New Results on Optimization in Shape Manifolds

Volker Schulz
TEAM:
K. Welker, M. Siebenborn

www.alop.uni-trier.de
Why should we care about shapes?

• Statistics on shapes is addressed, e.g., in computer vision or medicine (→ computational anatomy, Michael Miller@John Hopkins U., Baltimore)
  – Classification of diseased vs. healthy tissue

• Inference about shape from data $z$

\[
\min_{\Omega \in \mathcal{S}} \left( L(\Omega, z) + pr(\Omega) \right)
\]

→ Challenging problem structure: shapes do not live in a vector space!
Parabolic interface problem

- Motivation: structured inverse modeling of human skin cells
- combined estimation of coefficient and cell shapes
- discrete measurements in time and space
- millions of unknowns, especially on surfaces
- highly parallel solver based on UG4 (joined work with group of G. Wittum, GCSC Frankfurt)
\[
\min_{(y, k_1, k_2, \Omega)} \quad J(y, \Omega) = \frac{1}{2} \int_0^T \int_\Omega (y - z)^2 \, d\Omega \, dt \quad + \quad \mu \int_{\Gamma_{\text{int}}} 1 \, dS
\]

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

\begin{align*}
[y] &= 0 \quad \text{on} \quad \Gamma_{\text{int}} \times (0, T] \\
\begin{bmatrix} k \frac{\partial y}{\partial \vec{n}} \end{bmatrix} &= 0 \quad \text{on} \quad \Gamma_{\text{int}} \times (0, T] \\
y &= 1 \quad \text{on} \quad \Gamma_{\text{top}} \times (0, T] \\
\frac{\partial y}{\partial \vec{n}} &= 0 \quad \text{on} \quad \Gamma_{\text{bottom}} \cup \Gamma_{\text{left}} \cup \Gamma_{\text{right}} \times (0, T] \\
y &= y_0 \quad \text{in} \quad \Omega \times \{0\}
\end{align*}

\[
k = \begin{cases} 
  k_1 & \text{in} \quad \Omega_1 \\
  k_2 & \text{in} \quad \Omega_2 
\end{cases}
\]
Parametric versus nonparametric

- Finite shape parametrization in many industrial shape optimizations
  - Pro: vector space setting, fits in CAD framework
  - Con: complexity inevitably increases with number of parameters, mesh sensitivities can become expensive, set of reachable shapes is restricted

- Non-CAD approach built on shape calculus
  - Pro: avoids cons of parametric approach, can be very efficient
  - Con: no longer vector space setting, theoretically more challenging
Shape derivative in a nutshell

$$df(\Omega)[V] = \lim_{t \to 0} \frac{f(T_{t,V}(\Omega)) - f(\Omega)}{t}$$

$$df(\Omega)[V] = (\nabla f, < V, n >)_{\partial \Omega}$$ (Hadamard)

$$T_{t,V}(\Omega), \text{ where } T_{t,V}(x) = x + t \cdot V(x)$$
Example: simple objective, no PDE constraint

\[ f(\Omega_t) = \int_{\Omega_t} g(x) \, dx \quad \Omega_t = T_{t,V}(\Omega_0) \]

\[ T_{t,V}(x) = x + t \cdot V(x) \]

directional derivative in direction \( V \):

\[ df(\Omega_t)[V] := \left. \frac{d}{dt} \right|_{t=0} f(\Omega_t) = \left. \frac{d}{dt} \right|_{t=0} \int_{\Omega_t} g(x) \, dx \]

\[ = \left. \frac{d}{dt} \right|_{t=0} \int_{\Omega_0} g(T_{t,V}(x)) \mid \text{det}(DT_{t,V}(x)) \mid \, dx \]
\[
\begin{align*}
\Omega_0 & \quad \mathcal{W}_o \\
= & \quad \int_{\Omega_0} \left. \frac{d}{dt} \right|_{t=0} g(T_t, V(x)) | det(DT_{t,V}(x)) | dx \\
= & \quad \int_{\Omega_0} \nabla g(x)^\top V(x) + g(x) \cdot \text{tr}(DV(x)) dx \\
= & \quad \int_{\Omega_0} \text{div}(g(x)V(x)) dx \\
= & \quad \int_{\partial \Omega_0} V(x)^\top \tilde{n}(x) \cdot g(x) dx \quad \text{(Gauss)}
\end{align*}
\]
Important shape derivatives

\[
d\left( \int_{\Omega} g(x) \, dx \right) [W] = \int_{\partial \Omega} g(s)(W, n) \, ds
\]

\[
d\left( \int_{\partial \Omega} g(s) \, ds \right) [W] = \int_{\partial \Omega} \left( \frac{\partial g}{\partial \vec{n}}(s) + \kappa(s) g(s) \right) (W, \vec{n}) \, ds
\]

