The Level Set Method applied to Structural Topology Optimization

Dr Peter Dunning

22-Jan-2013
Structural Optimization

Level set based structural topology optimization

Increasing:
- No. design variables
- Opportunity for improvement
- Difficulty
Brief Introduction to Topology Optimisation

- Optimal distribution of material
- Discretize design space using Finite Elements
- Relax 0-1 problem → Allow partially filled elements (variable density)
The SIMP Method

- Force variable density solutions towards discrete variable solution
- Penalise intermediate densities by raising to a power:
  
  \[ \rho^p, \rho \in \{0,1\}, p > 1 \]

With sensitivity smoothing

No sensitivity smoothing
Boundary based Topology Optimization

• **Problem**: Element based methods.
  - results possess “fuzzy” boundaries & “grey” areas

• **Solution**: Boundary based methods.
  - combine shape optimization with provision to create/remove holes

*Level set based structural topology optimization*

Spline Based Methods

- Boundary represented using splines
- Design variables are control point positions
- Topological derivatives used to locate points to create new holes

However...

- Requires careful handling of splitting/merging boundaries
- Control points can bunch or spread → poor boundary representation
- Splines do not seem flexible enough for topology optimization
Implicit boundary representation

- Use a simple implicit function to define the boundary
- Assign a scalar value to each node of a discretized domain
- Boundary can naturally break and merge

\[ \phi(x) = 0 \]

\[ \phi(x) < 0 \]

\[ \phi(x) > 0 \]

\[
\begin{cases}
\phi(x) \geq 0, & x \in \Omega_S \\
\phi(x) = 0, & x \in \Gamma_S \\
\phi(x) < 0, & x \notin \Omega_S 
\end{cases}
\]
Level set method

- Developed to compute motions of implicitly represented interfaces
- Numerical solution to:

\[
\frac{\partial \phi(x, t)}{\partial t} + \nabla \phi(x, t) \frac{dx}{dt} = 0
\]

- Discretize and rearrange:

\[
\phi_{i}^{k+1} = \phi_{i}^{k} - \Delta t \left| \nabla \phi_{i}^{k} \right| V_{n,i}
\]

- \( \Delta t \) = time step
- \( V_{n,i} \) = velocity normal to boundary
- \( i \) = grid point
- \( k \) = current iteration
Level set based optimization

- Level set eq. can be used as an update rule in an optimization method
- Need to link the velocity function to the gradient of the objective
  - Shape sensitivity analysis
  - Decent method analogy

Basic Algorithm

1. Compute shape sensitivities along boundary
2. Define velocity function using descent method analogy
3. Move the boundary by numerically solving the level set equation
4. Repeat until a converged solution is found
Numerical considerations

- Analyse current structure to derive sensitivities → FEM
  - Fixed Grid (Eulerian) approach for efficiency
- Initialize $\phi(x)$ as signed distance function
  - Maintain to promote stability

Level set based structural topology optimization

\[ |\nabla \phi(x)| \gg 1 \]

\[ |\nabla \phi(x)| = 1 \]

\[ |\nabla \phi(x)| \ll 1 \]
Numerical considerations

- Stable time step defined by CFL condition: $\Delta t < \frac{h}{V_{\text{min}}}$
- Velocity function required at all grid nodes (not just at boundary)
  - Extension method to maintain signed distance function
- Numerically solve: $\nabla \phi_t \cdot \nabla V_{\text{ext}} = 0$
- Spacial gradient computation:
  - Upwind scheme + WENO (stability & robustness)
- Narrow band approach for efficiency:
  - Only update $\phi(x)$ with certain distance of initial boundary
- Occasion re-initialization of $\phi(x)$ to a signed distance function
Compliance minimisation problem

