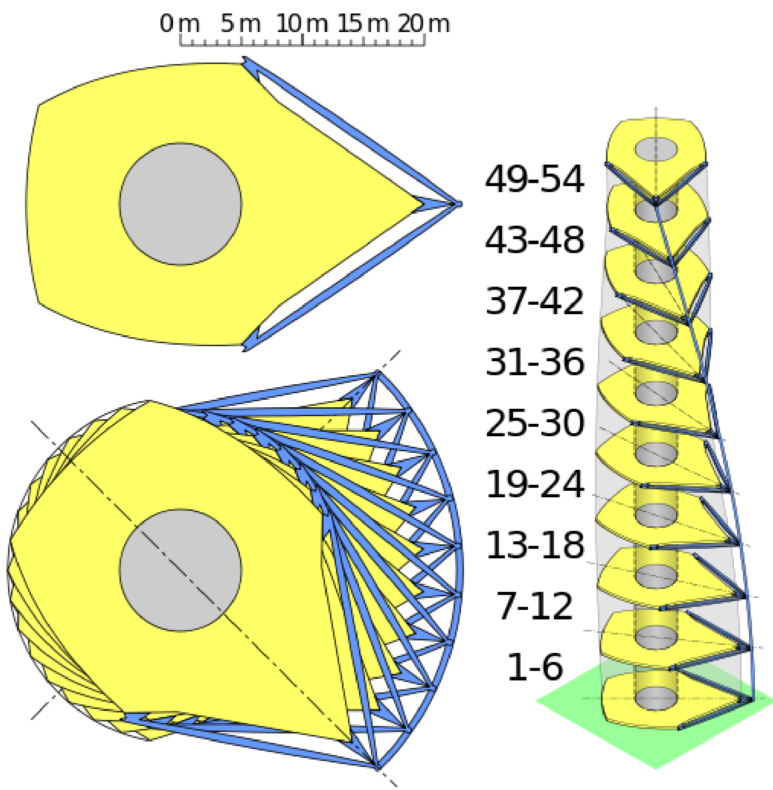
$$\begin{bmatrix} x_i' \\ y_i' \\ z_i' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & h \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

Extruded Shape

Architectural Example:

“Turning Torso”, the twisted tower in Malmo, Sweden.

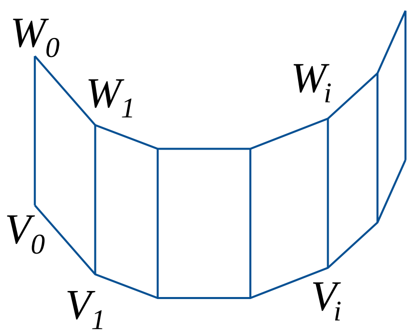
[Source: Wikipedia]



Extruded Surfaces Using Quad Strips

- The polygons surrounding an extruded surface can be generated using a quad strip or a triangle strip.

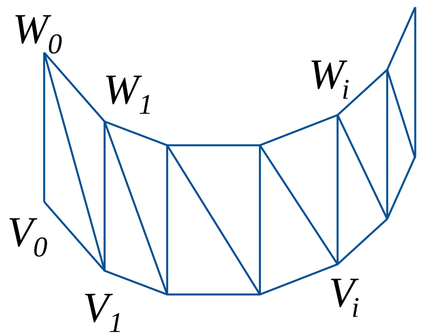
Quad Strip




```
glBegin(GL_QUAD_STRIP);
  for(int i = 0; i < N; i++)
  {
    glVertex3f(vx[i], vy[i], vz[i]);
    glVertex3f(wx[i], wy[i], wz[i]);
  }
glEnd();
```

- Quad strip: Every pair of points V_i, W_i ($i > 0$) generates a new quad $W_{i-1}V_{i-1}V_iW_i$. Define normal (in the above example) using points $V_{i-1}V_iW_i$.

Extruded Surfaces Using Triangle Strips




```
glBegin(GL_TRIANGLE_STRIP);
  for(int i = 0; i < N; i++)
  {
    glVertex3f(vx[i], vy[i], vz[i]);
    glVertex3f(wx[i], wy[i], wz[i]);
  }
glEnd();
```

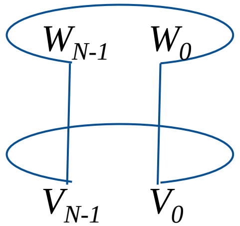
- Triangle strip: Every point V_i $(i > 0)$ generates a new triangle $W_{i\text{-}1}V_{i\text{-}1}V_i$. Every point W_i $(i > 0)$ generates a new triangle $W_{i\text{-}1}V_iW_i$.

Extruded Surfaces (Example)

- Quad Strip:
  - Number of vertices: *N*
  - Vertices: V_i = (vx[i], vy[i], vz[i]), W_i = (wx[i], wy[i], wz[i]),

```
glBegin(GL_QUAD_STRIP);
  for(int i = 0; i < N; i++)
  {
      if(i > 0) normal( vx[i-1], vy[i-1], vz[i-1],
                        vx[i], vy[i], vz[i],
                        wx[i], wy[i], wz[i] );
      glVertex3f(vx[i], vy[i], vz[i]);
      glVertex3f(wx[i], wy[i], wz[i]);
  }
glEnd();
```

Need to close the surface here.

