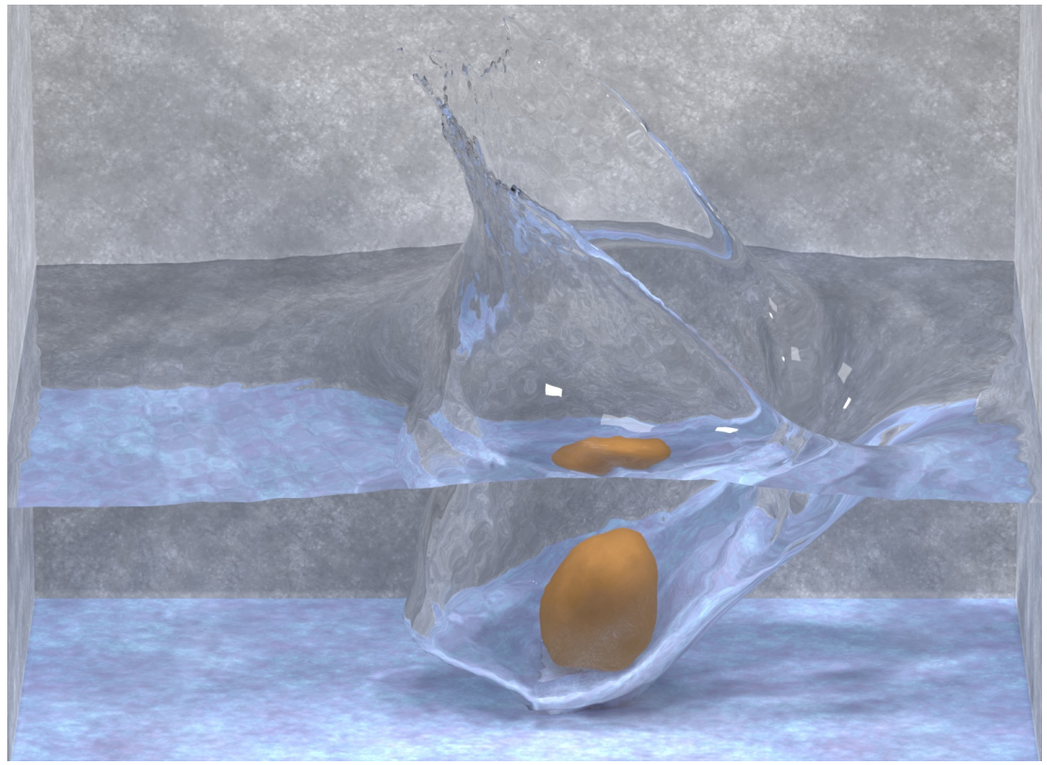
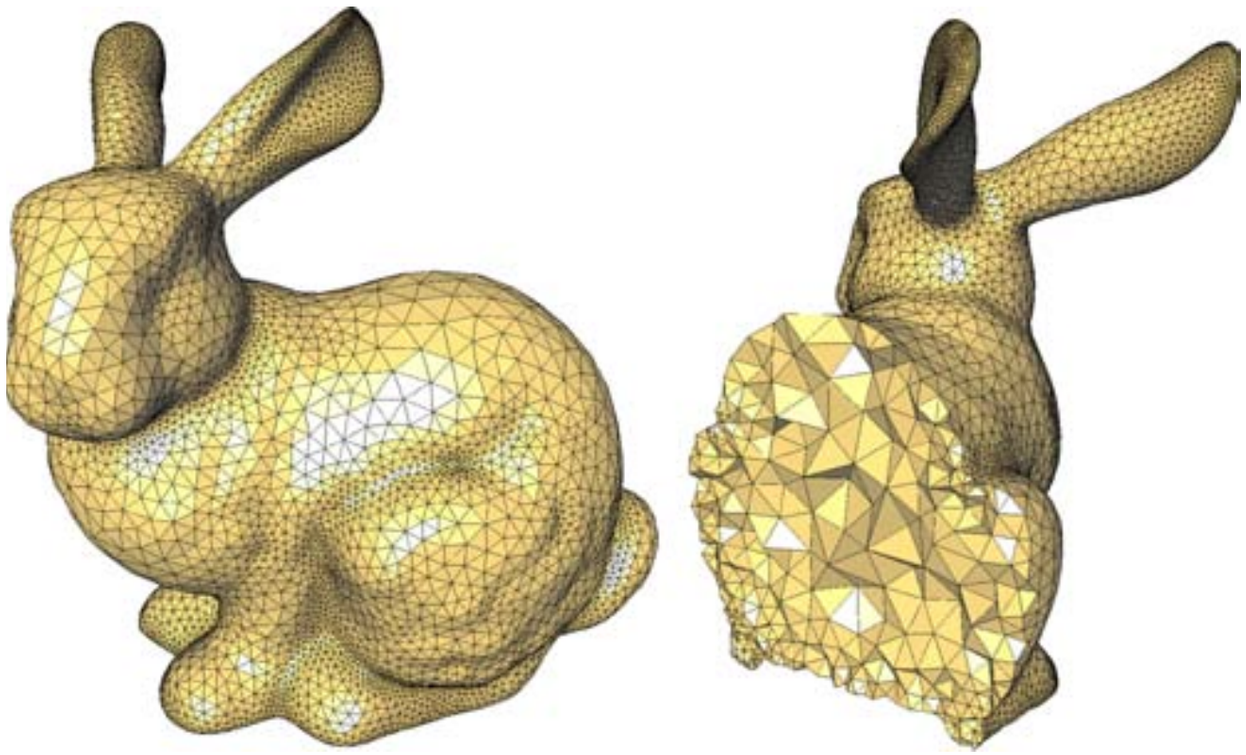
Discrete representations of geometric objects: Features, data structures and adequacy for dynamic simulation.





