Solution Techniques for Constrained Shape Optimization Problems in Shape Spaces

Kathrin Welker

Trier University

Seminar No. 125 on "Piecewise smooth system and optimization with piecewise linearization via algorithmic differentiation"

Shonan Village Center, Japan

June 25-28, 2018

Table of contents

$$\min_{\Omega} J(\Omega)$$
PDE or VI constraints

- J shape functional depending on a solution of a partial differential equation (PDE) or a variational inequality (VI)

  (Definition: Let $D \subset \mathbb{R}^d$ be non-empty and let $\mathcal{A} \subset \{\Omega \colon \Omega \subset D\}$ denote a set of subsets. A function $J \colon \mathcal{A} \to \mathbb{R}$, $\Omega \mapsto J(\Omega)$ is called a *shape functional*.)

- Ω shape

  $\rightsquigarrow$ *2D shape:* Simply connected, compact subset Ω of $\mathbb{R}^2$ with $\Omega \neq \emptyset$ and $\mathcal{C}^\infty$ boundary $\partial\Omega$

  $\rightsquigarrow$ *How is the set of all shapes defined?*

Space of smooth shapes

2D shape: Simply connected, compact subset Ω of $\mathbb{R}^2$ with $\Omega \neq \varnothing$ and $\mathcal{C}^\infty$ boundary $\partial\Omega$

2D shape: Simply connected, compact subset Ω of $\mathbb{R}^2$ with $\Omega \neq \emptyset$ and $\mathcal{C}^\infty$ boundary $\partial\Omega$

Shape space

$$\boxed{B_e := \text{Emb}(S^1, \mathbb{R}^2)/\text{Diff}(S^1)}$$

Space of smooth shapes

2D shape: Simply connected, compact subset Ω of $\mathbb{R}^2$ with $\Omega \neq \emptyset$ and $\mathcal{C}^\infty$ boundary $\partial\Omega$

Shape space

$$\boxed{B_e := \text{Emb}(S^1, \mathbb{R}^2)/\text{Diff}(S^1)}$$

- B_e is a manifold (cf. [1])
- *Generalization to higher dimensions:* $B_e(M,N) := \text{Emb}(M,N)/\text{Diff}(M)$, where M is a compact manifold and N denotes a Riemannian manifold with $\dim(M) < \dim(N)$

[1] A. Kriegl and P. Michor. *The Convient Setting of Global Analysis*, volume 53 of *Mathematical Surveys and Monographs*. American Mathematical Society, 1997.

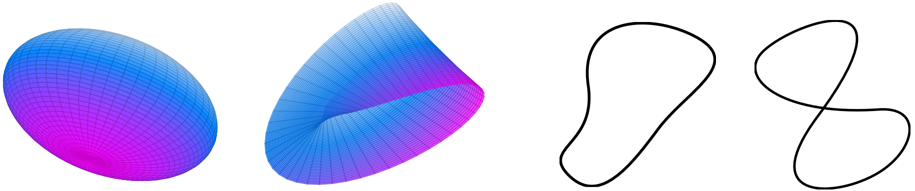
Space of smooth shapes

2D shape: Simply connected, compact subset Ω of $\mathbb{R}^2$ with $\Omega \neq \emptyset$ and $\mathcal{C}^\infty$ boundary $\partial\Omega$

Shape space

$$B_e := \text{Emb}(S^1, \mathbb{R}^2)/\text{Diff}(S^1)$$

- B_e is a manifold (cf. [1])
- *Generalization to higher dimensions:* $B_e(M,N) := \text{Emb}(M,N)/\text{Diff}(M)$, where M is a compact manifold and N denotes a Riemannian manifold with $\dim(M) < \dim(N)$



$b_1 \in B_e(S^2, \mathbb{R}^3)$ $b_2 \notin B_e(S^2, \mathbb{R}^3)$ $c_1 \in B_e(S^1, \mathbb{R}^2)$ $c_2 \notin B_e(S^1, \mathbb{R}^2)$

[1] A. Kriegl and P. Michor. *The Convient Setting of Global Analysis*, volume 53 of *Mathematical Surveys and Monographs*. American Mathematical Society, 1997.

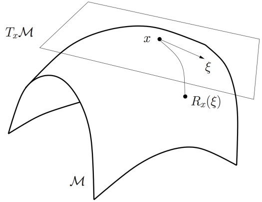
Optimization on manifolds (cf. [2])

- Line-search methods in $\mathbb{R}^n$: $x_{k+1} = x_k + t_k\xi_k$
- Manifolds are not necessarily linear spaces
  $\Rightarrow$ Select ξ_k as a tangent vector to $\mathcal{M}$ at x_k
  $\Rightarrow$ Points from the tangent space have to be mapped to the manifold

[2] P.-A. Absil, R. Mahony and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Optimization on manifolds (cf. [2])

- Line-search methods in $\mathbb{R}^n$: $x_{k+1} = x_k + t_k \xi_k$
- Manifolds are not necessarily linear spaces
  $\Rightarrow$ Select ξ_k as a tangent vector to $\mathcal{M}$ at x_k
  $\Rightarrow$ Points from the tangent space have to be mapped to the manifold



Example: Steepest descent method on B_e

Goal: Find a solution of $\min_{x \in B_e} J(x)$

Input: $x_0 \in B_e$

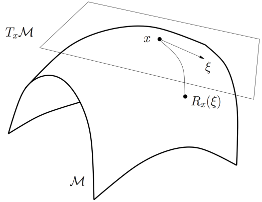
for $k = 0, 1, \ldots$ **do**

[1] Compute the increment $\xi_k := -\text{grad} J(x^k) \in T_{x_k} B_e$

[2] Set $x_{k+1} := \mathcal{R}_{x_k}(t_k \xi_k)$ for some steplength t_k and a retraction $\mathcal{R}$

[2] P.-A. Absil, R. Mahony and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

- Line-search methods in $\mathbb{R}^n$: $x_{k+1} = x_k + t_k\xi_k$
- Manifolds are not necessarily linear spaces
  $\Rightarrow$ Select ξ_k as a tangent vector to $\mathcal{M}$ at x_k
  $\Rightarrow$ Points from the tangent space have to be mapped to the manifold



Example: Steepest descent method on B_e

Goal: Find a solution of $\min\limits_{x \in B_e} J(x)$

Input: $x_0 \in B_e$

for $k = 0, 1, \ldots$ **do**

[1] Compute the increment $\xi_k := -\text{grad}J(x^k) \in T_{x_k}B_e$, where grad$J$ is a Riemannian shape gradient

[2] Set $x_{k+1} := \mathcal{R}_{x_k}(t_k\xi_k)$ for some steplength t_k and a retraction $\mathcal{R}$

[2] P.-A. Absil, R. Mahony and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

A *Riemannian metric* on a manifold M is a collection $g = (g_p)_{p \in M}$ of inner products

$$g_p \colon T_pM \times T_pM \to \mathbb{R}, \ (v, w) \mapsto g_p(v, w),$$

one for every $p \in M$, such that the map $M \to \mathbb{R}, \ p \mapsto g_p(X(p), Y(p))$ is smooth for every pair of vector fields X, Y on M.

A *Riemannian metric* on a manifold M is a collection $g = (g_p)_{p\in M}$ of inner products

$$g_p\colon T_pM \times T_pM \to \mathbb{R},\ (v,w) \mapsto g_p(v,w),$$

one for every $p \in M$, such that the map $M \to \mathbb{R},\ p \mapsto g_p(X(p), Y(p))$ is smooth for every pair of vector fields X, Y on M.

