SAND2019-0835C

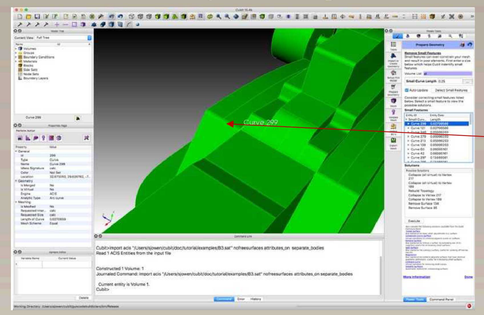
Machine Learning-based CAD Defeaturing for Tetrahedral Mesh Generation

Steve Owen, Tim Shead, Shawn Martin

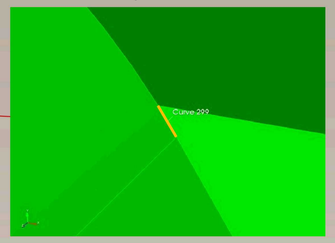
Overview

New machine-learning based methods for driving defeaturing of CAD models for tetrahedral meshing are proposed. The ability to predict mesh quality at geometric features of a CAD model prior to meshing is used to identify potential problem areas. A prioritized list of geometric operations can be presented to a user to improve meshing outcomes. New methods are introduced for generating training data based on both geometric and topological features of the CAD model with labels defined by local quality metrics. Implementation of the proposed machine learning driven defeaturing environment is demonstrated in Sandia's Cubit Geometry and Meshing Toolkit

Cubit Geometry and Meshing Toolkit



Close-up of small feature



Meshing software, such as Cubit, provide tools to identify dirty geometry and CAD operations to "fix" the issues. Geometry cleanup procedures are interactive and time consuming.

Approach

1. Find the most problematic features in the CAD Model
   - For a given local topology
     - Small Curves
     - Small Surfaces
     - Sharp angles
   - Predict local mesh quality
     - Scaled Jacobian
     - In-Radius
2. Find solution that yields best mesh quality
   - For a given local CAD geometry operation
     - Collapse Curve
     - Remove Surface
     - Blunt Tangency
     - Etc..
   - Predict local mesh quality after operation
     - Scaled Jacobian
     - In-Radius

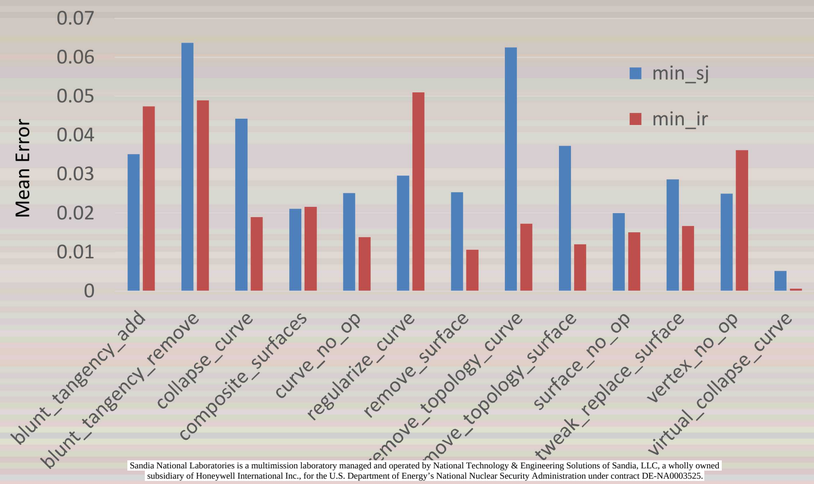
CAD Operations Trained

	Operation	Entity 1	Entity 2
1	Vertex No Operation	vertex	none
2	Curve No Operation	curve	none
3	Surface No Operation	surface	none
4	Remove Surface	surface	none
5	Tweak Replace Surface	surface	surface
6	Composite Surfaces	surface	surface
7	Collapse Curve	curve	vertex
8	Tweak Remove Topology Curve	curve	none
9	Virtual Collapse Curve	curve	vertex
10	Tweak Remove Topology Surface	surface	none
11	Regularize Curve	curve	none
12	Blunt Tangency Add	vertex	none
13	Blunt Tangency Remove	vertex	None

