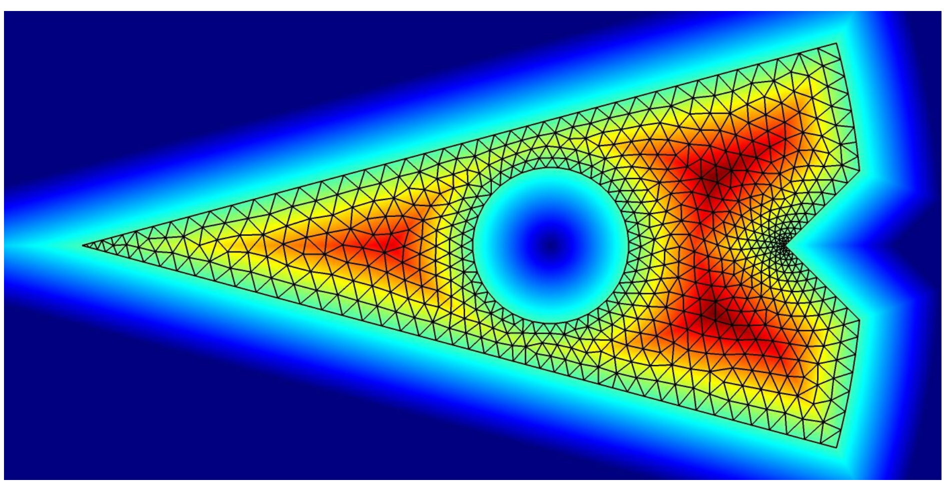
Adaptive Unstructured Mesh Generation using Distance Functions

Per-Olof Persson (persson@math.mit.edu)
Gilbert Strang (gs@math.mit.edu)

Department of Mathematics, MIT

Abstract

We present a simple and adaptable mesh generation algorithm for geometries specified implicitly by their signed distance functions. The Delaunay algorithm determines a topology, then we iteratively find a force equilibrium in the element edges, and position the boundary nodes using the distance function and its gradient. A given function specifies the element size distribution, and we show how geometry adaption can be obtained from a discretized distance function. The algorithm generalizes to any dimension, and we show examples of hybrid mesh generation and moving boundary problems in combination with the level set method.



Project web page (source code, documentation, examples):

http://math.mit.edu/~persson/mesh

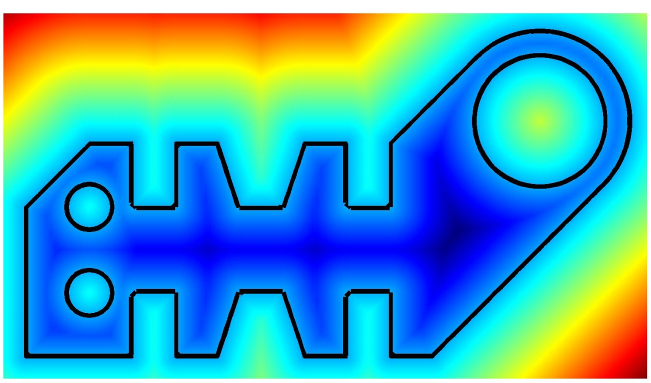
Introduction

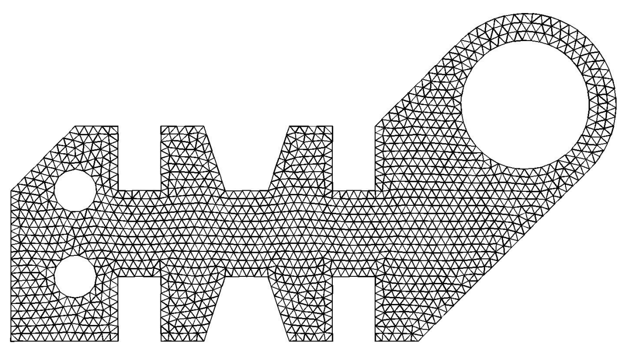
Distance functions

- Geometry boundary specified as zero level set of scalar function $\varphi(\boldsymbol{x})$
- *Signed distance function*: $\varphi(\boldsymbol{x}) \leq 0$ inside geometry, $|\nabla\varphi(\boldsymbol{x})| = 1$
- Given by:
  - Analytical function (simple geometries, boolean solid operations)
  - Procedural function (distance to polygons or curves)
  - Discretization on background mesh (Cartesian or unstructured, level set method)

