Announcements

- Assignment 1 (**due next Wednesday**)
- Midterm is **Wednesday October 15, 5-7pm**
- **Office Hours Monday 1-2 pm (again)**
- Homework questions?

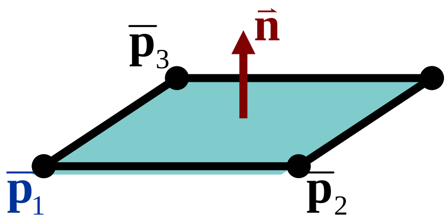
Last class

- **Coordinate Free Geometry**
  - Style of expressing geometric objects and relations that avoids reliance on coordinate systems
  - **Defined:** 9 basic CFG operations

- **3D Surfaces**
  - Implicit and parametric forms
  - **Basic surfaces:** plane, bilinear patch, cylinder
  - Tangents and Normals
  - Surface of revolution

Review: Basic Surfaces

Plane



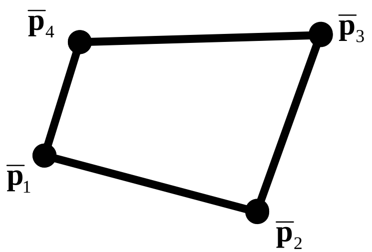
$$\bar{\mathbf{s}}(\alpha,\beta) = \bar{\mathbf{p}}_1 + \alpha\,\vec{\mathbf{a}} + \beta\,\vec{\mathbf{b}} \qquad \alpha,\beta \in \Re$$

Bilinear Patch

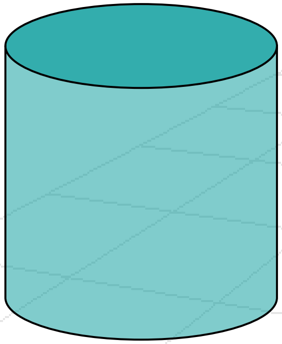
$$\bar{\mathbf{s}}(\alpha,\beta) = (1-\beta)\big((1-\alpha)\bar{\mathbf{p}}_1 + \alpha\bar{\mathbf{p}}_2\big) + \beta\big((1-\alpha)\bar{\mathbf{p}}_4 + \alpha\bar{\mathbf{p}}_3\big)$$

$$0 \le \alpha \le 1$$

$$0 \le \beta \le 1$$



Right Circular Cylinder



$$\bar{\mathbf{s}}(\alpha,\beta) = \big(\mathbf{r}\cos(\alpha), \mathbf{r}\sin(\alpha), \beta\big)$$

$$0 \le \alpha \le 2\pi$$

$$0 \le \beta \le 1$$

Computing a Normal for a Surface

- **Parametric Form**
  - The surface $\bar{\mathbf{s}}(\alpha,\beta) = \left(\mathbf{x}(\alpha,\beta), \mathbf{y}(\alpha,\beta), \mathbf{z}(\alpha,\beta)\right)$

    has **two tangents in a tangent plane** at a point

$$\left.\frac{\partial \bar{\mathbf{s}}(\alpha,\beta)}{\partial \alpha}\right|_{\alpha_0,\beta_0} \qquad \left.\frac{\partial \bar{\mathbf{s}}(\alpha,\beta)}{\partial \beta}\right|_{\alpha_0,\beta_0}$$

  - **Normal** to the surface at a point is then given by:

$$\vec{\mathbf{n}}(\alpha_0,\beta_0) = \left(\left.\frac{\partial \bar{\mathbf{s}}(\alpha,\beta)}{\partial \alpha}\right|_{\alpha_0,\beta_0}\right) \times \left(\left.\frac{\partial \bar{\mathbf{s}}(\alpha,\beta)}{\partial \beta}\right|_{\alpha_0,\beta_0}\right)$$

- **Implicit Form**

$$\vec{\mathbf{n}}(\bar{\mathbf{p}}_0) = \nabla \mathbf{f}(\bar{\mathbf{p}})\big|_{\bar{\mathbf{p}}_0} = \left(\left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{y},\mathbf{z})}{\partial \mathbf{x}}\right|_{\bar{\mathbf{p}}_0}, \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{y},\mathbf{z})}{\partial \mathbf{y}}\right|_{\bar{\mathbf{p}}_0}, \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{y},\mathbf{z})}{\partial \mathbf{z}}\right|_{\bar{\mathbf{p}}_0}\right)$$

Surface of Revolution

Demo

Quadrics

- Generalization of conic section 3D

$$\mathbf{a}\mathbf{x}^2 + \mathbf{b}\mathbf{y}^2 + \mathbf{c}\mathbf{z}^2 + \mathbf{d} = 0$$

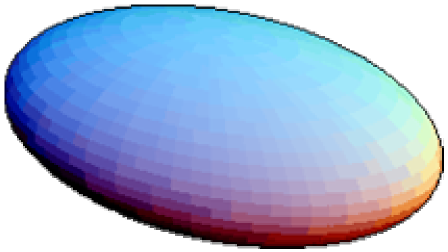
$$\mathbf{a}\mathbf{x}^2 + \mathbf{b}\mathbf{y}^2 + \mathbf{e}\mathbf{z} = 0$$

- Basic types of surface depend on signs of **a, b, c, d,** and **e** (i.e. -, +, 0).

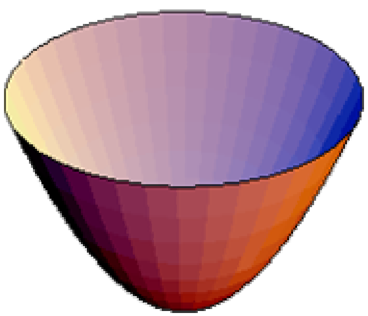
- Examples
  - Ellipsoid, elliptic cones
  - Hyperboloid of 1 sheet, of 2 sheets
  - Poraboloid
  - Hyperbolic poraboloid

