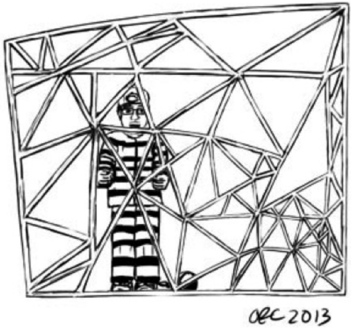
Inserting a Curve into an Existing Two Dimensional Unstructured Mesh



Daniel Zaide and Carl Ollivier-Gooch

Department of Mechanical Engineering

University of British Columbia

This work supported by Intel Corp.

International Meshing Roundtable

October 14th, 2013

Introduction

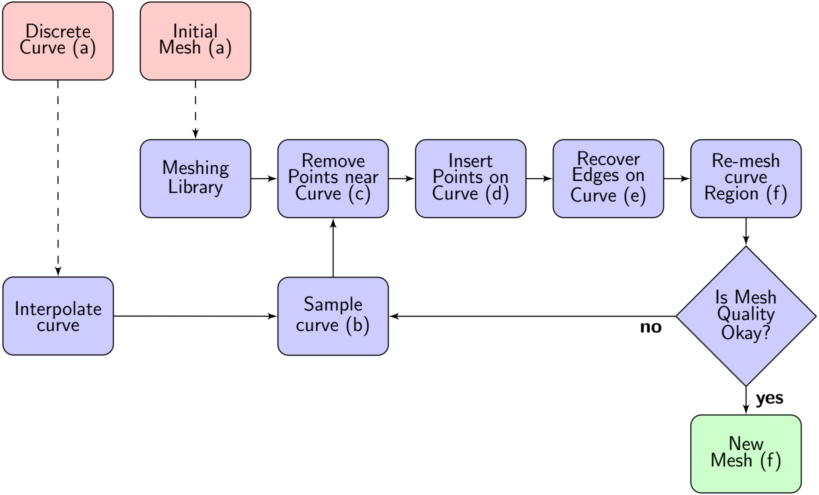
Problem: take a pre-existing mesh and inserting an internal boundary into its topology.

Goal: obtain specified internal boundary in a general, pre-existing mesh with minimal mesh modification.

Motivation: remesh to match the surface of a newly deposited layer of material in the simulation of the semiconductor device manufacturing process.

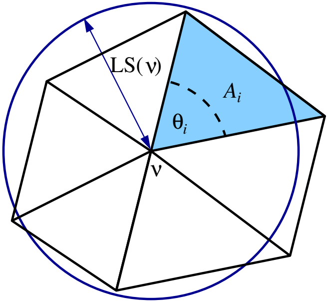
We first consider the 2D problem in this work.

Algorithm Overview



Length Scale

Define a vertex length scale, $LS(v) = \left(\frac{4}{3} \sum_{i=1}^{N} A_i \middle/ \sum_{i=1}^{N} \theta_i \right)^{\frac{1}{2}}$



The length scale at any point within a cell is then defined using barycentric interpolation.

Length Scale Equidistribution

To determine new vertex locations on the curve, we use equidistribution of length scale on a curve with arc length $\ell(x, y)$ on computational domain ξ.

The equidistribution equation is the Moving Mesh PDE,

$$\frac{\mathrm{d}}{\mathrm{d}\xi}\left(\frac{1}{LS(\ell)}\frac{\mathrm{d}\ell}{\mathrm{d}\xi}\right) = 0.$$

We solve approximately using Gauss-Seidel iteration as

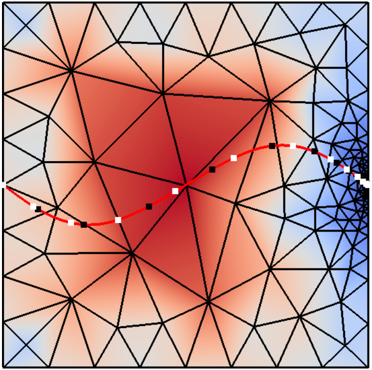
$$\ell_i^{n+1} = \frac{LS\left(\ell_{i-\frac{1}{2}}^{n+1}\right)\ell_{i+1}^{n} + LS\left(\ell_{i+\frac{1}{2}}^{n}\right)\ell_{i-1}^{n+1}}{LS\left(\ell_{i-\frac{1}{2}}^{n+1}\right) + LS\left(\ell_{i+\frac{1}{2}}^{n}\right)}$$

with initial point spacing determined using length scale based marching from one end of the curve.

