An exact and efficient 3D mesh intersection algorithm using only orientation predicates
Salles V. G. Magalhães1,2, W. Randolph Franklin2, Marcus V. A. Andrade1
1Universidade Federal de Viçosa, Brazil 2Rensselaer Polytechnic Institute, USA

Intersecting meshes
- Objective: Efficiently compute the exact intersection between two triangular meshes.
- Applications in CAD, GIS, Additive Manufacturing, etc.
- Example: 3D mesh may represent objects in a CAD system.
- Challenges:
  - Special cases and roundoff errors
  - Applications may give inconsistent results or even crash
  - People want exactness and performance.

The algorithm
- Tries to process triangles independently (parallelism)
- Intersect pairs of triangles
- Grid index
- Fast triangle-triangle intersection algorithm (Möller)
- Retessellation:
  - Triangle split at intersection edges
  - Polygonal subdivision is created and retriangulated (ear-clipping)

Special cases
- Triangular classification
  - Input and new triangles are classified.
  - If t was bounding objects (a,b) and is inside object c of the other mesh, in the output t will bound (a∩b∩c) (other booleans → similar strategy)
  - How to determine in what object of the other mesh t is? Traverse mesh and label accordingly
  - Start with an input vertex: point location → location of triangle containing it.
  - Two triangles share a "regular edge" → they are in the same object.
  - Two triangles share an edge generated from an intersection → they are in different objects (triangle labels give the locations).

Novelties
- Parallel: for multi-core computers
- Grid indexing: efficient parallel uniform grid
- Special cases: carefully treated using Simulation of Simplicity (SoS).
- All computation: exact (GMP rationals)
- For triangulated meshes:
  - Widely used
  - Simple representation
  - Supports multi-material and "internal structure"

Data representation
- Triangular soup:
  - Oriented triangles.
  - Each triangle stores the ids of the two objects it bounds (on the negative and positive sides).
- Supports:
  - Multiple components
  - Components with different ids ("materials")
  - Non-manifoldness
  - Nested components
- Self intersections → contradictions

Implementation
- Two versions of each algorithm: one using only orientation predicates.
- Tri-tri intersection: 5 3D orientations for each edge-triangle (Segura and Feito).
- Retessellation: sort intersection points along edges: 3D orientation
- Extracting faces from retesselation: 1D and 2D orientation.
- Ear-clipping: detecting convex vertices and point in triangle → 2D orientation
- Challenge: vertices generated from intersection may be argument of the predicates → represent them as pairs (edge,triangle).

Performance experiments
- Dual-core Xeon, 128 GB of RAM
- Algorithm still under development (can be improved)
- Comparison with LibiGL (exact algorithm, resolves self intersections)

Conclusions, limitations, future work
- Parallel and efficient machines → we can afford exact algorithms.
- Future work:
  - Improve efficiency
  - Validate results
  - Experiments with huge meshes, tetrahedral meshes, etc.
  - Compare with more methods (CGAL, QuickCGS, etc)
  - Floating-point input → exact and more efficient predicates
  - Result is valid for the symbolically perturbed input
  - If output is considered without the perturbation → it may contain polyhedra with volume 0, triangles with area 0, etc.
  - Perturbed output: also useful
  - Future work: how to remove perturbation from output?
  - Source code: freely available (soon on Github)

Acknowledgements

This work was partially supported by Capes (Caféda Sem Fronteiras), FAPEMIG and NExT grant IIS-1117277

Terms of Use: This document has been decomposed into its natural text and visual elements. The natural text is provided as a plain text representation, which may include formatted elements such as headings, lists, and paragraphs. The visual elements, such as tables, diagrams, and images, have been converted into a format that is compatible with the natural text. The overall goal is to present the document in a way that is easily readable and accessible to a wide audience, while maintaining the integrity and accuracy of the original content.