(First three examples bracketed:) **Surfaces of revolution**

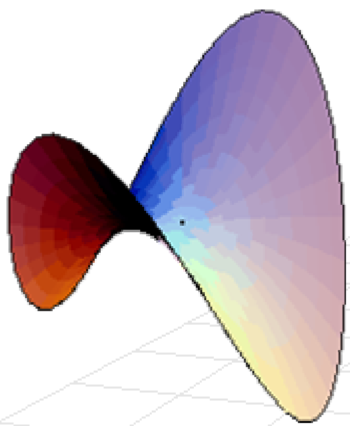
Quadrics



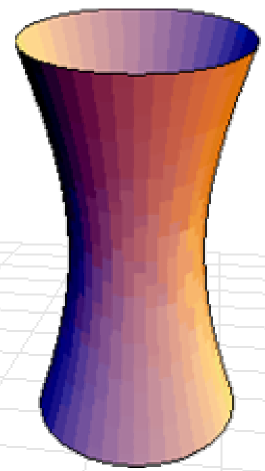
Ellipsoid



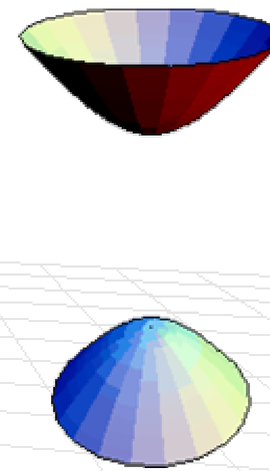
Elliptic poraboloid



Hyperbolic poraboloid



Hyperboloid of one sheet



Hyperboloid of two sheet

Example: Ellipsoid

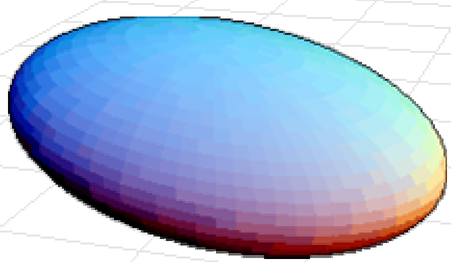
- **Parametric Form:**

$$\bar{\mathbf{s}}(\alpha,\beta) = \left(\underbrace{\mathbf{a}\cos(\alpha)}\sin(\beta),\ \underbrace{\mathbf{b}\sin(\alpha)}\sin(\beta),\ \mathbf{c}\cos(\beta)\right)$$

2D Ellipse

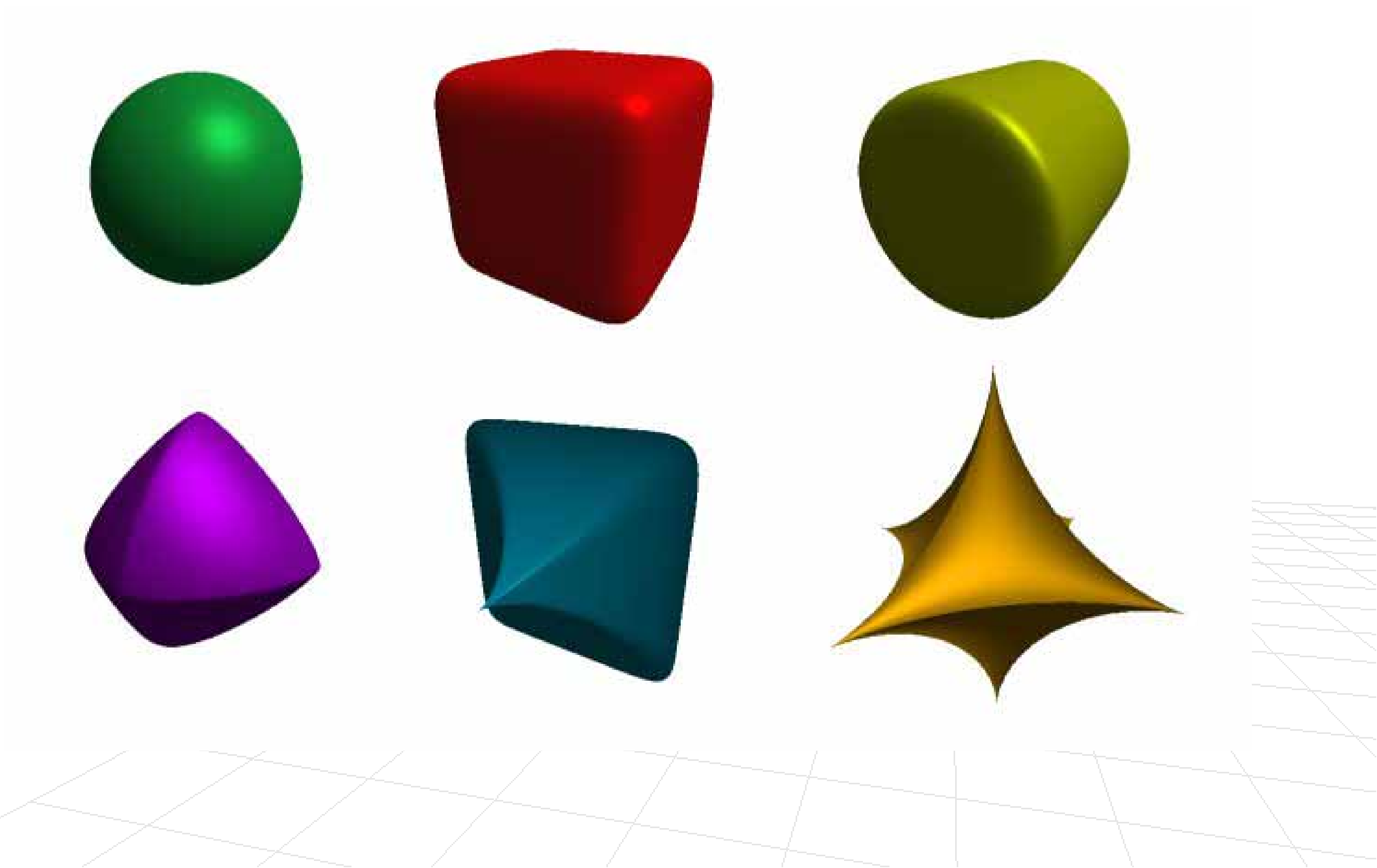
- **Implicit Form:**

$$\mathbf{f}(\mathbf{x},\mathbf{y},\mathbf{z}) = \frac{\mathbf{x}^2}{\mathbf{a}^2} + \frac{\mathbf{y}^2}{\mathbf{b}^2} + \frac{\mathbf{z}^2}{\mathbf{c}^2} - 1 = 0$$



Ellipsoid

Super quadrics



Polygonal Mesh

- **Polygons** are used to approximate curves
- **Polygonal meshes** are used to approximate surfaces

