GEOMETRICAL CONSTRAINTS
IN THE LEVEL SET METHOD FOR
SHAPE AND TOPOLOGY OPTIMIZATION

Grégoire ALLAIRE
CMAP, Ecole Polytechnique

Results obtained in collaboration with F. Jouve (LJLL, Paris 7), G. Michailidis (formerly Renault and CMAP).

RODIN project

CONTENTS

1. Introduction.
2. Shape and topology optimization by the level set method.
5. Molding constraints.

Ecole Polytechnique,
UPMC, INRIA,
Renault, EADS,
ESI group, etc.
-I- INTRODUCTION

Shape optimization: minimize an objective function over a set $\mathcal{U}_{ad}$ of admissibles shapes $\Omega$ (including possible constraints)

$$\inf_{\Omega \in \mathcal{U}_{ad}} J(\Omega)$$

The objective function is evaluated through a partial differential equation (state equation)

$$J(\Omega) = \int_{\Omega} j(u_\Omega) \, dx$$

where $u_\Omega$ is the solution of

$$PDE(u_\Omega) = 0 \quad \text{in} \quad \Omega$$

Topology optimization: the optimal topology is unknown.
The model of linear elasticity

A shape is an open set $\Omega \subset \mathbb{R}^d$ with boundary $\partial \Omega = \Gamma \cup \Gamma_N \cup \Gamma_D$.

For a given applied load $g : \Gamma_N \to \mathbb{R}^d$, the displacement $u : \Omega \to \mathbb{R}^d$ is the solution of

$$
\begin{aligned}
- \text{div} \left( A e(u) \right) &= 0 \quad \text{in } \Omega \\
u &= 0 \quad \text{on } \Gamma_D \\
(A e(u)) n &= g \quad \text{on } \Gamma_N \\
(A e(u)) n &= 0 \quad \text{on } \Gamma
\end{aligned}
$$

with the strain tensor $e(u) = \frac{1}{2} (\nabla u + \nabla^t u)$, the stress tensor $\sigma = A e(u)$, and $A$ an homogeneous isotropic elasticity tensor.

Typical objective function: the compliance

$$J(\Omega) = \int_{\Gamma_N} g \cdot u \, dx,$$
Example: the cantilever

Geometrical constraints in topology optimization

G. Allaire
The shape optimization problem is

$$\inf_{\Omega \in \mathcal{U}_{ad}} J(\Omega),$$

where the set of admissible shapes is typically

$$\mathcal{U}_{ad} = \left\{ \Omega \subset D \text{ open set such that } \Gamma_D \bigcup \Gamma_N \subset \partial \Omega \text{ and } \int_{\Omega} dx = V_0 \right\},$$

where $D \subset \mathbb{R}^d$ is a given “working domain” and $V_0$ is a prescribed volume.

- The boundary subsets $\Gamma_D$ and $\Gamma_N$ are fixed. Only $\Gamma$ is optimized (free boundary).

- Existence of optimal shapes is a delicate issue (typically, one needs further constraints in $\mathcal{U}_{ad}$).

- A nice numerical method is the level set algorithm since it allows for topology changes.
Tremendous progresses were achieved on academic research about shape and topology optimization.

There are several commercial softwares used by industry.

But manufacturability of the optimal shapes is not always guaranteed.
Goal of the present work

- We want to add geometrical constraints (for manufacturability), i.e., constraints on $\Omega$, not on the state $u_\Omega$.

- The level set framework is well suited for this because it relies on the distance function to the boundary.

- Issues to be addressed concerning geometrical constraints: modelling, shape differentiation, numerical implementation.

Before that, let’s review the state of the art about the level set method for shape and topology optimization.
-II- LEVEL SET METHOD

A new numerical implementation of an old idea...

☞ Framework of Hadamard’s method of shape variations.

☞ **Main tool**: the level set method of Osher and Sethian (JCP 1988).

☞ Shape capturing algorithm.

☞ Fixed mesh: low computational cost.

Shape tracking

Shape capturing
Shape capturing method on a fixed mesh of the “working domain” $D$.

A shape $\Omega$ is parametrized by a level set function

$$
\begin{aligned}
\psi(x) = 0 & \iff x \in \partial \Omega \cap D \\
\psi(x) < 0 & \iff x \in \Omega \\
\psi(x) > 0 & \iff x \in (D \setminus \Omega)
\end{aligned}
$$

Assume that the shape $\Omega(t)$ evolves in time $t$ with a normal velocity $V(t, x)$. Then its motion is governed by the following Hamilton Jacobi equation

$$
\frac{\partial \psi}{\partial t} + V |\nabla_x \psi| = 0 \quad \text{in } D.
$$
Example of a level set function
Advection velocity = shape gradient

The velocity $V$ is deduced from the shape gradient of the objective function. To compute this shape gradient we recall the well-known Hadamard’s method. Let $\Omega_0$ be a reference domain. Shapes are parametrized by a vector field $\theta$

$$\Omega = (\text{Id} + \theta)\Omega_0 \quad \text{with} \quad \theta \in C^1(\mathbb{R}^d; \mathbb{R}^d).$$
**Shape derivative**

**Definition:** the shape derivative of $J(\Omega)$ at $\Omega_0$ is the Fréchet differential of $\theta \to J((\text{Id} + \theta)\Omega_0)$ at 0.

