MVE555 Architectural Geometry, Lecture 3

MVE560, Lecture 3

Mathematical Sciences
CHALMERS

Outline

Set Operations

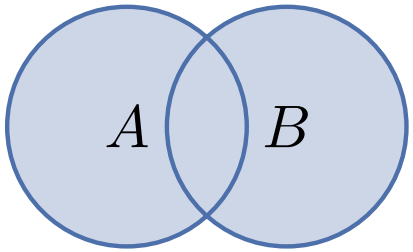
Offsets

Slice Deformations

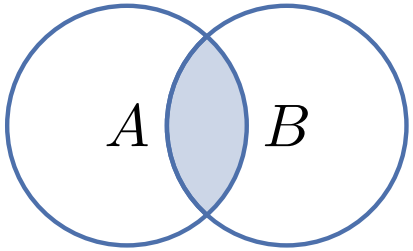
Union, Intersection, and Difference

It is often useful to think in terms of the common mathematical set operations:

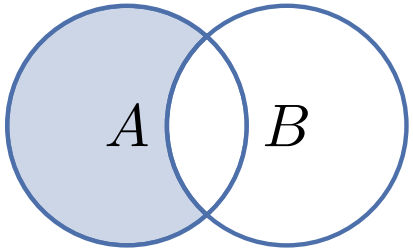
union: $A \cup B = \{x : x \in A \text{ or } x \in B\}$



intersection: $A \cap B = \{x : x \in A \text{ and } x \in B\}$



difference: $A \setminus B = \{x : x \in A \text{ and } x \notin B\}$



complement: $A^* = \{x : x \notin A\}$

Union, Intersection, and Difference (contd.)