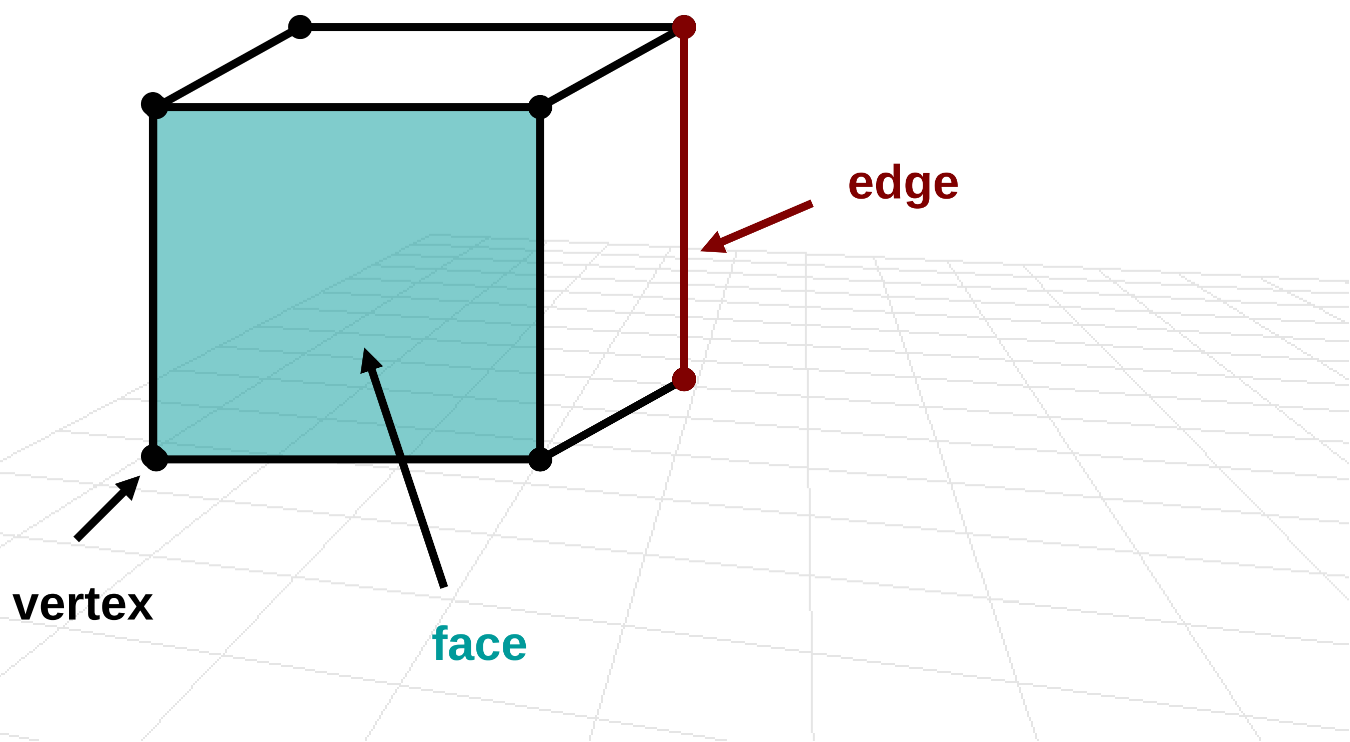
- **Polygonal mesh** - collection of polygons

- A **polyhedron** is a closed, connected polygonal mesh. Each edge must be shared by two faces.
- A **face** refers to a planar polygonal patch within a mesh.
- A mesh is **simple** when its topology is equivalent to that of a sphere. That is, it has no holes.

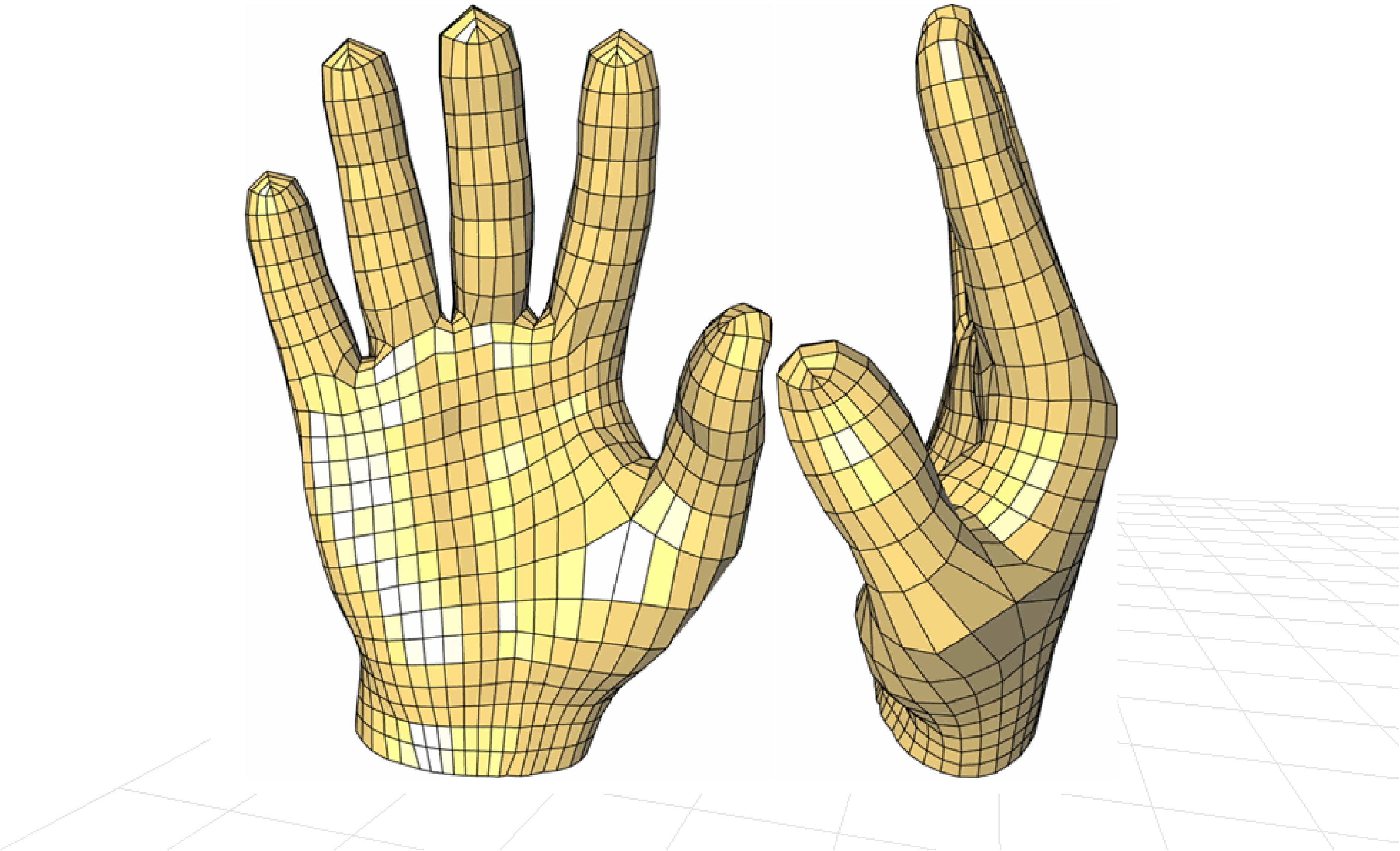
Polygonal Mesh

- **Polygons** are used to approximate curves
- **Polygonal meshes** are used to approximate surfaces

