Low Dimensional Shape Optimization

Julian Panetta

September 3, 2014

We seek to optimize a small set of shape parameters, $p \in \mathbb{R}^{n_p}$, that define an object $\Omega(p) \subset \mathbb{R}^n$. The objective, J, is a volume integral of some function, j, of the stress tensor under a particular loading scenario (using linear elasticity):

$$J(p) = \int_{\Omega(p)} j(x, \sigma(u)(x))\, \mathrm{d}x \tag{1}$$

$$\text{s.t.} \quad \begin{cases} -\nabla \cdot \sigma(u) = f(p) & \text{in}\, \Omega(p) \\ \sigma(u)\hat{n} = g(p) & \text{on}\, \Gamma_N(p) \\ u = 0 & \text{on}\, \Gamma_D \end{cases} \tag{2}$$

where $\sigma(u) = C : \epsilon(u),\ \epsilon(u) = \frac{1}{2}\left(\nabla u + (\nabla u)^T\right)$ (i.e. $\sigma_{ij} = C_{ijkl}\epsilon_{kl},\ \epsilon_{kl} = \frac{1}{2}\left(u_{k,l} + u_{l,k}\right)$), $\hat{n}$ is the outward-pointing normal, and $\Gamma_N(p) := \partial\Omega(p) \setminus \Gamma_D$. Notice the inclusion of a fixed Dirichlet boundary; we discuss in Section 4 how to optimize without this constraint. We focus on the task of computing this objective function's gradient with respect to the shape parameters, neglecting for now the terms that penalize shapes differing too greatly from the initial design, $\Omega(p_0)$.

Using a finite difference-based differentiation scheme is difficult due to the challenge of choosing a reasonable step size. Not only is a good step size, e.g. for a rectangle's rotation parameter, not immediately clear, but also as we shrink the step size to get a more accurate approximation, the noise/errors in the PDE constraint solution will dominate. Instead, we use the adjoint method [2] to compute the gradient. This will involve solving a single additional "adjoint" PDE of the same type as the state equation (2) and computing a volume and surface integral per parameter. The derivation here follows roughly that in [1], but it handles volume forces, it allows the entire Neumann boundary to evolve, and it is phrased in terms of shape parameters.

1 The Lagrangian and the Adjoint Method

We want to differentiate (1) with respect to parameter p_i. However, proceeding in a straight-forward way quickly hits a dead end:

$$\frac{\partial}{\partial p_i} J(p) = \frac{\partial}{\partial p_i} \int_{\Omega(p)} j(x, \sigma)\, \mathrm{d}x = \int_{\partial\Omega(p)} j(x, \sigma)(v_i(p) \cdot \hat{n})\, \mathrm{d}A + \int_{\Omega(p)} \frac{\partial}{\partial p_i} j(x, \sigma)\, \mathrm{d}x,$$

where $v_i(p)$ is the velocity of the boundary while varying p_i. This formula follows immediately from Reynold's transport theorem, but it also can be understood intuitively: the integral changes due to volume entering/leaving the integration domain (the first term) and due to the integrand changing (the second). The first term is easy to compute, but the second requires determining how the equilibrium displacement changes as the shape changes (hard). The adjoint method is a tool to sidestep that computation.

Following the traditional adjoint method derivation [2], we write the Lagrangian for the PDE-constrained optimization. We introduce Lagrange multipliers $\lambda \colon \overline{\Omega}(p) \to \mathbb{R}^n$ for the volume and Neumann boundary constraints in (2), but set $\lambda = 0$ on Γ_D since we do not need a term in the Lagrangian for the Dirichlet condition (we will always have u and all its perturbations vanish on the fixed Γ_D). The Lagrangian is thus:

$$\mathcal{L}(p,u,\lambda) = \int_{\Omega(p)} j(x,\sigma(u))\,\mathrm{d}x - \int_{\Omega(p)} \lambda \cdot [\nabla \cdot \sigma(u) + f(p)]\,\mathrm{d}x + \int_{\partial\Omega(p)} \lambda \cdot [\sigma(u)\hat{n} - g(p)]\,\mathrm{d}A,$$

