Mesh Generation using Level Set Methods and PDE-Based Size Control

Per-Olof Persson (persson@math.mit.edu)
Department of Mathematics, MIT

MIT EECS/ME/AA, March 15, 2004
Topics

1. Introduction

2. The New Meshing Algorithm

3. Automatic Mesh Size Control

4. Applications
Mesh Generation

- Given a geometry, determine node points and element connectivity
- Applications:
  - Numerical Solution of PDEs (FEM, FVM, BEM), Interpolation
  - Computer Graphics, Visualization
- Delaunay refinement, Advancing Front, Octree
  - Explicit boundary representations
The Level Set Method [Osher, Sethian]

- Represent geometry by *signed distance function* $\phi(x)$
- Discretize $\phi$ on Cartesian, unstructured, or Octree grid
- Propagate boundary with PDEs, e.g. $\phi_t + F|\nabla \phi| = 0$, $\phi_t + \mathbf{v} \cdot \nabla \phi = 0$
- Handles very large deformations
- Higher stability than explicit methods
- Fast Marching Method: Solve $|\nabla \phi| = 1$ in $\mathcal{O}(n \log n)$ time
Why Create Meshes on Level Sets?

- Level Set Methods superior for interface propagation with large deformations and topology changes

- But: Unstructured grids better for the physical problem
  - Better treatment of boundaries, graded meshes more efficient
  - FEM/FVM/BEM methods better suited

- Distance functions sometimes more natural
  - Image processing, MRI scans, biomechanics

- Easier to implement and manipulate than B-rep and Bézier patches
Topics

1. Introduction

2. The New Meshing Algorithm

3. Automatic Mesh Size Control

4. Applications
The New Meshing Algorithm

1. Distribute points using size function $h(x, y)$, reject points outside

2. Obtain topology by Delaunay triangulation

3. Find force equilibrium
Force Equilibrium

- Piecewise linear force function (current length $\ell$, desired length $\ell_0$):

$$f(\ell, \ell_0) = \begin{cases} k(\ell_0 - \ell) & \text{if } \ell < \ell_0, \\ 0 & \text{if } \ell \geq \ell_0. \end{cases}$$

- $\ell_0 = C_1 h(x, y)$ for non-uniform element sizes

- Larger desired lengths than actual lengths gives “internal pressure”

- Find force equilibrium using Forward Euler

$$p_{n+1} = p_n + \Delta t F(p_n)$$
Boundary Constraints

- Use distance function to project points to boundary
- Points can move tangentially along curve

Updated Node Location
\[ (x, y) - \nabla d(x, y) \cdot d(x, y) \]
MATLAB Demo
4-D Hypersphere

- \( d = r - 1 \) with \( r = \sqrt{\sum_{i=1}^{4} x_i^2} \)
- \( h_0 = 0.2 \) gives 3,458 nodes and 59,222 elements
- Mesh volume \( V_4 = 4.74 \) (expected \( \pi^2/2 \approx 4.93 \))
- Hyper-surface area \( S_4 = 19.2 \) (expected \( 2\pi^2 \approx 19.7 \))
- Poisson’s equation \(-\nabla^2 u = 1\), bnd cond’s \( u = 0 \)\( |r=1| \). Analytical solution \( u = (1 - r^2)/8 \), error \( \|e\|_\infty = 7.9 \cdot 10^{-4} \).
Mesh Quality

Delaunay Refinement with Laplacian Smoothing

Force Equilibrium by distmesh2d
Future Work

• Performance
  – Local Delaunay updates, 2-D easy, 3-D studied by [Devillers]
  – Hierarchical solver
  – Implicit instead of Forward Euler

• Robustness
  – Respect boundaries
  – Control logic and termination criterion involving element qualities

• Other ideas
  – Quad and Block meshing
  – Force-based sliver exudation
Topics

1. Introduction

2. The New Meshing Algorithm

3. **Automatic Mesh Size Control**

4. Applications
Automatic Mesh Size Control

- Important preprocessing step for many mesh generators
- Find mesh size function $h(x)$ considering 1. Curvature, 2. Local feature size, 3. Non-geometric data (user, numerical), and 4. Grading
Background Grids

- Discretize mesh size function $h(x)$ on a coarse background grid
- Mesh from previous iteration for adaptive FEM solvers
Curvature Adaptation

- On the boundaries $h(x) \leq 1/K|\kappa(x)|$, resolution $K$

- Curvature $\kappa$ computed explicitly or using distance function

Curvature Adaptation $h_{\text{curv}}$
Feature Size Adaptation

- $h(x) \leq \text{lfs}(x)/R$
- The local feature size $\text{lfs}$ is “width” of geometry
- Ruppert: $\text{lfs}(x)$ is “the larger distance from $x$ to the closest two non-adjacent polytopes [of the boundary]”
Feature Size Adaptation

- Alternative definition: $\text{lfs}(x)$ is equal to the smallest distance between the boundary and the point on the medial axis closest to $x$

- Algorithm:
  1. Identify medial axis by shock detection in $\phi(x)$ (skeletonization)
  2. Solve $\nabla \phi_{MA} \cdot \nabla \text{lfs} = 0$ with $\text{lfs}(x) = 2\phi(x)$ on medial axis
Shock Detection in Distance Function $\phi(x)$

Contours of $\phi(x, y)$ and shock

Contours of $\phi(x, j)$, $p_1(x)$, $p_2(x)$
Feature Size Adaptation - Example

Feature Size Adaptation $h_{lfs}$
Curvature + Feature Size Adaptation

\[
\text{Curvature & Feature Size} = \min ( h_{\text{curv}}, h_{\text{lfs}} )
\]
Gradient Limiting

