Tetrahedron Meshing

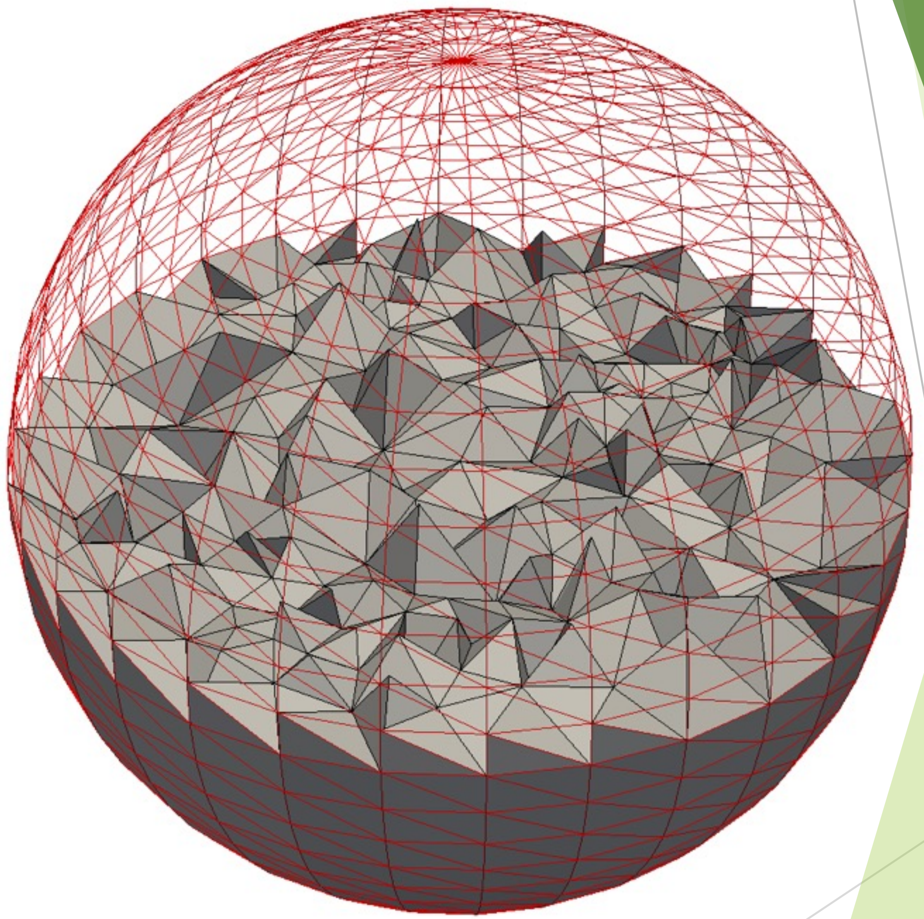
Nehar Poddar

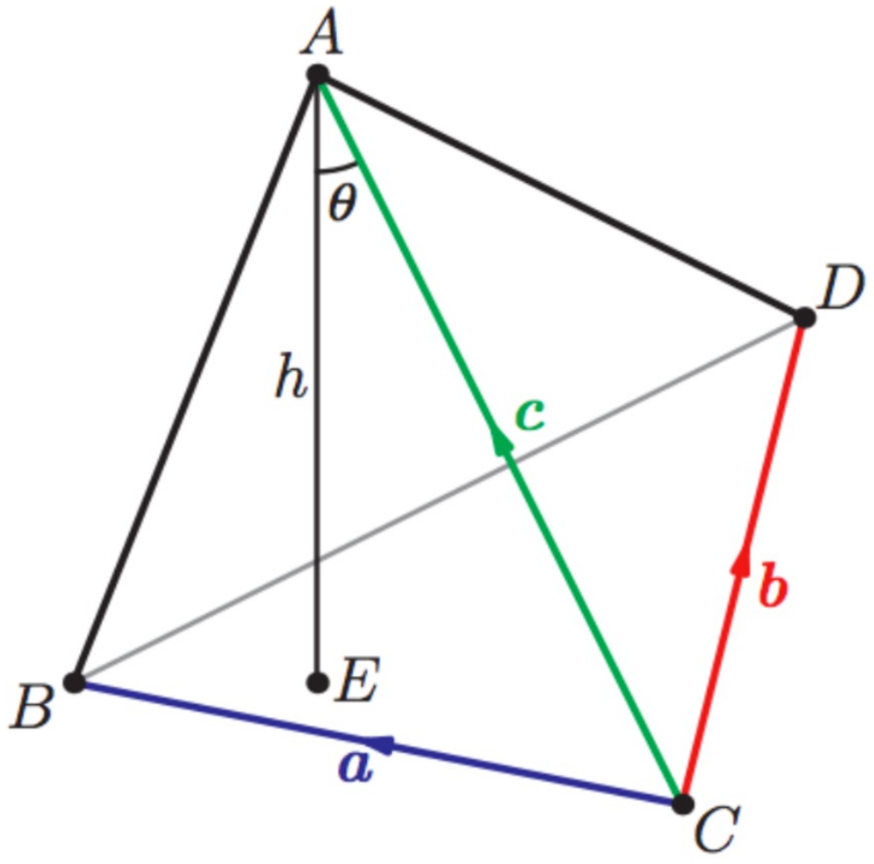
Numerical Analysis

Mini – Project 2

Tetrahedral Volume Meshing

- The growing importance of subject-specific modeling and simulation in medical applications has increased the need of automatic techniques for creating high-quality meshes directly from medical data.
- Tetrahedral mesh generation from volumetric data is one of the most intricate problems in modeling and simulation for medical applications.
- Difficulty comes from several complex steps that are prone to numerical error