where $u = 0$ on Γ_D is a displacement field independent of p (i.e. not necessarily the equilibrium displacement). Notice that if we take $u = u_*(p)$, the solution to the state equation, then $\mathcal{L}(p, u_*(p), \lambda) = J(p)$, independent of λ. Therefore:

$$\frac{\partial J}{\partial p_i}(p) = \frac{\mathrm{d}\mathcal{L}(p,u_*(p),\lambda)}{\mathrm{d}p_i} = \frac{\partial \mathcal{L}}{\partial p_i}(p,u_*(p),\lambda) + \left\langle \frac{\partial \mathcal{L}}{\partial u}(p,u_*(p),\lambda), \frac{\partial u_*}{\partial p_i}(p) \right\rangle,$$

where $\frac{\partial \mathcal{L}}{\partial u}$ is a linear functional that, when fed a vector field ϕ, computes the derivative of $\mathcal{L}$ in the ϕ direction (i.e. the Frechét derivative):

$$\left\langle \frac{\partial \mathcal{L}}{\partial u}(p,u_*(p),\lambda), \phi \right\rangle = \left. \frac{\partial \mathcal{L}(p, u_*(p) + h\phi, \lambda)}{\partial h} \right|_{h=0}.$$

Here is the key insight of the adjoint method: if we can choose $\lambda = \lambda_*(p)$ so that this linear functional is zero (vanishes on all admissible input vector fields), then we don't have to compute the problematic term $\frac{\partial u_*}{\partial p_i}(p)$. Then we just have:

$$\frac{\partial J}{\partial p_i}(p) = \frac{\partial \mathcal{L}}{\partial p_i}(p, u_*(p), \lambda_*(p)). \tag{3}$$

Finding such a $\lambda_*(p)$ is called the adjoint problem.

2 The Adjoint Problem

Soon we will derive the adjoint equations by setting

$$\left\langle \frac{\partial \mathcal{L}}{\partial u}(p,u_*(p),\lambda), \phi \right\rangle \stackrel{!}{=} 0 \quad \forall \phi,$$

but first we rewrite $\mathcal{L}$ in a more convenient form by restating in terms of strains and integrating by parts:

$$\begin{aligned}
\mathcal{L}(p,u,\lambda) &= \int_{\Omega(p)} j(x,\sigma(u))\,\mathrm{d}x - \int_{\Omega(p)} \lambda \cdot (\nabla \cdot [C : \epsilon(u)] + f(p))\,\mathrm{d}x + \int_{\partial\Omega(p)} \lambda \cdot ([C:\epsilon(u)]\,\hat{n} - g(p))\,\mathrm{d}A \\
&= \int_{\Omega(p)} j(x,\sigma(u))\,\mathrm{d}x + \overbrace{\int_{\Omega(p)} \epsilon(\lambda) : C : \epsilon(u)\,\mathrm{d}x - \cancel{\int_{\partial\Omega(p)} \lambda \cdot [C:\epsilon(u)]\,\hat{n}\,\mathrm{d}A}}^{\text{from I.B.P. identity (9) in Appendix A}} - \int_{\Omega(p)} \lambda \cdot f(p)\,\mathrm{d}x \\
&\qquad + \cancel{\int_{\partial\Omega(p)} \lambda \cdot [C:\epsilon(u)]\,\hat{n}\,\mathrm{d}A} - \int_{\partial\Omega(p)} \lambda \cdot g(p)\,\mathrm{d}A \\
&= \int_{\Omega(p)} j(x,\sigma(u))\,\mathrm{d}x + \int_{\Omega(p)} \epsilon(\lambda) : C : \epsilon(u)\,\mathrm{d}x - \int_{\Omega(p)} \lambda \cdot f(p)\,\mathrm{d}x - \int_{\partial\Omega(p)} \lambda \cdot g(p)\,\mathrm{d}A
\end{aligned} \tag{4}$$