Next 2 lectures:

Next 2 lectures:

- Describe a number of discrete representations used to encode geometric objects for modeling and simulation purposes
  - Meshes
  - Implicit surfaces
  - Point clouds

Next 2 lectures:

- Describe a number of discrete representations used to encode geometric objects for modeling and simulation purposes
  - Meshes
  - Implicit surfaces
  - Point clouds

- Discuss the features of these representations that are specific to simulation, as opposed to general geometry processing and rendering
  - Objects need to support *dynamic deformation*
  - Volumetric objects need internal structure
  - Discrete geometry needs to be *simulation-quality (well-conditioned)*

Next 2 lectures:

Next 2 lectures:

- Explain the features that make one representation better than another for certain tasks (e.g. meshes vs. implicit surfaces)
  - Static vs. dynamic topology (connectivity)
  - "Shape memory" and deformation drift
  - Regular, structured storage
  - Efficiency of geometric queries

Next 2 lectures:

- Explain the features that make one representation better than another for certain tasks (e.g. meshes vs. implicit surfaces)
  - Static vs. dynamic topology (connectivity)
  - "Shape memory" and deformation drift
  - Regular, structured storage
  - Efficiency of geometric queries
- Outline conversion methods between different geometric representations, e.g.
  - Tetrahedral meshing
  - Marching cubes, marching tetrahedra
  - MLS surface reconstruction, etc.