- Steps to generate a volumetric mesh :
  - Get a 3D object
  - Iso-surface extraction
  - Surface mask
  - Mesh orientation and verification
  - Oriented surface mesh
  - Tetrahedral Mesh Generation
  - Volumetric Mesh





Volume Of a Tetrahedron

Vertices:

$A = (x_1, y_1, z_1)$
$B = (x_2, y_2, z_2)$
$C = (x_3, y_3, z_3)$
$D = (x_4, y_4, z_4)$

Vectors :

$\overrightarrow{AB} = (x_2 - x_1)i + (y_2 - y_1)j + (z_2 - z_1)k$
$\overrightarrow{AC} = (x_3 - x_1)i + (y_3 - y_1)j + (z_3 - z_1)k$
$\overrightarrow{AD} = (x_4 - x_1)i + (y_4 - y_1)j + (z_4 - z_1)k$

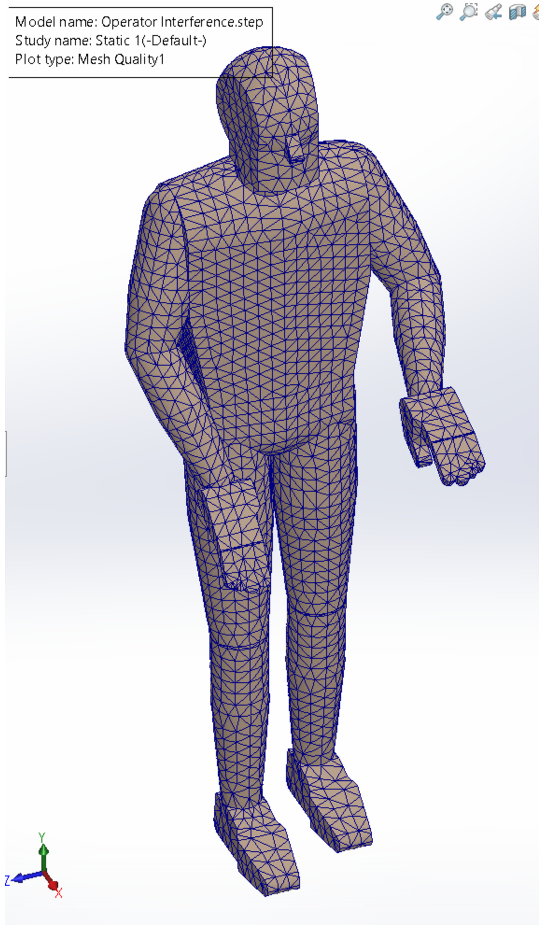
Volume of tetrahedron :