(cf. Delfour/Zolesio, 2001)

\[
d\left( \int_{\partial \Omega} h(s) \top n(s) \, ds \right) [W] = \int_{\partial \Omega} \text{div}(h(s))(W, n) \, ds
\]
Some observations on the usage of shape calculus

• Many publications on derivation of shape gradients and optimality conditions
• In numerical computations, mostly steepest descent is used with \( L^2 \) or \( H^1 \) representation

• We aim at transferring NLP to the shape calculus framework
Shape derivative due to Hadamard

\[
d f(\Omega)[V] = \lim_{t \to 0^+} \frac{f(T_{t,V}(\Omega)) - f(\Omega)}{t} = (\varphi, \langle V, n \rangle)_{\partial \Omega}
\]

- Example

\[
d(\int g(x) \, dx)[V] = \int_{\Omega} g(s) \langle V(s), n(s) \rangle \, ds
\]

- Primal representation via scalar product:

\[
L^2(\partial \Omega), -\Delta_{\partial \Omega}, \ldots ?
\]

- Hessian?
A closer look at the so-called shape Hessian

- **Symmetry**
  
  \[ d(df(\Omega)[W])[V] = \int_{\partial \Omega} \left( \frac{\partial g}{\partial \tilde{n}} + \kappa_c g \right) \langle W, \tilde{n} \rangle \langle V, \tilde{n} \rangle + g \langle DW V, \tilde{n} \rangle \, ds \]

- **Taylor series**

  \[ f(p) = f(p_0) + df(p_0)(p - p_0) + \frac{1}{2} d^2 f(p_0)(p - p_0, p - p_0) + O(\|p - p_0\|^3) \]

  ⇒ Sufficient conditions
  
  ⇒ Quadratic convergence of Newton method
• Distance concepts

„Morphing“: Riemannian length of shortest connecting path
The Riemannian metric of Michor and Mumford (2006)

Shape set

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

is a manifold with tangent space

\[ T_c B_e \cong \{ h \mid h = \alpha \vec{n}, \alpha \in C^\infty(S^1, \mathbb{R}) \} \]

Scalar product

\[ S^A(h, k) = \int_c \alpha \beta + A \alpha' \beta' ds \]

\[ G^A(h, k) = \int_c (1 + A \kappa_c^2) \alpha \beta ds, \quad A > 0 \]

defines a Riemannian manifold
A Riemannian view on shape optimization
(FoCM, 2014)

• Defining the action of a vector field as the shape derivative, we can unleash the Riemannian structure on shape optimization

\[ h(f)(c) = df(\Omega)[V], \quad V = h\nabla, \quad c = \partial \Omega \]

• Optimization on manifolds can be performed as in [Absil 2008] for matrix manifolds.
Consequences

- **Symmetric Riemannian shape Hessian**

\[
G^A(\text{Hess} f(\Omega)[V], W) := G^A(\nabla_V \text{grad} f(\Omega), W) = d(df(\Omega)[W])[V] - df(\Omega)[\nabla_V W]
\]

- **Taylor series expansion**

\[
f(\exp_\Omega(h)) = f(\Omega) + G^A(\text{grad} f(\Omega), h) + \frac{1}{2} G^A(\text{Hess} f(\Omega) h, h) + O(\|h\|^3)
\]
Newton convergence

• Riemannian variants of quadratic convergence results are possible.