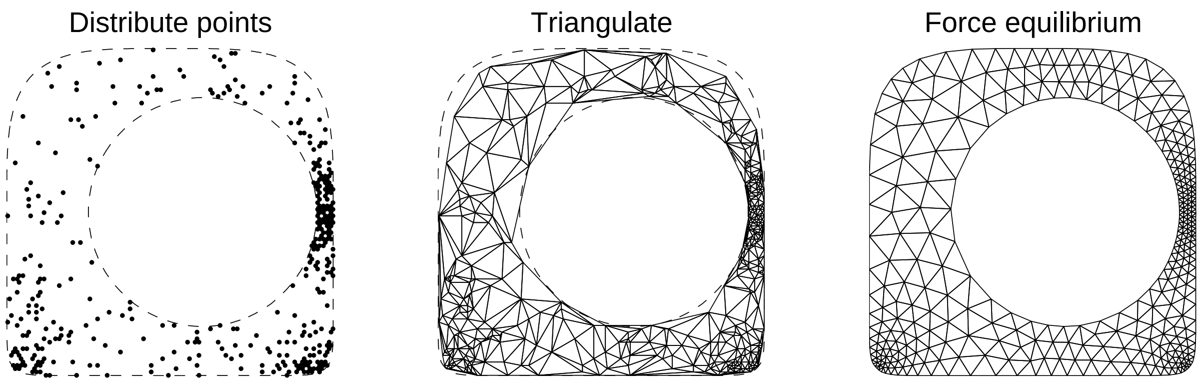
Mesh generation

- Traditional meshing algorithms (Delaunay refinement, Advancing front) first have to find and represent the boundary $\varphi(\boldsymbol{x})$ explicitly, which is inconvenient and expensive (in particular for 3-D geometries and moving boundaries).
- We use an iterative physically-based method, where the node locations are found by a force equilibrium in a truss structure. The boundaries are accessed indirectly by evaluations of $\varphi(\boldsymbol{x})$.





The Meshing Algorithm



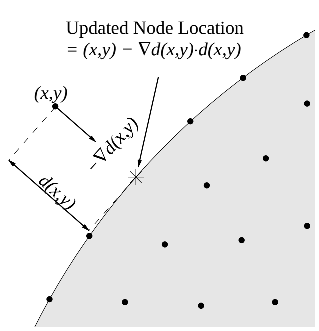
1. Distribute points inside the region according to size function $h(x, y)$, and reject points outside geometry ($\varphi(x, y) > 0$).
2. Obtain topology by Delaunay triangulation.
3. Find force equilibrium iteratively using Forward Euler, updating the topology when necessary.

$$\boldsymbol{p}_{n+1} = \boldsymbol{p}_n + \Delta t \boldsymbol{F}(\boldsymbol{p}_n)$$

Assign nonlinear forces in edges depending on current edge length ℓ and desired length ℓ_0:

