Triangular Meshing: Delaunay and Advancing Front

Saad Qadeer

April 26, 2016
The Meshing Problem

- Let $\Omega \subset \mathbb{R}^2$ be a polygon.
- Let $h > 0$ be a smooth function on $\Omega$. 
The Meshing Problem

- Let \( \Omega \subset \mathbb{R}^2 \) be a polygon.
- Let \( h > 0 \) be a smooth function on \( \Omega \).
- Need to implement meshing algorithms that size elements according to \( h \).
The Delaunay Method

- Choose nodes on the boundary first.
The Delaunay Method

- Choose nodes on the boundary first.

We require $h(x_0 + p\Delta x) = \Delta x$.

Assuming $\Delta x$ is small

$$h(x_0) + (\Delta x)p \cdot \nabla h(x_0) = \Delta x$$

$$\Rightarrow \Delta x = \frac{h(x_0)}{1 - p \cdot \nabla h(x_0)}.$$
Next, perform a Delaunay triangulation.
Next, perform a Delaunay triangulation.

To refine according to $h$, identify the “worst” triangle by the largest

$$\frac{\text{sum of edge lengths}}{\text{sum of } h\text{-values at the nodes}}$$

Find the circum-center $c$ and circum-radius $r$. Add $c$ to the list of points if

- $c \in \Omega$
- $\text{dist}(c, \partial \Omega) \geq \alpha r$
Next, perform a Delaunay triangulation.

To refine according to $h$, identify the “worst” triangle by the largest

$$\frac{\text{sum of edge lengths}}{\text{sum of } h\text{-values at the nodes}}$$

Find the circum-center $c$ and circum-radius $r$. Add $c$ to the list of points if

- $c \in \Omega$
- $\text{dist}(c, \partial\Omega) \geq \alpha r$

Typically, $\alpha = 0.5$ works great.

Otherwise take $c$ to be midpoint of most offensive edge.
The Bowyer-Watson Algorithm

- Delete all triangles whose circum-circles contain \( c \), but keep the nodes.
- Delaunay re-triangulate the void with \( c \) added to nodes.
Choose nodes on the boundary as before.
The Advancing Front

- Choose nodes on the boundary as before.
- Find the largest edge \((x_A, x_B)\) in the front. Choose \(x_C\) so \((x_A, x_B, x_C)\) is an equilateral triangle.
The Advancing Front

- Choose nodes on the boundary as before.
- Find the largest edge \((x_A, x_B)\) in the front. Choose \(x_C\) so \((x_A, x_B, x_C)\) is an equilateral triangle.
- Find \(x'_C = x_C + q\Delta s\) so that \(h(x'_C) = \|x'_C - x_B\| := L(\Delta s)\).
Choosing the New Node

- Assuming $\Delta s$ to be small

\[ h(x_C) + (\Delta s)\nabla h(x_C) \cdot q = L(0) + (\Delta s)L'(0) \]

\[ \Rightarrow \Delta s = \frac{L(0) - h(x_C)}{\nabla h(x_C) \cdot q - L'(0)} \]

where $L'(0) = \frac{q \cdot (x_C - x_B)}{L(0)}$.

- Scan the front to find any nodes within $\frac{1}{2} h(x'_C)$. 

Choosing the New Node

- Assuming $\Delta s$ to be small

\[
h(x_C) + (\Delta s)\nabla h(x_C) \cdot q = L(0) + (\Delta s)L'(0)
\]

\[
\Rightarrow \Delta s = \frac{L(0) - h(x_C)}{\nabla h(x_C) \cdot q - L'(0)}
\]

where $L'(0) = \frac{q \cdot (x_C - x_B)}{L(0)}$.

- Scan the front to find any nodes within $\frac{1}{2}h(x_C)$.

- If no points found, tentatively consider adding $x'_C$: check the resulting exterior angles.
Choosing the New Node

- Assuming $\Delta s$ to be small

\[
h(x_C) + (\Delta s)\nabla h(x_C) \cdot q = L(0) + (\Delta s)L'(0)
\]

\[
\implies \Delta s = \frac{L(0) - h(x_C)}{\nabla h(x_C) \cdot q - L'(0)}
\]

where $L'(0) = \frac{q \cdot (x_C - x_B)}{L(0)}$.

- Scan the front to find any nodes within $\frac{1}{2}h(x'_C)$.

- If no points found, tentatively consider adding $x'_C$: check the resulting exterior angles.

- If $x'_C$ is not favourable, extend search radius by 20% and keep scanning.
Choosing the New Node

- Assuming $\Delta s$ to be small

\[
h(x_C) + (\Delta s)\nabla h(x_C) \cdot q = L(0) + (\Delta s)L'(0)
\]

\[
\Rightarrow \Delta s = \frac{L(0) - h(x_C)}{\nabla h(x_C) \cdot q - L'(0)}
\]

where $L'(0) = \frac{q \cdot (x_C - x_B)}{L(0)}$.

- Scan the front to find any nodes within $\frac{1}{2}h(x'_C)$.

- If no points found, tentatively consider adding $x'_C$: check the resulting exterior angles.

- If $x'_C$ is not favourable, extend search radius by 20% and keep scanning.

- Any voids in the region are filled in recursively later on.

- Perform edge flips for triangles with angles $>120^\circ$ to weed out bad triangles.
Choosing the New Node

- Assuming $\Delta s$ to be small

\[
h(x_C) + (\Delta s)\nabla h(x_C) \cdot \mathbf{q} = L(0) + (\Delta s)L'(0)
\]

\[
\Rightarrow \Delta s = \frac{L(0) - h(x_C)}{\nabla h(x_C) \cdot \mathbf{q} - L'(0)}
\]

where $L'(0) = \frac{\mathbf{q} \cdot (x_C - x_B)}{L(0)}$.

- Scan the front to find any nodes within $\frac{1}{2}h(x'_C)$.
- If no points found, tentatively consider adding $x'_C$: check the resulting exterior angles.
- If $x'_C$ is not favourable, extend search radius by 20% and keep scanning.
- Any voids in the region are filled in recursively later on.
- Perform edge flips for triangles with angles $> 120^\circ$ to weed out bad triangles.