Now we're ready to differentiate with respect to u. Let ϕ be an arbitrary perturbation vanishing on Γ_D. Then:

$$\begin{aligned}
\left\langle \frac{\partial \mathcal{L}}{\partial u}(p,u,\lambda), \phi \right\rangle &= \left.\frac{\mathrm{d}}{\mathrm{d}h}\right|_{h=0} \left(\int_{\Omega(p)} j\big(x, \sigma(u+h\phi)\big) + \epsilon(\lambda) : C : \epsilon(u+h\phi)\,\mathrm{d}x \right) \\
&= \int_{\Omega(p)} \left.\frac{\partial}{\partial h}\right|_{h=0} j\big(x, \sigma(u) + h\sigma(\phi)\big) + \left.\frac{\partial}{\partial h}\right|_{h=0} \epsilon(\lambda) : C : [\epsilon(u) + h\epsilon(\phi)]\,\mathrm{d}x,
\end{aligned}$$

where we used the linearity of ϵ and σ. Now, defining j' to be the rank 2 tensor of partial derivatives of j wrt. σ_{ij}, we get by the chain rule:

$$\begin{aligned}\left\langle \frac{\partial \mathcal{L}}{\partial u}(p,u,\lambda),\phi \right\rangle &= \int_{\Omega(p)} j'\big(x,\sigma(u)\big) : \sigma(\phi) + \epsilon(\lambda) : C : \epsilon(\phi)\,\mathrm{d}x \\ &= \int_{\Omega(p)} \big[j'\big(x,\sigma(u)\big) + \epsilon(\lambda)\big] : C : \epsilon(\phi)\,\mathrm{d}x\end{aligned}$$

Integrating by parts (see (10) in Appendix A) to move the strain operator off ϕ we get:

$$\left\langle \frac{\partial \mathcal{L}}{\partial u}(p,u,\lambda),\phi \right\rangle = -\int_{\Omega(p)} \big(\nabla\cdot C : \big[j'\big(x,\sigma(u)\big) + \epsilon(\lambda)\big]\big)\cdot\phi\,\mathrm{d}x + \int_{\partial\Omega(p)} \big(C : \big[j'\big(x,\sigma(u)\big) + \epsilon(\lambda)\big]\big)\,\hat{n}\cdot\phi\,\mathrm{d}A$$

Finally, we derive the adjoint equation by plugging in $u_*(p)$ and requiring this derivative to be zero:

$$-\int_{\Omega(p)} \Big(\nabla\cdot C : \Big[j'\Big(x,\sigma\big(u_*(p)\big)\Big) + \epsilon(\lambda)\Big]\Big)\cdot\phi\,\mathrm{d}x + \int_{\partial\Omega(p)} \Big(C : \Big[j'\Big(x,\sigma\big(u_*(p)\big)\Big) + \epsilon(\lambda)\Big]\Big)\,\hat{n}\cdot\phi\,\mathrm{d}A \overset{!}{=} 0.$$

For this to be true for all $\phi = 0$ on Γ_D, we must have:

$$\begin{cases} -\nabla\cdot\sigma(\lambda) = \nabla\cdot\Big[C : j'\Big(x,\sigma\big(u_*(p)\big)\Big)\Big] & \text{in}\,\Omega(p) \\ -\sigma(\lambda)\hat{n} = \Big[C : j'\Big(x,\sigma\big(u_*(p)\big)\Big)\Big]\,\hat{n} & \text{on}\,\Gamma_N(p) \\ \lambda = 0 & \text{on}\,\Gamma_D \end{cases} \tag{5}$$

where the Dirichlet condition comes from the requirement that λ vanish on Γ_D. This is a well-posed elastostatic problem whose solution is called the "adjoint state," $\lambda_*(p)$.

3 The Gradient

We compute the right-hand side of (3) by applying Reynold's transport theorem and (11) from Appendix B to (4):

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial p_i} = &\int_{\partial\Omega(p)} \left[j\big(x,\sigma(u)\big) + \epsilon(\lambda) : C : \epsilon(u) - \lambda\cdot f(p) - H\lambda\cdot g(p) - \frac{\partial\big(\lambda\cdot g(p)\big)}{\partial\hat{n}} \right] (v_i(p)\cdot\hat{n})\,\mathrm{d}A \\ &- \int_{\partial\Omega(p)} \lambda\cdot\frac{\partial g(p)}{\partial p_i}\,\mathrm{d}A - \int_{\Omega(p)} \lambda\cdot\frac{\partial f(p)}{\partial p_i}\,\mathrm{d}x,\end{aligned}$$

where $H = \nabla\cdot\hat{n}$ is the curvature (twice the mean curvature in 3D).