- **Polygonal mesh** - collection of polygons

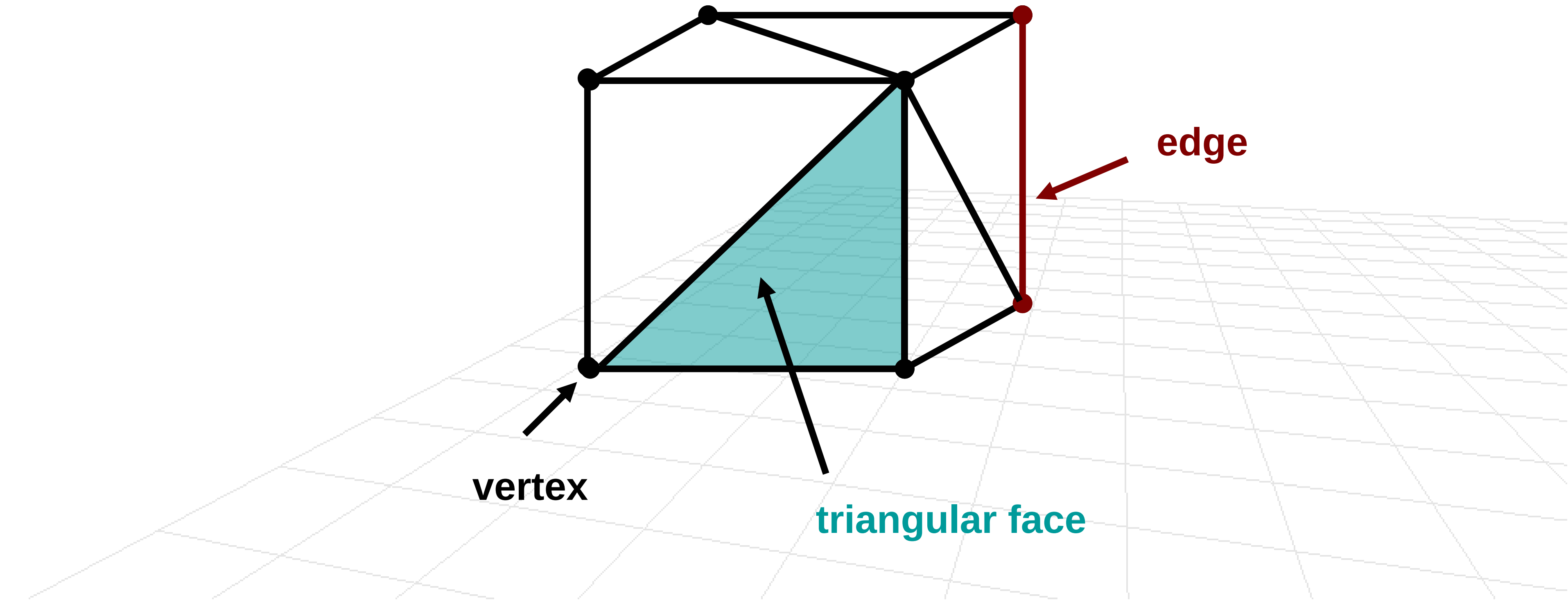


Polygonal Mesh: Example

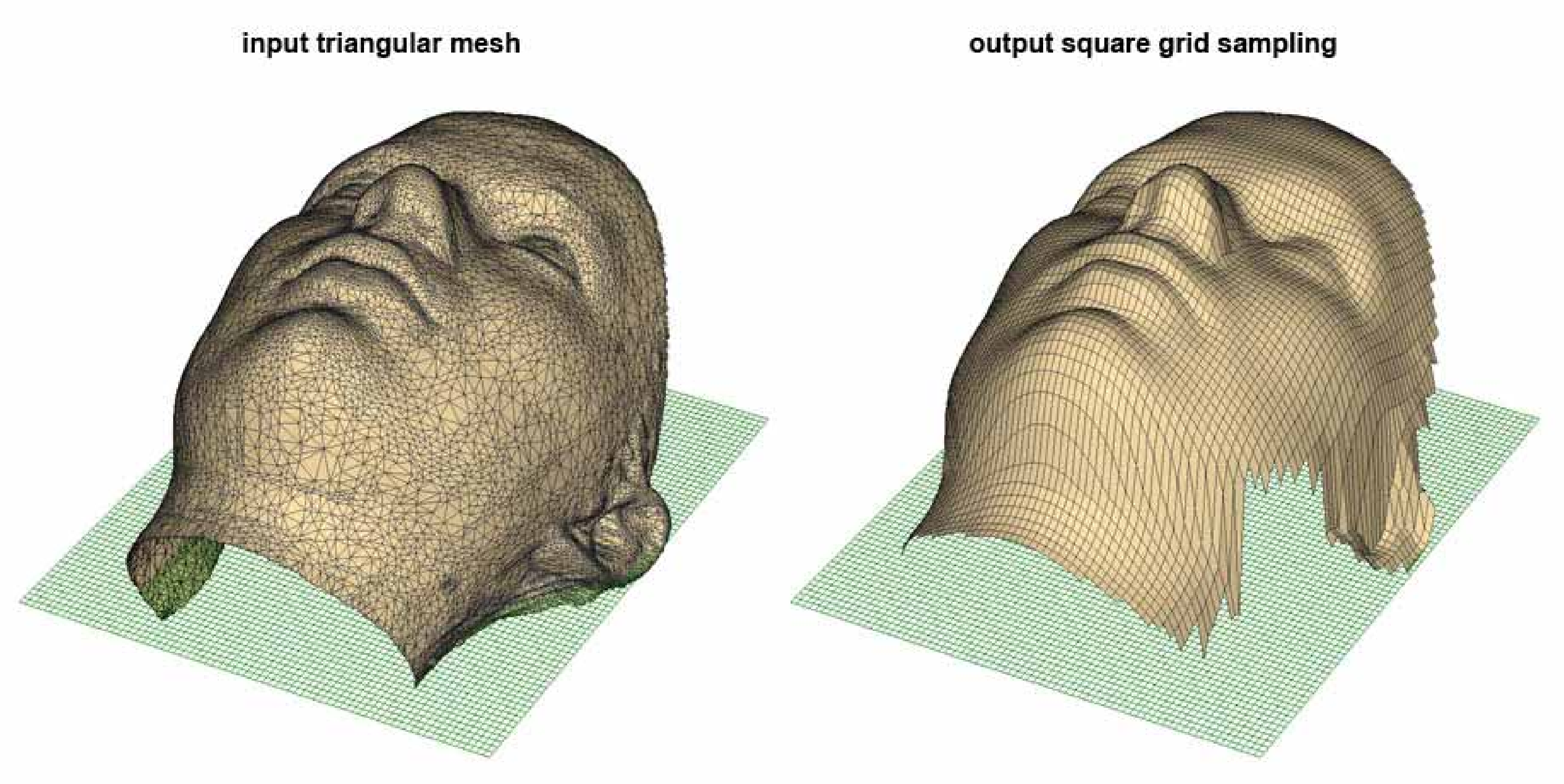


Triangular Mesh

- **Triangular mesh** - collection of triangles



Mesh Models: Example



3D Transformations

Computer Graphics, CSCD18

Fall 2007

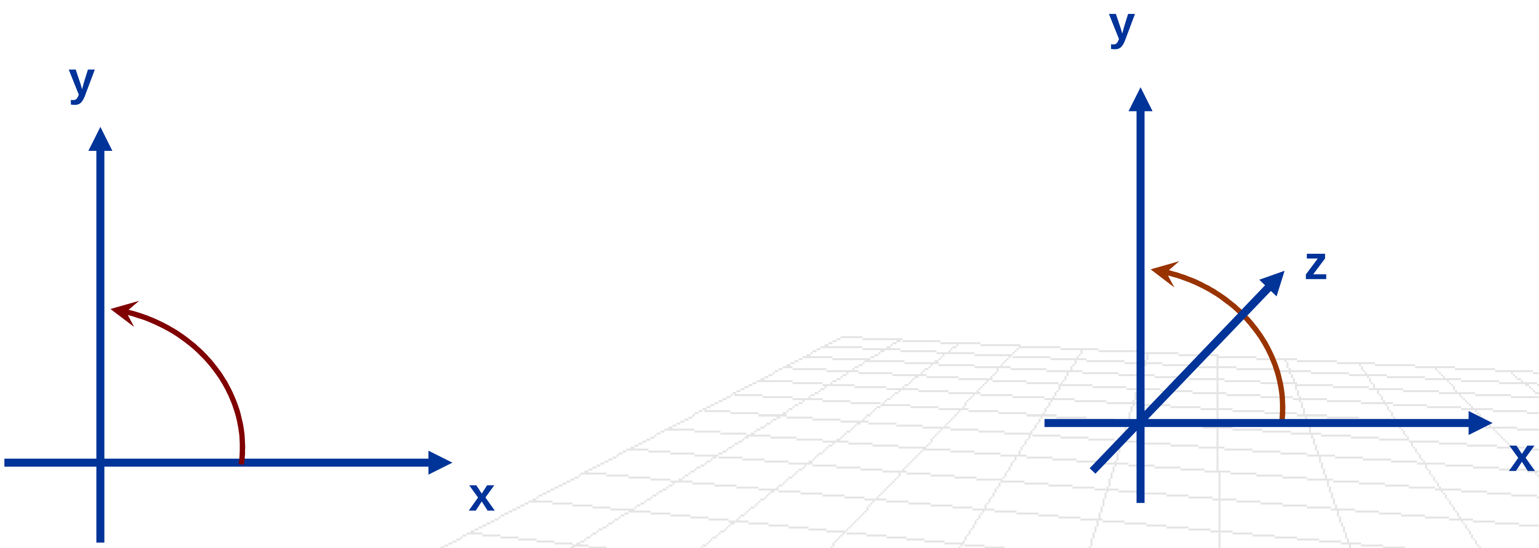
Instructor: Leonid Sigal

3D Transformations

- Why do we need them?
  - Coordinate transforms
  - Shape modeling (e.g. surfaces of revolution)