Length Scale Equidistribution

Example Spacing

The length scale is shown for the following mesh with a curve shown in red.

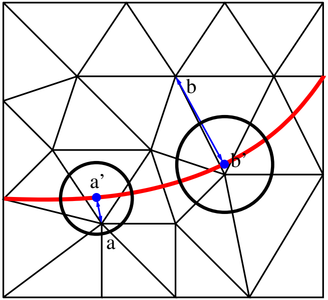


The initial points are in black, and final locations in white.

Curve Insertion and Mesh Cleanup

With new vertex locations determined, we first remove nearby vertices, defined as a half length scale away.

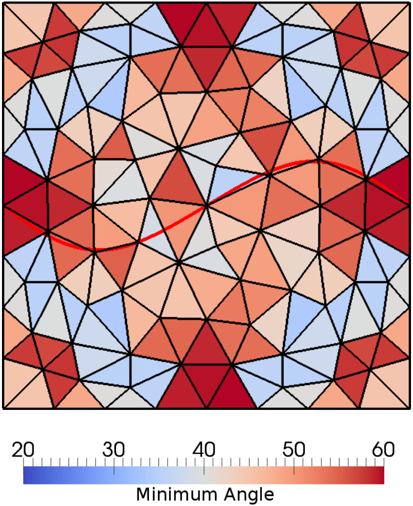
In this representative figure, we remove the vertex at a but keep the one at b.



After cleanup, mesh refinement is done using standard Delaunay circumcenter insertion and along with vertex smoothing.

A Simple Example

Example 1 - Uniform Mesh, Cubic Spline



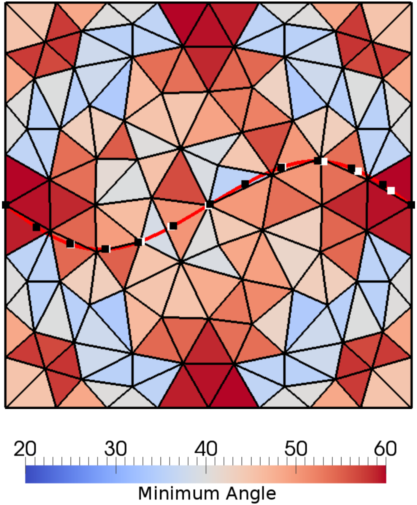
(a) Initial Mesh, 104 Vertices.

Minimum angle: 34.02°.

Cubic Spline in red.

A Simple Example

Example 1 - Uniform Mesh, Cubic Spline

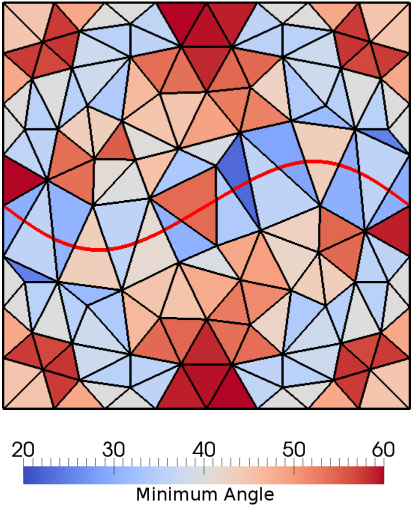


(b) Equidistribution of new vertices.

Initial vertex spacing in white, final vertex spacing in black.