Now, using (3) and plugging in equilibrium and adjoint states $u_*(p)$ and $\lambda_*(p)$, we get:

$$\begin{aligned}\frac{\partial J}{\partial p_i}(p) = &\int_{\partial\Omega(p)} \left[j\big(x,\sigma(u_*)\big) + \epsilon(\lambda_*) : C : \epsilon(u_*) - \lambda_*\cdot f(p) - H\lambda_*\cdot g(p) - \frac{\partial\big(\lambda_*\cdot g(p)\big)}{\partial\hat{n}} \right] (v_i(p)\cdot\hat{n})\,\mathrm{d}A \\ &- \int_{\partial\Omega(p)} \lambda_*\cdot\frac{\partial g(p)}{\partial p_i}\,\mathrm{d}A - \int_{\Omega(p)} \lambda_*\cdot\frac{\partial f(p)}{\partial p_i}\,\mathrm{d}x.\end{aligned} \tag{6}$$

This equation tells us how to compute each component of the gradient provided we have:

1. (Normal) boundary velocities induced by each parameter.

2. How f and g change with p_i.

3. How $\lambda_* \cdot g(p)$ changes in the normal direction.

Item 1 is relatively easy and will be discussed in the following subsections. Items 2 and 3 are more problematic. In particular, item 3 is not well-defined as stated since g is only defined on the boundary. As discussed in Appendix B, the *existence* of an extension into a neighborhood of the boundary is needed at the very least. As discussed there, it may be possible to drop the normal derivative term—the case here is less clear since λ_* already is defined inside the object. Another possible way to side-step this term is to blur all surface tractions into volume forces and set $g = 0$ (which is how the current meshfree discretization works anyway). However, we'd still have to figure out how to differentiate the resulting volume force, f, as mentioned in item 2.

3.1 Boundary Velocities for CSG Trees

In the case of objects represented by CSG trees, p is a vector of all the parameters appearing in any primitive of the tree. As claimed in [3], assuming primitives in a CSG object intersect transversely, the boundary can be decomposed into segments controlled by a single primitive's design parameters. Then, $v_i(p)$ is zero on every boundary segment not belonging to p_i's primitive(s). On each boundary segment where it is nonzero, $v_i(p)$ can be found by analyzing the corresponding primitive in isolation.

The boundary velocities for translation parameters are trivial. Rotation is not so hard either: for a rotation around axis $\hat{a}$ through point c, the velocity at a boundary point b is $\hat{a} \times (b - c)$. Size parameters for some primitives are easy (boxes), but for others they will require more thought (ellipsoids).

Incidentally, it is unclear how [3] maintains its fixed Dirichlet boundary. It must only consider examples where there are no parameters affecting the Dirichlet boundary. In our case this is too restrictive, so we will need to relax the requirement of a nonempty Dirichlet boundary (see Section 4).

3.2 Boundary Velocities for Parametrized Level Set Functions

Suppose now that we do not have the structure of a CSG tree and instead have only a parameterized level set function $\Phi(p, x)$ defining the object:

$$\begin{cases} \Phi(p,x) > 0 & \text{for } x \in \Omega(p) \\ \Phi(p,x) = 0 & \text{for } x \in \partial\Omega(p) \\ \Phi(p,x) < 0 & \text{for } x \in \mathbb{R}^n \setminus \overline{\Omega}(p) \end{cases}$$

We can find the boundary velocity by considering the position of a particular point on the boundary as a function, $x(p)$. The boundary velocity is, by definition, $v_i(p)(x) = \frac{\partial x}{\partial p_i}$, and the chain rule gives:

$$\Phi(p, x(p)) = 0 \quad \Longrightarrow \quad \frac{\partial \Phi(p, x(p))}{\partial p_i} = 0 = \frac{\partial \Phi}{\partial p_i}(p, x(p)) + \nabla\Phi \cdot v_i(p)$$