• In particular, if we eliminate expressions from the Hessian, which are zero at the solution, still quadratic convergence is achieved.
Optimization algorithms

\[ \Omega^{k+1} = \exp_{\Omega^k} (\alpha d^k) \]

\[ d^k = -\nabla f(\Omega^k) \text{ or } d^k = -\text{Hess}(\Omega^k)^{-1} \nabla f(\Omega^k) \]

Retraction
\[ \partial \Omega^{k+1} = \partial \Omega^k + \alpha d^k \]

Steepest descent  Newton method

linear conv.  quadratic conv.
A simple example

\[
\min_{\Omega} f(\Omega) := \int_{\Omega} g(x) dx
\]

with \( g(x) = x^\top x - 1 \)

\[
G^A(\text{Hess} f(\Omega)[V], W) = d(df(\Omega)[W])[V] - df(\Omega)[\nabla_V W]
\]

\[
= \int_{\partial \Omega} \left( \frac{\partial g}{\partial \bar{n}} + \frac{\kappa_c}{2} g - \frac{gA\kappa_c^3}{1 + A\kappa_c^2} \right) \langle V, \bar{n} \rangle \langle W, \bar{n} \rangle \, ds
\]

\[- \int_{\partial \Omega} gA\kappa_c \left( \langle V, \bar{n} \rangle \langle W, \bar{n} \rangle \right)_{tt} \, ds\]
Performance of optimization algorithms

Steepest descent with exact linesearch

<table>
<thead>
<tr>
<th>Iteration</th>
<th>objective</th>
<th>$d^A(\Omega^k, \hat{\Omega})$</th>
<th>line search</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-2.66624</td>
<td>3.4135</td>
<td>0.69</td>
</tr>
<tr>
<td>1</td>
<td>-4.18749</td>
<td>7.2458E-02</td>
<td>0.50</td>
</tr>
<tr>
<td>2</td>
<td>-4.18879</td>
<td>6.4371E-04</td>
<td>0.50</td>
</tr>
<tr>
<td>3</td>
<td>-4.18879</td>
<td>7.0061E-08</td>
<td></td>
</tr>
</tbody>
</table>

Equivalent to Newton method

-> example too simple
\[ g(x) = x_1^2 + 4x_2^2 \]

<table>
<thead>
<tr>
<th>Iteration</th>
<th>SD-objective</th>
<th>( d_{SC}(\Omega^k, \hat{\Omega}) )</th>
<th>SD-line search</th>
<th>NM-objective</th>
<th>( d_{NM}(\Omega^k, \hat{\Omega}) )</th>
<th>NM-line search</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-0.5571</td>
<td>.9222E+00</td>
<td>.5100E+00</td>
<td>-0.5571</td>
<td>.9222E+00</td>
<td>0.6400</td>
</tr>
<tr>
<td>1</td>
<td>-0.7630</td>
<td>.2142E+00</td>
<td>.3100E+00</td>
<td>-0.7775</td>
<td>.1359E+00</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>-0.7830</td>
<td>.5882E-01</td>
<td>.3700E+00</td>
<td>-0.7854</td>
<td>.4265E-02</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>-0.7852</td>
<td>.1662E-01</td>
<td>.3200E+00</td>
<td>-0.7854</td>
<td>.1037E-04</td>
<td>1.0000</td>
</tr>
<tr>
<td>4</td>
<td>-0.7854</td>
<td>.4459E-02</td>
<td>.3500E+00</td>
<td>-0.7854</td>
<td>.2590E-09</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-0.7854</td>
<td>.1356E-02</td>
<td>.3300E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-0.7854</td>
<td>.3951E-03</td>
<td>.3400E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-0.7854</td>
<td>.1218E-03</td>
<td>.3300E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-0.7854</td>
<td>.3643E-04</td>
<td>.3500E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-0.7854</td>
<td>.1194E-04</td>
<td>.3200E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-0.7854</td>
<td>.3421E-05</td>
<td>.3500E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>-0.7854</td>
<td>.1137E-05</td>
<td>.3200E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>-0.7854</td>
<td>.3305E-06</td>
<td>.3500E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>-0.7854</td>
<td>.1109E-06</td>
<td>.3000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>-0.7854</td>
<td>.2894E-07</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Towards a Lagrange-Newton approach in shape calculus

[S. /Siebenborn/Welker, ISNM 2015]

model problem:

$$\min_u J(y, u) \equiv \frac{1}{2} \int_{\Omega(u)} (y(x) - \bar{y}(x))^2 dx + \mu \int_u 1ds$$

s.t. $$- \Delta y(x) = f(x), \ \forall x \in \Omega(u)$$
$$y(x) = 0, \ \forall x \in \partial \Omega(u)$$

$$f(x) \equiv \begin{cases} f_1(x) = \text{const.}, \ \forall x \in \Omega_1(u) \\ f_2(x) = \text{const.}, \ \forall x \in \Omega_2(u) \end{cases}$$
Weak formulation

\[ \min_u \frac{1}{2} \int_{\Omega(u)} (y(x) - \bar{y}(x))^2 dx + \mu \int_u 1 ds \]

s.t. \( a_u(y, p) = b_u(p), \forall p \in H^1_0(\Omega(u)) \)

where \( a_u(y, p) := \int_{\Omega(u)} \nabla y(x)^\top \nabla p(x) dx \)

