# AN INTRODUCTION TO SHAPE AND TOPOLOGY OPTIMIZATION: HANDS-ON SESSION

C. DAPOGNY[1]

[1] *Univ. Grenoble Alpes, CNRS, Grenoble INP, LJK, 38000 Grenoble, France,*

Contents

The purpose of this exercise session is to guide the attendants through the implementation of three popular strategies for shape and topology optimization, namely density-based methods (such as the famous SIMP method), geometric optimization methods, and level set methods. 'Simple' shape and topology optimization problems are investigated, which are already quite relevant and interesting from the physical viewpoint.

The session are based on `FreeFem++` [18], an open-source environment which makes it possible to solve Partial Differential Equations (PDE) with the finite element method from the input of their variational formulation via a syntax close to that of `C++` .

The exercise session is organized as follows: the attendants are encouraged to start with the three exercises in Section 1, in which the three physical settings of interest in this course are introduced, namely those of the Laplace equation, the linearized elasticity system, and the Stokes equations. The main features and commands of `FreeFem++` are presented in the meantime; for the (many!) subsequent possibilities offered by this environment, the attendants will be referred to the exhaustive, albeit comprehensive manual [19], which may be downloaded at the following address:

`http://www.freefem.org/ff++/ftp/freefem++doc.pdf`

Part of the present material has been prepared in collaboration with Pascal Frey (ISCD & Laboratoire Jacques-Louis Lions, Université Paris VI) and Yannick Privat (CNRS & Laboratoire Jacques-Louis Lions), whose help is warmly acknowledged!

Attendants are then asked to select *one* of the three shape and topology optimization settings discussed during the course, and to work on the corresponding set of exercises: topology optimization using density-based methods are discussed in Section 2 while geometric optimization methods are considered in Section 3; finally, the level set method for shape and topology optimization is the main purpose of Section 4.

Throughout the following sections, more difficult theoretical or practical problems - tagged with a star $^*$ - are left as food for thought to the reader. The solutions to these are not required for the correct understanding of the targetted notions.

The present notes, as well as tentative corrections to most of the theoretical and practical questions they contain, may be downloaded from the following `GitHub` repository:

`https://github.com/dapogny/GDR-MOA-Course`

Obviously, any comment or judgement drawn from this series of exercises (flaws in such or such method, etc.) should apply only to the author's personal bias and implementations.

## 1. Getting started with FreeFem++

This first section is a short introduction to the basic features of `FreeFem++`. Attendants who are already familiar with this environment may get very quickly through this preliminary part, or even skip it altogether.

### 1.1. **A worked example: resolution of the Laplace equation**

Our first contact with `FreeFem++` arises in the context of the Laplace equation, which is a basic physical model for the stationary distribution of the temperature within a region, or for the voltage potential inside an electrically conductive medium.

Let $D$ be the L-shaped 2d domain depicted on Fig. 1 (left); $D$ is filled with a material with thermal conductivity $\gamma$; a heat source $f \in L^2(D)$ is acting inside $D$ and its boundary $\partial D$ is kept at temperature 0, so that the temperature field $u : D \to \mathbb{R}$ is the solution to the following Laplace equation:

$$\left\{ \begin{array}{rl} -\text{div}(\gamma \nabla u) = f & \text{in } D, \\ u = 0 & \text{on } \partial D. \end{array} \right. \tag{1.1}$$

The numerical resolution of (1.1) using `FreeFem++` relies on the finite element method, which we now briefly summarize, referring to classical monographs such as [2]for more exhaustive presentations. The starting point of this method is the associated *variational formulation* to (1.1): $u$ is sought in the functional space $V := H^1_0(D)$ of functions in $H^1(D)$ whose trace vanishes on $\partial D$; it is the unique function in this space that satisfies:

$$\forall v \in V, \;\; a(u,v) = \int_D fv \, dx, \;\; \text{where } a(u,v) := \int_D \gamma \nabla u \cdot \nabla v \, dx. \tag{1.2}$$

