Theoretical Background for OpenLSTO v1.0:

Open Source Level Set Topology Optimization

M2DO Lab[1,2]

[1]Cardiff University

[2]University of California, San Diego

August 2018

A brief description of theory of level set topology optimization is given here. For further details, the readers are referred to cited references, e.g. [OF03, WWG03, AJT04, DKM11, DK15].

Finite Element Analysis

Area Fraction Weighted Fixed Grid Approach

A fixed grid is generated by superimposing a rectangular grid of equal sized elements on the given structure instead of generating a mesh to fit the structure. Some of these elements are inside of the structure (I), some are outside (O) and some are on the boundary, namely neither-in-nor-out (NIO) elements. As O element is given a material property significantly less than an I element and the problem becomes a bimaterial one.

A NIO element is partially inside the structure and its material property value is not constant nor continuous over the element. Such an element is approximated by transforming the bimaterial element into a homogeneous isotropic element. The material property matrix of a NIO element is computed using:

$$[D(NIO)^e] = \alpha\,[D(I)^e] \tag{1}$$

where $[D(NIO)^e]$ is the elemental material property of a NIO element, $[D(I)^e]$ is for the elemental material property of inside, and α is the area ratio calculated by

$$\alpha = \frac{A_I}{A^e} \tag{2}$$

with A_I is the area inside the structure within the NIO element, and A^e is the total area of the e-th element.

Using these values of $[D(NIO)^e]$, the stiffness matrix can be computed and a standard finite element analysis can be applied to determine the displacements and hence stress values of elements.

Sensitivity Analysis

The principle of virtual work, also known as the principle of virtual displacement, is stated that for any quasi-static and admissible virtual displacement from an equilibrium configuration, the increment of strain energy stored is equal to the increment of work done by body force $\{\boldsymbol{b}\}$ in volume V and surface traction $\{\boldsymbol{t}\}$ on surface S. It is formulated as

$$\int \{\delta\boldsymbol{\varepsilon}\}^T \{\boldsymbol{\sigma}\}\, dV = \int \{\delta\boldsymbol{u}\}^T \{\boldsymbol{b}\}\, dV + \int \{\delta\boldsymbol{u}\}^T \{\boldsymbol{t}\}\, dS \tag{3}$$

where $\{\delta\boldsymbol{\varepsilon}\}$ is the vector of strains, $\{\delta\boldsymbol{u}\}$ is the virtual displacement. In textbook on topology optimization, the right and left hand sides are written as

$$\begin{aligned} a(\boldsymbol{u},\boldsymbol{v}) &= \int \{\delta\boldsymbol{\varepsilon}\}^T \{\boldsymbol{\sigma}\}\, dV \\ &= \int \{\delta\boldsymbol{\varepsilon}\}^T [\mathbf{E}] \{\boldsymbol{\varepsilon}\}\, d\Omega = \int \{\boldsymbol{\varepsilon}(\boldsymbol{v})\}^T [\mathbf{E}] \{\boldsymbol{\varepsilon}(\boldsymbol{u})\}\, d\Omega \qquad (4) \\ l(\boldsymbol{v}) &= \int \{\delta\boldsymbol{u}\}^T \{\boldsymbol{b}\}\, dV + \int \{\delta\boldsymbol{u}\}^T \{\boldsymbol{t}\}\, dS \\ &= \int \{\boldsymbol{v}\}^T \{\boldsymbol{b}\}\, dV + \int \{\boldsymbol{v}\}^T \{\boldsymbol{t}\}\, dS \qquad (5) \end{aligned}$$

where $[\mathbf{E}]$ is the constitutive matrix, $\boldsymbol{v}$ is the virtual displacement that is equivalent to $\{\delta\boldsymbol{u}\}$.

The Lagrange multiplier method is applied to solve the optimization problem, where the Lagrange function is:

$$\mathcal{L}(\Omega, \{\boldsymbol{u}\}, \{\boldsymbol{\lambda}\}) = \mathcal{J}(\Omega, \{\boldsymbol{u}\}) + a(\{\boldsymbol{u}\}, \{\boldsymbol{\lambda}\}) - l(\{\boldsymbol{\lambda}\}) \tag{6}$$

in which $\{\boldsymbol{\lambda}\}$ is the adjoint variable.