Recalling the basic level set result, $\hat{n} = -\frac{\nabla\Phi}{\|\nabla\Phi\|}$:

$$-\|\nabla\Phi\| \hat{n} \cdot v_i(p) = -\frac{\partial \Phi}{\partial p_i}(p, x(p)) \quad \Longrightarrow \quad \hat{n} \cdot v_i(p) = \frac{1}{\|\nabla\Phi\|} \frac{\partial \Phi}{\partial p_i}(p, x(p)).$$

This equation gives us the normal velocity for points on any boundary of the object.

4 Getting Rid of Γ_D

The fixed Dirichlet boundary Γ_D is incompatible with our goals. Unfortunately, all existing literature seems to rely on this boundary condition; though [1] mentions that "an equilibrium condition on g" is needed in the absence of the Dirichlet boundary, they do not discuss how to keep the adjoint problem well-posed. Ideally, we would instead use the no-rigid-motion constraints from our Worst-case Structural Analysis paper and allow the entire boundary to evolve:

$$J(p) = \int_{\Omega(p)} j(x, \sigma(u)(x))\,\mathrm{d}x$$
$$\text{s.t.} \quad \begin{cases} -\nabla \cdot \sigma(u) = f(p) & \text{in}\,\Omega(p) \\ \sigma(u)\hat{n} = g(p) & \text{on}\,\partial\Omega(p) \\ \displaystyle\int_{\Omega(p)} u\,\mathrm{d}x = 0, \quad \int_{\Omega(p)} u \times x\,\mathrm{d}x = 0 \end{cases} \tag{7}$$

Introducing the Lagrange multiplier *vectors*, λ_t and λ_r, for the zero net translation and zero net rotation constraints respectively, the Lagrangian becomes:

$$\mathcal{L}(p, u, \lambda, \lambda_t, \lambda_r) = \int_{\Omega(p)} j(x, \sigma(u))\,\mathrm{d}x - \int_{\Omega(p)} \lambda \cdot [\nabla \cdot \sigma(u) + f(p)]\,\mathrm{d}x + \int_{\partial\Omega(p)} \lambda \cdot [\sigma(u)\hat{n} - g(p)]\,\mathrm{d}A +$$
$$\lambda_t \cdot \int_{\Omega(p)} u\,\mathrm{d}x + \lambda_r \cdot \int_{\Omega(p)} u \times x\,\mathrm{d}x$$

Again, $\mathcal{L}$ is independent of λ, λ_t, λ_r when u is chosen as a solution to (7). This allows us to still choose choose the Lagrange multipliers so that:

$$\left\langle \frac{\partial \mathcal{L}}{\partial u}(p, u, \lambda, \lambda_t, \lambda_r), \phi \right\rangle = 0,$$

which can be shown to happen when $\lambda_t = 0$, $\lambda_r = 0$, and λ satisfies

$$\begin{cases} -\nabla \cdot \sigma(\lambda) = \nabla \cdot \Big[C : j'\Big(x, \sigma\big(u_*(p)\big)\Big)\Big] & \text{in}\,\Omega(p) \\ \sigma(\lambda)\hat{n} = \Big[C : j'\Big(x, \sigma\big(u_*(p)\big)\Big)\Big]\,\hat{n} & \text{on}\,\partial\Omega(p) \end{cases} \tag{8}$$

This, however, is *not* a well-posed elastostatic problem: λ_* is determined only up to a rigid motion offset. While this offset doesn't affect the terms of the gradient where the adjoint state enters though the strain operator, it does affect the terms like $\lambda_* \cdot f$. More thought is needed to determine how to proceed.

5 Contributions

So far, if we solve the remaining problems, the contributions would be:

- Minimize stress-based objectives *with respect to shape parameters* (ideally, parameters more general than those controlling a single primitive).
- Optimize without Dirichlet constraints (with no-rigid-motion constraints).
- Handle design-dependent loads. There isn't so much literature on this. The closest thing I've seen is handling constant pressure Neumann conditions on the evolving boundary, but more of a literature search is needed.
- Penalize visual dissimilarity instead of the traditional volume penalties/constraints.