In practice, $D$ is discretized by means of a computational mesh $\mathcal{T}$ (for instance composed of triangles); see Fig. 1 (right). Such a mesh may be generated in `FreeFem++` from the input of the boundary $\partial D$ as a set of parametrized curves, according to Listing 1.

```
/* Declaration of the boundary curves of the domain */
border left(t=0,1.0){ x=0.0 ; y=1.0-t; label=1;};
border bot(t=0,1.0){ x=t ; y=0.0; label=1;};
border right(t=0,0.5){ x=1.0 ; y=t; label=1;};
border ang1(t=0,0.5){ x=1.0-t ; y=0.5; label=1;};
border ang2(t=0,0.5){ x=0.5 ; y=0.5+t; label=1;};
border top(t=0,0.5){ x=0.5-t ; y=1.0; label=1;};

/* Build mesh, display and save as a .mesh file */
mesh Th = buildmesh(left(10)+bot(10)+right(5)+ang1(5)+ang2(5)+top(5));
plot(Th,wait=1);
savemesh(Th,"Lshape.mesh");
```

Listing 1. *Generation of a triangular mesh for the L-shaped domain of Section 1.1.*

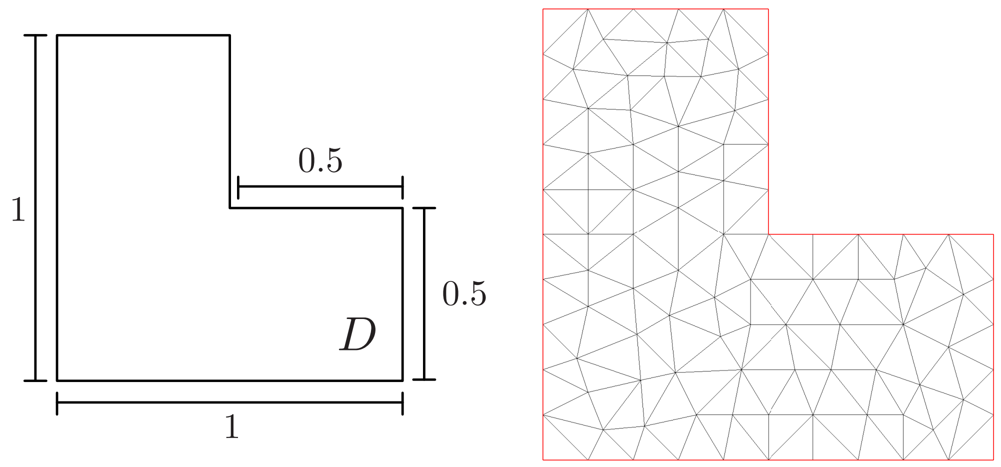


FIGURE 1. *(Left) The L-shaped domain $D$ considered in Section 1.1 and (right) an associated triangular mesh generated using* `FreeFem++`.

Thence, the (continuous) variational formulation (1.2) is discretized on the mesh $\mathcal{T}$: a finite element space $V_h \subset V$ is chosen, which is indexed by the size $h$ of the mesh $\mathcal{T}$. For instance $V_h$ may be chosen as the set of $\mathbb{P}_1$ Lagrange finite element functions on $\mathcal{T}$, that is:

$$\left\{u \in H^1_0(D),\ \forall T \in \mathcal{T},\ u|_T \text{ is a bivariate first-order polynomial function}\right\}, \tag{1.3}$$

or as the set of $\mathbb{P}_2$ Lagrange finite element functions on $\mathcal{T}$

$$\left\{u \in H^1_0(D),\ \forall T \in \mathcal{T},\ u|_T \text{ is a bivariate second-order polynomial function}\right\}. \tag{1.4}$$

The discrete counterpart to (1.2) reads: search for $u_h \in V_h$ such that:

$$\forall v_h \in V_h,\ \ a(u_h, v_h) = \int_D f v_h \, dx. \tag{1.5}$$

Introducing a basis $\{\varphi_i\}_{i=1,\dots,N_h}$ of $V_h$, this simply rewrites as the linear system:

$$AU = b \tag{1.6}$$

where the unknown vector $U = (u_{h,1}, \dots, u_{h,N_h})^T \in \mathbb{R}^{N_h}$ gathers the coordinates of $u_h$ in the basis $\{\varphi_i\}$:

$$u_h = \sum_{i=1}^{N_h} u_{h,i} \varphi_i, \tag{1.7}$$

and $A$ (resp. $b$) is the $N_h \times N_h$ matrix (resp the vector of size $N_h$) whose entries are given by:

$$\forall i, j = 1, \dots, N_h,\ \ A_{ij} = a(\varphi_j, \varphi_i),\ \text{and } b_i = \int_D f \varphi_i \, dx.$$