\( b_u(p) := \int_{\Omega(u)} f(x) p(x) dx \)

Gives rise to the Lagrangian

\[ \mathcal{L}(y, u, p) := J(y, u) + a_u(y, p) - b_u(p) \]
The adjoint problem

\[ \int_{\Omega} \nabla z^T \nabla p \, dx = - \int_{\Omega} (y - \bar{y})z \, dx, \quad \forall \bar{y} \in H^1_0(\Omega(u)) \]

allows for a comfortable computation of the shape derivative

\[ dJ(y(u), u)[V] = d\mathcal{L}(y, u, p)[V] \]

Resulting KKT-conditions

\[
\frac{\partial}{\partial y} \mathcal{L}(y, u, p) = 0, \forall y \quad \text{(adjoint)}
\]

\[
d\mathcal{L}(y, u, p)[V] = 0, \forall V \quad \text{(design)}
\]

\[
\frac{\partial}{\partial p} \mathcal{L}(y, u, p) = 0, \forall p \quad \text{(state)}
\]

What about applying Newton's method to these equations?
- or even multigrid?

But the function spaces for \( y \) and \( p \) vary with the shape!
Remedy: yet another structure from differential geometry: vector bundles

\[ u := \Gamma \in \mathcal{N} := B^0_e([0, 1], \mathbb{R}^2) := \text{Emb}^0([0, 1], \mathbb{R}^2) / \text{Diff}([0, 1]) \]
Vector bundle $E := \{(H(u), u) \mid u \in \mathcal{N}\}$

local diffeomorphisms:

$$\tau_i : \pi^{-1}(U_i) \to H_0 \times U_i$$

$$\tau_i(u) : \pi^{-1}(u) = H(u) \to H_0$$

domain deformation
Variational Newton step $h$

$$G(\text{Hess}\mathcal{L}(y,u,p)h,k) = -G(\nabla\mathcal{L}(y,u,p),k), \forall k \in TE$$

After omitting terms in the Hessian, which are zero at the solution, this can be identified with the linear-quadratic optimal control problem:

$$\min_{(z,w)} \int_{\Omega(u)} \frac{z^2}{2} + (y - \bar{y})z \, dx + \mu \int_{U} \kappa_1 w \, ds + \frac{1}{2} \int_{U} (f_2 - f_1) \frac{\partial p}{\partial n_1} w^2 + \mu \left( \frac{\partial w}{\partial \tau} \right)^2 \, ds$$

$$\text{s.t.} \quad \int_{\Omega(u)} \nabla z^\top \nabla \bar{q} \, dx + \int_{U} (f_2 - f_1) \bar{q} w \, ds = -\int_{\Omega_0} \nabla y^\top \nabla \bar{q} \, dx + \int_{\Omega(u)} f \bar{q} \, dx,$$

$$\forall \bar{q} \in H^1_0(\Omega(u))$$

-> can be efficiently solved by multigrid methods (e.g. Borzi/S., or Ascher, or Zulehner, or Nash)
Strong form of QP

\[
\min_{(z,w)} \int_{\Omega(u)} \frac{z^2}{2} + (y - \bar{y})zdx + \mu \int_u \kappa_1 w ds + \frac{1}{2} \int_u (f_2 - f_1) \frac{\partial p}{\partial n_1} w^2 + \mu \left( \frac{\partial w}{\partial \tau} \right)^2 ds
\]

s.t. \[-\Delta z = \Delta y + f_1 \quad \text{in} \quad \Omega_1(u)\]
\[-\Delta z = \Delta y + f_2 \quad \text{in} \quad \Omega_2(u)\]
\[
\frac{\partial z}{\partial n_1} = f_1 w \quad \text{on} \quad u
\]
\[
\frac{\partial z}{\partial n_2} = f_2 w \quad \text{on} \quad u
\]
\[
z = 0 \quad \text{on} \quad \partial \Omega(u)
\]
Numerical results

[0]

[1]

[2]
Quadratic convergence?