$$\frac{1}{6} \times |\textit{scalar triple product of the three vectors}|$$

$$\frac{1}{6} \times |\{(\overrightarrow{AB} \times \overrightarrow{AC}) \cdot \overrightarrow{AD}\}|$$

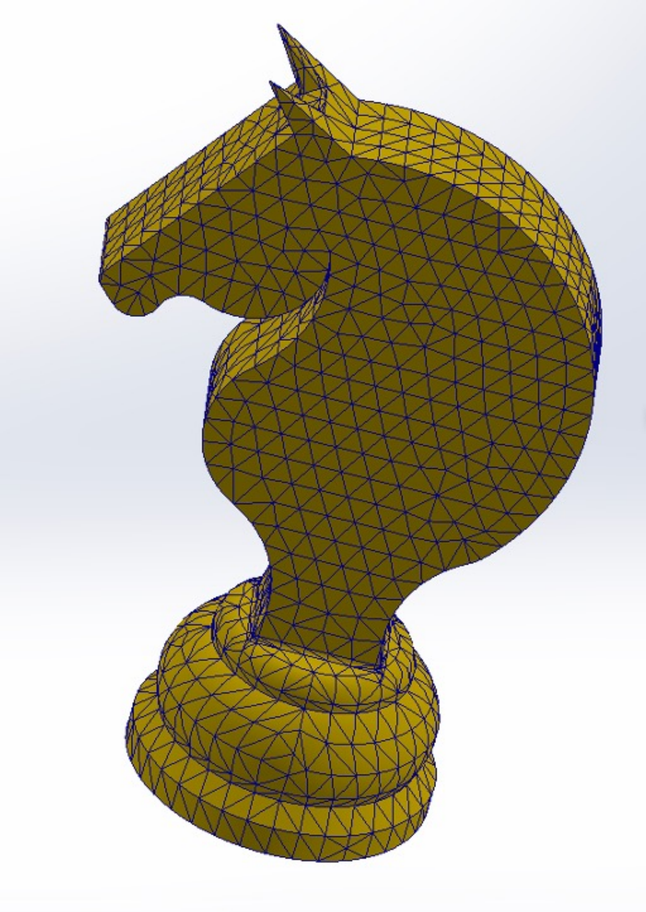
$$\frac{1}{6} \times \det \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ x_4 - x_1 & y_4 - y_1 & z_4 - z_1 \end{bmatrix}$$

Algorithm

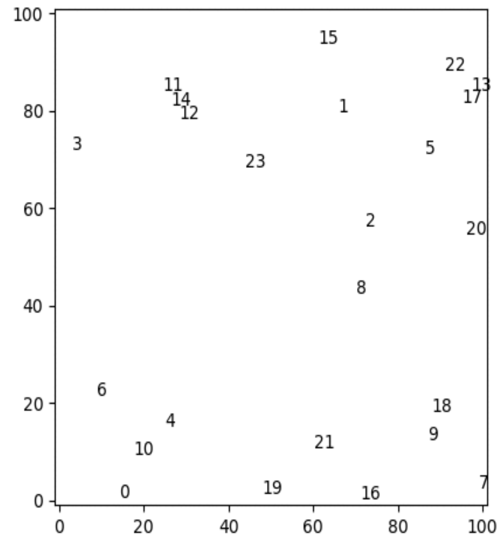


1. Download **CAD** file
2. **Surface mesh** object in SolidWorks
3. Export to **.STL** file
4. Download **Pymesh** Library in Python
5. From Surface Mesh extract **Faces & Vertexes**
6. Using **TETGEN** generate the Tetrahedral Volume Mesh
7. From Volume Mesh read the **Voxels & Vertexes**
8. Iterate through each Voxel (it comprises of 4 Vertices)
9. Go to the index of the Vertex get X,Y,Z coordinates of all 4 Vertexes
10. From the Coordinates get the **3 Vectors**
11. Do **Scalar product** of the vectors
12. Calculate the **Determinant** of the Matrix and divide by 6 to get the volume of each Voxel/Tetrahedron
13. Add up all the Volumes to get **Total Volume**