In `FreeFem++`, finite element spaces (and functions) of the form (1.3) and (1.4) are declared as in Listing 2.

```
/* Definition of finite element spaces and functions */
fespace Vh(Th,P1); // or P2
Vh u,v;

/* Other parameters */
real gamma = 1.0;

/* Source term */
func real f() {
  return(1.0);
}
```

LISTING 2. *Definition of finite element spaces and functions in `FreeFem++`.*

Finally, the variational formulation (1.5) is programmed and solved in `FreeFem++` according to the self-explanatory syntax of Listing 3.

```
/* Stationary heat equation (solver = Conjugate Gradient) */
problem laplace(u,v,solver=CG) = int2d(Th)(gamma*(dx(u)*dx(v)+dy(u)*dy(v)))
                                - int2d(Th)(f()*v)
                                + on(1,u=0.0);   // Homogeneous Dirichlet BC

/* Resolution of the problem */
laplace;

/* Display of the result */
plot(Th,u,fill=1);
```

LISTING 3. *Resolution of the Laplace equation (1.5) in `FreeFem++`.*

**Question 1.1.1.** Implement the above listings. You may want to try out different sets of physical parameters, and in particular:

- To modify the shape of the computational domain $D$;
- To select different mesh sizes;
- To choose different values for the conductivity $\gamma$ and for the heat source $f$ (e.g. space-dependent ones).

Unfortunately, the above numerical resolution yields disappointing results, as depicted on Fig. 2 (left). This is partly due to the fact that the solution $u$ to (1.1) is not much more regular than just $H^1(D)$ near the reentrant corner of $D$ (in particular, it is not of class $\mathcal{C}^2$), while the mesh used for the numerical resolution is very coarse in this region (see Fig. 1, (right)). On the other hand, it is not desirable to work out with a very fine mesh (i.e. one composed of a lot of triangles) since the size of the linear system (1.6) would then dramatically increase, and thereby the required CPU cost.

One remedy to this concern consists in *adapting* the computational mesh, so that it is selectively refined where needed (i.e. at the regions where the computed solution $u_h$ presents large variations).

**Question 1.1.2.** Implement the procedure of Listing 4 to achieve this feature.

The result is that presented on Fig. 2 (right).

```
/* Multiple resolutions of the Laplace equation,
                intertwined with mesh adaptation steps */
for (int it=0; it<maxit; it++) {
  laplace;
  Th = adaptmesh(Th,u,err=eps);
  plot(Th,u,fill=1);
  eps *= 0.5;
}
```

LISTING 4. *Iterative resolution of (1.5) using mesh adaptation.*

**Question 1.1.3.** There is one way to vizualize the results of your computations which offers more flexibility than the standard `plot` command from `FreeFem++`. This uses the `medit` software, which is integrated into `FreeFem++` under the form of `ffmedit`. A documentation of this software may be consulted at the address:

`https://www.ljll.math.upmc.fr/frey/logiciels/Docmedit.dir/index.html`

In particular, `medit` makes it possible to save pictures from your results, etc.

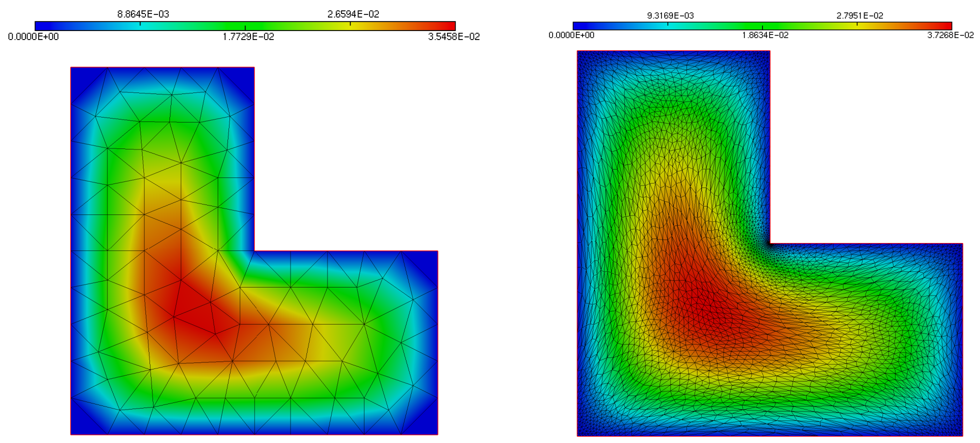


FIGURE 2. *Temperature field $u$ obtained in the resolution of (1.5) using* `FreeFem++` *with $\mathbb{P}_1$ elements (left) on the mesh of Fig. 1, (right) on an iteratively adapted mesh.*