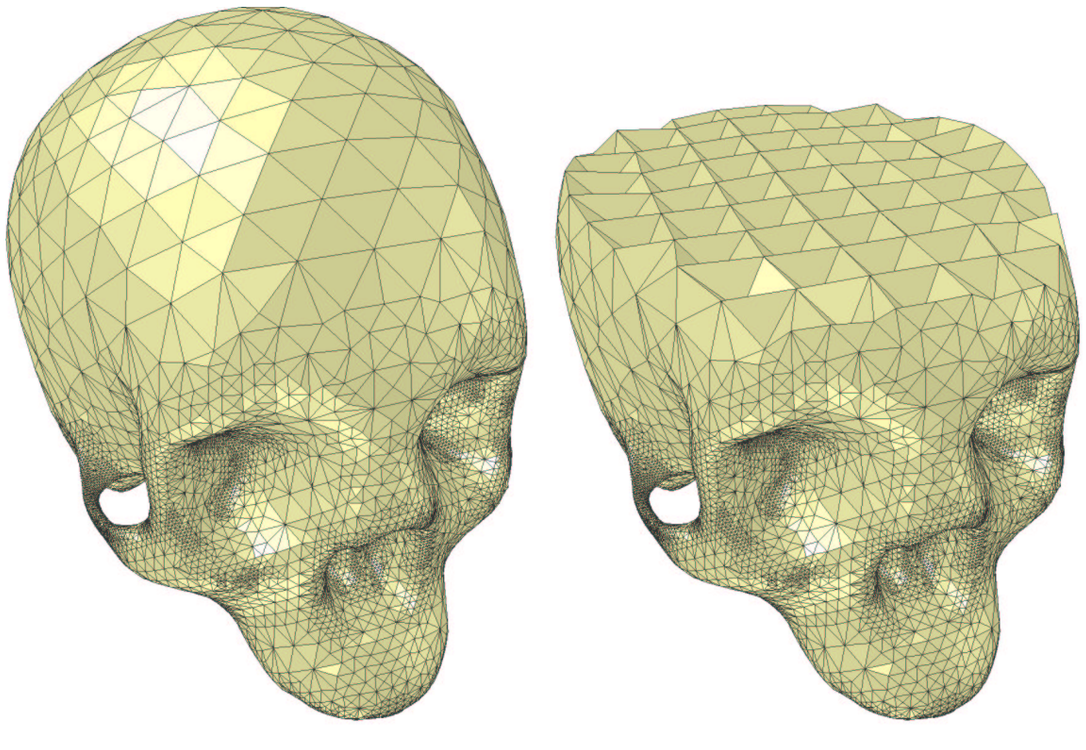
Next 2 lectures:

- Explain the features that make one representation better than another for certain tasks (e.g. meshes vs. implicit surfaces)
  - Static vs. dynamic topology (connectivity)
  - "Shape memory" and deformation drift
  - Regular, structured storage
  - Efficiency of geometric queries
- Outline conversion methods between different geometric representations, e.g.
  - Tetrahedral meshing
  - Marching cubes, marching tetrahedra
  - MLS surface reconstruction, etc.