Sobolev metric (cf. [3])

$$\boxed{g^1\colon T_cB_e \times T_cB_e \to \mathbb{R},\ (h,k) \mapsto \langle (I - A\triangle_c)\alpha, \beta\rangle_{L^2(c)} \text{ with } A > 0}$$

Here $h = \alpha n$ and $k = \beta n$ denote two elements of the tangent space

$$T_cB_e \cong \{\psi \mid \psi = \alpha n,\ \alpha \in \mathcal{C}^\infty(S^1, \mathbb{R})\}.$$

[3] M. Bauer, P. Harms, and P. Michor. Sobolev metrics on shape space of surfaces. *JGM*, 3(4):389–438, 2011.

Riemannian metrics on B_e

A *Riemannian metric* on a manifold M is a collection $g = (g_p)_{p\in M}$ of inner products

$$g_p \colon T_pM \times T_pM \to \mathbb{R},\ (v,w) \mapsto g_p(v,w),$$

one for every $p \in M$, such that the map $M \to \mathbb{R},\ p \mapsto g_p(X(p), Y(p))$ is smooth for every pair of vector fields X, Y on M.

Sobolev metric (cf. [3])

$$\boxed{g^1 \colon T_cB_e \times T_cB_e \to \mathbb{R},\ (h,k) \mapsto \langle (I - A\triangle_c)\alpha, \beta\rangle_{L^2(c)} \text{ with } A > 0}$$

Here $h = \alpha n$ and $k = \beta n$ denote two elements of the tangent space

$$T_cB_e \cong \{\psi \mid \psi = \alpha n,\ \alpha \in \mathcal{C}^\infty(S^1, \mathbb{R})\}.$$

Riemannian shape gradient w.r.t. g^1

Representation of $DJ(x)$ such that $\quad g^1(\mathrm{grad}J(x), h) = DJ(x)[h] \quad \forall h \in T_xB_e$

[3] M. Bauer, P. Harms, and P. Michor. Sobolev metrics on shape space of surfaces. *JGM*, 3(4):389–438, 2011.

Shape derivative

Definition

Let $D \subset \mathbb{R}^d$ be non-empty and open and let $\Omega \subset D$ be measurable. The *shape derivative* of a shape functional J at Ω in direction $V \in \mathcal{C}_0^k(D, \mathbb{R}^d)$, $k \in \mathbb{N} \cup \{\infty\}$, is defined by the *Eulerian derivative*

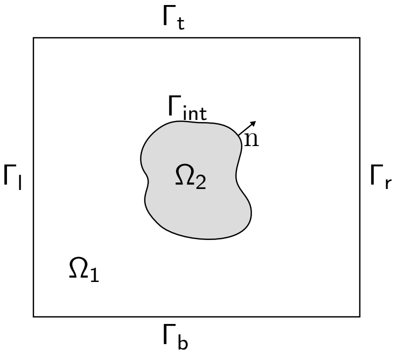
$$DJ(\Omega)[V] = \lim_{t \to 0^+} \frac{J(\Omega_t) - J(\Omega)}{t} \qquad (\Omega_t = \{x + t \cdot V(x) : x \in \Omega\}).$$

- Shape derivative of a shape differentiable functional is a directional derivative in the direction of a differentiable vector field
- ***Hadamard Structure Theorem*** (cf. Theorem 2.17 in [4]):
  - Only the normal part of a vector field on the boundary Γ of a domain Ω has an impact on the value of the shape derivative
  - Existence of a scalar distribution r on Γ
  - If $r \in L^1(\Gamma)$, then $DJ(\Omega)[V] = \int_\Gamma r \langle V, n \rangle \, ds$
- r is often called the *shape gradient* but gradients depend on chosen scalar products defined on the space under consideration

[4] J. Sokolowski and J.-P. Zolésio. *Introduction to Shape Optimization*, volume 16 of *Computational Mathematics* Springer, 1992.

Contents

- Diffusion process
- $\Omega \subset \mathbb{R}^2$ bounded Lipschitz domain
- Two materials $\Omega_1, \Omega_2 \subset \Omega$ with different permeability
- Fixed outer boundary $\partial\Omega = \Gamma_b \,\dot\cup\, \Gamma_l \,\dot\cup\, \Gamma_r \,\dot\cup\, \Gamma_t$
- Variable boundary $\Gamma_{int} \in B_e(S^1, \mathbb{R}^2)$ $\Rightarrow$ Fit to measured concentration
- Homogeneous concentration in Ω for $t = 0$
- Higher concentration on the top in the beginning



Diffusion problem

For **concentration** y and **diffusion coefficient** k:

$$\min_{\Gamma_{\text{int}}} J(y,\Omega) = j(y,\Omega) + j^{\text{reg}}(\Omega) := \frac{1}{2}\int_0^T\int_\Omega (y-\bar{y})^2 dx\, dt + \mu \int_{\Gamma_{\text{int}}} 1\, ds$$

$$\text{s.t. } \frac{\partial y}{\partial t} - \operatorname{div}(k\nabla y) = f \quad \text{in } \Omega \times (0,T]$$

$$y = 1 \quad \text{on } \Gamma_t \times (0,T]$$

$$\frac{\partial y}{\partial n} = 0 \quad \text{on } (\Gamma_b \cup \Gamma_l \cup \Gamma_r) \times (0,T]$$

$$y = y_0 \quad \text{in } \Omega \times \{0\}$$

- Data measurements: $\bar{y} \in L^2\left(0,T; L^2(\Omega)\right)$
- Right hand-side: $f = \text{const.} \quad \text{in } \Omega \times (0,T]$
- Jumping coefficient: $k = \begin{cases} k_1 = \text{const.} & \text{in } \Omega_1 \\ k_2 = \text{const.} & \text{in } \Omega_2 \end{cases}$
- Transmission conditions: $[\![y]\!] = 0, \quad \left[\!\!\left[k\frac{\partial y}{\partial n}\right]\!\!\right] = 0 \quad \text{on } \Gamma_{\text{int}} \times (0,T]$

Shape derivative

Shape derivative of the regularization term j^{reg}

$$Dj^{\text{reg}}(\Omega)[V] = \mu \int_{\Gamma_{\text{int}}} \langle V, n \rangle \, \kappa \, ds$$

Shape derivative

Shape derivative of the regularization term j^{reg}

$$Dj^{\text{reg}}(\Omega)[V] = \mu \int_{\Gamma_{\text{int}}} \langle V, n \rangle \, \kappa \, ds$$

Shape derivative of the objective functional j

$$\boxed{Dj_\Omega(y,\Omega)[V] = Dj(y,\Omega)[V] = Dj_{\Gamma_{\text{int}}}(y,\Omega)[V]}$$

- $$Dj_\Omega(y,\Omega)[V] = \int_0^T \int_\Omega -k\nabla y^\top \left(\nabla V + \nabla V^\top\right) \nabla p - p \nabla f^\top V + \operatorname{div}(V) \left(\frac{1}{2}(y-\overline{y})^2 + \frac{\partial y}{\partial t} p + k \nabla y^\top \nabla p - fp\right) dx\, dt$$
- $$Dj_{\Gamma_{\text{int}}}(y,\Omega)[V] = \int_0^T \int_{\Gamma_{\text{int}}} [\![k]\!] \, \nabla y_1^\top \nabla p_2 \, \langle V, n \rangle \, ds\, dt$$