A Simple Example

Example 1 - Uniform Mesh, Cubic Spline

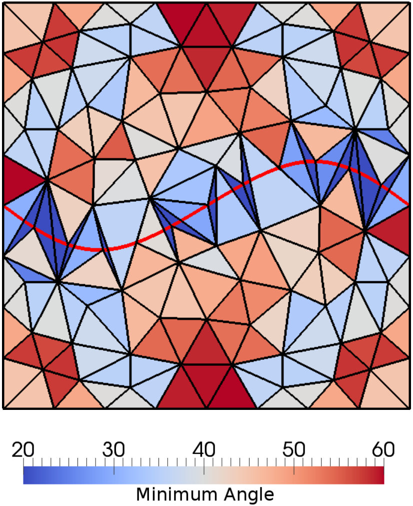


(c) Removing vertices near curve.

With the locations of the new vertices on the curve known, nearby vertices are cleared out, one at a time to maintain a connected mesh.

A Simple Example

Example 1 - Uniform Mesh, Cubic Spline

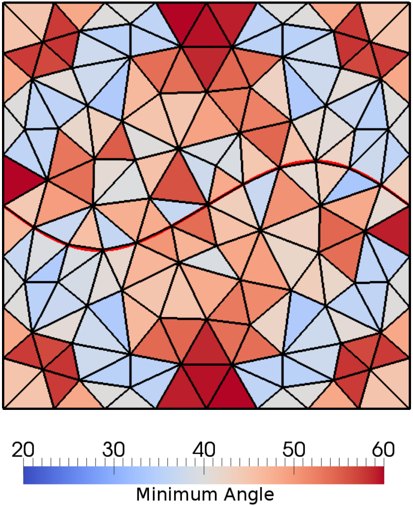


(d) Inserting new vertices on curve.

With vertices near the curve cleared out, the new vertices are inserted into the the mesh, either inside a cell or on an edge.

A Simple Example

Example 1 - Uniform Mesh, Cubic Spline



(e,f) Curve recovery and final mesh.

In this example, no refinement or smoothing is needed.

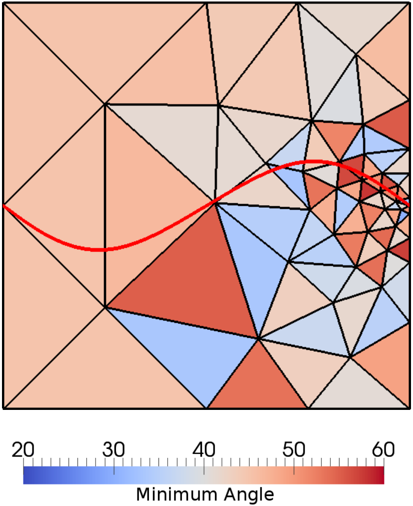
The final mesh has 108 Vertices.

Minimum angle: 33.31°.

91% of initial vertices unchanged.

Results

Example 2 - Graded Mesh, Cubic Spline



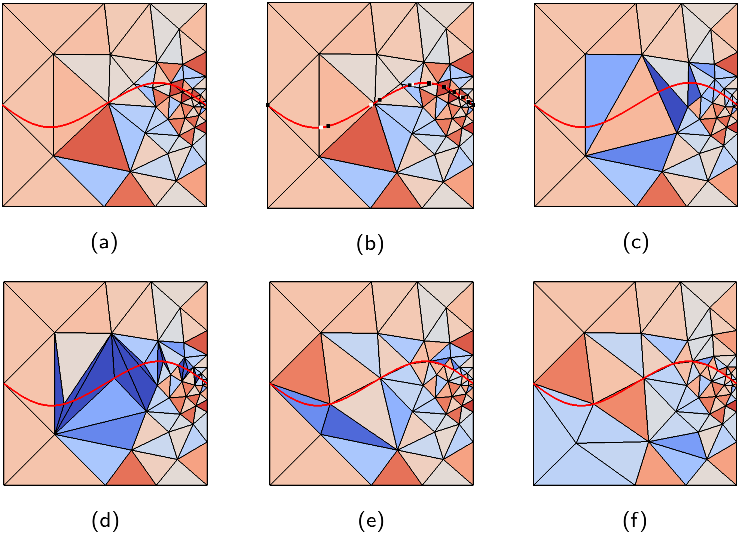
Initial Mesh, 49 Vertices.