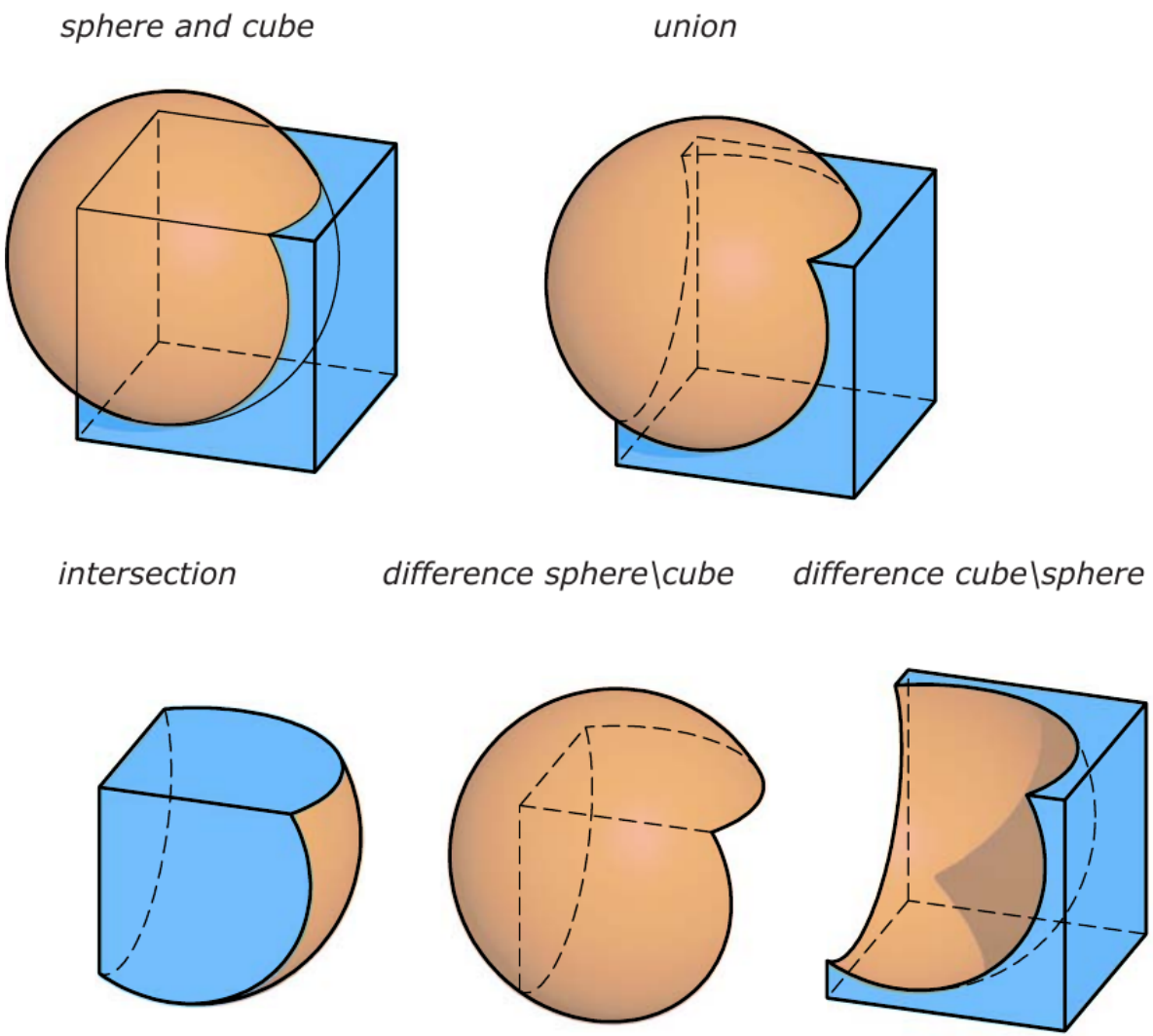


Image source: Pottmann *et al.*

Union, Intersection, and Difference (contd.)

Mathematically, set operations have many useful properties, e.g.

- $A \cup B = B \cup A$ and $A \cap B = B \cap A$ (*commutativity*)
- $A \cap (B \cap C) = (A \cap B) \cap C$ and $A \cup (B \cup C) = (A \cup B) \cup C$ (*associativity*)
- $A \setminus B = A \cap B^*$
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ (distributivity)
- $(A \cup B)^* = A^* \cap B^*$ (De Morgan)
- ...

Union, Intersection, and Difference (contd.)

Mathematically, set operations have many useful properties, e.g.

- $A \cup B = B \cup A$ and $A \cap B = B \cap A$ (*commutativity*)
- $A \cap (B \cap C) = (A \cap B) \cap C$ and $A \cup (B \cup C) = (A \cup B) \cup C$ (*associativity*)
- $A \setminus B = A \cap B^*$
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ (distributivity)
- $(A \cup B)^* = A^* \cap B^*$ (De Morgan)
- ...
- Look up or think through when you need them!

Implementing Set Operations

Set operations for *solids* are straightforward, but we represent our geometry as *surfaces*! To obtain the expected result for surfaces, we must

- find the intersection curve(s) of the surfaces,
- split the surfaces along the intersection curves,
- decide which parts to save and which parts to delete, and
- join the remaining parts along the intersection curves.

Offset Curves

An *offset curve* to a curve $\boldsymbol{c}(t) = (x(t), y(t))$ is a (usually two-branched) curve 'parallel' to $\boldsymbol{c}(t)$.
Mathematically, the offset curve is given by $\boldsymbol{c}_d(t) = \boldsymbol{c}(t) \pm d\boldsymbol{n}(\boldsymbol{t})$, where $\boldsymbol{n}(t)$ is the unit normal at $\boldsymbol{c}(t)$ and d is the offset distance.