Shape derivative

Shape derivative of the regularization term j^{reg}

$$Dj^{\text{reg}}(\Omega)[V] = \mu \int_{\Gamma_{\text{int}}} \langle V, n \rangle \, \kappa \, ds$$

Shape derivative of the objective functional j

$$\boxed{Dj_{\Omega}(y,\Omega)[V] = Dj(y,\Omega)[V] = Dj_{\Gamma_{\text{int}}}(y,\Omega)[V]}$$

- $$Dj_{\Omega}(y,\Omega)[V] = \int_0^T \int_{\Omega} -k \nabla y^{\top} \left(\nabla V + \nabla V^{\top} \right) \nabla p - p \nabla f^{\top} V + \operatorname{div}(V) \left(\frac{1}{2} (y - \overline{y})^2 + \frac{\partial y}{\partial t} p + k \nabla y^{\top} \nabla p - fp \right) dx \, dt$$
- $$Dj_{\Gamma_{\text{int}}}(y,\Omega)[V] = \int_0^T \int_{\Gamma_{\text{int}}} [\![k]\!] \, \nabla y_1^{\top} \nabla p_2 \, \langle V, n \rangle \, ds \, dt$$
  - Volume formulation: $y \in L^2\left(0, T; H^1(\Omega)\right), p \in W\left(0, T; H^1(\Omega)\right)$
  - Surface formulation: H^2-regularity in space is necessary

Shape derivative of the regularization term j^{reg}

$$Dj^{\text{reg}}(\Omega)[V] = \mu \int_{\Gamma_{\text{int}}} \langle V, n\rangle \, \kappa \, ds$$

Shape derivative of the objective functional j

$$\boxed{Dj_{\Omega}(y,\Omega)[V] = Dj(y,\Omega)[V] = Dj_{\Gamma_{\text{int}}}(y,\Omega)[V]}$$

- $$Dj_{\Omega}(y,\Omega)[V] = \int_0^T \int_{\Omega} -k\nabla y^{\top}\left(\nabla V + \nabla V^{\top}\right)\nabla p - p\nabla f^{\top} V + \operatorname{div}(V)\left(\frac{1}{2}(y-\overline{y})^2 + \frac{\partial y}{\partial t}p + k\nabla y^{\top}\nabla p - fp\right) dx\, dt$$
- $$Dj_{\Gamma_{\text{int}}}(y,\Omega)[V] = \int_0^T \int_{\Gamma_{\text{int}}} [\![k]\!]\, \nabla y_1^{\top} \nabla p_2 \, \langle V, n\rangle \, ds\, dt = g^1(\operatorname{grad}^{B_e} j(y,\Omega), V) \;\; \forall V \in T_{\Gamma_{\text{int}}} B_e$$

  - Volume formulation: $y \in L^2\left(0,T;H^1(\Omega)\right), p \in W\left(0,T;H^1(\Omega)\right)$
  - Surface formulation: H^2-regularity in space is necessary

Algorithm on (B_e, g^1) (cf. [5])

Evaluate measurements

↓

Solve the state and adjoint equation

↓

Assemble the Dirichlet boundary condition of the linear elasticity equation:

1. Compute the Riemannian shape gradient with respect to g^1 from the surface shape derivative $D_\Gamma \mathcal{L}[V] = \int_\Gamma r \langle V, n \rangle \, ds$ $(r \in L^1(\Gamma))$:

$$\operatorname{grad}^{B_e} \mathcal{L} = qn \text{ mit } (I - A\triangle_\Gamma)q = r \quad (\Gamma \in B_e, A > 0, q \in \mathcal{C}^\infty(\Gamma))$$

2. Compute a Riemannian limited-memory BFGS update

↓

Solve the linear elasticity equation with source term equals zero

↓

Apply the resulting deformation to the finite element mesh

(loop back to: Evaluate measurements)

[5] V. Schulz, M. Siebenborn and K. W. Structured inverse modeling in parabolic diffusion problems. *SICON*, 53(6):3319-3338, 2015.

Contents

Steklov-Poincaré: Motivation and aims

- Derivation of surface shape derivative formulations is a time-consuming process
  $\Rightarrow$ ***Aim:*** *Usage of volume shape derivative expressions*

- Gradient representation and afterwards mesh deformation
  $\Rightarrow$ ***Aim:*** *Gradient representation and mesh deformation all at once*

- $\mathcal{C}^\infty$-shapes
  $\Rightarrow$ ***Aim:*** *Weaken the assumption of $\mathcal{C}^\infty$-shapes*

- Derivation of surface shape derivative formulations is a time-consuming process
  $\Rightarrow$ ***Aim:*** *Usage of volume shape derivative expressions*
- Gradient representation and afterwards mesh deformation
  $\Rightarrow$ ***Aim:*** *Gradient representation and mesh deformation all at once*
- $\mathcal{C}^\infty$-shapes
  $\Rightarrow$ ***Aim:*** *Weaken the assumption of $\mathcal{C}^\infty$-shapes*

Definition of an inner product based on volume formualtions

- Inner product should be derived from the second shape derivative
- Second shape derivative can be related to the Steklov-Poincaré operator (cf. [6])

$\Rightarrow$ Definition of a *Steklov-Poincaré type metric*

[6] S. Schmidt and V. Schulz. Impulse response approximations of discrete shape Hessians with application in CFD. *SICON*, 48(4):2562–2580, 2009.

Definition

$$g^S \colon H^{1/2}(\Gamma) \times H^{1/2}(\Gamma) \to \mathbb{R},\ (\alpha,\beta) \mapsto \int_\Gamma \alpha \cdot (S^{pr})^{-1} \beta \, ds$$

Here S^{pr} denotes the projected Poincaré-Steklov operator and is given by

$$S^{pr} \colon H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma),\ \xi \mapsto (\gamma_0 U)^\top n,$$

where $U \in H_0^1(\Omega, \mathbb{R}^d)$ solves $a(U,V) = \int_\Gamma \xi \cdot (\gamma_0 V)^\top n \, ds \ \forall V \in H_0^1(\Omega, \mathbb{R}^d)$.

Definition

$$g^S : H^{1/2}(\Gamma) \times H^{1/2}(\Gamma) \to \mathbb{R},\ (\alpha, \beta) \mapsto \int_\Gamma \alpha \cdot (S^{pr})^{-1} \beta \, ds$$

Here S^{pr} denotes the projected Poincaré-Steklov operator and is given by

$$S^{pr} : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma),\ \xi \mapsto (\gamma_0 U)^\top n,$$

where $U \in H_0^1(\Omega, \mathbb{R}^d)$ solves $a(U, V) = \int_\Gamma \xi \cdot (\gamma_0 V)^\top n \, ds \ \forall V \in H_0^1(\Omega, \mathbb{R}^d)$.

