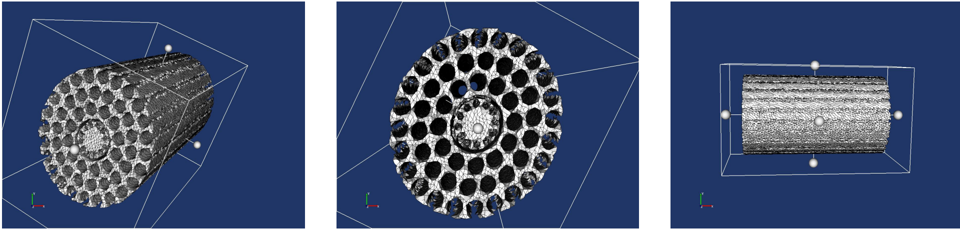
Constrained Delaunay Tetrahedralization(CDT) for boundary preserving mesh generation

Pranav Kant Gaur

Computer Division,
Bhabha Atomic Research Centre, Mumbai, India

Motivation

- CFD analysis for Nuclear reactor component modelling:
  - Designing reactor core:



(a) (b) (c)

Figure: Fuel bundle

Motivation(contd.)

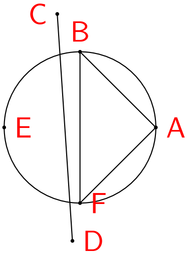
- Anupravaha
  - General purpose CFD solver over hybrid unstructured grid.
  - Collaborative project between BARC and other academic institutions.
  - Integrated mesh generation and CFD solver:
    - Currently, no open source software with iterative feedback based hybrid mesh generation capability.
    - Need to have tighter control over configurability of mesh generation and solver tools.

Motivation(contd.)

- Resulting mesh should have elements suitable for finite element calculations:
  - Constrained Delaunay Tetrahedralization is optimal domain discretization approach for Finite element applications [1].
    - Preserves input domain in output mesh.
    - Shares characterstics with Delaunay triangulation [2]

Constrained Delaunay Tetrahedralization

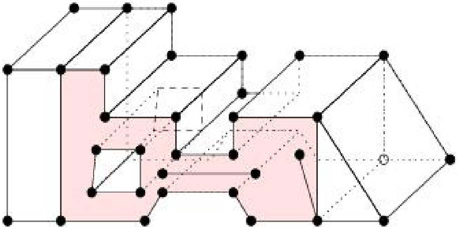
- Simplical complex: A topological space formed by gluing together points, line segments, triangles and their *n*-dimensional counterparts.
- A simplical complex is *constrained Delaunay* if there exist a circumsphere which encloses no other *visible* point inside it.



- Constrained Delaunay Tetrahedralization is a simplical complex of constrained Delaunay cells.

Theoretical underpinnings

- Piecewise linear complex(PLC):
  - Representation of input constraints.
  - Polyhedra are a special case.
  - Can contain holes, isolated vertices.

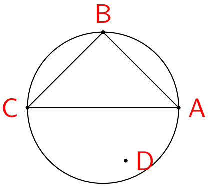


(a) A PLC

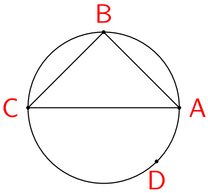
- Given input PLC, compute its CDT.

Theoretical underpinnings(contd.)

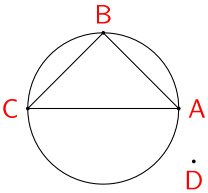
- Schewchuck's CDT existence theorem [2]:
  - A PLC X has a d-dimensional CDT if each k-dimensional constraining facet in X with $k \leq d - 2$ is a union of strongly Delaunay k-simplices.



(b) D is inside: Not strongly Delaunay



(c) D is on boundary: Not strongly Delaunay



(d) Strongly Delaunay

Theoretical underpinnings(contd.)

- Hang Si's Theorem [3]
  - If X has no local degeneracy and DT of X contains all segments of X, then CDT of X exists.

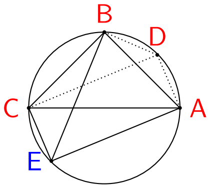


Figure: **Local degeneracy**: *ABCD* and *ABCE* are neighbouring Delaunay tetrahedrons

  - Does not require constraints to be strongly Delaunay *globally*.

Hang Si's algorithm

1. Transform X into a *topologically equivalent* X' which satisfies preconditions for Si's theorem:
    1. Compute initial Delaunay Tetrahedralization
    2. Recover constraint segments:
        - Split constraint segments such that resulting subsegments are *Delaunay*.
    3. Remove local degeneracies:
        - Use symbolic perturbation scheme.
2. Recover constraint facets:
    - Cavity retetrahedralization.

Implementation

CDTGenerator class:

- Data members:
  - plc : input piece-wise linear complex
  - DT : intermediate Delaunay triangulation
  - cdtMesh : output mesh
- Member functions:
  - generate() : *public interface*
    1. readInputPLC()
    2. computeDelaunayTetrahedralization()
    3. recoverConstraintSegments()
    4. removeLocalDegeneracies()
    5. recoverConstraintFacets()
    6. removeExteriorTetrahedrons()

Implementation(contd.)

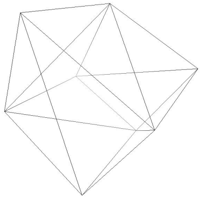
- Unit tests using Google test:
  - Constraint segment recovery and Constraint facet recovery:
    - Checks if all constraint segments and constraint facets are recovered.
- Doxygen inline code documentation
- Continuous integration testing on Travis CI
- Github repository:
  https://github.com/pranavkantgaur/CDTGenerator

Implementation: Using CGAL

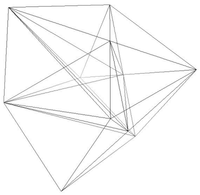
- Geometric object generator package:
    - Generating random ray for performing inside-outside test.
- Combinatorial Maps and Linear Cell complex package:
    - Representing Piece-wise linear complex and output mesh.
- 3D Triangulations package:
    - Representing intermediate Delaunay triangulations.
- 2D and 3D Linear Geometry Kernel:
    - Points, Triangles, Rays.

Platform: Core i3, 4GB RAM, 3.2 GHZ CPU running Ubuntu 14.04 X86_64.

- Cube model:
  - Passes unit tests, executes.



(a) Input: Cube surface mesh



(b) Output: Cube tetrahedral mesh

- Sphere model:
  - Does not terminate in call to remove exterior tetrahedrons from output mesh.
  - Takes more than 10 minutes in facet recovery step(*unacceptable*).

TODOs

- Rewrite facet recovery code.
- Devise efficient algorithm for checking if a 3-cell is inside Linear cell complex.
- Extend the code for adaptive mesh generation with feedback loop from CFD solver.
- Generalize the solution to support Hybrid meshing.

Thank you!!

References

[1] J Shewchuk. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). *University of California at Berkeley*, 73, 2002.

[2] Jonathan Richard Shewchuk. A condition guaranteeing the existence of higher-dimensional constrained delaunay triangulations. In *Proceedings of the fourteenth annual symposium on Computational geometry*, pages 76–85. ACM, 1998.

[3] Hang Si and Klaus Grtner. Meshing piecewise linear complexes by constrained delaunay tetrahedralizations. In ByronW. Hanks, editor, *Proceedings of the 14th International Meshing Roundtable*, pages 147–163. Springer Berlin Heidelberg, 2005.