Minimum angle: 30.40°.

Mesh has variable length scale.

Results

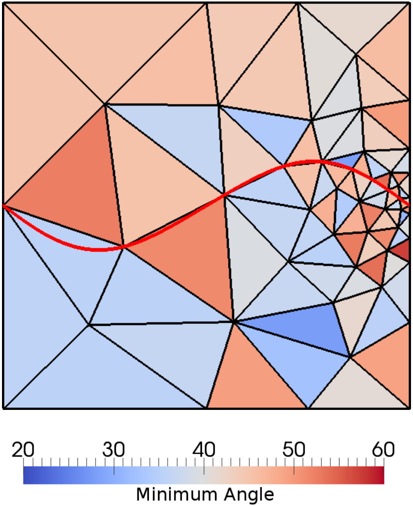
Example 2 - Graded Mesh, Cubic Spline



(a) (b) (c) (d) (e) (f)

Results

Example 2 - Graded Mesh, Cubic Spline



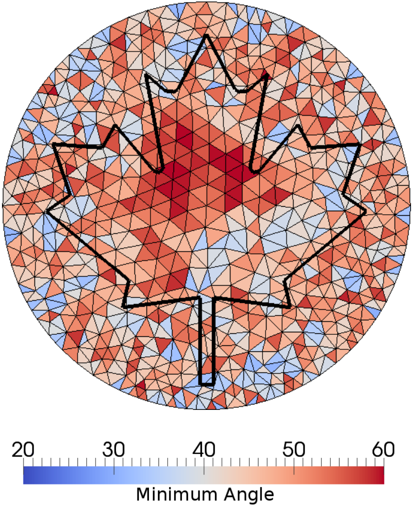
Final Mesh, 49 Vertices.

Minimum angle: 30.18°.

71% of initial vertices unchanged.

Results

Example 3 - Uniform Mesh, Maple Leaf



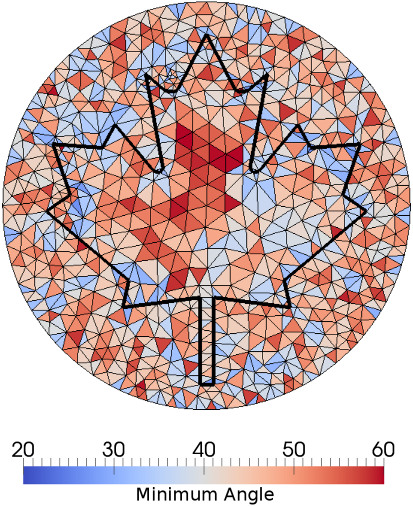
Initial Mesh, 614 Vertices.

Minimum angle: 30.01°.

Set of lines to insert, variable lengths.

Results

Example 3 - Uniform Mesh, Maple Leaf



Final Mesh, 697 Vertices.

Minimum angle: 25.84°.

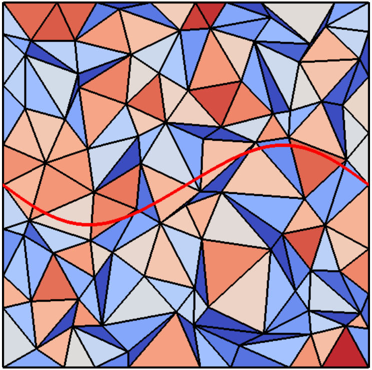
76% of initial vertices unchanged.