Shape derivative in surface formulation $DJ_\Gamma[V] = \int_\Gamma r \langle V, n \rangle \, ds$

$\rightsquigarrow$ **Shape gradient with respect to g^S** is given by $h \in T_\Gamma B_e \cong \mathcal{C}^\infty(\Gamma)$ s.t.

$$g^S(\psi, h) = \langle r, \psi \rangle_{L^2(\Gamma)} \qquad \forall \psi \in \mathcal{C}^\infty(\Gamma)$$

Definition

$$\boxed{g^S\colon H^{1/2}(\Gamma)\times H^{1/2}(\Gamma)\to\mathbb{R},\ (\alpha,\beta)\mapsto\int_\Gamma \alpha\cdot\left(S^{\mathrm{pr}}\right)^{-1}\beta\, ds}$$

Here S^{pr} denotes the projected Poincaré-Steklov operator and is given by

$$S^{\mathrm{pr}}\colon H^{-1/2}(\Gamma)\to H^{1/2}(\Gamma),\ \xi\mapsto(\gamma_0 U)^\top n,$$

where $U\in H^1_0(\Omega,\mathbb{R}^d)$ solves $a(U,V)=\int_\Gamma \xi\cdot(\gamma_0 V)^\top n\, ds\ \forall V\in H^1_0(\Omega,\mathbb{R}^d)$.

Shape derivative in surface formulation $DJ_\Gamma[V]=\int_\Gamma r\langle V,n\rangle\, ds$

$\rightsquigarrow$ **Shape gradient with respect to g^S** is given by $h\in T_\Gamma B_e\cong\mathcal{C}^\infty(\Gamma)$ s.t.

$$g^S(\psi,h)=\langle r,\psi\rangle_{L^2(\Gamma)}\quad\left(\Leftrightarrow\int_\Gamma\psi\cdot\left(S^{\mathrm{pr}}\right)^{-1}h\, ds=\int_\Gamma r\psi\, ds\right)\quad\forall\psi\in\mathcal{C}^\infty(\Gamma)$$

Definition

$$g^S \colon H^{1/2}(\Gamma) \times H^{1/2}(\Gamma) \to \mathbb{R},\ (\alpha,\beta) \mapsto \int_\Gamma \alpha \cdot (S^{pr})^{-1} \beta \, ds$$

Here S^{pr} denotes the projected Poincaré-Steklov operator and is given by

$$S^{pr} \colon H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma),\ \xi \mapsto (\gamma_0 U)^\top n,$$

where $U \in H_0^1(\Omega,\mathbb{R}^d)$ solves $a(U,V) = \int_\Gamma \xi \cdot (\gamma_0 V)^\top n \, ds \ \forall V \in H_0^1(\Omega,\mathbb{R}^d)$.

Shape derivative in surface formulation $DJ_\Gamma[V] = \int_\Gamma r \langle V, n\rangle \, ds$

$\rightsquigarrow$ **Shape gradient with respect to g^S** is given by $h \in T_\Gamma B_e \cong \mathcal{C}^\infty(\Gamma)$ s.t.

$$g^S(\psi,h) = \langle r,\psi\rangle_{L^2(\Gamma)} \quad \left(\Leftrightarrow \int_\Gamma \psi \cdot (S^{pr})^{-1} h \, ds = \int_\Gamma r \psi \, ds\right) \quad \forall \psi \in \mathcal{C}^\infty(\Gamma)$$

$\Rightarrow h = S^{pr} r = (\gamma_0 U)^\top n$, where $U \in H_0^1(\Omega,\mathbb{R}^d)$ solves

$$a(U,V) = \int_\Gamma r \cdot (\gamma_0 V)^\top n \, ds \qquad \forall V \in H_0^1(\Omega,\mathbb{R}^d)$$

Definition

$$g^S\colon H^{1/2}(\Gamma)\times H^{1/2}(\Gamma)\to\mathbb{R},\ (\alpha,\beta)\mapsto\int_\Gamma \alpha\cdot(S^{\mathrm{pr}})^{-1}\beta\, ds$$

Here S^{pr} denotes the projected Poincaré-Steklov operator and is given by

$$S^{\mathrm{pr}}\colon H^{-1/2}(\Gamma)\to H^{1/2}(\Gamma),\ \xi\mapsto(\gamma_0 U)^\top n,$$

where $U\in H^1_0(\Omega,\mathbb{R}^d)$ solves $a(U,V)=\int_\Gamma \xi\cdot(\gamma_0 V)^\top n\, ds\ \forall V\in H^1_0(\Omega,\mathbb{R}^d)$.

Shape derivative in surface formulation $DJ_\Gamma[V]=\int_\Gamma r\langle V,n\rangle\, ds$

$\rightsquigarrow$ **Shape gradient with respect to g^S** is given by $h\in T_\Gamma B_e\cong\mathcal{C}^\infty(\Gamma)$ s.t.

$$g^S(\psi,h)=\langle r,\psi\rangle_{L^2(\Gamma)}\quad\left(\Leftrightarrow\int_\Gamma\psi\cdot(S^{\mathrm{pr}})^{-1}h\, ds=\int_\Gamma r\psi\, ds\right)\quad\forall\psi\in\mathcal{C}^\infty(\Gamma)$$

$\Rightarrow h=S^{\mathrm{pr}}r=(\gamma_0 U)^\top n$, where $U\in H^1_0(\Omega,\mathbb{R}^d)$ solves

$$a(U,V)=\int_\Gamma r\cdot(\gamma_0 V)^\top n\, ds\ =DJ_\Gamma[V]=DJ_\Omega[V]\quad\forall V\in H^1_0(\Omega,\mathbb{R}^d)$$

Construction of the g^S-metric yields

$$g^S(u,v) = DJ_\Gamma[V] = DJ_\Omega[V] = a(U,V) \quad \forall V \in H_0^1(\Omega,\mathbb{R}^d) \text{ and } v = (\gamma_0 V)^\top n$$

$\Rightarrow$ *Shape gradient representation* $u = (\gamma_0 U)^\top n$ *and mesh deformation* $U \in H_0^1(\Omega,\mathbb{R}^d)$

Results

Construction of the g^S-metric yields

$$g^S(u,v) = DJ_\Gamma[V] = DJ_\Omega[V] = a(U,V) \quad \forall V \in H_0^1(\Omega,\mathbb{R}^d) \text{ and } v = (\gamma_0 V)^\top n$$

⇒ *Shape gradient representation $u = (\gamma_0 U)^\top n$ and mesh deformation $U \in H_0^1(\Omega,\mathbb{R}^d)$*

We have to solve the ***deformation equation***:

$$a(U,V) = b(V) \quad \forall V \in H_0^1(\Omega,\mathbb{R}^d)$$

- $a(\cdot,\cdot)$ symmetric and coercive bilinearform
- $b(V) := DJ_{\text{vol}}(\Omega)[V] + DJ_{\text{surf}}(\Omega)[V]$
  - $J_{\text{vol}}(\Omega)$ parts of J leading to volume shape derivative expressions
  - $J_{\text{surf}}(\Omega)$ parts of J leading to surface shape derivative expressions

⇒ Combination of surface and volume formulation of shape derivatives

Algorithm on (B_e, g^S) (cf. [7])

Evaluate measurements

↓

Solve the state and adjoint equation

↓

Assemble the deformation equation:

- Assemble $DJ_{\text{vol}}(\Omega)[V]$ for V with $\Gamma \cap \text{supp}(V) \neq 0$ as source term
- Assemble $DJ_{\text{surf}}(\Omega)[V]$ as Neumann boundary conditions

↓

Solve the deformation equation

↓

Compute a Riemannian limited-memory BFGS update

↓

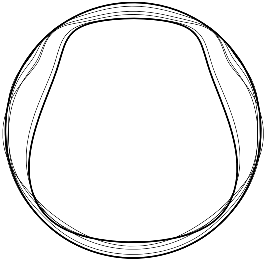
Apply the resulting deformation to the finite element mesh

(↺ back to "Evaluate measurements")

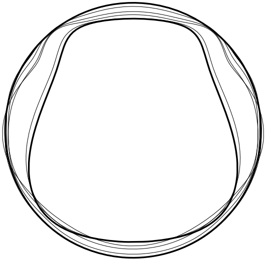
[7] V. Schulz, M. Siebenborn and K. W. Efficient PDE constrained optimization based on Steklov-Poincaré type metrics. *SIOPT*, 26(4):2800-2819, 2016.