Offset Curves

An *offset curve* to a curve $\boldsymbol{c}(t) = (x(t), y(t))$ is a (usually two-branched) curve 'parallel' to $\boldsymbol{c}(t)$.
Mathematically, the offset curve is given by $\boldsymbol{c}_d(t) = \boldsymbol{c}(t) \pm d\boldsymbol{n}(\boldsymbol{t})$, where $\boldsymbol{n}(t)$ is the unit normal at $\boldsymbol{c}(t)$ and d is the offset distance.
Since $\boldsymbol{n}(t) = \frac{(-y'(t),x'(t))}{\sqrt{x'(t)^2+y'(t)^2}}$,

$$\boldsymbol{c}_d(t) = \boldsymbol{c}(t) \pm d \cdot \frac{(-y'(t), x'(t))}{\sqrt{x'(t)^2 + y'(t)^2}}$$

Offset Curves (Examples)

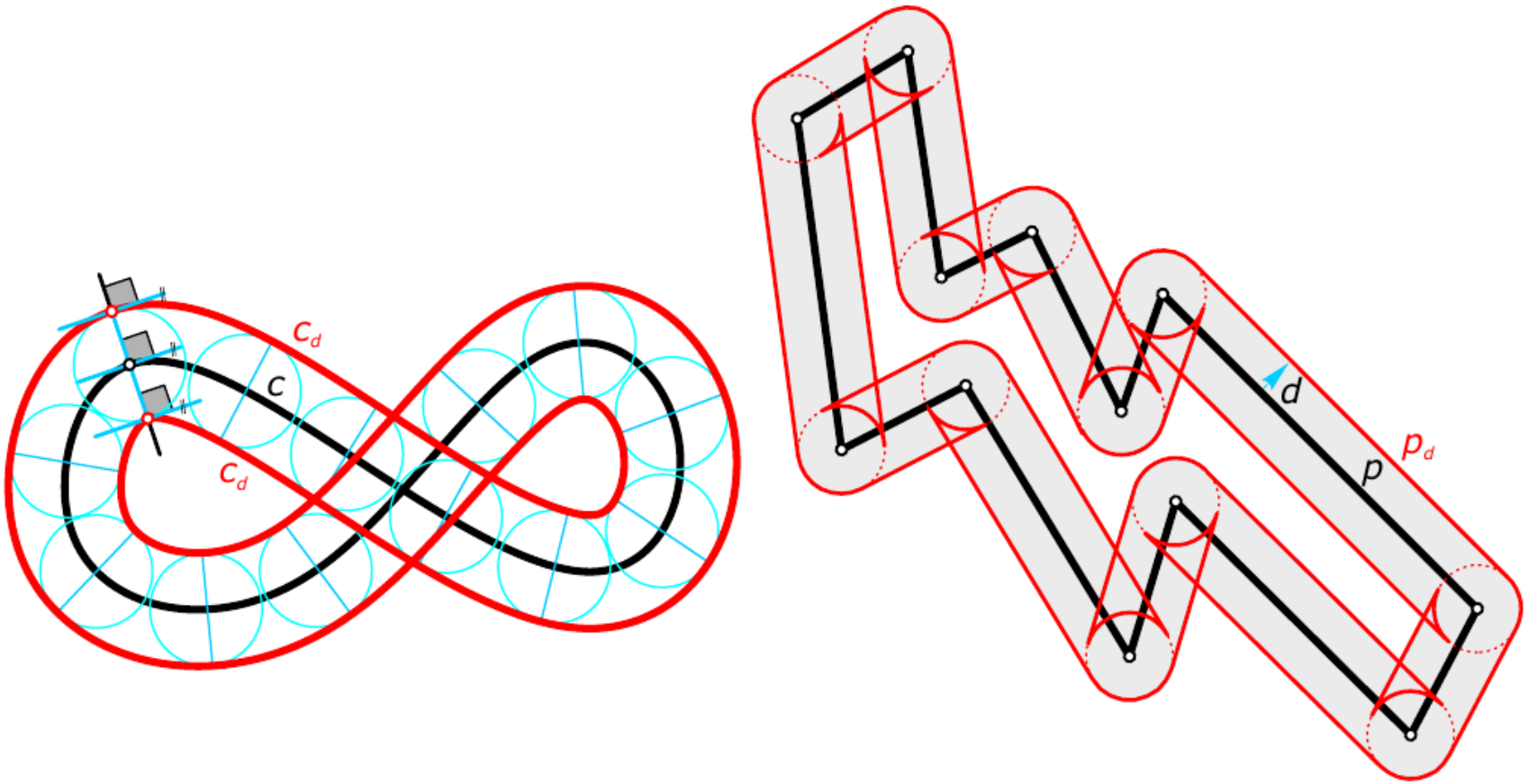


Image source: Pottmann *et al*.