• Max size ratio $G$ between elements ($h_i \leq G h_j$ for neighboring $i, j$)

• Continuous analogue: $|\nabla h(x)| \leq G - 1 \equiv G_c$

• Traditional methods:
  – Repeated discrete refinements [Borouchaki et al]
  – Balanced Octrees [Blacker, Stephenson]

• New continuous formulation:
  – Steady-state solution to Hamilton-Jacobi equation
Gradient Limiting

- The Gradient Limiting Equation:
  \[
  \frac{\partial h}{\partial t} + |\nabla h| = \min(|\nabla h|, G_c), \quad h(0) = h_0
  \]

- Analytical solution (convex domains, use Lax’ theorem):
  \[
  h(x) = \min_y (h_0(y) + G_c \|x - y\|).
  \]

- PDE-based formulation gives straightforward extension to:
  - Higher order
  - Unstructured meshes
  - Space and solution dependent \( G_c(x, h) \)

- Solve in \( O(n \log n) \) time using modified Fast Marching Method
Gradient Limiting in 1-D

Maximum gradient $G_c = 4$

Maximum gradient $G_c = 2$

Maximum gradient $G_c = 1$

Maximum gradient $G_c = 0.5$
Gradient Limiting - Example

Mesh Size Function $h$
Mesh Size Function in 2-D

Curvature Adaptation $h_{\text{curv}}$

Feature Size Adaptation $h_{\text{lfs}}$

Mesh Size Function $h$

Triangular Mesh

0 0.05 0.1 0.15 0.2 0.25 0.3
MATLAB Demo
Mesh Size Function in 3-D

Curvature Adaptation $h_{\text{curv}}$

Feature Size Adaptation $h_{\text{lf}}$

Mesh Size Function $h = \text{gradlim}(\min(h_{\text{curv}}, h_{\text{lf}}))$

Tetrahedral Mesh
Unstructured Gradient Limiting

- Solve \( \frac{\partial h}{\partial t} + |\nabla h| = \min(|\nabla h|, G_c) \) on triangulated domains using explicit positive coefficient scheme [Barth, Sethian]
Unstructured Gradient Limiting

- Solve $\frac{\partial h}{\partial t} + |\nabla h| = \min(|\nabla h|, G_c)$ on triangulated domains using explicit positive coefficient scheme [Barth, Sethian]

Background Mesh | Gradient Limited $h(x,y)$ | Generated Mesh
Unstructured Gradient Limiting

- Solve $\frac{\partial h}{\partial t} + |\nabla h| = \min(|\nabla h|, G_c)$ on triangulated domains using explicit positive coefficient scheme [Barth, Sethian]
Topics

1. Introduction

2. The New Meshing Algorithm

3. Automatic Mesh Size Control

4. Applications
Numerical Adaptation

- Model problem: \(-\Delta u = 0\) in domain, \(u(r, \theta) = \sin(4\theta/7)\) on boundary
- Refinement using energy norm error estimate
- Interpolate size function \(h(x)\) from error indicator on unstructured mesh from previous iteration
Numerical Adaptation

Adaptive Size Function $h(x,y)$

Longest Edge Refinement

Gradient Limited $h(x,y)$

Force Equilibrium
Numerical Adaptation - Linear Elasticity

- Adaptation using energy norm error estimate
- Gradient limiting on unstructured mesh with $G_c = 0.3$
Meshes with Moving Interfaces

- Use level set method on background mesh for interface propagation
- Solve physical problem on unstructured mesh
- Last mesh for initial node points, density control
MATLAB Demo
Example - Structural Vibration Control

- Consider eigenvalue problem

\[-\Delta u = \lambda \rho(x)u, \quad x \in \Omega\]
\[u = 0, \quad x \in \partial \Omega.\]

with

\[\rho(x) = \begin{cases} 
\rho_1 & \text{for } x \notin S \\
\rho_2 & \text{for } x \in S.
\end{cases}\]

- Solve the optimization

\[
\min_S \lambda_1 \text{ or } \lambda_2 \text{ subject to } \|S\| = K.
\]
Example - Structural Vibration Control

- Model problem in shape optimization
- Level set formulation by Osher and Santosa:
  - Calculate descent direction $\delta \phi = -v(x)|\nabla \phi|$ using solution $\lambda_i, u_i$
  - Linearized constraint or Newton’s method for Lagrange multiplier
  - Represent interface implicitly, propagate using level set method
- Use finite elements on unstructured mesh for eigenvalue problem
  - More accurate interface condition
  - Arbitrary geometries
  - Graded meshes
  - Better calculation of subdomain and curve integrals
MATLAB Demo
Example - Structural Design Improvement

- Linear elastostatics, minimize compliance

\[ \int_{\partial \Omega} g \cdot u \, ds = \int_{\Omega} A\varepsilon(u) \cdot \varepsilon(u) \, dx \text{ subject to } ||\Omega|| = K. \]

- Boundary variation methods by Murat and Simon, Pironneau, Homogenization method by Bendsoe and Kikuchi

- Level set approaches by Sethian and Wiegmann (Immersed interface method) and Allaire et al \((Ersatz \ material, \ Young’s \ modulus \ \varepsilon)\)
MATLAB Demo
Future Work

- More applications
  - Multiphase and free surface flow
  - Fluid-structure interaction
  - More shape optimization, e.g. photonic bandgap maximization
- Non-uniform meshes, compute $h(x)$ as described previously
- Higher dimensions
- Remove background grid - use last mesh for $\phi$
Conclusions

- Mesh generation for level sets
- Automatic PDE-based size functions
- Combine FEM and level set methods
- Applications with moving interfaces and topology changes