	No. Of Vertices Of 2D Mesh	No. of Faces Of 2D Mesh	No. Of Vertices Of 3D Mesh	No. Of Faces Of 3D Mesh	No. of Tetrahedrons/ Voxels	Volume
Sphere	642	1280	699	1280	1969	4.152 (4.18879)
Cylinder	114	224	253888	79418	1500998	783751.355 (785398.163)
Knight	2321	4638	93977	13048	370066	25608.972
Person	4461	8918	786452	89367	789226	732991.271
Cow	2903	5804	638971	37827	628321	111562.498



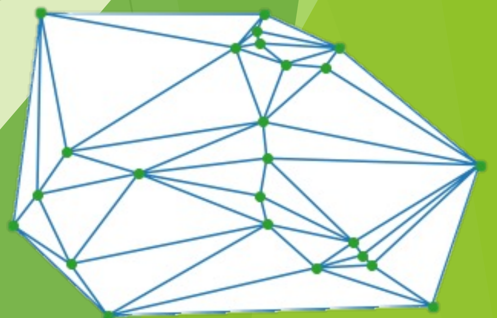
Delaunay Tetrahedralization



Assume that V is a finite point set on a two-dimensional real number field, edge **e** is a closed line segment composed of points in the point concentration as the end point, and **E** is a set of e. Then a triangulation T=(V,E) of the point set V is a plane graph G, which satisfies the conditions:

- Except for the endpoints, the edges in the plane graph do not contain any points in the point set.
- There are no intersecting edges.
- All faces in the plan view are triangular faces, and the collection of all triangular faces is the convex hull of the scattered point set V.

- The most commonly used triangulation in practice is Delaunay triangulation, which is a special kind of triangulation that meet the following properties:
  - Delaunay triangulation is unique , in Delaunay triangulation there will be no other points within the circumcircle of any triangle
  - A triangle is formed by the three nearest points, and each line segment does not intersect.
  - No matter where the area starts from, the final result will be consistent.
  - If the diagonals of the convex quadrilateral formed by any two adjacent triangles are interchangeable, then the smallest angle among the six internal angles of the two triangles will not become larger.
  - if the smallest angle of each triangle in the triangulation is arranged in ascending order, the arrangement of the Delaunay triangulation will get the largest value.
  - Adding, deleting, or moving a vertex will only affect the adjacent triangle.
  - The outermost boundary of the triangular mesh forms a convex polygon shell.
- Many algorithms for computing Delaunay triangulations rely on fast operations for detecting when a point is within a triangle's circumcircle and an efficient data structure for storing triangles and edges.