6 What is Needed/Concerns

So far, it appears the additional derivation/code needed to compute gradients of stress-based objectives is:

- Computing boundary velocity per primitive.
- Computing the divergence of the tensor field $C : j'\Big(x, \sigma\big(u_*(p)\big)\Big)$. This is likely to have numerical robustness/accuracy issues due to cancellation.
- Predicting how worst-case volume forces or surface tractions change with boundary movement.

The additional theory questions/points that need to be addressed are:

- How do we reformulate with no-rigid-motion constrains instead of Dirichlet boundary conditions while keeping the adjoint problem well-posed?
- Can we drop the normal derivative term appearing in (6)?
- How do we model loading scenarios changing as the boundary changes?
- Find a more rigorous derivation for Appendix B (e.g., see footnote).

Finally, it is worrying that the gradient specified by (6) has the term $\int_{\partial\Omega(p)} j\big(x, \sigma(u_*)\big) v_i(p) \cdot \hat{n} \, \mathrm{d}A$, which encourages removal of high-stress material. While this term makes sense in context of the formal derivation, intuitively it is counterproductive; removing high-stress regions only redistributes their stresses inward and will in fact weaken the object further. The term must be compensated for by the rest of the gradient's terms (involving the adjoint state), but it makes me worry that the stress field doesn't satisfy some of the differentiability properties we implicitly assume while carrying out the formal computations. This could possibly invalidate the derivations.

Appendix A Integration By Parts Formulas

Here we derive the two integration by parts formulas used in this writeup. The derivations are straightforward once broken down to their components using index notation. First, we derive the one for:

$$\int_{\Omega(p)} \lambda \cdot (\nabla \cdot [C : \epsilon(u)]) \, \mathrm{d}x = \int_{\Omega(p)} \lambda_i \frac{\partial}{\partial x_j} \big[C_{ijkl} \epsilon_{kl}\big] \, \mathrm{d}x$$

From product rule:

$$\int_{\Omega(p)} \lambda_i \frac{\partial}{\partial x_j} C_{ijkl} \epsilon_{kl}(u) \, \mathrm{d}x = \int_{\Omega(p)} \frac{\partial}{\partial x_j} \big[\lambda_i C_{ijkl} \epsilon_{kl}(u)\big] - \frac{\partial \lambda_i}{\partial x_j} C_{ijkl} \epsilon_{kl}(u) \, \mathrm{d}x$$

Notice that $\frac{\partial \lambda_i}{\partial x_j} C_{ijkl} = \epsilon_{ij}(\lambda)$ by the symmetry $C_{ijkl} = C_{jikl}$. Applying Gauss' theorem to the term on the left, we get:

$$\int_{\partial\Omega(p)} \lambda_i C_{ijkl} \epsilon_{kl}(u) \hat{n}_j \, \mathrm{d}A - \int_{\Omega(p)} \epsilon_{ij}(\lambda) C_{ijkl} \epsilon_{kl}(u) \, \mathrm{d}x$$

Reinterpreting these in coordinate-free notation, we arrive at the desired expression,

$$\int_{\partial\Omega(p)} \lambda \cdot \big[C : \epsilon(u)\big] \hat{n} \, \mathrm{d}A - \int_{\Omega(p)} \epsilon(\lambda) : C : \epsilon(u) \, \mathrm{d}x. \tag{9}$$

Next, we derive the identity used to move the strain operator off ϕ in

$$\int_{\Omega(p)} [j'(x, \sigma(u)) + \epsilon(\lambda)] : C : \epsilon(\phi) \, \mathrm{d}x$$

To simplify notation, let $\gamma(x, u, \lambda)$ be the rank 2 tensor $j'(x, \sigma(u)) + \epsilon(\lambda)$. We want to show:

$$\int_{\Omega(p)} \gamma(x,u,\lambda) : C : \epsilon(\phi)\,\mathrm{d}x = -\int_{\Omega(p)} \big(\nabla \cdot \big[C : \gamma(x,u,\lambda)\big]\big) \cdot \phi\,\mathrm{d}x + \int_{\partial\Omega(p)} \big[C : \gamma(x,u,\lambda)\big]\hat{n} \cdot \phi\,\mathrm{d}A \tag{10}$$