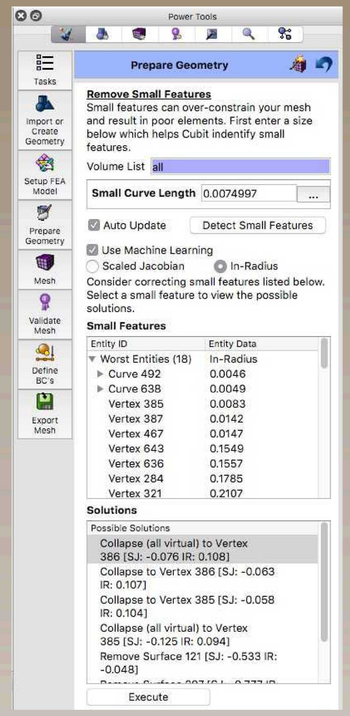
13 different CAD operations were trained. Relevant operations are identified for each small curve, surface or sharp angle. Features computed and mesh metrics are used to label resulting operation.

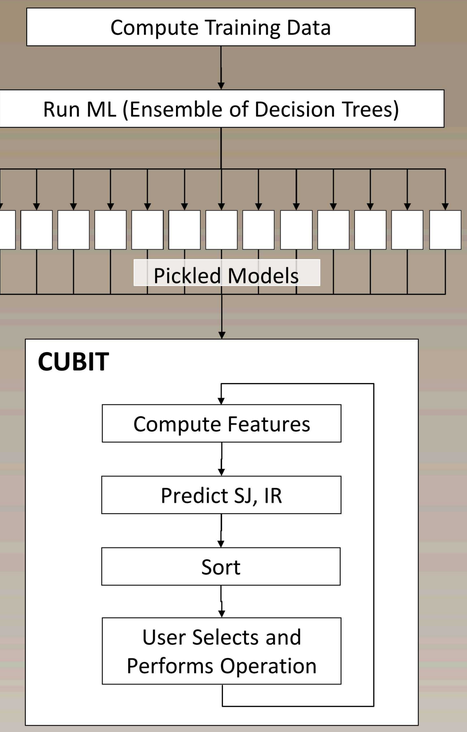
Prediction Accuracy

Many CAD models were trained. Accuracy was determined by training on 50% of training data and validating the remaining 50%. Table shows expected mean error for each trained CAD operation.

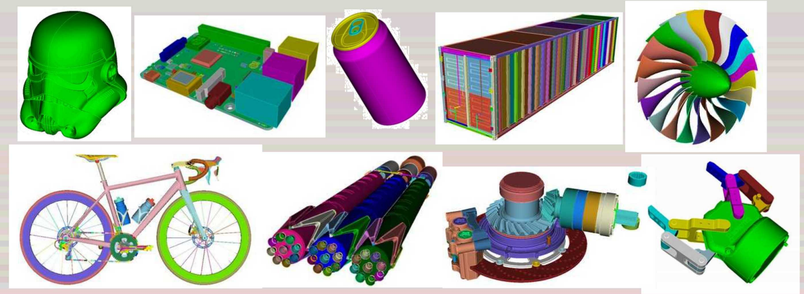


Application



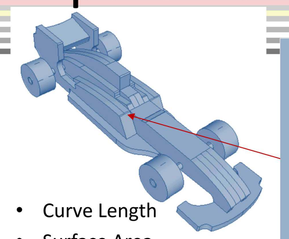


ML-based defeaturing is currently deployed as part of the Cubit Wizard tool for preparing geometry for meshing. Pickled data is loaded once and queried run-time to predict mesh quality at small features. Best CAD operations are presented to the user for "fixing" issues based on predicted metrics.