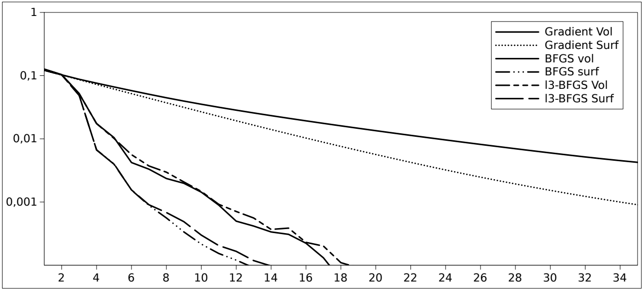
- Diffusion problem with $T = 20$, $k = \begin{cases} k_1 = 1 \text{ in } \Omega_1 \\ k_2 = 0.001 \text{ in } \Omega_2 \end{cases}$
- $A = 0.001$
- Data $\overline{y}$ are generated from a solution of the state equation for the setting $\Omega_2 = \{x \colon \|x\|_2 < 0.5\}$

- Diffusion problem with $T = 20$, $k = \begin{cases} k_1 = 1 \text{ in } \Omega_1 \\ k_2 = 0.001 \text{ in } \Omega_2 \end{cases}$
- $A = 0.001$
- Data $\overline{y}$ are generated from a solution of the state equation for the setting $\Omega_2 = \{x\colon \|x\|_2 < 0.5\}$

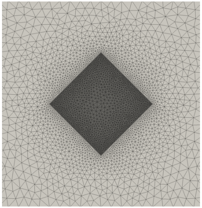


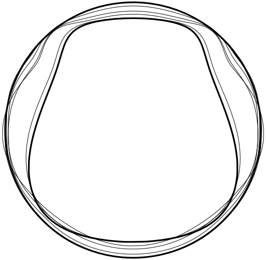
- Diffusion problem with $T = 20$, $k = \begin{cases} k_1 = 1 \text{ in } \Omega_1 \\ k_2 = 0.001 \text{ in } \Omega_2 \end{cases}$
- $A = 0.001$
- Data $\overline{y}$ are generated from a solution of the state equation for the setting $\Omega_2 = \{x \colon \|x\|_2 < 0.5\}$

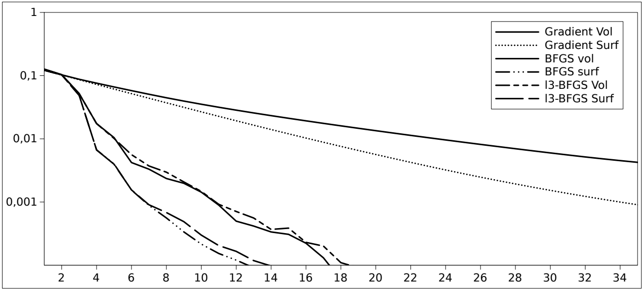




- Diffusion problem with $T = 20$, $k = \begin{cases} k_1 = 1 \text{ in } \Omega_1 \\ k_2 = 0.001 \text{ in } \Omega_2 \end{cases}$
- $A = 0.001$
- Data $\overline{y}$ are generated from a solution of the state equation for the setting $\Omega_2 = \{x \colon \|x\|_2 < 0.5\}$







Contents

Space of $H^{1/2}$-shapes

$$\boxed{\mathcal{B}^{1/2}(\Gamma_0, \mathbb{R}^d) := \mathcal{H}^{1/2}(\Gamma_0, \mathbb{R}^d) / \sim}$$

- $\Gamma_0 \subset \mathbb{R}^d$ d-dimensional Lipschitz shape
  (Definition: A d-dimensional *Lipschitz shape* is defined as the boundary $\Gamma_0 = \partial \mathcal{X}_0$ of a compact Lipschitz domain $\mathcal{X}_0 \subset \mathbb{R}^d$ with $\mathcal{X}_0 \neq \emptyset$.)
- $\mathcal{H}^{1/2}(\Gamma_0, \mathbb{R}^d)$
  $:= \{w \colon w \in H^{1/2}(\Gamma_0, \mathbb{R}^d)$ injective, continuous; $w(\Gamma_0)$ Lipschitz shape$\}$
- $w_1 \sim w_2 \Leftrightarrow w_1(\Gamma_0) = w_2(\Gamma_0)$, where $w_1, w_2 \in \mathcal{H}^{1/2}(\Gamma_0, \mathbb{R}^d)$

Space of $H^{1/2}$-shapes

$$\mathcal{B}^{1/2}(\Gamma_0, \mathbb{R}^d) := \mathcal{H}^{1/2}(\Gamma_0, \mathbb{R}^d) / \sim$$

- $\Gamma_0 \subset \mathbb{R}^d$ d-dimensional Lipschitz shape
  (Definition: A d-dimensional *Lipschitz shape* is defined as the boundary $\Gamma_0 = \partial \mathcal{X}_0$ of a compact Lipschitz domain $\mathcal{X}_0 \subset \mathbb{R}^d$ with $\mathcal{X}_0 \neq \emptyset$.)
- $\mathcal{H}^{1/2}(\Gamma_0, \mathbb{R}^d)$
  $:= \{w \colon w \in H^{1/2}(\Gamma_0, \mathbb{R}^d)$ injective, continuous; $w(\Gamma_0)$ Lipschitz shape$\}$
- $w_1 \sim w_2 \Leftrightarrow w_1(\Gamma_0) = w_2(\Gamma_0)$, where $w_1, w_2 \in \mathcal{H}^{1/2}(\Gamma_0, \mathbb{R}^d)$

Challenges

- Properties of $w \in H^{1/2}(\Gamma_0, \mathbb{R}^d)$ such that $w(\Gamma_0)$ Lipschitz shape
- Independence of the definition of $\mathcal{B}^{1/2}\left(\Gamma_0, \mathbb{R}^d\right)$ from the Lipschitz shape Γ_0
- Structure of $\mathcal{B}^{1/2}\left(\Gamma_0, \mathbb{R}^d\right)$

Space of $H^{1/2}$-shapes

$$\boxed{\mathcal{B}^{1/2}(\Gamma_0,\mathbb{R}^d) := \mathcal{H}^{1/2}(\Gamma_0,\mathbb{R}^d)/\sim}$$

- $\Gamma_0 \subset \mathbb{R}^d$ d-dimensional Lipschitz shape
  (Definition: A d-dimensional *Lipschitz shape* is defined as the boundary $\Gamma_0 = \partial\mathcal{X}_0$ of a compact Lipschitz domain $\mathcal{X}_0 \subset \mathbb{R}^d$ with $\mathcal{X}_0 \neq \emptyset$.)
- $\mathcal{H}^{1/2}(\Gamma_0,\mathbb{R}^d)$
  $:= \{w\colon w \in H^{1/2}(\Gamma_0,\mathbb{R}^d)$ injective, continuous; $w(\Gamma_0)$ Lipschitz shape$\}$
- $w_1 \sim w_2 \Leftrightarrow w_1(\Gamma_0) = w_2(\Gamma_0)$, where $w_1, w_2 \in \mathcal{H}^{1/2}(\Gamma_0,\mathbb{R}^d)$