    - Alex will do this in the tutorial next week
  - Hierarchical object models
  - Camera modeling

3D Coordinate Frame

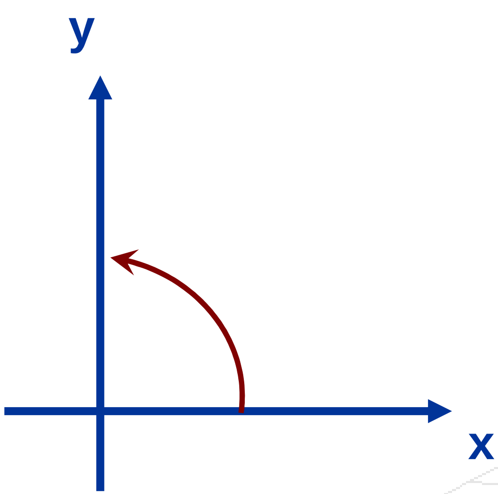
- In 3D there are two conventions for coordinate frames

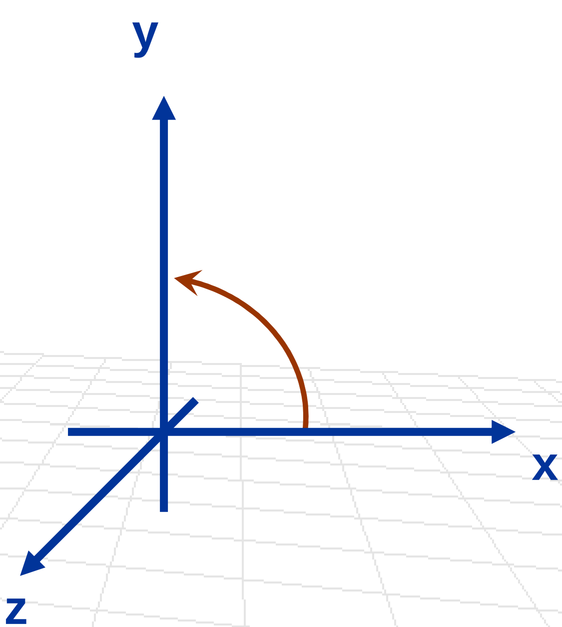


Left-handed Coordinate System

3D Coordinate Frame

- In 3D there are two conventions for coordinate frames





Right-handed Coordinate System
(OpenGL uses this convention … so will we)