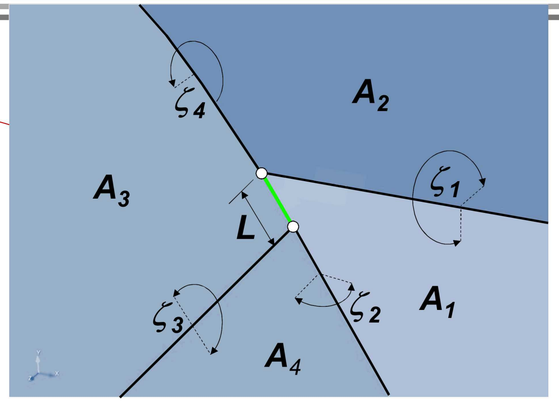


Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Expert Training Features



- Curve Length
- Surface Area
- Angle at vertex
- Angle at curves
- Valence at vertex
- Number of loops
- Hydraulic Radius
- Etc…



Features characterize the local topology for small curves, surfaces and sharp angles in the model. Features are specific to one of 13 trained CAD operations

Training Features

A fundamental challenge applying machine learning to any meshing problem is identifying a useful representation – because a CAD model includes both *topology* (connectivity among vertices) and *geometry* (the embedding of those vertices in 3D space), it cannot be easily shoehorned into typical machine learning algorithms, which expect fixed-length feature vectors as inputs.

The problem is further complicated by questions of locality: when supplying features, should they be based on the entire geometry, a small neighborhood of the geometry, or something in between?

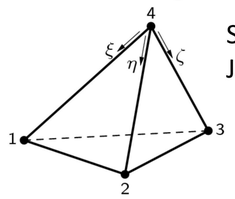
Good features must be able to work within these constraints while providing useful properties such as position and rotation invariance. We are currently evaluating several types of feature as part of this work:

Expert Features are hand-picked by experts based on the individual CAD operation to be performed, and include a small fixed set of local properties of a CAD model, such as curve lengths, adjacent surface angles, and adjacent vertex valences. (shown at left)

Aggregate Features use statistical methods such as histograms to reduce arbitrarily-complex meshes to fixed length feature vectors.

Graph Features explicitly represent the structure of the mesh using undirected graphs with vertex attributes for the geometry. The graphs can be embedded in a relatively low dimension feature space using techniques based on text analysis (*node2vec*) or spectral graph theory. These features can be used for training directly, or with convolutional neural nets for automated feature generation.

Training Labels

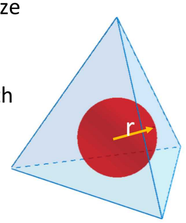


Scaled Jacobian

- Function of angles at vertices
- Scaled -1.0 to 1.0
- Independent of mesh size

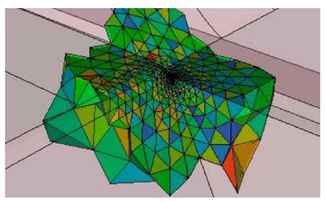
In-Radius Ratio

- Ratio with equilateral tet with target edge length
- Scaled 0.0 to 1.0
- Sensitive to mesh size



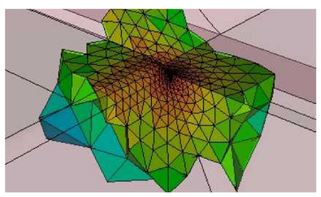
CAD model is meshed and minimum Scaled Jacobian and In-radius extracted near small feature as training labels. Metrics are computed before and after performing CAD operation.