Summary

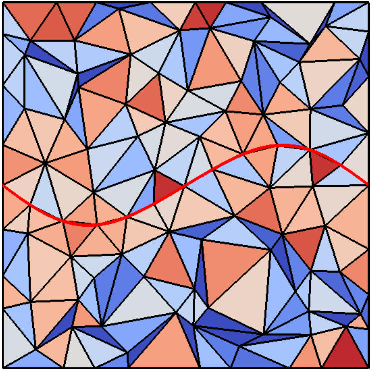
	Initial Mesh		Final Mesh		
Example	Verts	Min $\angle$	Verts	Min $\angle$	% Unch.
1 - Coarse	104	34.02°	108	33.31°	91.3
1 - Medium	8315	30.24°	8356	30.01°	98.6
1 - Fine	40315	30.24°	40411	30.04°	99.3
2 - Coarse	49	30.40°	49	30.18°	71.4
2 - Medium	8254	30.17°	8287	29.04°	98.5
2 - Fine	33952	30.06°	34054	30.06°	99.0
3 - Coarse	614	30.01°	697	25.84°	76.2
3 - Medium	14601	30.07°	14843	26.63°	95.5
3 - Fine	57077	30.30°	57499	26.78°	97.7

Results

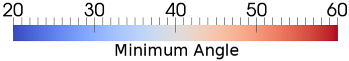
Example 4 - Bad Uniform Mesh, Cubic Spline



(g)

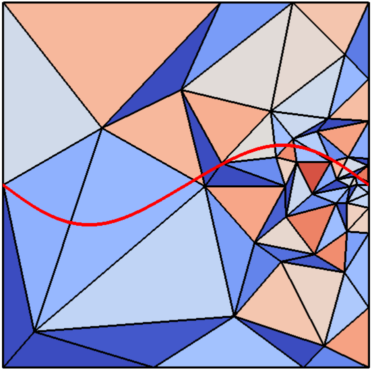


(f)

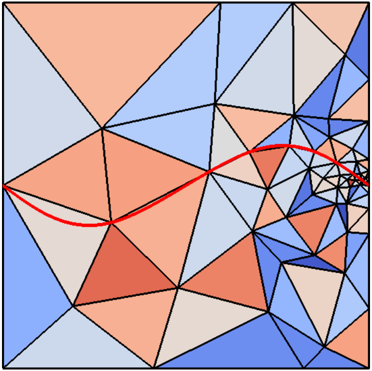


Results

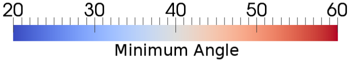
Example 4 - Bad Graded Mesh, Cubic Spline



(g)

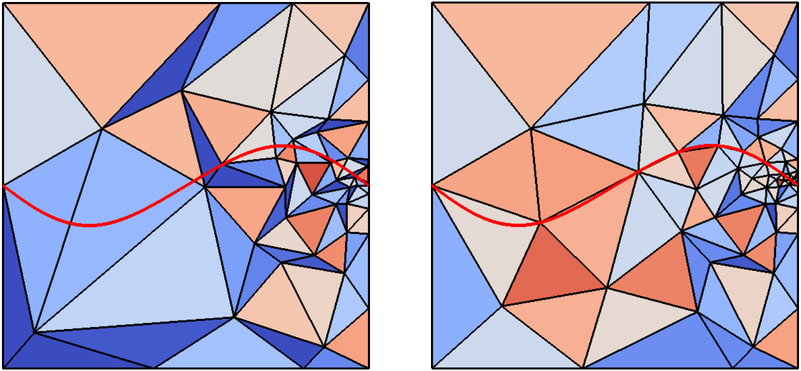


(f)

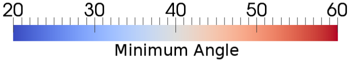


Results

Example 4 - Bad Graded Mesh, Cubic Spline

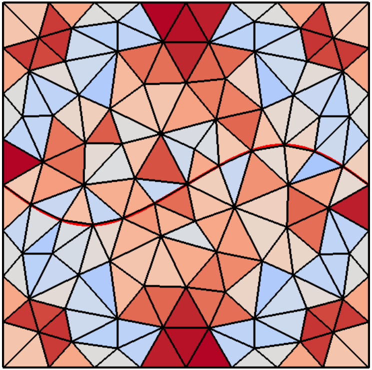


(g) (f)