Affine Transformations

- Affine transformations in 3D look the same as in 2D

$$F(\vec{p}) = A\vec{p} + \vec{t}$$

$\vec{p}$ - point mapped, $\in \boldsymbol{R}^3$

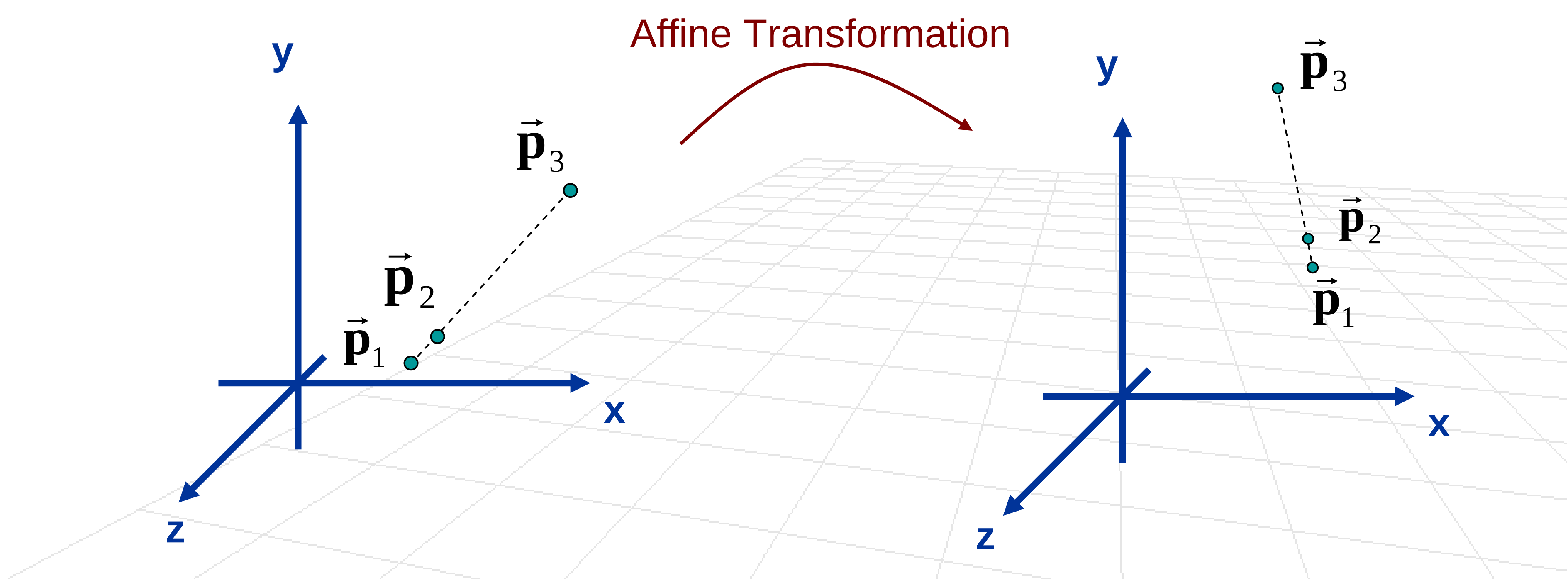
$\vec{t}$ - translation, $\in \boldsymbol{R}^3$

A - transformation matrix, $\in \boldsymbol{R}^{3x3}$

- Many of the transformations we will talk about today are of this type

Properties of Affine Transformations

- Collinearity of points is preserved
- Ratio of distances along the line is preserved
- Concatenation of affine transformations is also an affine transformation



Homogeneous Affine Transformations

- We can rewrite the affine transformation

$$\mathbf{F}(\vec{\mathbf{p}}) = \mathbf{A}\vec{\mathbf{p}} + \vec{\mathbf{t}}$$

in homogeneous coordinates as follows:

$$\boxed{\mathbf{F}(\hat{\mathbf{p}}) = \mathbf{M}\hat{\mathbf{p}}}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \vec{\mathbf{t}} \\ [0,0,0] & 1 \end{bmatrix} \qquad \hat{\mathbf{p}} = \begin{bmatrix} \vec{\mathbf{p}} \\ 1 \end{bmatrix}$$

- This has nice properties, as we have seen before (and will see again)

3D Translation

- Simple extension of the 2D translations

$$\mathbf{T}_{2\mathbf{D}} = \begin{bmatrix} 1 & 0 & \mathbf{t_x} \\ 0 & 1 & \mathbf{t_y} \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{T}_{3\mathbf{D}} = \begin{bmatrix} 1 & 0 & 0 & \mathbf{t_x} \\ 0 & 1 & 0 & \mathbf{t_y} \\ 0 & 0 & 1 & \mathbf{t_z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D Scaling

- Simple extension of the 2D translations

$$\mathbf{S}_{2\mathbf{D}} = \begin{bmatrix} \mathbf{s_x} & 0 & 0 \\ 0 & \mathbf{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}_{3\mathbf{D}} = \begin{bmatrix} \mathbf{s_x} & 0 & 0 & 0 \\ 0 & \mathbf{s_y} & 0 & 0 \\ 0 & 0 & \mathbf{s_z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D Rotation

- In general, rotations in 3D are much more complicated then 2D rotations
  - There is typically no unique rotation that does what you want
  - You can specify rotations in variety of ways that are convenient for different tasks (e.g. Euler angles, Axis/Angle, Quaternion, Exponential Map)
- We will only consider elementary rotations (Euler Angles)

3D Rotation

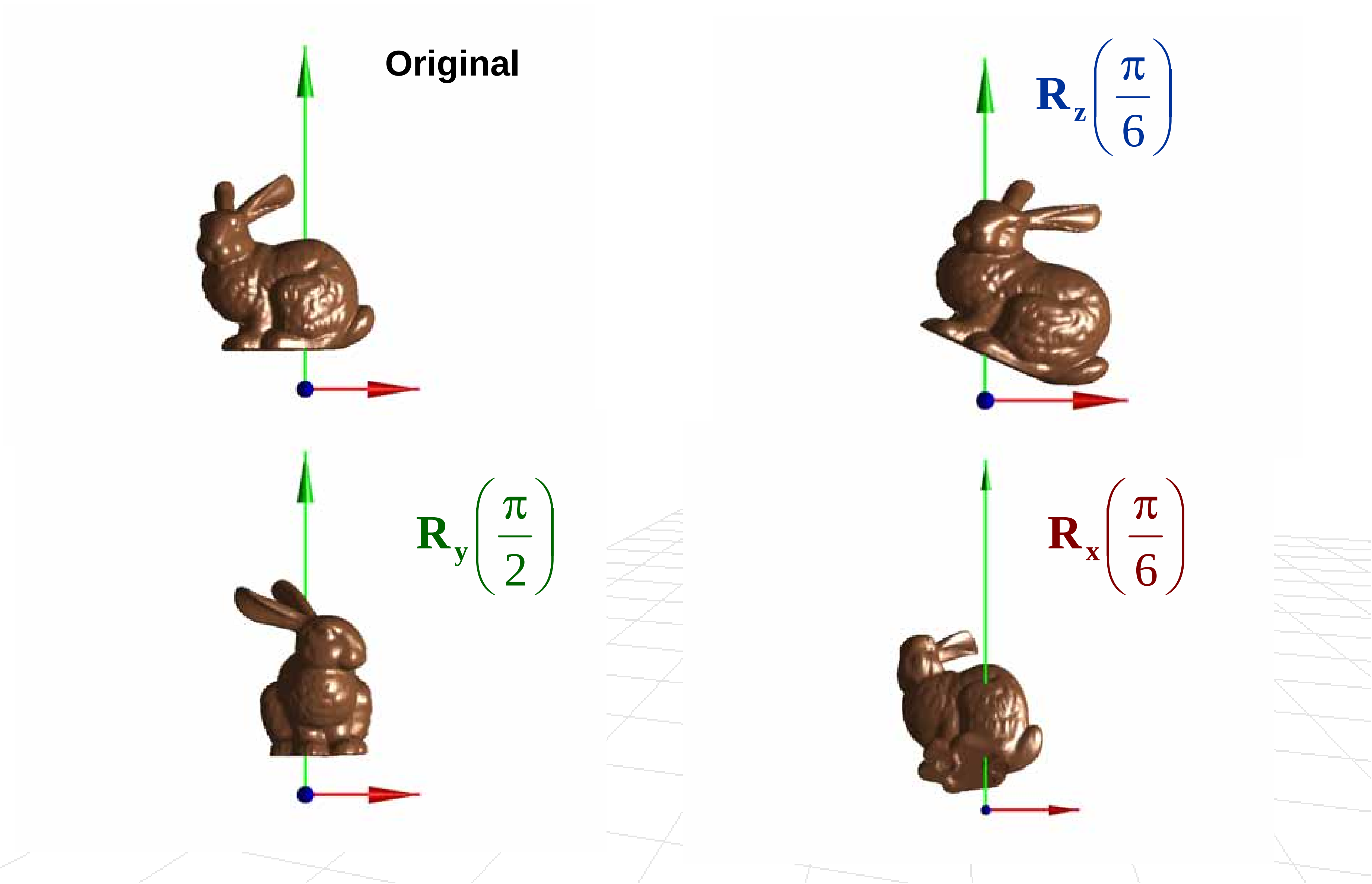
- 2D rotation introduced previously is simply a 3D rotation about the Z-axis