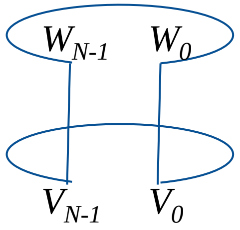


Extruded Surfaces (Example)

- Triangle Strip:

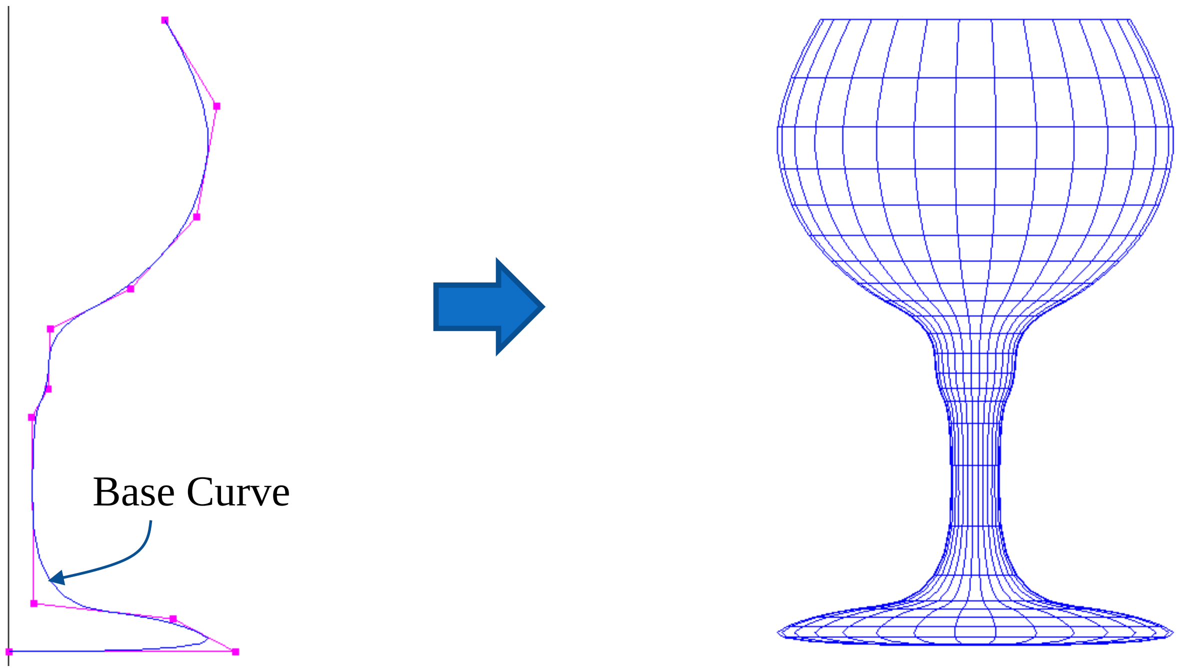
```
glBegin(GL_TRIANGLE_STRIP);
  for(int i = 0; i < N; i++)
  {
      if(i > 0) normal( wx[i-1], wy[i-1], wz[i-1],
                        vx[i-1], vy[i-1], vz[i-1],
                        vx[i], vy[i], vz[i] );
      glVertex3f(vx[i], vy[i], vz[i]);
      if(i > 0) normal( wx[i-1], wy[i-1], wz[i-1],
                        vx[i], vy[i], vz[i],
                        wx[i], wy[i], wz[i] );
      glVertex3f(wx[i], wy[i], wz[i]);
  }
glEnd();
```

Need to close the surface here.