Next topic : Introduction to PhysBAM data structures and scene layout

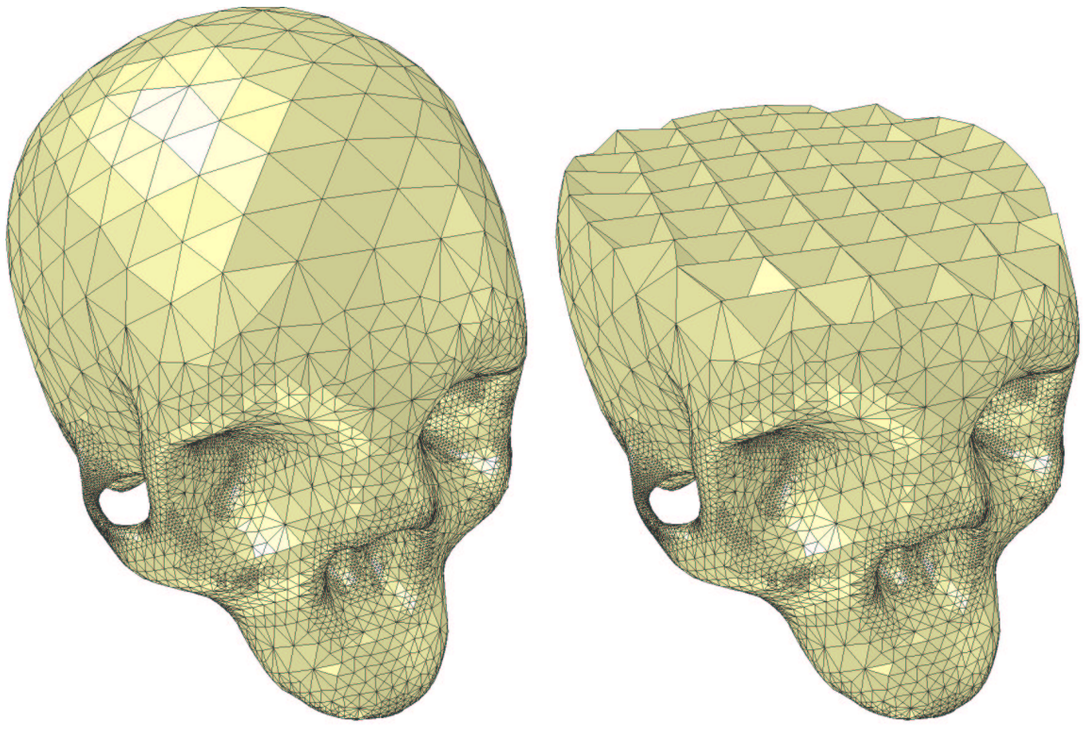
Meshes

Meshes



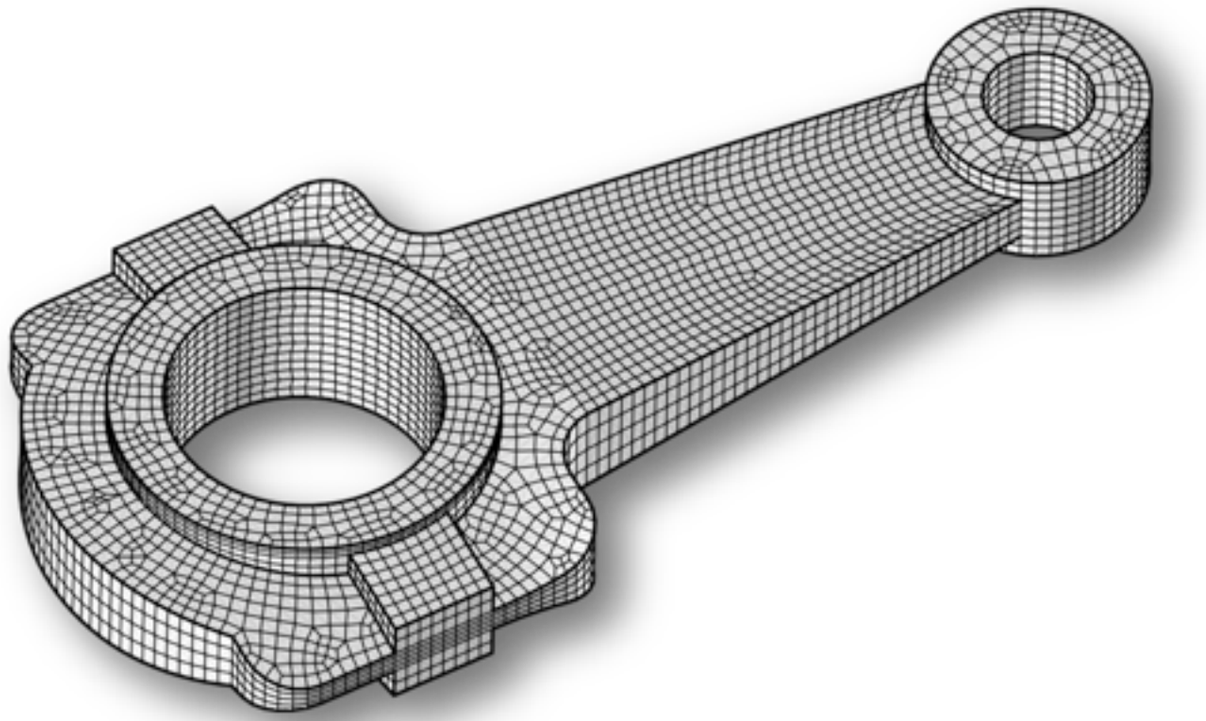
Tetrahedral meshes
(volumetric)

Meshes



Tetrahedral meshes
(volumetric)

Hexahedral meshes
(volumetric)