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- But we also have rotations about the X- and Y-axis

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D Rotation - Examples



Composing Rotations

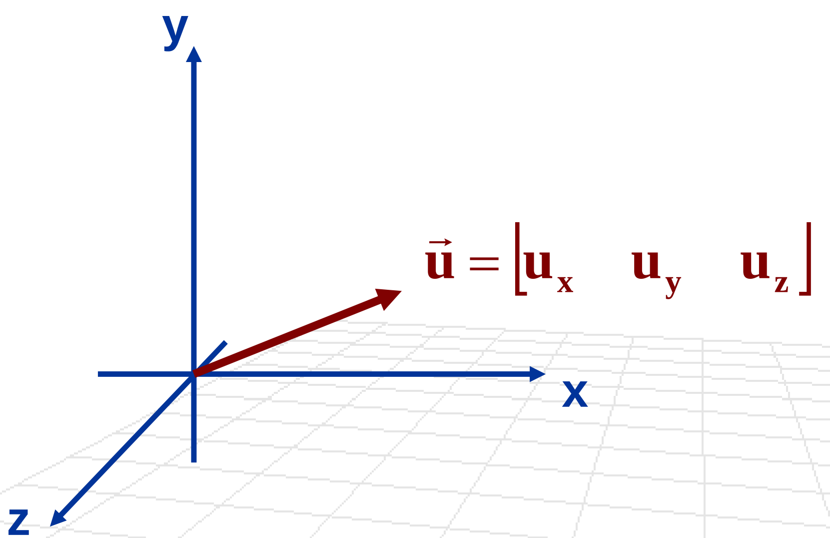
- Rotation order matters !!!
- For example,

$$\mathbf{R_z}(\theta_\mathbf{z})\mathbf{R_x}(\theta_\mathbf{x}) \neq \mathbf{R_x}(\theta_\mathbf{x})\mathbf{R_z}(\theta_\mathbf{z})$$

- So one needs to be careful

Rotation about Arbitrary Axis

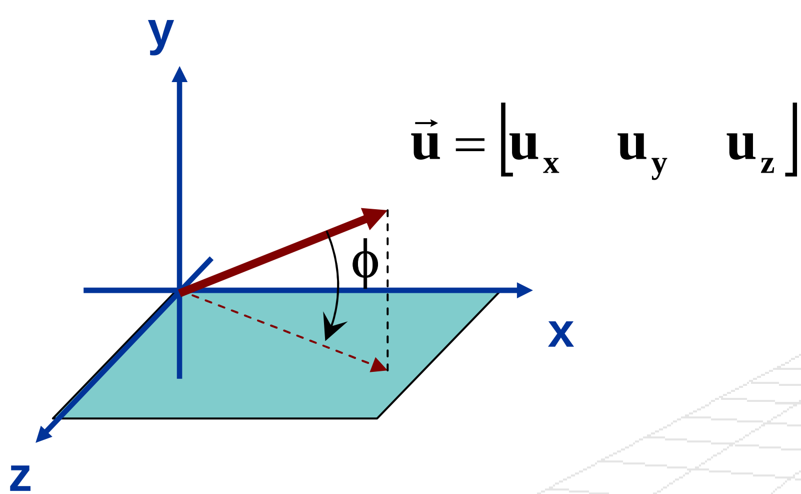
- In general we want to rotate a point or an object about arbitrary axis $\vec{\mathbf{u}}$ by some θ
- How do we do this using what we already know?

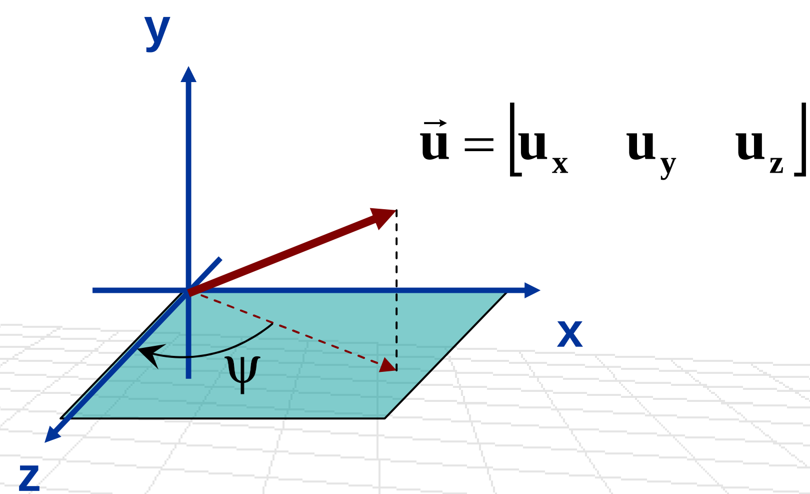


- **Hint:** Can be done by composing elementary rotations

Rotation about Arbitrary Axis

- **Idea:** Align $\vec{\mathbf{u}}$ with z-axis, then rotate about z-axis by desired angle θ





1) Rotate $\vec{\mathbf{u}}$ into x-z plane $\mathbf{R}_z(\phi)$

2) Rotate $\vec{\mathbf{u}}$ in x-z plane $\mathbf{R}_y(\psi)$

3) Rotate by θ about z-axis

4) Undo (1) and (2), i.e. $\left(\mathbf{R}_z(\phi)\mathbf{R}_y(\psi)\right)^{-1} = \left(\mathbf{R}_y(\psi)\right)^{-1}\left(\mathbf{R}_z(\phi)\right)^{-1} = \mathbf{R}_y(-\psi)\mathbf{R}_z(-\phi)$

Rotation about Arbitrary Axis

- Hence rotation about an arbitrary axis can always be expressed as a series of elementary rotations

$$\mathbf{R}(\vec{\mathbf{u}},\theta)=\mathbf{R_z}(\phi)\mathbf{R_x}(\psi)\mathbf{R_z}(\theta)\mathbf{R_x}(-\psi)\mathbf{R_z}(-\phi)$$

- How do we obtain values for angles ϕ, ψ ?