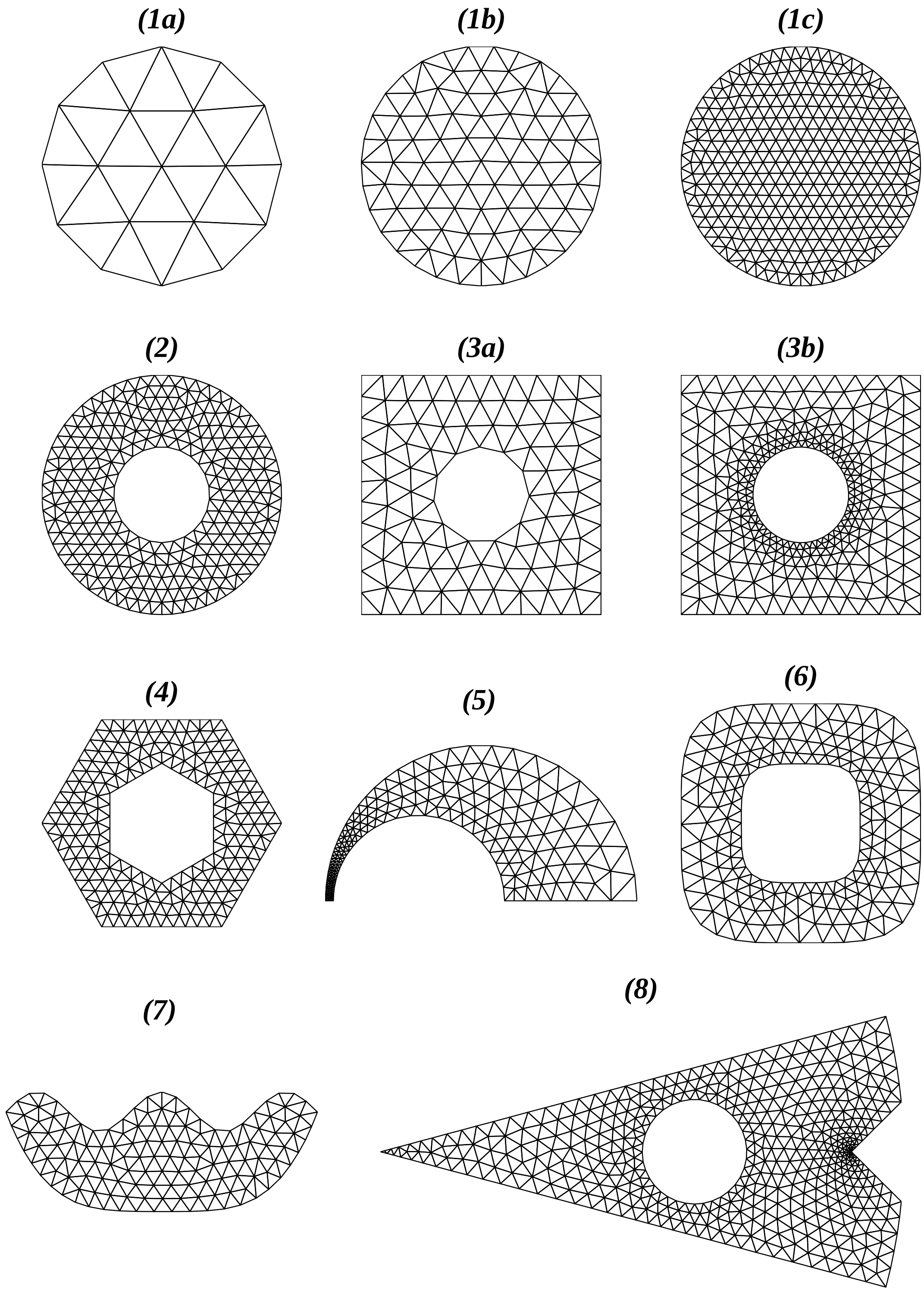
$$f(\ell, \ell_0) = \begin{cases} 1 - \left(\frac{\ell}{\ell_0}\right)^2 & \text{if } \ell < \ell_0, \\ 0 & \text{if } \ell \geq \ell_0. \end{cases}$$

Assign reaction forces at boundaries, by repositioning node points after each step using distance function: $\boldsymbol{x}_{\text{new}} \leftarrow \boldsymbol{x} - \nabla\varphi(\boldsymbol{x}) \cdot \varphi(\boldsymbol{x})$



Results (2-D)

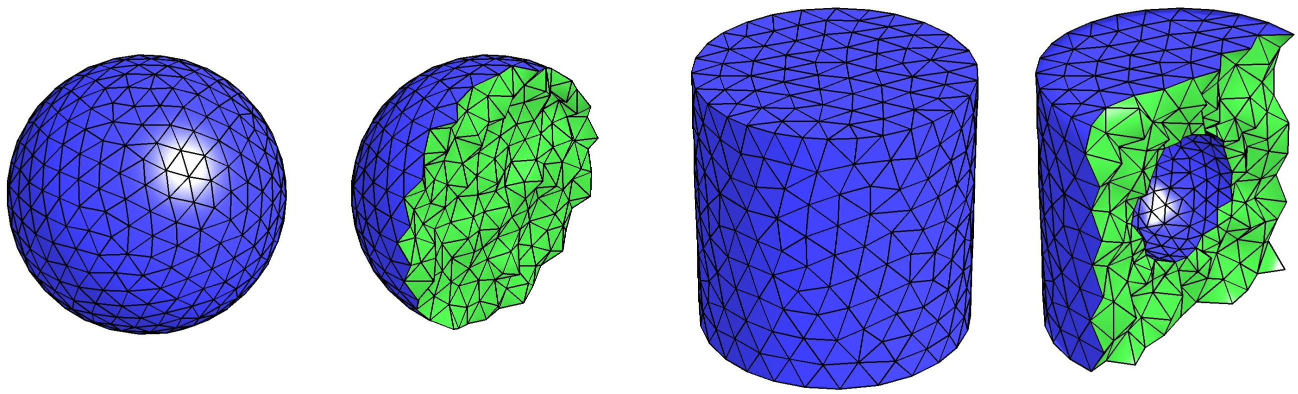
2-D Meshes



Results (3-D and 4-D)