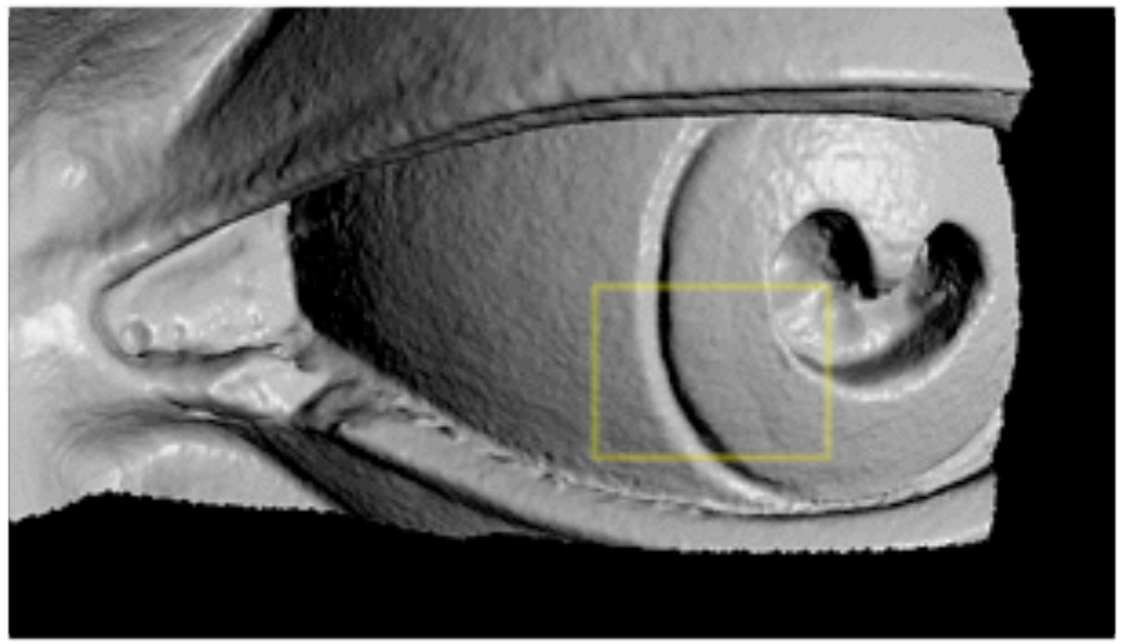


Meshes

Meshes

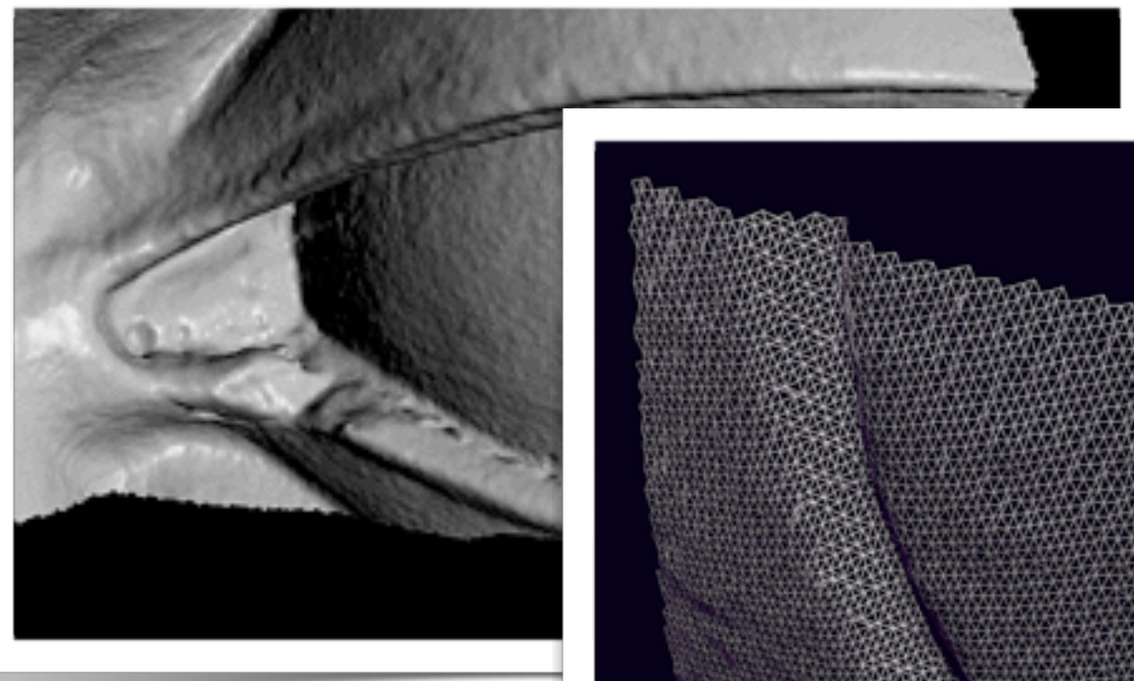
Triangular *surface* meshes
(not volumetric)