In component notation we have, by symmetry of constant $C_{ijkl} = C_{ijlk}$,

$$\int_{\Omega(p)} \gamma_{ij} C_{ijkl} \epsilon_{kl}(\phi)\,\mathrm{d}x = \int_{\Omega(p)} \gamma_{ij} C_{ijkl} \frac{\partial \phi_k}{\partial x_l}\,\mathrm{d}x = \int_{\Omega(p)} \gamma_{ij} \frac{\partial C_{ijkl}\phi_k}{\partial x_l}\,\mathrm{d}x$$

By product rule the expression becomes:

$$\int_{\Omega(p)} \frac{\partial}{\partial x_l}\big[\gamma_{ij} C_{ijkl} \phi_k\big] - \frac{\partial \gamma_{ij}}{\partial x_l} C_{ijkl}\phi_k\,\mathrm{d}x.$$

Applying Gauss' theorem to the left term, we get:

$$\int_{\partial\Omega(p)} \gamma_{ij} C_{ijkl} \phi_k \hat{n}_l\,\mathrm{d}A - \int_{\Omega(p)} \frac{\partial \gamma_{ij}}{\partial x_l} C_{ijkl}\phi_k\,\mathrm{d}x.$$

Using the major symmetry $C_{ijkl} = C_{klij}$ and the fact C_{ijkl} is constant, this is:

$$\int_{\partial\Omega(p)} \gamma_{kl} C_{ijkl} \phi_i \hat{n}_j\,\mathrm{d}A - \int_{\Omega(p)} \frac{\partial \gamma_{kl} C_{ijkl}}{\partial x_j} \phi_i\,\mathrm{d}x.$$

Reinterpreting in coordinate-free notation, gives the desired formula, (10).

Appendix B Derivative of a Surface Integral

Here we compute the derivative of a boundary integral. Unlike the volume integral case, to which Reynold's transport theorem applies, there doesn't seem to be a classical theorem available. We derive a slightly more general statement than those appearing without proof in many of the topology optimization papers (e.g. [1]). This more general version has a chance of handling design-dependent loads, but there are still problems to be solved as we will soon see.

The task is to compute:

$$\frac{\partial}{\partial p_i} \int_{\partial\Omega(p)} h(p,x)\,\mathrm{d}A$$

We use Gauss' theorem to rewrite as a volume integral so that we can apply Reynold's transport theorem:

$$\frac{\partial}{\partial p_i} \int_{\partial\Omega(p)} h(p,x)\,\mathrm{d}A = \frac{\partial}{\partial p_i} \int_{\partial\Omega(p)} \tilde{h}(p,x)\tilde{n} \cdot \hat{n}\,\mathrm{d}A = \frac{\partial}{\partial p_i} \int_{\Omega(p)} \nabla \cdot \big[\tilde{h}(p,x)\tilde{n}\big]\,\mathrm{d}x$$

Here $\tilde{h}$ and $\tilde{n}$ represent an extension of the surface fields h and $\hat{n}$ into the volume ($\tilde{h} = h$ and $\tilde{n} = \hat{n}$ on $\partial\Omega(p)$). It will turn out that $\tilde{n}$ is only ever evaluated on the boundary, so only its existence is needed. However, *we will end up needing to compute normal derivatives of* $\tilde{h}$. In [4] a similar extension is needed, and they discuss using "transfinite interpolation" with approximate distance fields.