Before Curve Collapse Operation



Min SJ = 0.250

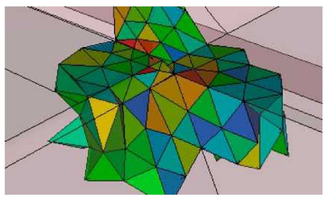
Scaled Jacobian



Min IR = 0.027

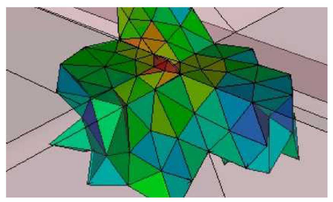
In-radius

After Curve Collapse Operation



Min SJ = 0.227

Scaled Jacobian

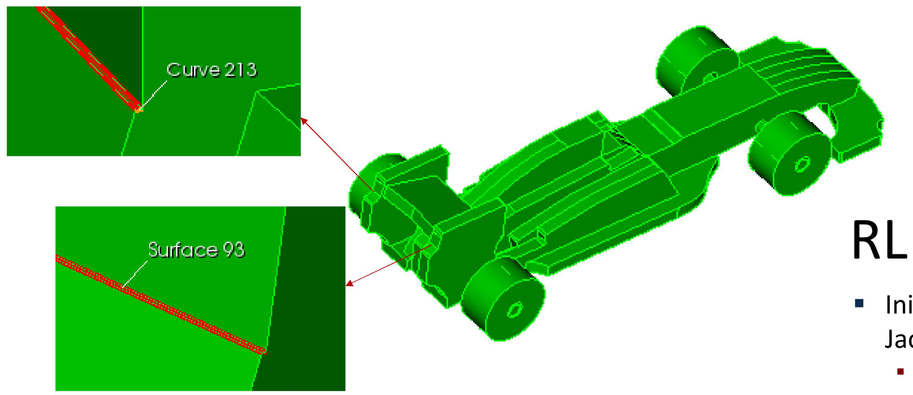


Min IR = 0.267

In-radius

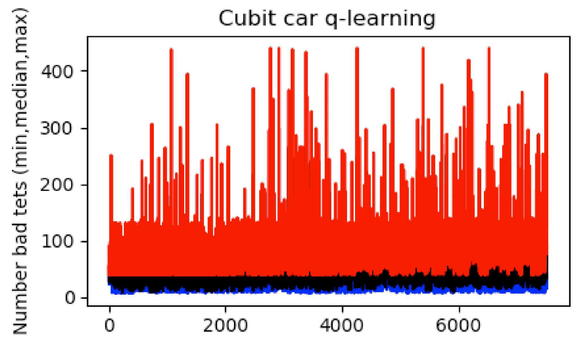
Reinforcement Learning (RL)

- Can we automate the Cubit wizard machine learning approach beyond ranking the current options available to the user?
  - Analogous to GPS mapping software
    - Turn left on Eubank in 100 yds, turn right on Moon in ¼ mile, etc.
    - If you turn the wrong way, new suggestions are made.
    - The user is in control, but the software continues to offer instructions.
- To start, we have implemented separate q-learning/cubit python modules to experiment with automating cubit operations.
  - Modules can be swapped out to provide more advanced q-learning (e.g. deep learning), or different cubit operations (e.g. small surfaces).
  - Our initial results are applied to a CAD model of a toycar, using small curves.



RL Details

- Currently representing state as a binary vector describing whether a small curve has or has not had operation applied.
  - For the car, there are 2^12 = 4096 states
- Currently representing actions on small curves according to category (“remove”, “collapse”, “composite”, “tweak”).
  - For the car there are 4*12 = 48 possible actions.
- Ran q-learning algorithm for 7,500 iterations.



RL Using Small Curve Operations

- Initial q-learning effort reduced number of tets with Scaled Jacobian < 0.2 from 40 to 8 using two cubit commands:
  - `tweak remove_topology curve 213 small_curve_size 0.358497 backoff 3.25907`
  - `remove surface 93 extend`

U.S. DEPARTMENT OF ENERGY
National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525. SAND No. SAND2019-xxxx C