Challenges

- Properties of $w \in H^{1/2}(\Gamma_0,\mathbb{R}^d)$ such that $w(\Gamma_0)$ Lipschitz shape
- Independence of the definition of $\mathcal{B}^{1/2}\left(\Gamma_0,\mathbb{R}^d\right)$ from the Lipschitz shape Γ_0
- Structure of $\mathcal{B}^{1/2}\left(\Gamma_0,\mathbb{R}^d\right)$: Diffeological structure

Diffeological structure of $\mathcal{B}^{1/2}$ (cf. [8])

$$\mathcal{B}^{1/2}(\Gamma_0, \mathbb{R}^d) := \mathcal{H}^{1/2}(\Gamma_0, \mathbb{R}^d) / \sim$$

- $\mathcal{H}^{1/2}(\Gamma_0, \mathbb{R}^d) = \{w : w \in H^{1/2}(\Gamma_0, \mathbb{R}^d)$ injective, continuous; $w(\Gamma_0)$ Lipschitz shape$\}$
- $w_1 \sim w_2 \Leftrightarrow w_1(\Gamma_0) = w_2(\Gamma_0)$, where $w_1, w_2 \in \mathcal{H}^{1/2}(\Gamma_0, \mathbb{R}^d)$

- $\mathcal{H}^{1/2}(\Gamma_0, \mathbb{R}^d)$ is obviously a subset of $H^{1/2}(\Gamma_0, \mathbb{R}^d)$
- $H^{1/2}(\Gamma_0, \mathbb{R}^d)$ is a Banach space and, thus, a manifold
  $\Rightarrow$ We can view $H^{1/2}(\Gamma_0, \mathbb{R}^d)$ with the corresponding diffeology

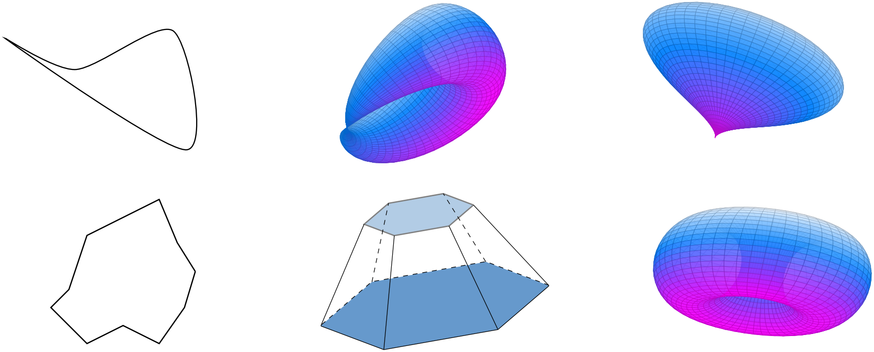
Results in [9]:

- Every subset of a diffeological space carries a natural *subset diffeology*, which is defined by the *pullback* of the ambient diffeology by the *natural inclusion*
- Every quotient of a diffeological space carries a natural *quotient diffeology* defined by the *pushforward* of the diffeology of the source space to the quotient by the *canonical projection*

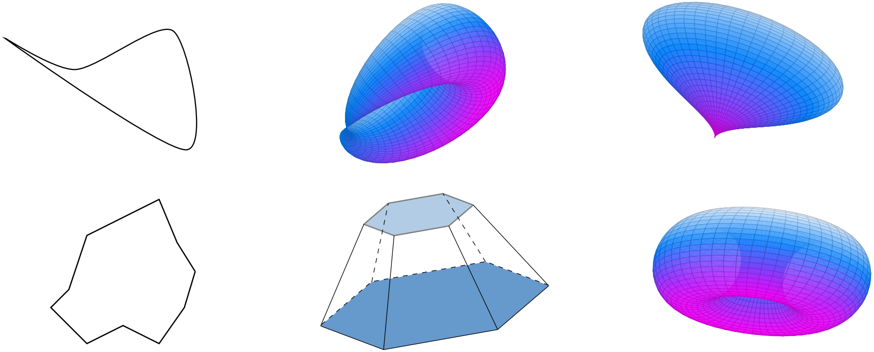
$\Rightarrow$ We can construct diffeologies on $\mathcal{H}^{1/2}(\Gamma_0, \mathbb{R}^d)$ and $\mathcal{B}^{1/2}(\Gamma_0, \mathbb{R}^d)$ □

[8] K. W. Suitable spaces for shape optimization, 2017. (arXiv:1702.07579)
[9] P. Iglesias-Zemmour. *Diffeology*. Volume 185, American Mathematical Society, 2013.

Diffeological space



Diffeological space



- No theory for shape optimization on diffeological spaces so far
- Diffeological structure suffices for many differential-geometric tools used in shape optimization techniques
- Riemannian structures can be used to measure shape distances and state convergence properties

Contents

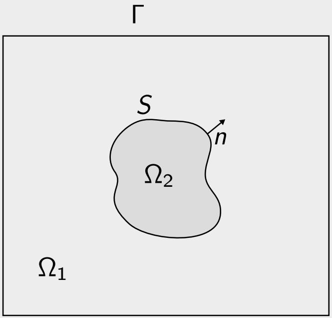
VI constrained shape optimization

- No explicit dependence on the domain in classical VIs
- In VI constrained shape optimization problems: Unavoidable source of non-linearity and non-convexity due to the non-linear and non-convex nature of shape spaces

- No explicit dependence on the domain in classical VIs
- In VI constrained shape optimization problems: Unavoidable source of non-linearity and non-convexity due to the non-linear and non-convex nature of shape spaces

Setting

- $\Omega \subset \mathbb{R}^2$ bounded Lipschitz domain
- Two subdomains $\Omega_1, \Omega_2 \subset \Omega$
- Fixed outer boundary $\partial\Omega = \Gamma_b \,\dot\cup\, \Gamma_l \,\dot\cup\, \Gamma_r \,\dot\cup\, \Gamma_t$
- Variable boundary $S \in B_e(S^1, \mathbb{R}^2)$ or $S \in \mathcal{B}^{1/2}(\Gamma_0, \mathbb{R}^2)$
- Outer normal vector n to Ω_2



VI constrained problem

$$\min_S \; J(y,S) = j(y,S) + j^{\text{reg}}(S) := \frac{1}{2}\int_{\Omega(S)} (y-\bar{y})^2 dx + \mu \int_S 1\, ds$$

$$\begin{aligned}
\text{s.t.} \quad -\Delta y + \lambda &= f && \text{in } \Omega \\
y &\leq \psi && \text{in } \Omega \\
\lambda &\geq 0 && \text{in } \Omega \\
\lambda(y-\psi) &= 0 && \text{in } \Omega \\
y &= 0 && \text{on } \Gamma
\end{aligned}$$