Offset Surfaces

The 3D counterpart to an offset curve is an *offset surface*.
If $\boldsymbol{S}(u, v)$ is a parametrisation of the surface and $\boldsymbol{n}(u, v)$ is the unit normal at (u, v), the offset surface is given by

$$\boldsymbol{S}_d(u, v) = \boldsymbol{S}(u, v) \pm d\boldsymbol{n}(u, v).$$

Here, $\boldsymbol{n}(u, v)$ may be computed as $\boldsymbol{n}(u, v) = \frac{\boldsymbol{S}'_u(u,v) \times \boldsymbol{S}'_v(u,v)}{\|\boldsymbol{S}'_u(u,v) \times \boldsymbol{S}'_v(u,v)\|}$.

Implementing Offset Surfaces

When computing offset surfaces to meshes, keep in mind that:

- meshes are not differentiable in most places, so we need to find the normal in some other way
- the offset surface may intersect itself — we may have to find intersections, delete parts, etc.

Slice Deformations in General

Slice deformations work by slicing an object up into infinitely thin parallel slices, and then transforming each entire slice in some way.

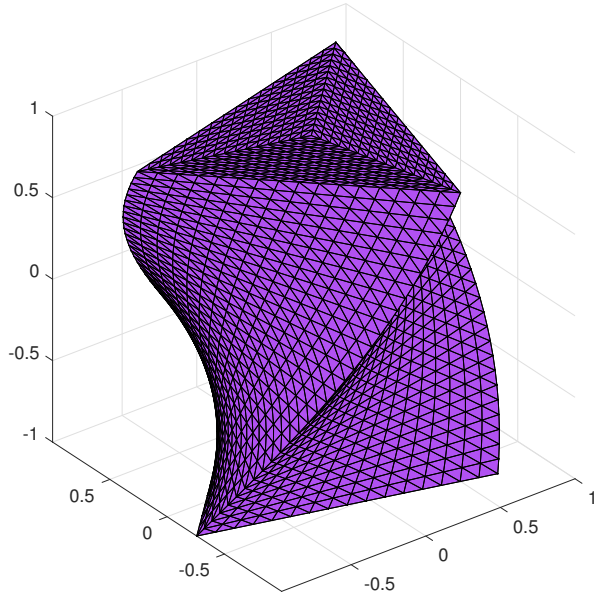
It is most convenient to express slice deformations with slices parallel to e.g. the xy-plane. We need to define what our three coordinate axes for the transformation are, and then perform the operation in this coordinate frame.

One way is to specify a rotation matrix $\boldsymbol{R}$ (new directions of the three coordinate axes) and a point $\boldsymbol{p}_0$ (new origin). Then the transform-local coordinates of $\boldsymbol{x}$ will be

$$\boldsymbol{x}_{\text{local}} = \boldsymbol{R}^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{p}_0).$$

Twisting

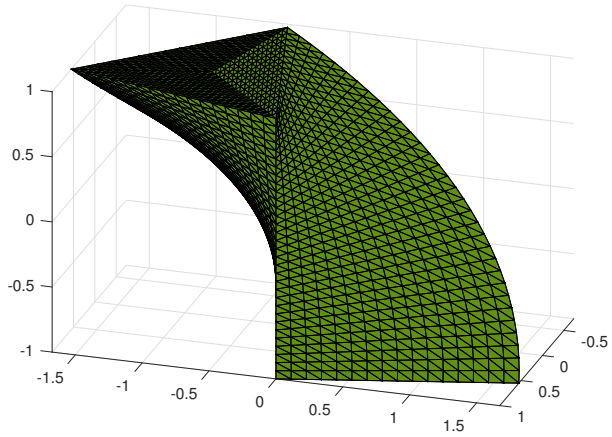
By slicing an object into (infinitely) many thin slices parallel to a fixed base plane, and then rotating each slice about a fixed axis orthogonal to the base plane, we obtain the *twist* deformation.