- Objective to minimise structural compliance (or maximise stiffness)
- Upper limit on amount of available material
- Prescribed loading and boundary conditions

\[
\min : C(u) = \int_{\Omega_s} A \varepsilon(u) \varepsilon(u) \, d\Omega_s \quad \text{← Total structural compliance}
\]

\[
s.t. : \int_{\Omega_s} A \varepsilon(u) \varepsilon(v) \, d\Omega_s = \int_{\Gamma_s} fv \, ds \quad \text{← Static equilibrium}
\]

\[
\int_{\Omega_s} \, d\Omega_s \leq Vol^* \quad \text{← Material volume constraint}
\]

\[
\min : \bar{C}(u) = C(u) + \lambda \left[ \int_{\Omega_s} d\Omega_s - Vol^* \right] \quad \text{← Unconstrained problem}
\]

\[
\bar{C}'(u) = \int_{\Gamma_0} \left( A \varepsilon(u) \varepsilon(u) - \lambda \right) V_n \, d\Gamma_0 \quad \text{← Shape sensitivity}
\]
Cantilever Beam Example

Level set based structural topology optimization
Challenges for Level Set Based Optimization

- Efficiency:
  - Constrained by CFL condition
  - Optimization only interested in final solution
- New hole creation:
  - Front tracking paradigm cannot create new holes
  - Solution is dependent on initial no. holes
- Accurate sensitivity computation:
  - Fixed grid elements do not match the boundary
  - Largest errors occur at the boundary
- Constraint handling:
  - Limited research to date – mainly concerned with a volume constraint
Fixed grid FEA

- “Topology” of FE mesh remains constant
- Structural boundary can cut though an element
- Discontinuous elements require special treatment
- Simplest approach to weight stiffness by a volume fraction

\[ K_{cut} = \left( \frac{A_{in}}{A_{full}} \right) \times K_{full} \]

- Very efficient → Attractive property in structural optimization
- However, does not capture exact boundary position → Instability
Fixed grid sensitivity computation - example

Sensitivities obtained using simple nodal averaging
Solution – Weighted Least Squares approach

• Observations:
  • Generally, error in sensitivities increased as volume fraction decreased
  • Strains often more accurate when computed at gauss points

• Solution:
  • Compute sensitivities using a weighted least squares approach

• Numerical investigations:
  • Basis polynomial order
  • Support radius
  • Number of evaluations / element
  • Weighting function
Proposed Weighted LS sensitivity computation

- 2\textsuperscript{nd} order basis function: 
  \[ \zeta(x, y) = c_0 + c_1 x + c_2 y + c_3 xy + c_4 x^2 + c_5 y^2 \]
- Weighted by volume fraction and inverse distance
Fixed grid sensitivity computation example - revisited

Level set based structural topology optimization
Hole creation method

- Moving just the boundary does not allow new holes to emerge.
- Could use topological derivatives:
  - Insert new hole every $n$ iterations.
  - Arbitrary, hole creation not linked to boundary movement.
- New holes can emerge in 3D by an overlap of two approaching fronts.

Level set based structural topology optimization
Hole creation method

- Exploit 3D hole creation mechanism in 2D:
  - Introduce 2\textsuperscript{nd} implicit function to represent a pseudo 3\textsuperscript{rd} dimension
  - Update 2\textsuperscript{nd} function using shape sensitivities to define velocity

\[
\phi^0_{h}(x) = \begin{cases} 
  +\hat{h} & \text{if } x \in \Omega_s \\
  -\hat{h} & \text{if } x \notin \Omega_s 
\end{cases}
\]

\[
\phi^{k+1}_{h}(x_i) = \phi^{k}_{h}(x_i) - \Delta t V_{n,i} \quad \leftarrow \text{Initialize 2\textsuperscript{nd} implicit function, } \hat{h} = \text{pseudo thickness}
\]

\[
\phi^{k+1}_{h}(x_i) < 0 \quad \text{if } x \in \Omega_s \quad \leftarrow \text{Criteria for creating a new hole}
\]

- Holes can emerge naturally during optimization
- Hole creation is linked to boundary optimization
Hole creation method – Cantilever example

Level set based structural topology optimization
Applications

Multi-material

Compliant mechanisms

Robust optimization

Eigenfrequency maximisation
Future Work

• Multidisciplinary optimization (Aero-structural)
• Topology optimization of wing box structure (minimise weight)
• Heterogenous materials (composites, FGM)
• Realistic constraints:
  • Stress
  • Aeroelastic (divergence, flutter)
Any questions?
References