## 1.2. **The linearized elasticity system**

The second context considered in this course is that of *linearized elasticity*, a system of equations which is widely used to model structures (e.g. beams, struts, etc.) undergoing small deformations when external stresses are at play.

In this section, the structure $\Omega$ of interest is a two-dimensional bridge, as depicted on Fig. 3 (left): $\Omega$ is clamped on a region $\Gamma_D$ of its boundary $\partial\Omega$, and surface loads $g = (0, -0.01)$ are applied on a disjoint subset $\Gamma_N \subset \partial\Omega$, modelling the impact of pedestrians or cars on the upper deck of the bridge. The remaining part $\Gamma = \partial\Omega \setminus (\overline{\Gamma_D} \cup \overline{\Gamma_N})$ is traction-free.

The considered bridge $\Omega$ is filled with a linearly elastic material characterized by its Hooke's tensor $A$:

$$\forall e \in \mathcal{S}(\mathbb{R}^2), \;\; Ae = 2\mu e + \lambda \text{tr}(e)\text{I}; \tag{1.8}$$

in the above equation, $\lambda$ and $\mu$ are the Lamé coefficients, characterizing the elastic behavior of the material. In the present $2d$ plane stress situation, these are related to the more physical quantities $E$ (the Young's modulus, appraising the resistance of the material to traction stresses) and $\nu$ (the Poisson's ratio, accounting for its resistance to shear) via the relations:

$$\lambda = \frac{E}{2(1+\nu)}, \text{ and } \mu = \frac{\nu E}{2(1+\nu)(1-\nu)}.$$

For the applications of this exercise session, the values

$$E = 1 \text{ and } \nu = \frac{1}{3} \tag{1.9}$$

may be used. The displacement $u = (u_1, u_2) : \Omega \to \mathbb{R}^2$ of the structure in these circumstances belongs to the functional space

$$H^1_{\Gamma_D}(\Omega)^2, \text{ where } H^1_{\Gamma_D}(\Omega) := \left\{ u \in H^1(\Omega)^2, \;\; u = 0 \text{ on } \Gamma_D \right\}, \tag{1.10}$$

and it is the unique solution in the latter to the linearized elasticity system:

$$\left\{ \begin{array}{cl} -\text{div}(Ae(u)) = 0 & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_D, \\ Ae(u)n = g & \text{on } \Gamma_N, \\ Ae(u)n = 0 & \text{on } \Gamma, \end{array} \right. \tag{1.11}$$

where

$$e(u) = \frac{1}{2}(\nabla u + \nabla u^T), \;\; (e(u))_{ij} = \frac{1}{2}\left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \;\; i,j = 1,2, \tag{1.12}$$

is the strain tensor (i.e a matrix with size $2 \times 2$) associated to the displacement $u$.

### 1.2.1. *Solving the linearized elasticity system using `FreeFem++`*

We first focus on the numerical resolution of (1.11) using `FreeFem++` on the geometry $\Omega$ of Fig. 3 (left).

**Question 1.2.1.** Write down the variational formulation for (1.11).

**Question 1.2.2.** Implement this formulation in `FreeFem++`; in order to create the geometry of $\Omega$ in `FreeFem++`, you may use the supplied `Subjects/EVBridge/creabridge1.edp` file.