Typically, the amount each slice is rotated varies linearly along the axis, i.e.

$$\alpha(z) = \alpha_{\text{max}} \cdot \frac{z - z_{\text{min}}}{z_{\text{max}} - z_{\text{min}}},$$

where z_{max} is the largest 'z-value' of the object, and z_{min} is the smallest, but other choices of $\alpha(z)$ are also possible.

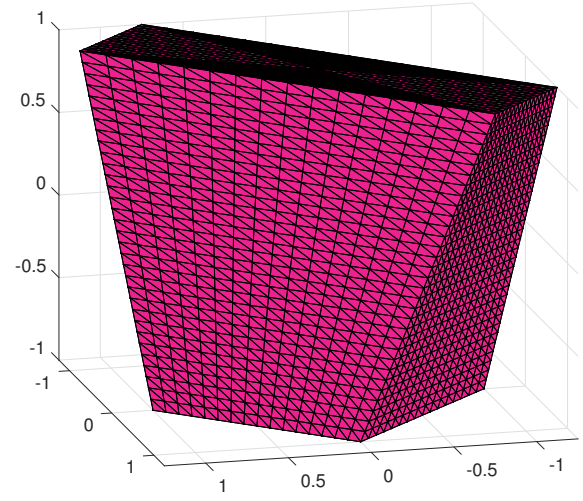


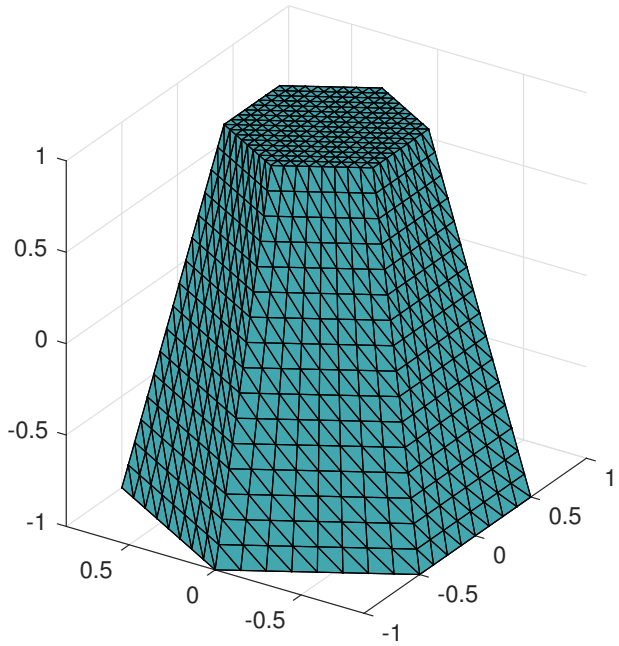
Tapering

The *tapering* deformation scales each slice some prescribed amount along two perpendicular directions in the slice plane.

The most common variant is the *linear tapering*, where the two scaling factors are proportional to the height above the base plane.

To specify a linear tapering, we select the two desired maximum scale factors (along with the new origin $\boldsymbol{p}_0$ and a rotation matrix $\boldsymbol{R}$).





Shearing

The *shear* deformation applies a translation in each slice plane.

For a shearing, we need to specify by a complete coordinate frame and a parametric curve $[0,1] \mapsto \mathbb{R}^3$:

$$\boldsymbol{c}(t) = (c_x(t), c_y(t), 0), \quad t \in [0,1].$$

The shear transformation will then be

$$\boldsymbol{x} \mapsto \boldsymbol{x} + \boldsymbol{c}(t),$$

where $t = \dfrac{z - z_{\text{min}}}{z_{\text{max}} - z_{\text{min}}}$.