- Obstacle: $\psi \in H^4(\Omega)$ with $0 < \psi \leq M$ for $M > 0$
- Jumping coefficient: $f = \begin{cases} f_{\text{int}} = \text{const.} & \text{in } \Omega_{\text{int}} \\ f_{\text{out}} = \text{const.} & \text{in } \Omega_{\text{out}} \end{cases}$
- Transmission conditions: $[\![y]\!] = 0, \quad \left[\!\!\left[\frac{\partial y}{\partial n}\right]\!\!\right] = 0 \quad \text{on } S$

$$\min_S \; J(y,S) = j(y,S) + j^{reg}(S) := \frac{1}{2}\int_{\Omega(S)} (y-\bar{y})^2 dx + \mu \int_S 1\, ds$$

$$\text{s.t.} \quad \left.\begin{aligned} -\Delta y + \lambda &= f \quad \text{in } \Omega \\ y &\le \psi \quad \text{in } \Omega \\ \lambda &\ge 0 \quad \text{in } \Omega \\ \lambda(y-\psi) &= 0 \quad \text{in } \Omega \end{aligned}\right\}$$

$$y = 0 \quad \text{on } \Gamma$$

Regularized version [10]: $\boxed{-\Delta y + \lambda_c = f \quad \text{in } \Omega}$

- $\lambda_c = \max\{0, \overline{\lambda} + c(y-\psi)\}^2$
- $c > 0,\ 0 \ge \overline{\lambda} \in L^4(\Omega)$

- Obstacle: $\psi \in H^4(\Omega)$ with $0 < \psi \le M$ for $M > 0$
- Jumping coefficient: $f = \begin{cases} f_{\text{int}} = \text{const.} & \text{in } \Omega_{\text{int}} \\ f_{\text{out}} = \text{const.} & \text{in } \Omega_{\text{out}} \end{cases}$
- Transmission conditions: $[\![y]\!] = 0, \quad \left[\!\!\left[\frac{\partial y}{\partial n}\right]\!\!\right] = 0 \quad \text{on } S$

[10] M. Hintermüller and A. Laurain. Optimal shape design subject to elliptic variational inequalities. *SICON*, 49(3):1015-1047, 2011.

Regularized state equation

Idea: Adapt the primal-dual active set (PDAS) algorithm in [11] to our problem

Modified PDAS (mPDAS) algorithm

1. Choose y_0, $k = 0$ and $\lambda_0 = 0$
2. $\mathcal{A}_{k+1} := \{x \colon [\lambda_k + c(y - \psi)](x) > 0\}$ and $\mathcal{I}_{k+1} := \Omega \backslash \mathcal{A}_{k+1}$
3. Compute $y_{k+1} \in H_0^1(\Omega)$ as solution of

$$a(y_{k+1}, v) + \left([\lambda_k + c(y_{k+1} - \psi)]^2, \mathcal{X}_{\mathcal{A}_{k+1}} v \right) = (f, v) \qquad \forall v \in H_0^1(\Omega) \qquad (*)$$

4. $\lambda_{k+1} := \begin{cases} 0 & \text{if } x \in \mathcal{I}_{k+1} \\ \lambda_k + c(y_{k+1} - \psi) & \text{if } x \in \mathcal{A}_{k+1} \end{cases}$
5. Stop or $k := k + 1$ and go to 2.

[11] K. Ito and K. Kunisch. Semi–smooth Newton methods for variational inequalities of the first kind. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(1):41-62, 2003.

Regularized state equation

Idea: Adapt the primal-dual active set (PDAS) algorithm in [11] to our problem

Modified PDAS (mPDAS) algorithm

1. Choose y_0, $k = 0$ and $\lambda_0 = 0$
2. $\mathcal{A}_{k+1} := \{x \colon [\lambda_k + c(y - \psi)](x) > 0\}$ and $\mathcal{I}_{k+1} := \Omega \backslash \mathcal{A}_{k+1}$
3. Compute $y_{k+1} \in H_0^1(\Omega)$ as solution of

$$a(y_{k+1}, v) + \left([\lambda_k + c(y_{k+1} - \psi)]^2, \mathcal{X}_{\mathcal{A}_{k+1}} v \right) = (f, v) \qquad \forall v \in H_0^1(\Omega) \qquad (*)$$

4. $\lambda_{k+1} := \begin{cases} 0 & \text{if } x \in \mathcal{I}_{k+1} \\ \lambda_k + c(y_{k+1} - \psi) & \text{if } x \in \mathcal{A}_{k+1} \end{cases}$
5. Stop or $k := k + 1$ and go to 2.

[11] K. Ito and K. Kunisch. Semi–smooth Newton methods for variational inequalities of the first kind. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(1):41-62, 2003.

Linear mPDAS algorithm (cf. [12])

Step 3 in mPDAS algorithm

Compute $y_{k+1} \in H_0^1(\Omega)$ as solution of

$$a(y_{k+1}, v) + \left([\lambda_k + c(y_{k+1} - \psi)]^2, \mathcal{X}_{\mathcal{A}_{\mathbf{k+1}}} v\right) = (f, v) \qquad \forall v \in H_0^1(\Omega) \qquad (*)$$

Problem: $(*)$ is not linear

$\rightsquigarrow$ **Idea:** Compute $\Delta y := y_{k+1} - y_k$ instead of y_{k+1} and use the linearization

$$(\lambda_k + c(y_k + \Delta y - \psi))^2 \doteq (\lambda_k + c(y_k - \psi))^2 + 2c\Delta y(\lambda_k + c(y_k - \psi))$$

Step 3 in linear mPDAS algorithm

a) Compute Δy as solution of

$$a(\Delta y, v) + \left(2c\Delta y\,[\lambda_k + c(y_k - \psi)], \mathcal{X}_{\mathcal{A}_{\mathbf{k+1}}} v\right)$$
$$= (f, v) - a(y_k, v) - \left([\lambda_k + c(y_k - \psi)]^2, \mathcal{X}_{\mathcal{A}_{\mathbf{k+1}}} v\right) \qquad \forall v \in H_0^1(\Omega)$$

b) $y_{k+1} := y_k + \Delta y$

[12] B. Führ, V. Schulz and K. W. Shape optimization for interface identification with obstacle problems. Appears in: *Vietnam Journal of Mathematics*, 2018.

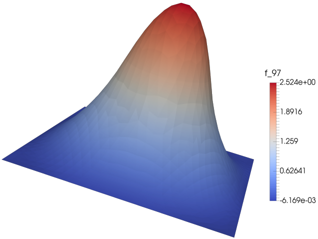
Solution of the regularized state equation

$$\boxed{-\Delta y + \lambda_c = f \quad \text{in } \Omega}$$

- $\lambda_c = \max\{0, \overline{\lambda} + c(y - \psi)\}^2$
- $0 \geq \overline{\lambda} \in L^4(\Omega),\ c = 5,\ f = \begin{cases} f_1 = -10 & \text{in } \Omega_1 \\ f_2 = 100 & \text{in } \Omega_2 \end{cases}$

$$\boxed{-\Delta y + \lambda_c = f \quad \text{in } \Omega}$$

- $\lambda_c = \max\{0, \overline{\lambda} + c(y - \psi)\}^2$
- $0 \geq \overline{\lambda} \in L^4(\Omega),\ c = 5,\ f = \begin{cases} f_1 = -10 & \text{in } \Omega_1 \\ f_2 = 100 & \text{in } \Omega_2 \end{cases}$