• Distance to solution on several grids

<table>
<thead>
<tr>
<th>It.-No.</th>
<th>$\Omega^1_h$</th>
<th>$\Omega^2_h$</th>
<th>$\Omega^3_h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0705945</td>
<td>0.070637</td>
<td>0.0706476</td>
</tr>
<tr>
<td>1</td>
<td>0.0043115</td>
<td>0.004104</td>
<td>0.0040465</td>
</tr>
<tr>
<td>2</td>
<td>0.0003941</td>
<td>0.000104</td>
<td>0.0000645</td>
</tr>
</tbody>
</table>

• Major advantages over steepest descent:
  – natural step scaling of 1 (vs. 10000)
  – Optimal control technology (multigrid!) can be exploited
Parabolic interface problem

- Motivation: structured inverse modeling of human skin cells
- combined estimation of coefficient and cell shapes
- discrete measurements in time and space
- millions of unknowns, especially on surfaces
\[
\min_{(y, k_1, k_2, \Omega)} J(y, \Omega) = \frac{1}{2} \int_0^T \int_\Omega (y - z)^2 \, d\Omega \, dt + \mu \int_{\Gamma_{\text{int}}} 1 \, dS
\]

s.t. \[
\frac{\partial y}{\partial t} - \nabla \cdot k \nabla y = f \quad \text{in} \quad \Omega \times (0, T]
\]
\[
[y] = 0 \quad \text{on} \quad \Gamma_{\text{int}} \times (0, T]
\]
\[
[k \frac{\partial y}{\partial \vec{n}}] = 0 \quad \text{on} \quad \Gamma_{\text{int}} \times (0, T]
\]
\[
y = 1 \quad \text{on} \quad \Gamma_{\text{top}} \times (0, T]
\]
\[
\frac{\partial y}{\partial \vec{n}} = 0 \quad \text{on} \quad \Gamma_{\text{bottom}} \cup \Gamma_{\text{left}} \cup \Gamma_{\text{right}} \times (0, T]
\]
\[
y = y_0 \quad \text{in} \quad \Omega \times \{0\}
\]
\[
k = \begin{cases} 
k_1 & \text{in} \quad \Omega_1 \\
k_2 & \text{in} \quad \Omega_2
\end{cases}
\]
Shape derivative

- Hadamard form

\[ dJ(y, \Omega)[\mathcal{V}] = - \int_0^T \int_{\Gamma_{int}} [k] \langle \nabla y_2, \nabla p_1 \rangle \langle \mathcal{V}, \vec{n} \rangle \ dS \ dt + \int_{\Gamma_{int}} \mu \kappa \langle \mathcal{V}, \vec{n} \rangle \ dS \]

- Very similar to results by Ito/Kunisch 2008 and Paganini 2014 for the elliptic case
Basic approach

- Use shape derivative in a steepest descent manner.

- In literature, mostly Riesz representations based on $L^2$ or Laplace-Beltrami are used, i.e.

$$d = (I - \sigma \Delta \Gamma)^{-1} g$$

- Shape manifold framework allows even for quasi-Newton algorithms (S. JFCM, 2014)
Adaptive meshes

- shape gradient as boundary condition in linear elasticity mesh deformation
- shape calculus depends on fine grid near variable boundary
Riemannian Quasi-Newton

quasi–Newton method – k. iteration:
1. compute update formula $H_k$ for $\text{Hess} J(c_k)$ and the increment $\Delta c = H_k^{-1} \text{grad} J(c_k)$
2. increment $c_{k+1} := \exp_{c_k}(\alpha_k \Delta c)$ for some steplength $\alpha^k$

- $\text{grad} J$ Riemannian shape gradient
- update formula $H_k$ for the Riemannian shape Hessian $\text{Hess} J(c_k)$ is based on the secant condition
  \[ H_k s_k = w_k \]
where $s_k$ denotes the difference between iterated shapes and $w_k$ the difference between iterated shape gradients
Riemannian limited BFGS