**Question 1.2.3.** Visualize the displacement of the shape. To achieve this, you may use the `movemesh` command, as is exemplified in Listing 5; see Fig. 3 (right) for the result.

```
/* Move each vertex (x,y) of Th to (x+ux(x,y),y+uy(x,y)) */
Thn = movemesh(Th,[x+ux,y+uy]);
plot(Thn);
savemesh(Thn,"brn.mesh");
```

LISTING 5. *Sample use of the `movemesh` command.*

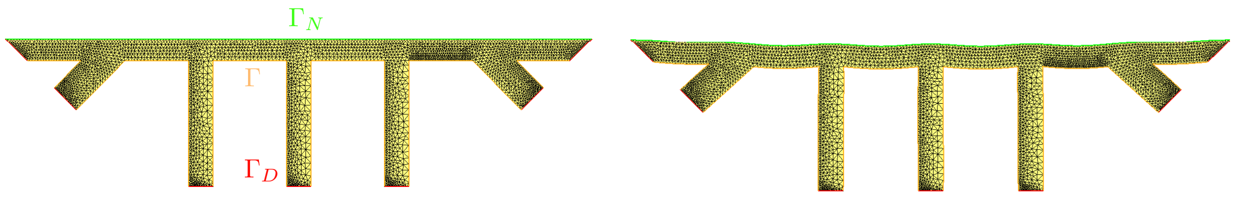


FIGURE 3. *(Left) Geometry and mesh of the bridge considered in Section 1.2; (right) deformed configuration of the bridge in the situation of (1.11).*

### 1.2.2. *A fictitious domain method: the ersatz material approximation*

In several realistic applications - such as those discussed in Section 4 -, the domain $\Omega$ may not be known via a mesh (which could be difficult to construct in practice), but by quite different means. It is then no longer possible to rely on the finite element method to solve a PDE of the form (1.11) on $\Omega$, in the same way as in Section 1.2.1.

To be more precise, let us consider the following context: a large, 'hold-all' domain $D$ is given, and it is equipped with a triangular mesh $\mathcal{T}$. The domain $\Omega$ of interest is a subdomain of $D$; no mesh of $\Omega$ is available, but $\Omega$ is known via an associated *level set function* $\phi : D \to \mathbb{R}$. By this, we mean that $\Omega$ coincides with the negative subdomain of the scalar function $\phi$, i.e.:

$$\forall x \in D, \quad \begin{cases} \phi(x) < 0 & \text{if } x \in \Omega, \\ \phi(x) = 0 & \text{if } x \in \partial\Omega, \\ \phi(x) > 0 & \text{if } x \in D \setminus \overline{\Omega}. \end{cases}$$

In practice, $\phi$ is (for instance) a Lagrange $\mathbb{P}_1$ fonction on the mesh $\mathcal{T}$ of $D$; see Fig. 4 for an illustration.

The calculation of the solution $u$ to (1.11) on $\Omega$ when $\Omega$ is given by the pair $(D, \phi)$ is achieved thanks to a so-called *fictitious domain method*: the *ersatz material* approximation: the 'void' part $D \setminus \overline{\Omega}$ is filled with a very soft material, with Hooke's law $\varepsilon A$, where $\varepsilon \ll 1$ (in practice, $\varepsilon \approx 1e^{-3}$). The system (1.11) is then approximated by:

$$(1.13) \qquad \begin{cases} -\text{div}(A_\varepsilon e(u_\varepsilon)) = 0 & \text{in } D, \\ A_\varepsilon e(u_\varepsilon) n = g & \text{on } \Gamma_N, \\ A_\varepsilon e(u_\varepsilon) n = 0 & \text{on } \Gamma, \\ u_\varepsilon = 0 & \text{on } \Gamma_D, \end{cases} \quad \text{where } A_\varepsilon(x) = \begin{cases} A & \text{if } x \in \Omega \Leftrightarrow \phi(x) < 0; \\ \varepsilon A & \text{if } x \in D \setminus \Omega \Leftrightarrow \phi(x) \geq 0, \end{cases}$$

It is indeed possible to prove that the solution $u_\varepsilon$ to (1.13) is a close approximation to that $u$ of (1.11) (see Question* 1.2.5).