Mesh Generation Tools

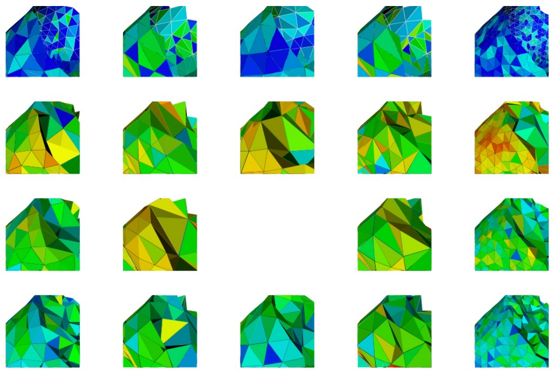
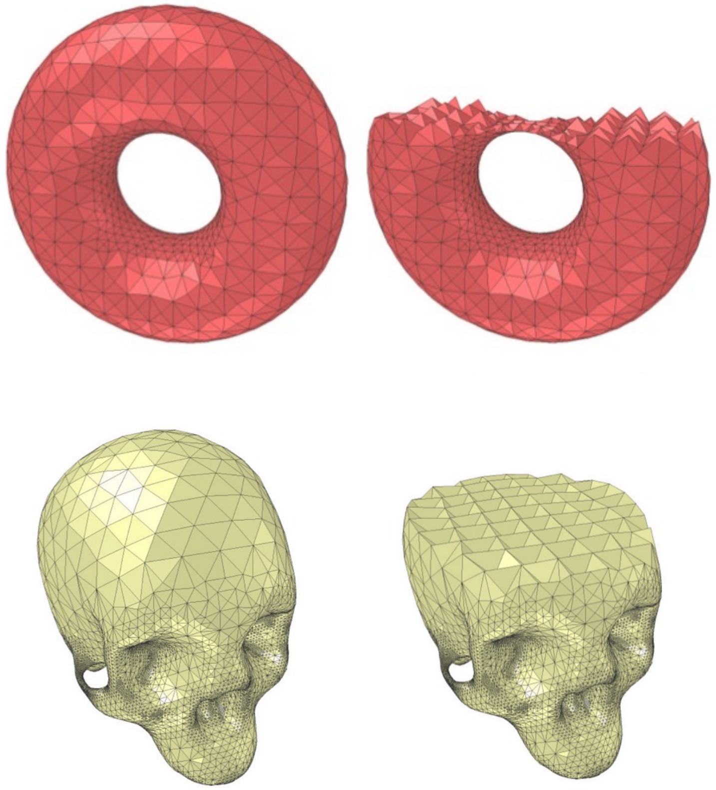


Figure 3: Silicium meshes from left to right: Afront, Macet, Dual Contouring, Marching Cubes, CGAL. Top to botton: Surface mesh, zoomed surface mesh, TetGen, NetGen,

- TetGen
  - TetGen is a suite of techniques used to generate different tetrahedral meshes from three-dimensional point sets or domains with piecewise linear boundaries.
  - The algorithm it uses is called Constrained Delaunay Tetrahedralization (CDT).
  - The TetGen implementation of the CDT is based on the incremental edge flipping algorithm.
  - This algorithm preserves triangles in the boundary, thus making the edge flipping test is very sensitive to the quality of the input mesh, and in some situations numerical problems are so severe that no CDT is generated.

- NetGen
  - NetGen is an automatic 3D advancing-front tetrahedral mesh generator that accepts input from constructive solid geometry (CSG) or boundary representations (BRep) from the STL file format.
  - NetGen contains modules for mesh optimization and hierarchical mesh refinement.
  - NetGen is open source for Unix/Linux and Windows.
- CAMAL
  - The CUBIT Adaptive Meshing Algorithm Library (CAMAL) contains several of the CUBIT project's mesh generation algorithms.
  - CUBIT's goal is robust and unattended mesh generation of complex geometries, scalable to millions of elements and thousands of parts.
  - CUBIT is best known for its pioneering work on automated quadrilateral and hexahedral mesh generation, but also maintains robust triangle and tetrahedral meshing technologies.
  - The tetrahedral mesh generation tool included in CAMAL is TetMesh-GHS3D, a package to automatically create tetrahedral meshes from of closed triangular surface meshes, with little or no user interaction.
  - The implementation is based on the algorithm that corresponds to another variation of a Constrained Delaunay Triangulation, and is, therefore, sensitive to the quality of the input mesh and may be prone to generate tetrahedral meshes with slivers if the boundary mesh does not maintain high quality.

Tools Used and References



- SolidWorks
- MeshLab
- Python Library – PyMesh
- C++ Library – TetGen
- http://www.sci.utah.edu/publications/lizier08/meshing-comp.pdf
- https://graphics.stanford.edu/papers/meshing-sig03/meshing.pdf
- https://towardsdatascience.com/delaunay-triangulation-228a86d1ddad