\[
\begin{align*}
\rho_k & \leftarrow \frac{1}{g^1(w_k, s_k)} \\
q & \leftarrow \text{grad} J(c_k) \\
& \text{for } i = k - 1, \ldots, k - m \text{ do} \\
& \quad s_i \leftarrow T s_i \\
& \quad w_i \leftarrow T w_i \\
& \quad \alpha_i \leftarrow \rho_i g^1(s_i, q) \\
& \quad q \leftarrow q - \alpha_i d_i \\
& \text{end for} \\
z \leftarrow \text{grad} J(c_k) \\
q \leftarrow \frac{g^1(w_{k-1}, s_{k-1})}{g^1(w_{k-1}, w_{k-1})} \text{grad} J(c_k) \\
& \text{for } i = k - m, \ldots, k - 1 \text{ do} \\
& \quad \beta_i \leftarrow \rho_i g^1(w_i, z) \\
& \quad q \leftarrow q + (\alpha_i - \beta_i) s_i \\
& \text{end for} \\
& \text{return } q = H_k^{-1} \text{grad} J(c_k)
\end{align*}
\]

- implements quasi-Newton update formula for $H_k^{-1}$
- computes the BFGS update $H_k^{-1} \text{grad} J(c_k)$
- memory contains only $m$ shape gradients
- $s_k$ distance between iterated shapes
- $w_k$ difference of iterated shape gradients
- $T$ vector transport of elements in tangential space to updated shape
L-BFGS

3D Scalability study up to 320M el. on 64k processors @ HLRS Stuttgart (CVS 2015)
(Hazel Hen: #9 on top 500 supercomputer list, June 2016)

Noisy data do not affect method!
Scalability issues

- Hadamard form of gradient lives on the boundary only
- Laplace-Beltrami metric lives on the boundary only

=> load balancing has to take into account explicitly boundary nodes
Domain integral formulation

\[ dJ(y, \Omega)[V] = \int_0^T \int_\Omega -k \nabla y^T \left( \nabla V + \nabla V^T \right) \nabla p - p \nabla f^T V \]
\[ + \text{div}(V) \left( \frac{1}{2} (y - \bar{y})^2 + \dot{y} p + k \nabla y^T \nabla p - f p \right) dx \, dt. \]

- often intermediate step in derivation
- is shown as asymptotically more accurate (Berggren 2010, Paganini, 2014)
- Volumetric operations do not change load balancing in parallelization.
- \( \Rightarrow \) improves scalability of optimization algorithm, if scalar product can also be transferred to the volume.
Harmonizing shape metric

\[ S^p : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma), \quad a(V, W) = \int_{\Gamma} \alpha \cdot (\gamma_0 W)^\top n ds, \quad \forall W \in H^1_0(\Omega, \mathbb{R}^d) \]

\[ \alpha \mapsto (\gamma_0 V)^\top n \]

\[ g^S(\alpha, \beta) := \langle \alpha, (S^p)^{-1} \beta \rangle = \int_{\Gamma} \alpha(s) \cdot [(S^p)^{-1} \beta](s) ds \]

- \( S^p \): projected Poincaré-Steklov
- inherits positivity from full P.-S.
- enables combination of gradient representation and mesh deformation

[S./ Siebenborn/Welker, SIOPT 2016]
Resulting method

• Compute mesh deformation $U$ from elasticity equation

\[ a(U, V) = DJ_\Gamma[V] = DJ_\Omega[V], \quad \forall V \in H^1_0(\Omega, \mathbb{R}^d) \]

• Use $U$ also in I-BFGS

• $U$ is typically used as **boundary deformation** in steepest descent
First algorithmic performance

mesh

r.h.s. in elasticity
Zoom

\[ D J_\Omega [V_h] \]
\[ V_h \in H^1_0(\Omega, \mathbb{R}^d) \]

Remedy:
Force 0 outside boundary!
nice convergence even for coarse grids
even kinky starts are possible
Stokes study:

Large deformation for Stokes shape optimization

S./Siebenborn, CMAM 2016
Algorithmic performance

Approximative geodesic distance to solution
Algorithmic aspects

• Multigrid is used for
  – each timestep of forward problem
  – each timestep of adjoint problem
  – evaluation of shape gradient based on Laplace-Beltrami or Steklov-Poincare metric
  – mesh deformation based on elasticity equation

• Classical forward-backward loop in each optimization iteration

• L-BFGS updates in domain representation
Skin Problem in 2D

Start

Goal
State solution
Iterations

Goal
Coarse 3D-results

1.8 Mio elements, 10 time steps, 20 optimization steps
Conclusions and outlook

• Structured inverse modeling involves challenging shape optimization problem.

• Poincaré-Steklov type metric allows less smoothness assumptions and improves scalability

• Future work is devoted variational inequalities as system models.