**Hadamard structure theorem:** the shape derivative of $J(\Omega)$ can always be written (in a distributional sense)

$$J'(\Omega_0)(\theta) = \int_{\partial\Omega_0} \theta(x) \cdot n(x) j(x) \, ds$$

where $j(x)$ is an integrand depending on the state $u$ and an adjoint $p$.

We choose the velocity $V = \theta \cdot n$ such that $J'(\Omega_0)(\theta) \leq 0$.

**Simplest choice:** $V = \theta \cdot n = -j$ but other ones are possible (including regularization).
SHAPE DERIVATIVE OF THE COMPLIANCE

\[ J(\Omega) = \int_{\Gamma_N} g \cdot u_{\Omega} \, ds = \int_{\Omega} A e(u_{\Omega}) \cdot e(u_{\Omega}) \, dx, \]

where \( u_{\Omega} \) is the state variable in \( \Omega \).

\[ J'(\Omega)(\theta) = -\int_{\Gamma} A e(u_{\Omega}) \cdot e(u_{\Omega}) \theta \cdot n \, ds, \]

Remarks:

1. self-adjoint problem (no adjoint state is required),

2. taking into account the volume constraint add a fixed Lagrange multiplier \( \lambda - A e(u_{\Omega}) \cdot e(u_{\Omega}) \).
1. Initialization of the level set function $\psi_0$ (including holes).

2. Iteration until convergence for $k \geq 1$:
   
   (a) Compute the elastic displacement $u_k$ for the shape $\psi_k$.
   Deduce the shape gradient = normal velocity = $V_k$

   (b) Advect the shape with $V_k$ (solving the Hamilton Jacobi equation) to obtain a new shape $\psi_{k+1}$.

For numerical examples, see the web page:
http://www.cmap.polytechnique.fr/~optopo/level_en.html
Examples of results with complex topologies
We focus on thickness control because of

- manufacturability,
- uncertainty in the microscale (MEMS design),
- robust design (fatigue, buckling, etc.).

Existing works:

- Several approaches in the framework of the SIMP method to ensure minimum length scale (Sigmund, Poulsen, Guest, etc.).
- In the level-set framework: Chen, Wang and Liu implicitly control the feature size by adding a ”line” energy term to the objective function; Alexandrov and Santosa kept a fixed topology by using offset sets.
- Many works in image processing.
Definition. Let $\Omega \subset \mathbb{R}^d$ be a bounded domain. The signed distance function to $\Omega$ is the function $\mathbb{R}^d \ni x \mapsto d_\Omega(x)$ defined by:

$$d_\Omega(x) = \begin{cases} 
-d(x, \partial\Omega) & \text{if } x \in \Omega \\
0 & \text{if } x \in \partial\Omega \\
d(x, \partial\Omega) & \text{if } x \in \mathbb{R}^d \setminus \Omega 
\end{cases}$$

where $d(\cdot, \partial\Omega)$ is the usual Euclidean distance.
**Constraint formulations**

**Maximum thickness.**

Let $d_{\text{max}}$ be the maximum allowed thickness. The constraint reads:

$$d_{\Omega}(x) \geq -d_{\text{max}}/2 \quad \forall x \in \Omega$$

**Minimum thickness**

Let $d_{\text{min}}$ be the minimum allowed thickness. The constraint reads:

$$d_{\Omega}(x - d_{\text{off}}n(x)) \leq 0 \quad \forall x \in \partial\Omega, \forall d_{\text{off}} \in [0, d_{\text{min}}]$$

**Remark:** similar constraints for the thickness of holes.
For minimum thickness we rely on the classical notion of offset sets of the boundary of a shape, defined by

$$\{ x - d_{\text{off}} n(x) \quad \text{such that } x \in \partial \Omega \}$$
Caution with minimum thickness!

Writing a constraint for a single (large) value of $d_{\text{off}}$ does not work!

This is the reason why all values of $d_{\text{off}}$ between 0 and $d_{\text{min}}$ are taken into account.
Quadratic penalty method

We reformulate the pointwise constraint into a global one denoted by $P(\Omega)$.