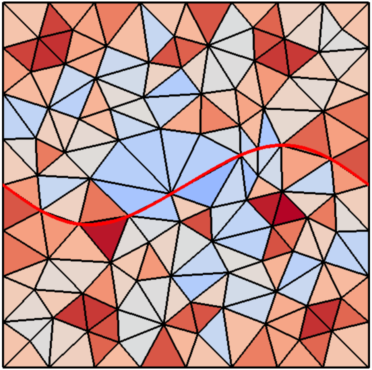


Results

Comparison Between Methods - Example 1



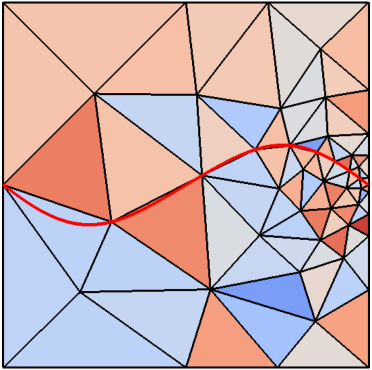
Initial mesh generated, then curve inserted.



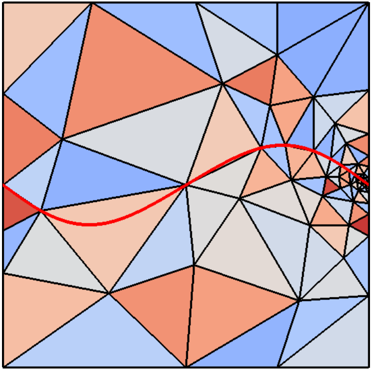
Mesh initially generated with internal boundary.

Results

Comparison Between Methods - Example 2



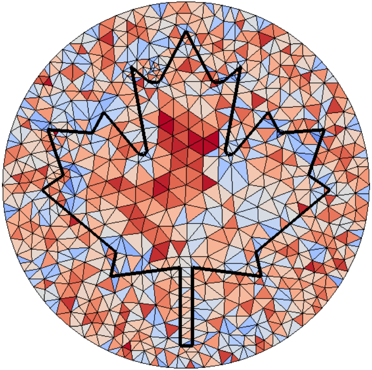
Initial mesh generated, then curve inserted.



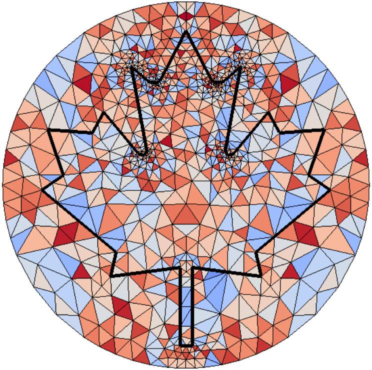
Mesh initially generated with internal boundary.

Results

Comparison Between Methods - Example 3



Initial mesh generated, then curve inserted.



Mesh initially generated with internal boundary.

Conclusions

We have developed an algorithm for curve insertion into an existing mesh.

We use equidistribution of length scale to determine the location of new vertices on the curve.

This results in only local changes to the mesh, near the curve.

With new vertices inserted, existing mesh refinement and smoothing techniques are used to produce good quality mesh.

Future Work

In three dimensions, the the problem is inserting a surface into an existing tetrahedral mesh.

The algorithm remains similar, with a length scale based surface mesh generated and the initial mesh cleaned up and reconnected around it, followed by smoothing and refinement.

MMPDEs on a surface are similar to two-dimensional metric-based mesh adaptation.

In our algorithm, we can define metric based on length scale of initial mesh.

Questions?
cfd@mech.ubc.ca, unstructuredmesh.com