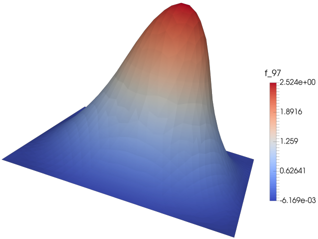


y with $\lambda_c = 0$

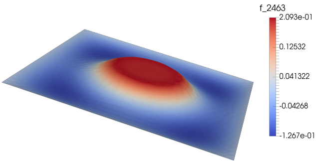
Solution of the regularized state equation

$$\boxed{-\Delta y + \lambda_c = f \quad \text{in } \Omega}$$

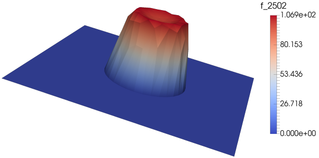
- $\lambda_c = \max\{0, \overline{\lambda} + c(y - \psi)\}^2$
- $0 \geq \overline{\lambda} \in L^4(\Omega),\ c = 5,\ f = \begin{cases} f_1 = -10 & \text{in } \Omega_1 \\ f_2 = 100 & \text{in } \Omega_2 \end{cases}$



y with $\lambda_c = 0$



y with $\psi = 0.2$



Corresponding λ_c

Algorithm (based on g^S)

Evaluate objective

↓

Solve the state equation (with the linear mPDAS algorithm)

↓

Solve the adjoint equation

↓

Assemble the deformation equation:

- E.g., choose the weak form of the linear elasticity equation as left hand-side of the deformation equation
- Assemble $DJ_{\text{vol}}(y, X)[V]$ for all test functions V whose support includes S as a source term
- Assemble $DJ_{\text{surf}}(y, X)[V]$ as Neumann boundary conditions

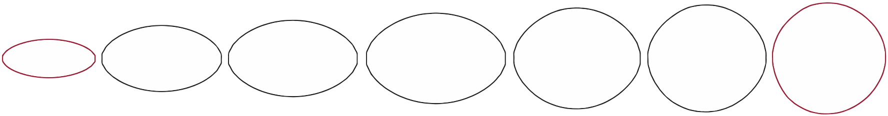
↓

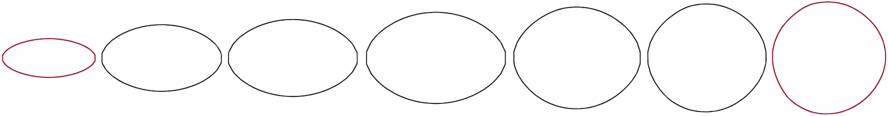
Solve the deformation equation

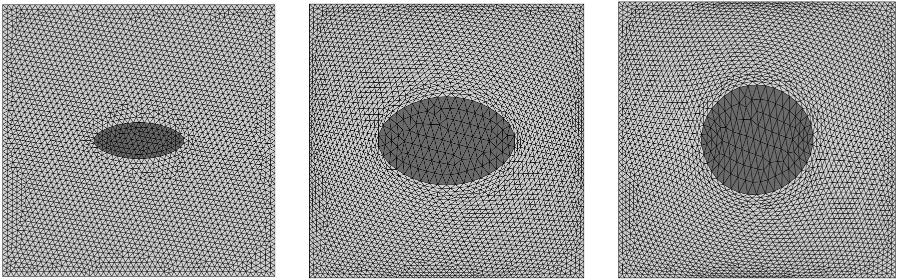
↓

Apply the resulting deformation to the finite element mesh

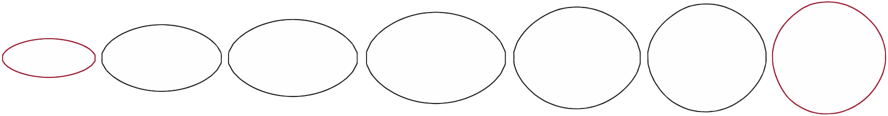
(loop back to: Evaluate objective)

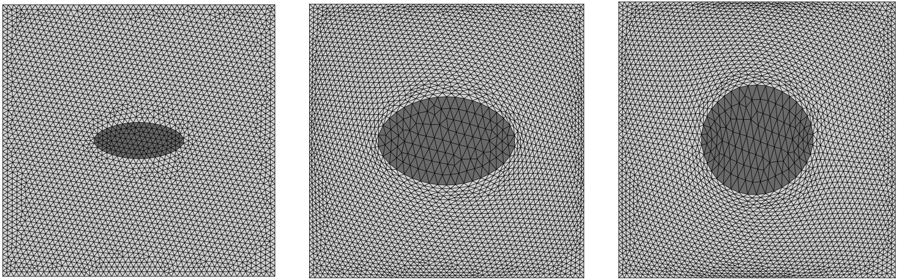




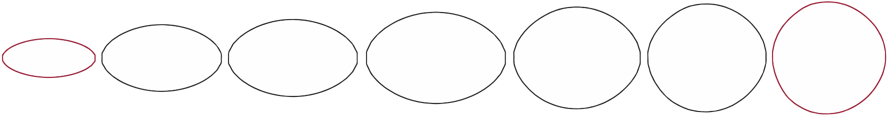


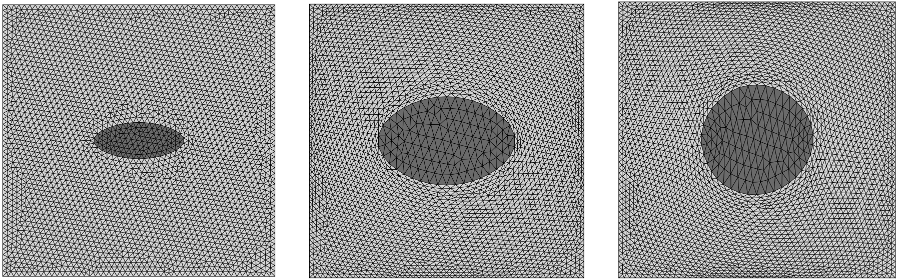
- Computations on unstructured grids with about 1 500 up to 6 000 triangles





- Computations on unstructured grids with about 1 500 up to 6 000 triangles
- In the interior domain, elements are magnified by the mesh deformations
  $\rightsquigarrow$ Choose locally adapted meshes or re-mesh after a few iterations
- Largest deformations at the beginning of the iteration process





- Computations on unstructured grids with about 1 500 up to 6 000 triangles
- In the interior domain, elements are magnified by the mesh deformations
  $\rightsquigarrow$ Choose locally adapted meshes or re-mesh after a few iterations
- Largest deformations at the beginning of the iteration process
- Algorithmic performance deteriorates if the obstacle problem is strongly binding
- More iterations for tighter obstacles, i.e., small values of ψ (494 iterations for $\psi = 0.5$ vs. 22 iterations for $\psi = 10$)

Contents

Conclusion

- Optimization in the space of smooth shapes
- PDE constrained shape optimization problem
- Algorithm based on surface shape derivative formulations
  $\rightsquigarrow$ Application to a parabolic diffusion problem
- Steklov-Poincaré metric
  $\rightsquigarrow$ Algorithm based on volume shape derivative formulations
  $\rightsquigarrow$ Application to a parabolic diffusion problem
- Diffeological shape space
- VI constrained shape optimization problem
- Linear modified primal-dual active set (lmPDAS) algorithm
  $\rightsquigarrow$ Application to a VI constrained shape optimization problem

Conclusion

- Optimization in the space of smooth shapes
- PDE constrained shape optimization problem
- Algorithm based on surface shape derivative formulations
  $\rightsquigarrow$ Application to a parabolic diffusion problem
- Steklov-Poincaré metric
  $\rightsquigarrow$ Algorithm based on volume shape derivative formulations
  $\rightsquigarrow$ Application to a parabolic diffusion problem
- Diffeological shape space
- VI constrained shape optimization problem
- Linear modified primal-dual active set (lmPDAS) algorithm
  $\rightsquigarrow$ Application to a VI constrained shape optimization problem

Thank you for your attention!