The shape derivation of the objective function is obtained by differentiating

$$\mathcal{J}(\Omega) = \mathcal{L}(\Omega, \{\boldsymbol{u}\}, \{\boldsymbol{\lambda}\}) \tag{7}$$

which, by the chain rule theorem, reduces to the partial derivative of $\mathcal{L}$ with respect to Ω in the direction θ

$$\mathcal{J}'(\Omega)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}. \tag{8}$$

Applying **Lemma** 4 and 5 in [AJT04] to $\mathcal{L}$ and substituting $\mathcal{J}$ for compliance, we obtain:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \Omega} &= \frac{\partial}{\partial \Omega}\left(\int \{\boldsymbol{u}\}^T\{\boldsymbol{b}\}dV + \int \{\boldsymbol{u}\}^T\{\boldsymbol{t}\}dS\right. \\
&\quad + \int \{\boldsymbol{\varepsilon}(\boldsymbol{\lambda})\}^T[\mathbf{E}]\{\boldsymbol{\varepsilon}(\boldsymbol{u})\}\,d\Omega \\
&\quad \left. - \int \{\boldsymbol{\lambda}\}^T\{\boldsymbol{b}\}\,dV - \int \{\boldsymbol{\lambda}\}^T\{\boldsymbol{t}\}\,dS\right) \\
&= \int_{\partial\Omega} \theta \cdot n \left(\{\boldsymbol{u}\}^T\{\boldsymbol{b}\} + \{\boldsymbol{\varepsilon}(\boldsymbol{\lambda})\}^T[\mathbf{E}]\{\boldsymbol{\varepsilon}(\boldsymbol{u})\} - \{\boldsymbol{\lambda}\}^T\{\boldsymbol{b}\}\right) dS \\
&\quad + \int_{\partial\Omega} \theta \cdot n \left(\frac{\partial \{\boldsymbol{u}\}^T\{\boldsymbol{t}\}}{\partial n} + H\{\boldsymbol{u}\}^T\{\boldsymbol{t}\}\right) dS \\
&\quad - \int_{\partial\Omega} \theta \cdot n \left(\frac{\partial \{\boldsymbol{\lambda}\}^T\{\boldsymbol{t}\}}{\partial n} + H\{\boldsymbol{\lambda}\}^T\{\boldsymbol{t}\}\right) dS
\end{aligned} \tag{9}$$

In the case of a self-adjoint problem, e.g. compliance, it has $\{\boldsymbol{\lambda}\} = -\{\boldsymbol{u}\}$.

Level Set Method

Implicit Surfaces

In three spatial dimensions, the lower-dimensional interface is a surface that separates $\mathcal{R}^3$ into separate subdomains with nonzero volumes. We consider only closed surfaces with clearly defined interior and exterior regions.

For complicated surfaces with no analytical representation, we again need to use a discretization. In three spatial dimensions the explicit representation can be quite difficult to discrete. One needs to choose a number of points on the two-dimensional surface and record their connectivity. If the exact surface and its connectivity are known, it is simple to tile the surface with triangles whose vertices lie on the interface and whose edges indicate connectivity. On the other hand, if connectivity is not known, it can be quite difficult to determine.

Connectivity can change for dynamic implicit surfaces, i.e., surface that are moving around.

Signed Distance Function

A distance function $d(\boldsymbol{x})$ is defined as

$$d(\boldsymbol{x}) = \min\left(|\boldsymbol{x} - \boldsymbol{x}_I|\right), \quad \forall \boldsymbol{x}_I \in \partial\Omega, \tag{10}$$

implying that $d(\boldsymbol{x}) = 0$ on the boundary where $\boldsymbol{x} \in \partial\Omega$. Geometrically, d may be constructed as follows. If $\boldsymbol{x} \in \partial\Omega$, then $d(\boldsymbol{x}) = 0$. Otherwise, for a given point $\boldsymbol{x}$, find the point on the boundary set $\partial\Omega$ closest to $\boldsymbol{x}$, and label this point $\boldsymbol{x}_C$. Then $d(\boldsymbol{x}) = |\boldsymbol{x} - \boldsymbol{x}_c|$.

A signed distance functions is an implicit function ϕ with $|\phi\left(\boldsymbol{x}\right)| = d(\boldsymbol{x})$ for all $\boldsymbol{x}$. Thus, $\phi\left(\boldsymbol{x}\right) = d(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \partial\Omega$, $\phi\left(\boldsymbol{x}\right) = -d(\boldsymbol{x})$ for all $\boldsymbol{x} \in \Omega^-$, and $\phi\left(\boldsymbol{x}\right) = d(\boldsymbol{x})$ for all $\boldsymbol{x} \in \Omega^+$. In a compact for, it is written as

$$\phi\left(\boldsymbol{x}\right) = \begin{cases} -d(\boldsymbol{x}), & \forall \boldsymbol{x} \in \Omega^- \\ 0, & \forall \boldsymbol{x} \in \partial\Omega \\ d(\boldsymbol{x}), & \forall \boldsymbol{x} \in \Omega^+ \end{cases} \tag{11}$$

. Signed distance functions share all the properties of implicit functions. In addition, there are a number of new properties that only signed distance functions possess. For example,

$$|\nabla\phi| = 1, \tag{12}$$

which is known as the eikonal equation.