3-D Meshes

- Tetrahedral meshes of unit ball (left) and cylinder with hole (right)
- Surface mesh plots and "split views"



4-D Hypersphere

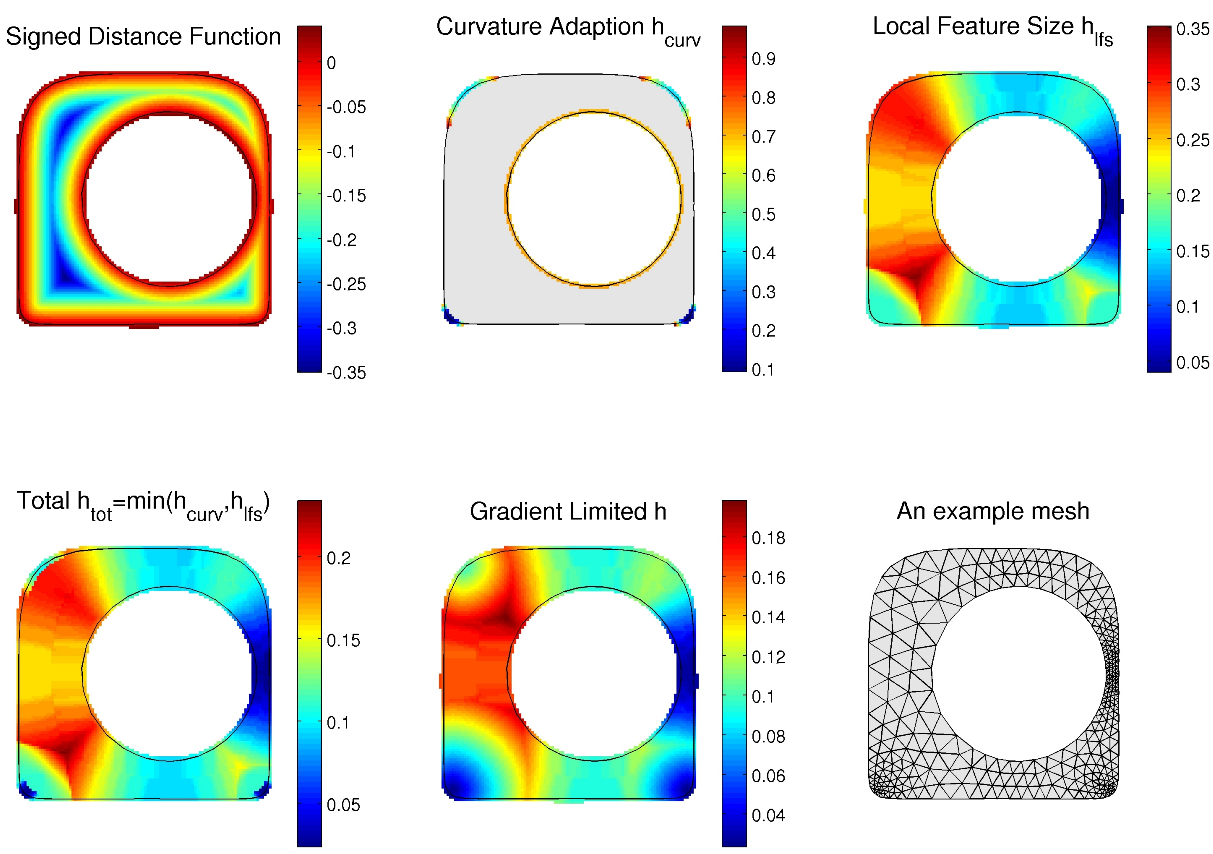
- $\varphi(\boldsymbol{x}) = r - 1$ with $r = \sqrt{\sum_{i=1}^{4} x_i^2}$
- $h_0 = 0.2$ gives $3,458$ nodes and $59,222$ elements
- No plots, hard to visualize! Instead indirect verifications:
  - Mesh volume $V_4 = 4.74$ (expected value $\pi^2/2 \approx 4.93$)
  - Hyper-surface area $S_4 = 16.3$ (surface area $2\pi^2 \approx 19.7$ of a 4-D ball). Deviations because of the approximation of the curved surface with simplices.
  - Poisson's equation $-\nabla^2 u = 1$, bnd cond's $u = 0|_{r=1}$. Analytical solution $u = (1 - r^2)/8$, linear FEM error $\|e\|_\infty = 7.9 \cdot 10^{-4}$.