**Maximum thickness**

$$P(\Omega) = \int_{\Omega} \left[ (d_{\Omega}(x) + \frac{d_{\text{max}}}{2})^2 \right] dx$$

**Minimum thickness**

$$P(\Omega) = \int_{\partial \Omega} \int_{0}^{d_{\text{min}}} \left[ (d_{\Omega}(x) - d_{\text{off}} n(x))^2 \right] dx \, dd_{\text{off}}$$

where $f^+ = \max(f, 0)$ and $f^- = \min(f, 0)$. 
Property of the signed distance function

The signed distance function has a tent-like shape.
- The **skeleton** (or ridge) is made of the points \( x \in \Omega \) where there are multiple minimizers for

\[
d(x, \partial \Omega) = \min_{y \in \partial \Omega} \|x - y\|.
\]

- Equivalently, the skeleton is the set of points where \( n = \nabla d_{\Omega} \) is discontinuous.

- Equivalently (Huygens principle) the skeleton is the geometric location of centers of maximal disks.
Rays and skeleton (Ctd.)

The ray issued from \( x \in \partial \Omega \) is the integral curve of \( n = \nabla d_{\Omega} \).

The rays are straight lines because \( \dot{x}(t) = n(x(t)) \) implies \( t = d_{\Omega}(x(t)) \).
Shape derivative of the signed-distance function

Lemma. Fix \( x \in \Omega \setminus \text{Skeleton} \). Define \( p_{\partial \Omega}(x) \) the unique point on \( \partial \Omega \) such that

\[
d(x, \partial \Omega) = \| x - p_{\partial \Omega}(x) \|.
\]

Then, the "pointwise" shape derivative is

\[
d'_{\Omega}(\theta)(x) = \left( \theta \cdot n \right) (p_{\partial \Omega}(x)).
\]

Remarks.

- The computation of the shape derivative of the signed-distance function is classical (e.g. Delfour and Zolesio).
- The shape derivative \( d'_{\Omega}(\theta) \) remains constant along the normal and rays (but is discontinuous on the skeleton).
Lemma. For a function $j \in C^1(\mathbb{R})$ define

$$ J(\Omega) = \int_D j(d_\Omega(x)) \, dx. $$

Then $J$ is shape differentiable and

$$ J'(\Omega)(\theta) = -\int_D j'(d_\Omega(x)) \left( \theta \cdot n \right) (p_{\partial\Omega}(x)) \, dx $$

or equivalently for a $C^2$ domain $\Omega$ (by using a coarea formula)

$$ J'(\Omega)(\theta) = -\int_{\partial\Omega} \left( \theta \cdot n \right)(y) \left( \int_{\text{ray}(y)} j'(d_\Omega(x)) \prod_{i=1}^{d-1} (1 + d_\Omega(x)\kappa_i(y)) \, ds \right) \, dy $$

with $\kappa_i$ the principal curvatures of $\partial\Omega$, $\text{ray}(y) = \{ x = y - s \, n(y) \}$ and $s$ the curvilinear abcissa.

In numerical practice we approximate the Jacobian by 1.
All the geometrical computations (skeleton, offset, projection, etc.) are standard and very cheap (compared to the elasticity analysis).

All our numerical examples are for compliance minimization (except otherwise mentioned).

Optimization: we use an augmented Lagrangian method.

At convergence, the geometrical constraints are exactly satisfied.

All results have been obtained with our software developed in the finite element code SYSTUS of ESI group.
Maximum thickness (MBB, solution without constraint)
Maximum thickness (solution with increasing constraint)
Maximum thickness (3d MBB beam)
Geometrical constraints in topology optimization

G. Allaire
Minimum thickness (MBB beam)
Geometrical constraints in topology optimization

G. Allaire
Minimum thickness (3d)
Moldability condition: $d_i \cdot n(x) \geq 0$, $\forall x \in \Gamma_i$. 

Parting surfaces $\Gamma_i$ and draw directions $d_i$: castable (left), not castable (right).
Sufficient conditions for molding

Starting from a **castable** initial design:

Xia et al. (SMO 2010) proposed to project the velocity

\[ \theta_i(x) = \lambda(x) d_i, \quad \forall x \in \Gamma_i. \]

Starting from a **non-castable** initial design:

we suggest the constraint

\[ d_\Omega (x + \xi d_i) \geq 0 \quad \forall x \in \Gamma_i, \quad \forall \xi \in [0, \text{dist}(x, \partial D)]. \]
No constraint (top), vertical draw direction (bottom).
Parting surface fixed at bottom (left) and free (right).
Industrial test case (courtesy of Renault): no molding constraint (left), out of plane draw direction (right).
Conclusion

- Work still going on.
- Other penalizations of the geometrical constraints.
- Should we apply the constraints from the start or near the end?
- What if we want to stay feasible at each iteration?
- Handling several constraints simultaneously.
- Better optimization algorithm: sequential linear programming with trust region.