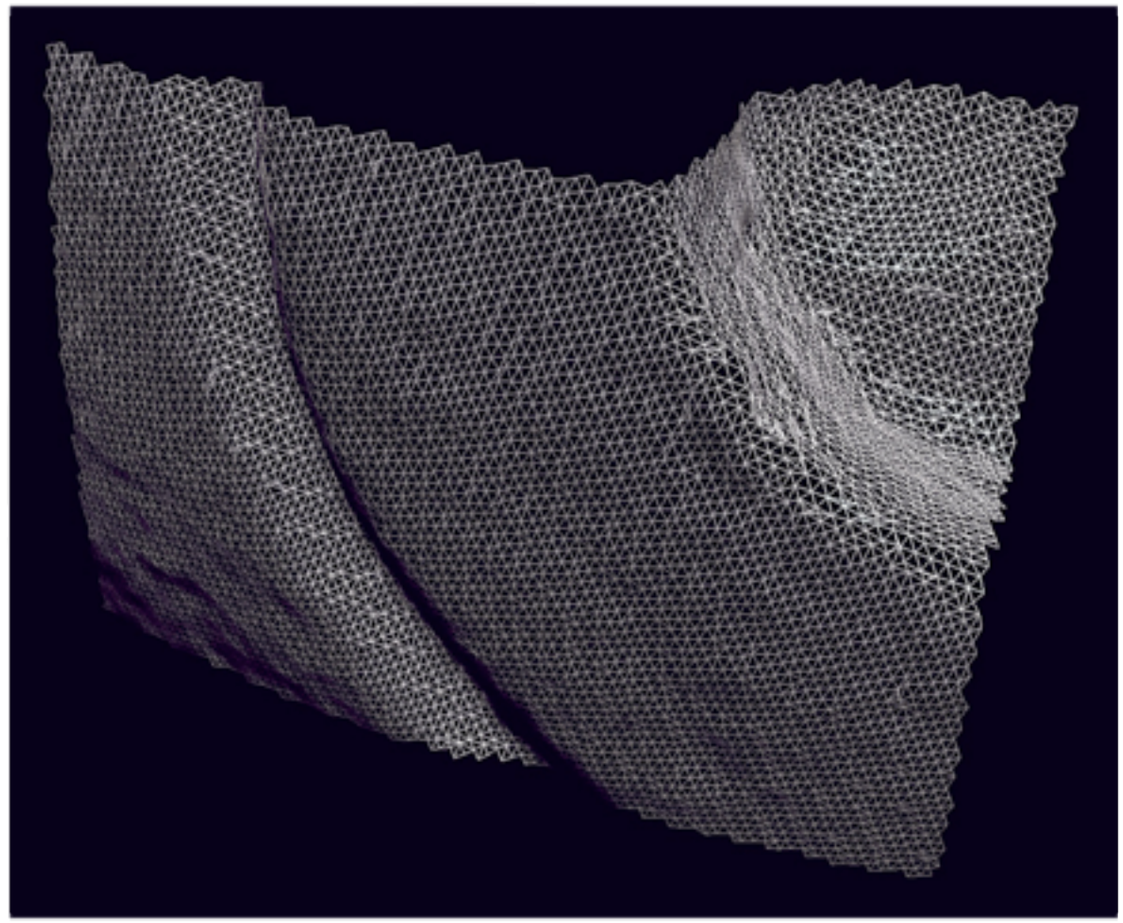
Meshes



Triangular *surface* meshes
(not volumetric)

Meshes





Triangular *surface* meshes
(not volumetric)