Non-Linear Transformations

- Affine transformations

$$\mathbf{F}(\vec{\mathbf{p}}) = \mathbf{A}\vec{\mathbf{p}} + \vec{\mathbf{t}}$$

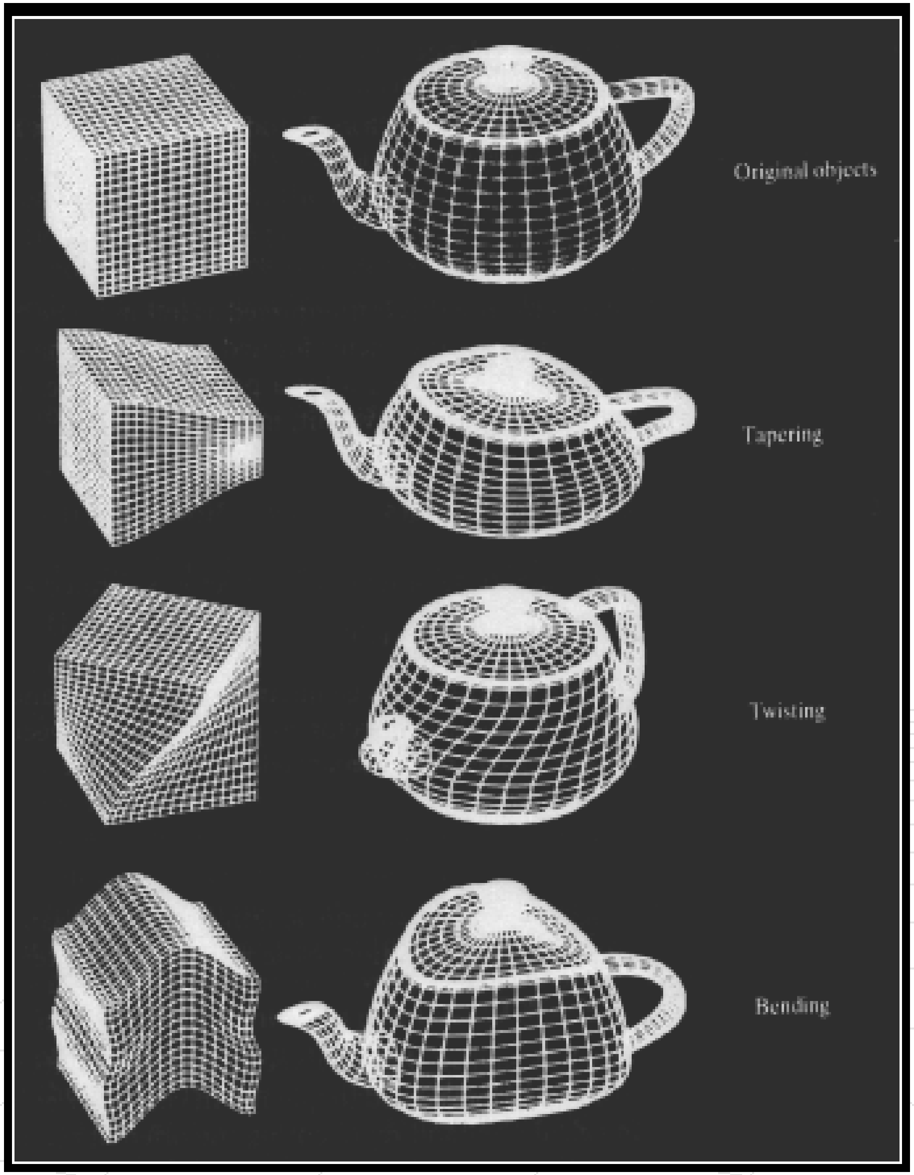
  are 1st order shape deformations

- Higher order deformations are also possible, let's consider general differentiable deformation $\mathbf{F}(\vec{\mathbf{p}})$ then we can express deformation as a Taylor series

$$\mathbf{F}(\vec{\mathbf{p}}) = \vec{\mathbf{t}} + \mathbf{A}\vec{\mathbf{p}} + \mathbf{B}\vec{\mathbf{p}}^{2} + \ldots$$

- Common non-linear transformations: tapering, twisting, bending

Non-Linear Transformations



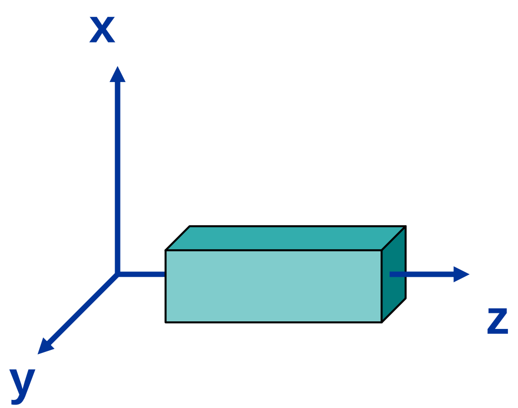
Original

Tapering

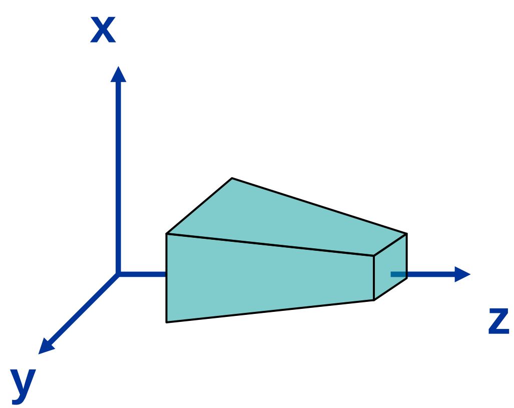
Twisting

Bending

Tapering



Scaling



Linear Taper

$$\mathbf{S}_{3D} = \begin{bmatrix} \mathbf{s_x} & 0 & 0 & 0 \\ 0 & \mathbf{s_y} & 0 & 0 \\ 0 & 0 & \mathbf{s_z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Taper}_{3D} = \begin{bmatrix} \mathbf{s_x}(\mathbf{p_z}) & 0 & 0 & 0 \\ 0 & \mathbf{s_y}(\mathbf{p_z}) & 0 & 0 \\ 0 & 0 & \mathbf{s_z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{s_x}(\mathbf{p_z}) = \alpha_0 + \alpha_1 \mathbf{p_z}$$

$$\mathbf{s_y}(\mathbf{p_z}) = \alpha_0 + \alpha_1 \mathbf{p_z}$$