MAP562 Optimal design of structures (École Polytechnique)

by Grégoire Allaire, Beniamin Bogosel

Session 6: Feb 13th, 2019 – Geometric Optimization

Exercise 1 Optimize an inclusion

In this exercise we consider an elastic body Ω which is clamped on Γ_D and is subjected to some loadings on part Γ_N. We assume that $\partial\Omega = \Gamma_D \cup \Gamma_N$. We suppose that there is a hole $\omega \subset \Omega$ which is traction-free will be subject to optimization. As usual, the displacements u are computed using the linearized elasticity equation

$$\begin{cases} -\operatorname{div}\sigma &= 0 & \text{in } \Omega \setminus \omega \\ u &= 0 & \text{on } \Gamma_D \\ \sigma n &= \sigma_0 n & \text{on } \Gamma_N \\ \sigma n &= 0 & \text{on } \partial\omega \end{cases}.$$

The objective function is the compliance

$$J(\omega) = \int_{\Gamma_N} \sigma_0 n \cdot u.$$

1. Show that an alternate formula for the compliance is
$$J(\omega) = \int_{\Omega\setminus\omega} \sigma \cdot e(u) dx$$
where σ is the stress tensor $2\mu e(u) + \lambda \operatorname{div} u$.
2. Compute the shape derivative with respect to ω, when ω is a general inclusion in the direction given by a general vector field θ. (Note that the outer boundary $\partial\Omega$ does not move.)

In the rest of the exercise we suppose that the inclusion ω is an ellipse, which will simplify the numerical aspects. The ellipse is characterized by its center $\mathbf{c} = (x_0, y_0)$ its principal axes a, b and its orientation α.

3. What are the vector fields θ which correspond to the following transformations T of $\partial\omega$:
   - Translations $T\mathbf{x} = \mathbf{x} + \mathbf{v}$ with $\mathbf{v} = (v_x, v_y)$
   - Scalings $T\mathbf{x} = (\alpha, \beta) \cdot (\mathbf{x} - \mathbf{c}) + \mathbf{c}$, with $\alpha, \beta \in \mathbb{R}_+$
   - **(Optional)** Rotations $T\mathbf{x} = \mathbf{A}(\mathbf{x} - \mathbf{c}) + \mathbf{c}$ where $\mathbf{A} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$ for some angle α.

   Deduce the partial derivatives of $J(\omega)$ with respect to the parameters defining the ellipse: x_0, y_0, a, b (**(Optional)** α).
4. Implement a `FreeFem++` algorithm which performs the optimization of $J(\omega)$, when $|\omega|$ is fixed, in the following cases:
   - $\partial\omega$ is a circle ($a = b = r$) and the center (x_0, y_0) is variable.
   - $\partial\omega$ is an ellipse with fixed center and variable axes
   - $\partial\omega$ is a general ellipse with axes oriented parallel to the coordinate axes
   - **(Optional)** $\partial\omega$ is a general ellipse (including variable orientation)
5. **(Homework)** Consider the following types of parametric curves:
   1. **Stadiums:** rectangles of dimensions $a \times b$ with a half disk of diameter b glued to the two opposite sides of length b. Optimization variables: a, b and the center.
   2. **Superellipse:** parametric curve defined by $\frac{|x|^p}{a^p} + \frac{|y|^p}{b^p} = 1$ where $p = 4$. Optimization variables: a, b and the center.
   3. **Generic parametric curve:** curve with given radial parametrization $\rho(\beta) = 1 + a\cos(2\beta) + b\cos(3\beta)$. Recall that given a radial function $[0, 2\pi] \ni \beta \mapsto \rho(\beta) \in (0, \infty)$ the corresponding parametric curve is defined by $(x(\beta), y(\beta)) = (\rho(\beta)\cos\beta, \rho(\beta)\sin\beta)$. (notice that this is one of the standard ways to define a curve in `FreeFem++`) You may suppose that $|a| \le 0.5, |b| \le 0.5$ and the bounding box is the square $(-2.5, 2.5)^2$. For this type of curve you may assume

that the center for the radial parametrization is fixed. The optimization variables are: a, b. Note that in the computation of the derivative you may need the value of the angle β on the boundary of the curve $\partial\omega$. You may use the `FreeFem++` function `atan2` to do this (an example will be posted on Moodle).

Choose one of the categories of curves defined above and compute the partial derivatives of $J(\omega)$ with respect to the relevant parameters. As above, it is useful to find what is the corresponding vector field θ when varying the desired parameter and then use the general shape derivative formula found at Question 1.

Correction of part 3

Determination of a vector field corresponding to the deformation given by the change of one of the parameters of the ellipse.

As we saw in class, the shape derivative of the compliance with respect to perturbations given by vector fields which deform $\partial\omega$ is given by

$$J'(\omega)(\theta) = -\int_{\partial\omega} \sigma(u) \cdot e(u)\theta.nds.$$

Here we used the fact that the problem associated to the compliance is self-adjoint and the adjoint is simply $p = -u$. In a general gradient algorithm, once we have the shape derivative we would like to modify the boundary such that $J'(\omega)(\theta)$ is negative. This comes to the choice of a vector-field θ such that $J'(\omega)(\theta) < 0$.

In the simplified case presented in the above exercise, the shape ω is an ellipse depending on the parameters $\mathbf{c} = (x_0, y_0)$ and the length of principal axes a, b. The equation of the ellipse can be written as

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1.$$

In order to write an optimization algorithm we need to find the derivative of $J(\omega)$ with respect to the parameters x_0, y_0, a, b. This is done by observing what are the associated deformation fields when perturbing the given parameters.