Automatic Generation of Size Functions

- Systematic method for computation of $h(\boldsymbol{x})$ on a *background mesh* (for example Cartesian grid, but also unstructured meshes)
- Satisfies curvature, feature size, numerical, and grading constraints
- Curvature given by distance function, $\kappa = \nabla \cdot \frac{\nabla\varphi}{|\nabla\varphi|}$
- PDE-based approach for computation of medial axis transform and local feature size
- PDE-based approach for gradient limiting:

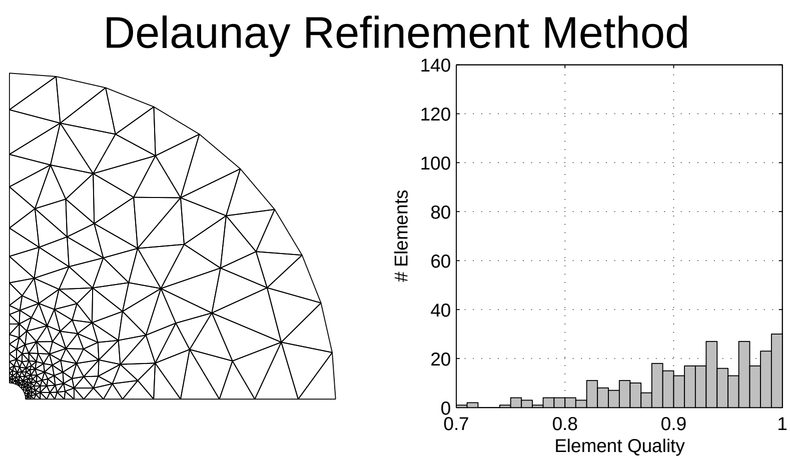
$$h(\boldsymbol{x}) = \min_{\boldsymbol{x}'}(h_0(\boldsymbol{x}') + C\|\boldsymbol{x} - \boldsymbol{x}'\|)$$

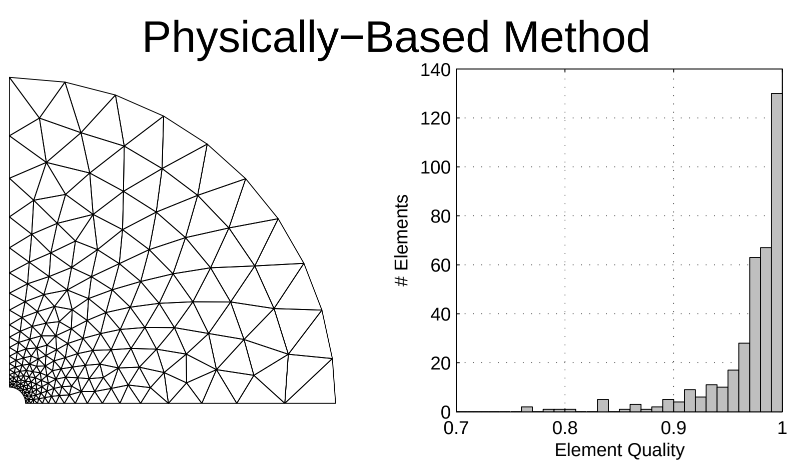
- Numerical adaptive solver easily incorporated
- Generalizes to higher dimensions without modifications



Element Quality

- Our physically-based method tends to generate meshes with very high element qualities, see plots below
- Histograms show "radius ratio" quality measure, $q = nr/R$ where n is the dimension, r is the inradius, and R is the circumradius of the element
- Standard Laplacian smoothing of the node positions in the Delaunay Refinement mesh will *not* give the same high quality, topology changes also required
- Similar results in 3-D, where face swapping and edge flipping are applied in both cases