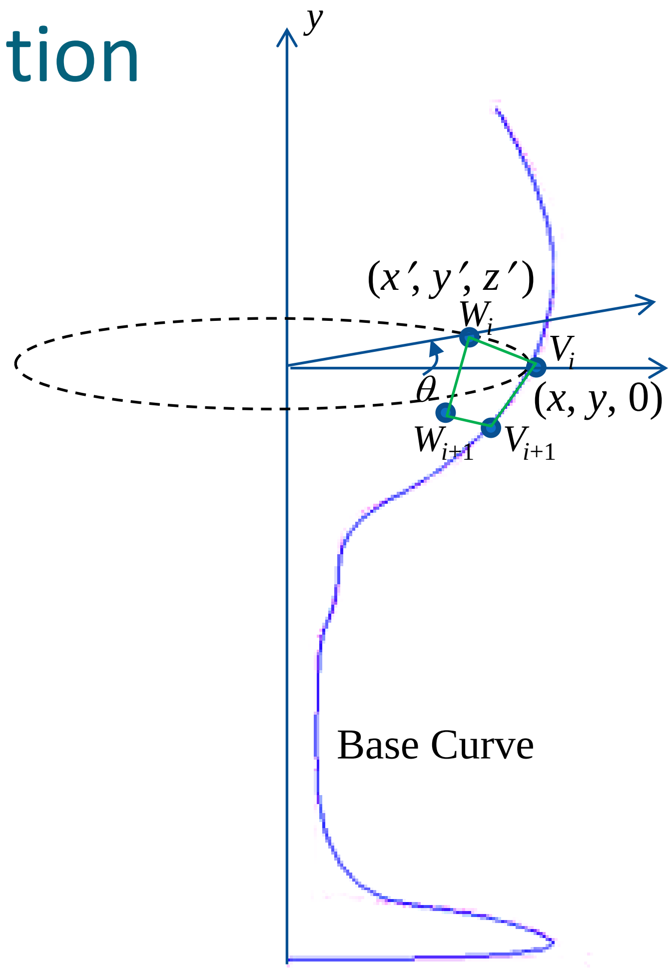
Surface of Revolution

- A surface of revolution is obtained by revolving a planar curve about an axis, typically the *y*-axis.
- The curve (called the base curve) is usually generated using a parametric equation, such as Bezier polynomials or other types of splines (eg. Basis splines).



Surface of Revolution

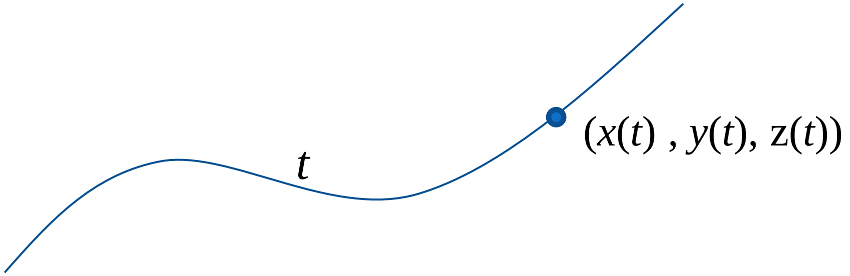
- The vertices on the base curve are given by $V_i = (x_i, y_i, 0)$, $i = 0.. n_v-1$.
- Transform the above vertices using a rotation matrix $\boldsymbol{M}$ to get the vertices of the revolved shape:
  $W_i = \boldsymbol{M}V_i, \quad i = 0.. n_v-1.$
- Generate the boundary surface using quadrilaterals $V_i\, V_{i+1}\, W_{i+1}\, W_i$ or quad/triangle strips $V_i\, W_i$
- Transform the surface normal vectors also by the same matrix $\boldsymbol{M}$.



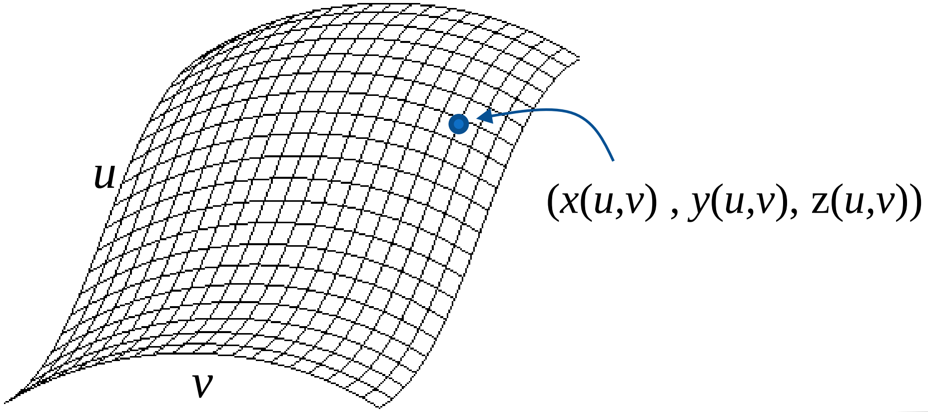
$$\begin{bmatrix} x_i' \\ y_i' \\ z_i' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 0 \\ 1 \end{bmatrix}$$

Parametric Forms of Curves and Surfaces

- The parametric form of a curve in three-dimensional space gives point coordinates on the curve in terms of a single parameter *t*.



- A parametric surface requires two independent parameters to express points on the surface.



Quadrics

- A quadric surface in three dimensions is given by an algebraic equation of the form

$$ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d = 0.$$

- Quadric surfaces can also be expressed in parametric form: $x(u, v)$, $y(u, v)$, $z(u, v)$
  - Sphere: $(R \sin\alpha \cos\delta,\ R \sin\delta,\ R \cos\alpha \cos\delta)$,

    $0 \le \alpha \le 360, \quad -90 \le \delta \le 90$
  - Cylinder: $(R \sin\alpha,\ \ y,\ \ R \cos\alpha)$

    $0 \le \alpha \le 360, \quad 0 \le y \le h$
  - Cone: $((h-y)R \sin\alpha/h,\ \ y,\ \ (h-y)R \cos\alpha/h)$

    $0 \le \alpha \le 360, \quad 0 \le y \le h$