In general, if $\mathcal{F}(\omega) = \mathcal{F}(\mathbf{a})$ is a shape function depending on the parameter $\mathbf{a}$, the derivative with respect to $\mathbf{a}$ in the direction $\mathbf{q}$ can be understood as the directional derivative

$$\lim_{t\to 0} \frac{F(\mathbf{a} + t\mathbf{q}) - F(\mathbf{a})}{t}.$$

In order to turn this into something related to the shape derivative we should find a vector field θ such that $\mathcal{F}(\mathbf{a} + t\mathbf{q}) = \mathcal{F}((Id + \theta)(\Omega))$. Then, the above derivative will be the shape derivative evaluated for the vector field θ

$$\langle \nabla\mathcal{F}(\mathbf{a}), \mathbf{q}\rangle = \mathcal{F}'(\Omega)(\theta)$$

Case 1. **Translations.** When translating an object in the direction $\mathbf{v}$ the vector field θ is obviously equal to $\mathbf{v}$. This shows that the gradient of the compliance with respect to (x_0, y_0) in the direction $\mathbf{v}$ is equal to

$$J'(\omega)(\mathbf{v}) = \int_{\partial\omega} (-\sigma \cdot e(u))(\mathbf{v}.n)ds.$$

In particular, the partial derivative with respect to x_0 can be found by choosing $\mathbf{v} = (1, 0)$ and the partial derivative with respect to y_0 is found for $\mathbf{v} = (0, 1)$.

Case 2. **Scalings.** Since we want to be able to perturb both a and b independently, we will consider scalings in x and y separately. Let's suppose that the parameter a is changed to $a + t$ and see what is a corresponding vector field θ. Let's note that since b does not change, it is more practical to fix the variable y and to look what is ne new x. Note the new coordinates with x', y'. Therefore we have

$$\frac{(x' - x_0)^2}{(a + t)^2} + \frac{(y' - y_0)^2}{b^2} = 1 = \frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2}.$$

Since we agreed that $y' = y$ we are left with

$$\frac{x' - x_0}{a + t} = \frac{x - x_0}{a}$$

which is equivalent to

$$x' - x = \frac{t}{a}(x - x_0)$$

and since $(x' - x, 0) = t\theta$ we have

$$\theta_x = \frac{x' - x}{t} = \frac{x - x_0}{a}.$$

This gives us $\theta = (\frac{x-x_0}{a}, 0)$.

In a similar manner the derivative of J with respect to b is obtained by choosing $\theta = (0, \frac{y-y_0}{b})$.

Second method. If x and y can be written explicitly in terms of the parameters, then the vector field associated to the variation of one of the parameters can be obtained by differentiating x and y with respect to the corresponding parameter.

For example, in the case of the ellipse we have

$$\begin{cases} x = x_0 + a\cos\theta \\ y = y_0 + b\sin\theta \end{cases}$$

We can compute immediately the partial derivatives of $\mathbf{x} = (x, y)$ with respect to all the parameters

- $\displaystyle\frac{\partial \mathbf{x}}{\partial x_0} = (1, 0)$
- $\displaystyle\frac{\partial \mathbf{x}}{\partial y_0} = (0, 1)$
- $\displaystyle\frac{\partial \mathbf{x}}{\partial a} = (\cos\theta, 0) = (\frac{x - x_0}{a}, 0)$
- $\displaystyle\frac{\partial \mathbf{x}}{\partial x_0} = (0, \sin\theta) = (0, \frac{y - y_0}{b})$

You may observe that we get the same results as above. This method is faster, but it needs an explicit parametrization of the shape in terms of all the variables.

Similar arguments can be used to solve the homework questions. The PDE remains the same, but the shapes and parameters change.

Remarks regarding the code

1. Once the derivatives of (x_0, y_0, a, b) are found and are denote with (dx_0, dy_0, da, db) we perform a gradient type update

$$\begin{cases} x_0 &= x_0 - \gamma dx_0 \\ y_0 &= y_0 - \gamma dy_0 \\ a &= a - \gamma da \\ b &= b - \gamma db \end{cases}$$

   where γ is a step size. In the ellipse problem given above, you may note that this does not work very well: in some cases, the translation derivatives do not seem to have any effect. If you output the derivatives you will notice that the derivatives with respect to (x_0, y_0) are small relative to the derivatives with respect to (a, b). Therefore, the algorithm, in this form, will not manage to capture the change of center.

   An alternative procedure is to use different step sizes for each of the derivatives. This is justified by the following argument: if A is a positive definite matrix then if $\vec{d}$ gives a descent direction then $A\vec{d}$ is also a descent direction. Thus, a modified gradient algorithm reads

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma A\vec{d}.$$

   A particular positive definite matrix is a diagonal matrix with positive entries. This amounts to choosing different scalings for the step size when looking at different variables.

   The variant proposed in the code on Moodle consists of choosing the step size normalized to the size of the gradient for the (x_0, y_0) part and again for the (a, b) part.

2. The ellipse ω should have constant volume, if not, it will shrink to a point. We can preserve the volume at each iteration by doing a rescaling: once we found the new a and the new b compute the area of the new ellipse ($ab\pi$) and find the scaling factor δ so that $\delta^2 ab\pi$ gives the desired area.

3. The ellipse ω should not touch the exterior boundary $\partial\Omega$, so in the course of the algorithm you should make sure that a, b, x_0, y_0 remain within some bounds. Note that `FreeFem++` will not be able to mesh your domain and will give an error if you don't pay attention at this. The idea is to simply project a, b, x_0, y_0 so that a, b are smaller than half the length of the square and bigger than 0, and (x_0, y_0) is in a certain region of the Ω which allows you to build the ellipse.

   This considerations should be updated when changing the geometry of the inclusion for the Homework questions.