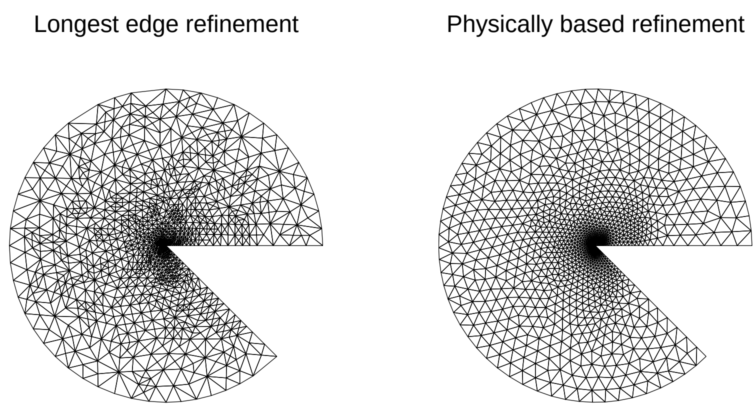




Applications

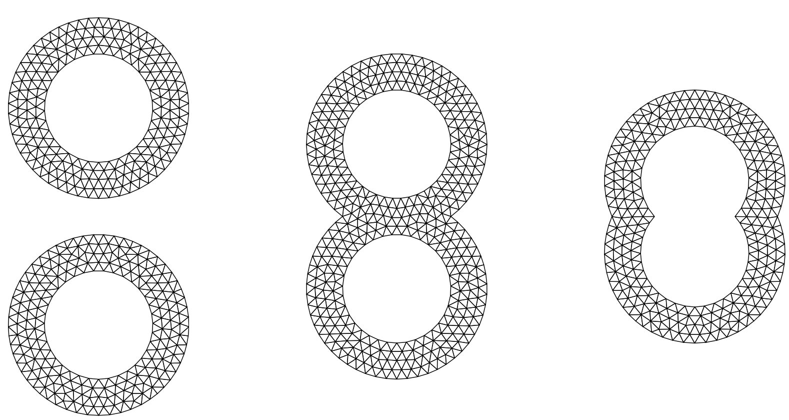
Numerical Adaptivity

- Model problem: $-\Delta u = 0$ in domain, $u(r,\theta) = \sin(4\theta/7)$ on boundary, refinement based on energy norm error estimate
- Interpolate size function $h(\boldsymbol{x})$ from error indicator on unstructured mesh from previous iteration
- Use previous mesh as initial condition in iterations, less expensive than remeshing from scratch



Moving Interfaces with Topology Changes

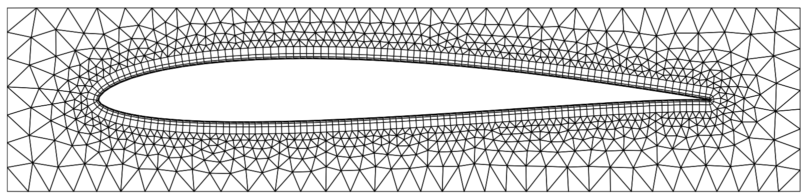
- Implicit functions handle topology changes in any dimension
- Use level set method on background mesh for interface propagation
- Multiphase flow, fluid-structure interact., shape optimization, etc

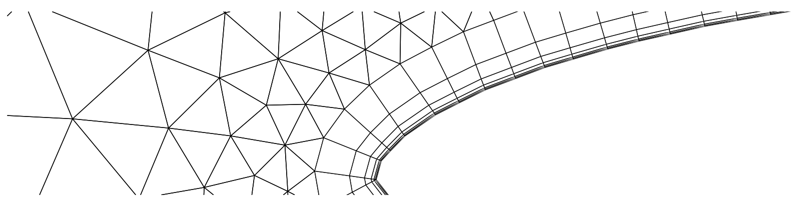


Applications

Semi-structured Hybrid Meshing

- Unstructured mesh for *offset curve* $\varphi(\boldsymbol{x}) - \delta$
- The structured grid is easily created with the distance function





Conclusions

- Unstructured mesh generation for geometries represented by their signed distance functions
- No explicit representation of boundary curves/surfaces
- Generalizes to higher dimensions (even $>$ 3-D)
- Automatic PDE-based generation of size function
- Easy to implement (50 lines of MATLAB code, see web page)
- Very high element qualities
- Applications: Simple mesh generation, moving interfaces with topology changes, hybrid meshes, etc