Hamilton Jacobi Equation

An implicitly defined boundary of the structure used during level-set method is updated by solving pseudo time-dependent Hamilton-Jacobi equation as shown:

$$\frac{\partial\phi(x,t)}{\partial t} + \nabla\phi(x,t)\frac{dx}{dt} \tag{13}$$

where t is time, $\nabla\phi(x,t)$ is the gradient of the level set function. To put the equation simply, a smooth boundary $\Gamma = x|\phi(x) = 0$ at given time x and space t is changed by its normal velocity.

Being a hyperbolic partial differential equation, Hamilton-Jacobi equation is often solved numerically. In practice, an upwind scheme with high-order differentiation is used with a constraint from CFL (Courant–Friedrichs–Lewy) condition is a common practice to obtain a stable solution.

Boundary Evolution

Since the structural boundary is the zero value level set, so the boundary evolution is implicitly achieved by the updating of level set function.

In numerical implementations, the structural boundary is discretized and approximately represented by a series of boundary points which satisfies $\phi_(\boldsymbol{x}) = 0$. If two adjacent nodes have ϕ-values with opposite signs then there is a boundary point between them, which is taken to lie on the edge between the nodes, with a position determined by linear interpolation. The boundary points form a set of closed curves, which provide a discrete representation of the boundary.

Update and reinitialization

The level set function in each node can be updated by the following discretized H-J equation with the use of the up-wind differential scheme:

$$\phi_i(t + \Delta t) = \phi_i(t) - v_i^n \Delta t \cdot |\nabla \phi(t)|_i \tag{14}$$

where $v_i^n \Delta t$ is the boundary movement obtained by solving a sublevel linearised programme; the gradient filed is estimated for each node using the Hamilton–Jacobi weighted essentially non-oscillatory method (HJ-WENO) described in [OF03].

Instead of updating the level set values in the whole field, we only restrict the update to nodes within a narrow band close to the boundary. This improves the efficiency of the method, but it means that ϕ_i is given by the signed distance to the boundary only within the narrow band. To correct for this effect, it is common that all of the ϕ_i variables are periodically reinitialised to be consistent with a signed distance function, i.e., satisfying the following equation:

$$|\nabla \phi| = 1 \tag{15}$$

This reinitialisation uses the same fast-marching implementation used for the velocity extension.

Level Set Method Based Topology Optimization

Sequential linear programming level set topology optimization

The velocities required for the level set update are obtained by solving an optimization problem. A generic optimization problem can be formulated using the position of the structural boundary as the design variable:

$$\begin{aligned} \underset{\Omega}{\text{minimize}} \quad & f(\Omega) \\ \text{subject to} \quad & g_i(\Omega) \leq 0 \end{aligned} \tag{16}$$

where $f(\Omega)$ is the objective function and g_i is the i^{th} inequality constraint function. The objective and constraint functions are linearized about the design variables at each k^{th} iteration using a first-order Taylor expansion:

$$\begin{aligned} \underset{\Delta\Omega^k}{\text{minimize}} \quad & \frac{\partial f}{\partial \Omega^k} \cdot \Delta\Omega^k \\ \text{subject to} \quad & \frac{\partial g_i}{\partial \Omega^k} \cdot \Delta\Omega^k \leq -\bar{g}_i^k \end{aligned} \tag{17}$$

where $\Delta\Omega^k$ is the update for the design domain Ω and $\bar{g}_i^k$ is the change in the i^{th} constraint at iteration k.

In the level-set description of the boundary, shape derivatives provide information about how a function changes over time with respect to a movement of the boundary point. They usually take the form of boundary integrals [AJT04]. In this case,

$$\frac{\partial f}{\partial \Omega} \cdot \Delta\Omega = \Delta t \int_\Gamma s_f V_n d\Gamma, \tag{18}$$

$$\frac{\partial g_i}{\partial \Omega} \cdot \Delta\Omega = \Delta t \int_\Gamma s_{gi} V_n d\Gamma, \tag{19}$$

where s_f and s_{g_i} are the shape sensitivity functions for the objective and the i^{th} constraints. Discretizing the boundary at nb points, one can rewrite:

$$\frac{\partial f}{\partial \Omega} \cdot \Delta\Omega \approx \sum_{j=1}^{nb} \Delta t V_{nj} s_{f,j} l_j = \mathbf{C}_f \cdot \mathbf{V}_n \Delta t, \tag{20}$$

$$\frac{\partial g_i}{\partial \Omega} \cdot \Delta\Omega \approx \sum_{j=1}^{nb} \Delta t V_{nj} s_{gi,j} l_j = \mathbf{C}_{gi} \cdot \mathbf{V}_n \Delta t, \tag{21}$$

where l_j is the discrete length of the boundary around the boundary point j, $\mathbf{C}_f$ and $\mathbf{C}_{gi}$ are vectors containing integral coefficients and $\mathbf{V}_n$ is the vector of normal velocities. For a constrained problem, one can write