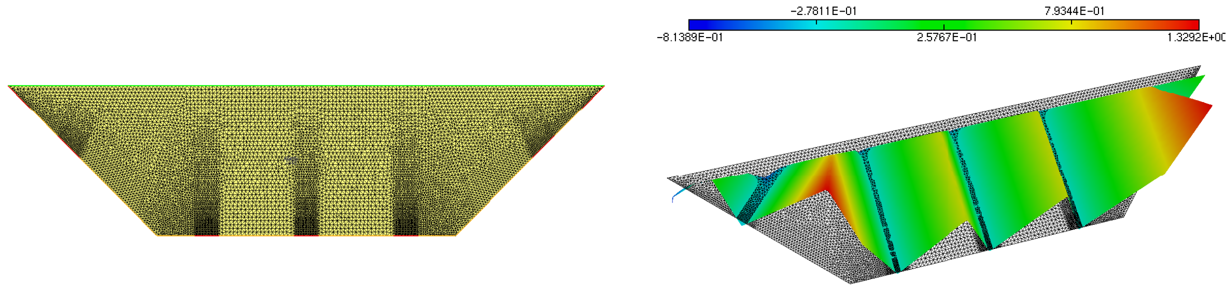


FIGURE 4. *(Left) Mesh of the hold-all domain $D$ considered in Section 1.2.2; (right) graph of one level set function $\phi : D \to \mathbb{R}$ for the bridge $\Omega$ of Section 1.2.2.*

**Question 1.2.4.** Implement the numerical resolution of the above ersatz material system (1.13); compare the resulting displacement $u_\varepsilon$ with that $u$ obtained in Section 1.2.1, possibly varying the value of the ersatz material parameter $\varepsilon$. the computational mesh of $D$ (resp. the level set function $\phi$ for $\Omega$) is given in the file `Subjects/EVBridge/box.mesh` (resp. `Subjects/EVBridge/box.sol`). Both files may be loaded in `FreeFem++` as in Listing 6, provided the supplied file `Subjects/EVBridge/tools.idp` is added to your current folder (the latter contains pre-implemented routines that do not need to be modified).

*[Hint: You may take advantage of the macro `volfrac` implemented in the file `tools.idp`, whose use is detailed in Listing 6, in order to calculate the approximate tensor $A_\varepsilon$ featured in (1.13). ]*

```
/* Load the macro file */
include "tools.idp"

/* Load the mesh */
mesh Th = readmesh("box.mesh");

/* Finite Element spaces */
fespace Vh(Th,P1);
fespace Vh0(Th,P0);

/* Parameters and functions */
real ers = 1.e^-3;
Vh phi;
Vh0 Achi;

/* Read the level set function in the external file */
readsol("box.sol",Th,phi);

/* Calculation of the P0 function Achi
   = volume fraction of material and ersatz inside each triangle */
volfrac(Achi,phi,ers);
```

LISTING 6. *Calculation of the ersatz material tensor $A_\varepsilon$ by using a personnalized macro file in Section 1.2.2.*

A glimpse of the result is displayed on Fig. 5.

**Question* 1.2.5.** The purpose of this question is to investigate the consistency of the ersatz material method on a simplified version of the model (1.11) and (1.13).

In this question, $\Omega \Subset D$ is strongly contained in the fixed hold-all domain $D$, and $f$ is a given function in $L^2(D)$ with compact support inside $\Omega$; $u$ is the unique solution in $H^1(\Omega)$ to the elliptic equation:

$$\left\{ \begin{array}{cl} -\Delta u + u = f & \text{in } \Omega, \\ \frac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega, \end{array} \right. \tag{1.14}$$

and its intended approximation $u_\varepsilon$ is the unique solution in $H^1(D)$ to the system

$$\left\{ \begin{array}{cl} -\text{div}(A_\varepsilon \nabla u_\varepsilon) + A_\varepsilon u_\varepsilon = f & \text{in } D, \\ \frac{\partial u_\varepsilon}{\partial n} = 0 & \text{on } \partial D. \end{array} \right. \quad \text{where } A_\varepsilon(x) = \left\{ \begin{array}{cl} A & \text{if } x \in \Omega, \\ \varepsilon A & \text{if } x \in D \setminus \Omega. \end{array} \right. \tag{1.15}$$

Prove that:

$$||u - u_\varepsilon||_{H^1(\Omega)} \xrightarrow{\varepsilon \to 0} 0.$$

**Remark 1.1.** *That $u_\varepsilon$, the 'ersatz material' solution to (1.15) (resp. (1.13)), be a close approximation to that $u$ of the exact system (1.14) (resp. (1.11)) depends crucially on the fact that the $\Gamma$ of $\partial\Omega$ which is not discretized in $D$ bears traction-free (i.e. homogeneous Neumann) boundary conditions. For instance, a totally different fictitious domain method would have to be designed if $\Gamma$ were equipped with homogeneous Dirichlet boundary conditions.*