Applying Reynold's transport theorem, the derivative becomes:

$$\int_{\partial\Omega(p)} \nabla \cdot \big[\tilde{h}(p,x)\tilde{n}\big]\big(v_i(p) \cdot \hat{n}\big)\,\mathrm{d}A + \int_{\Omega(p)} \frac{\partial}{\partial p_i} \nabla \cdot \big[\tilde{h}(p,x)\tilde{n}\big]\,\mathrm{d}x$$

Applying the product rule $\nabla \cdot \big[\tilde{h}(p,x)\tilde{n}\big] = \nabla\tilde{h}(p,x) \cdot \tilde{n} + \tilde{h}(p,x)\nabla \cdot \tilde{n}$ on the left and exchanging the order of differentiation on the right:

$$\int_{\partial\Omega(p)} \left[\frac{\partial \tilde{h}(p,x)}{\partial \hat{n}} + h(p,x)H\right]\big(v_i(p) \cdot \hat{n}\big)\,\mathrm{d}A + \int_{\Omega(p)} \nabla \cdot \left[\frac{\partial \tilde{h}}{\partial p_i}(p,x)\tilde{n} + \tilde{h}(p,x)\frac{\partial \tilde{n}}{\partial p_i}\right]\,\mathrm{d}x,$$

where we replaced tilde quantities evaluated on the boundary with there pre-extension counterparts and set $H = \nabla \cdot \tilde{n}|_{\partial\Omega(p)}$, which should be the trace of the shape operator (the sum of the principle curvatures) for any reasonable extension $\tilde{n}$. We claim the boundary normal and its extension change negligibly with p_i so we can drop the last term[1]. Applying Gauss' theorem:

$$\int_{\partial\Omega(p)} \left[\frac{\partial \tilde{h}(p,x)}{\partial \hat{n}} + h(p,x)H\right] \big(v_i(p) \cdot \hat{n}\big) \, \mathrm{d}A + \int_{\partial\Omega(p)} \left[\frac{\partial \tilde{h}}{\partial p_i}(p,x)\tilde{n}\right] \cdot \hat{n} \, \mathrm{d}A,$$

so finally we get the identity:

$$\frac{\partial}{\partial p_i} \int_{\partial\Omega(p)} h(p,x) \, \mathrm{d}A = \int_{\partial\Omega(p)} \left[\frac{\partial \tilde{h}(p,x)}{\partial \hat{n}} + h(p,x)H\right] \big(v_i(p) \cdot \hat{n}\big) \, \mathrm{d}A + \int_{\partial\Omega(p)} \frac{\partial h}{\partial p_i}(p,x) \, \mathrm{d}A. \tag{11}$$

Though this requires a careful handling of $\tilde{h}$, each term has an intuitive meaning: the first accounts for the boundary moving to encounter new scalar field values, the second handles dilation/contraction of the boundary (since $Hv_i(p) \cdot \hat{n}$ gives the change in surface area due to normal motion), and the third handles the parameterized changes of h (in our case, the changes in the design-dependent load).

If we can define $\tilde{h}$ so that $\left.\frac{\partial \tilde{h}(p,x)}{\partial \hat{n}}\right|_{\partial\Omega(p)} = 0$ (the most natural choice, which should be possible to do in a neighborhood though I don't have a proof), then the first term vanishes and we are left with:

$$\frac{\partial}{\partial p_i} \int_{\partial\Omega(p)} h(p,x) \, \mathrm{d}A = \int_{\partial\Omega(p)} h(p,x) H v_i(p) \cdot \hat{n} \, \mathrm{d}A + \int_{\partial\Omega(p)} \frac{\partial h}{\partial p_i}(p,x) \, \mathrm{d}A. \tag{12}$$

References

[1] Gregoire Allaire and Francois Jouve. Minimum stress optimal design with the level set method. *Engineering Analysis with Boundary Elements*, 32(11):909 – 918, 2008. Shape and Topological Sensitivity Analysis: Theory and Applications.

[2] Andrew Bradley. PDE-constrained optimization and the adjoint method, June 2013. `http://www.stanford.edu/~ambrad/adjoint_tutorial.pdf`.

[3] Jiaqin Chen, Michael Freytag, and Vadim Shapiro. Shape sensitivity of constructive representations. In *Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling*, SPM '07, pages 85–95, New York, NY, USA, 2007. ACM.

[4] Jiaqin Chen, Vadim Shapiro, Krishnan Suresh, and Igor Tsukanov. Shape optimization with topological changes and parametric control. *International journal for numerical methods in engineering*, 71(3):313–346, 2007.

[1] I do not have a proof for this, but it seems sensible...