$$\mathbf{V}_n \Delta t = \alpha \mathbf{d}, \tag{22}$$

where $\mathbf{d}$ is the search direction for the boundary update and $\alpha > 0$ is the actual distance of the boundary movement. Then, the optimization formulation to obtain the optimal boundary velocities can be written as:

$$\begin{aligned} \underset{\alpha^k, \breve{}^k}{\text{minimize}} \quad & \Delta t \mathbf{C}_f^k \cdot \mathbf{V}_n^k(\alpha^k, \boldsymbol{\lambda}^k) \\ \text{subject to} \quad & \Delta t \mathbf{C}_i^k \cdot \mathbf{V}_n^k(\alpha^k, \boldsymbol{\lambda}^k) \leq -\bar{g}_i^k \\ & \mathbf{V}_{n,\min}^k \leq \mathbf{V}_n^k \leq \mathbf{V}_{n,\max}^k \end{aligned} \tag{23}$$

where $\boldsymbol{\lambda}$ are Lagrange multipliers for each constraint function. This optimization problem is solved at every iteration k. More details can be found in [DK15, SDK16].

Velocity extension

Velocity function defined in previous equation is only computed at points along the structural boundary. In order to update the level set function, velocity values at all grid points are required. Thus, the velocity function must be extended or extrapolated to grid points away from the boundary. Natural velocity extension schemes compute strain and sensitivity fields over the entire design domain. Methods that achieve this include filling the void part with a fictitious weak material [AJT04], or smoothing the velocity field over the discontinuity at the boundary edge [WW06]. However, the implicit function often becomes too steep or flat around the boundary, which leads to potential stability issues. Thus, these schemes usually require frequent reinitializing of the implicit function to a signed distance function to maintain stability [AJT04, WW06].

To avoid frequent reinitialization, we employ an extension velocity technique designed to maintain the signed distance function [AS99]. This technique ensures the preservation of the signed distance by using the efficient fast marching method to solve the following equation:

$$\nabla \phi_t \nabla V_{\text{ext}} = 0 \tag{24}$$

where ϕ_t is a temporary signed distance implicit function and V_{ext} is the extended velocity function. The extended velocity function is constrained to maintain the values already computed along the boundary.

Gradient computation

An accurate estimation of the gradient $\nabla\phi$ is essential in solving Hamilton-Jacobi equation using upwind scheme. A current gradient computation evoked internally during level-set update utilizes 5th order Hamilton-Jacobi WENO (weighted essentially non-oscillatory) method. It gives out a better approximation when compared with ENO.

Spatial derivatives of ϕ and spatial stencils are obtained depending on the sign of the velocity and the location of the node in the domain of interest. In any case, five stencil values are obtained with same distances $(v_1, v_2, v_3, v_4, v_5)$. Based on the smoothness of each stencil, we calculate normalized weights (w_1, w_2, w_3) whose sum is equal to unity.

$$\nabla\phi = w_1(2v_1 - 7v_2 + 11v_3) + w_2(5v_3 - v_2 + 2v_4) + w_3(2v_3 + 5v_4 - v_5) \tag{25}$$

Detailed numerical steps can be found in the reference [OF03]

Convergence criterion

The convergence criterion is computed if the volume constraint is satisfied and is defined using the maximum change in compliance over the previous 5 iterations:

$$\Delta C^k = \max\left(|C^k - C^m|/C^k\right), \quad m \in [k-5, k-1] \tag{26}$$

where C^k is the compliance computed at iteration k and the optimization process is terminated if $\Delta C^k < 1e-3$.

Bibliography

[AJT04] Grégoire Allaire, François Jouve, and Anca-Maria Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194(1):363–393, 2004.

[AS99] D Adalsteinsson and J. A Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148(1):2–22, 1999.

[DK15] Peter D. Dunning and H. Alicia Kim. Introducing the sequential linear programming level-set method for topology optimization. *Structural and Multidisciplinary Optimization*, 51(3):631–643, 2015.

[DKM11] Peter D. Dunning, H. Alicia Kim, and Glen Mullineux. Investigation and improvement of sensitivity computation using the area-fraction weighted fixed grid fem and structural optimization. *Finite Elements in Analysis and Design*, 47(8):933–941, 2011.

[OF03] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer, London, 2003.

[SDK16] Raghavendra Sivapuram, Peter D. Dunning, and H. Alicia Kim. Simultaneous material and structural optimization by multiscale topology optimization. *Structural and Multidisciplinary Optimization*, pages 1–15, 2016.

[WW06] Shengyin Wang and Michael Y. Wang. A moving superimposed finite element method for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 65(11):1892–1922, 2006.

[WWG03] Michael Yu Wang, Xiaoming Wang, and Dongming Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1–2):227–246, 2003.