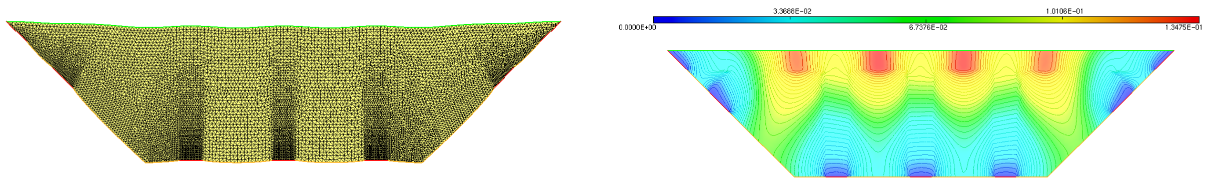


FIGURE 5. *(Left) Deformed configuration of the computational domain with the ersatz material approximation to simulate (1.11) on the implicitly-defined shape $\Omega$; (right) norm of the displacement.*

1.2.3. *An eigenvalue problem*

We now turn to another key aspect in structural mechanics, namely the analysis of vibration modes, or *eigenmodes*.

Mathematically, a real value $\lambda \in \mathbb{R}$ (resp. a $H^1_{\Gamma_D}(\Omega)^2$ function $u$ which is not identically 0) is an eigenvalue (resp. an associated eigenfunction) for the linearized elasticity system (1.11) if:

$$\left\{ \begin{array}{cl} -\text{div}(Ae(u)) = \lambda u & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_D, \\ Ae(u)n = 0 & \text{on } \Gamma_N, \\ Ae(u)n = 0 & \text{on } \Gamma. \end{array} \right. \tag{1.16}$$

It is a classical, albeit non trivial result from spectral theory (see for instance [2], Chap. 7) that these eigenvalues form a sequence $\lambda_n$ of positive, non decreasing real numbers going to $+\infty$ as $n \to \infty$. The corresponding eigenfunctions $u_n$ are usually normalized by the condition $\int_\Omega u_n^2 \, dx = 1$.

From the physical point of view, eigenvalues represent those values of the frequency at which imposed time-harmonic stresses may entail a dramatically large response of the structure, thus jeopardizing its integrity; such *resonance* phenomena have been responsible for the collapse of multiple buildings in the past, and notably of that of the Tacoma bridge; see for instance [1].

**Question 1.2.6.** Write down the (continuous) variational formulation for (1.16).

**Question 1.2.7.** Using the notations of Section 1.1, the discretized version of this formulation relies on the choice of an adequate finite element space $V_h$. Write this discretized version under the form:

$$\text{Search for } \lambda_h \in \mathbb{R} \text{ s.t. } AU = \lambda_h M, \tag{1.17}$$

where the unknown vector $U$ is given by (1.7), and $A$ and $M$ are two $N_h \times N_h$ matrices which are respectively called *stiffness* and *mass* matrices.

**Question 1.2.8.** Implement the numerical calculation of the first (i.e. the lowest) eigenpairs $(\lambda_1, u_1)$, $(\lambda_2, u_2)$,... of the system (1.16) in `FreeFem++`.

*[Hint: the calculation of the eigenvalues of partial differential operators in* `FreeFem++` *relies on a discretized variational formulation such as that established in Question 1.2.6 and 1.2.7; this involves the assembly of the above stiffness and mass matrices $A$ and $M$; see the sketch in Listing 7 for useful sample commands and Fig. 6 for a look at the result.]*

```
real sigma = 0.0; // shift value, around which eigenvalues are calculated
int nev = 5; // number of computed eigenpairs
real[int] ev(nev);
Vh2[int] [eVx,eVy](nev);
int ier;

/* Bilinear form associated to the linearized elasticity system,
                  shifted by sigma*bilinear form for the rhs of the eigenproblem */
varf elas([ux,uy],[vx,vy],solver=Crout) =  int2d(Th)( 2.0*mu*
           (dx(ux)*dx(vx) + 0.5*(dy(ux)+dx(uy))*(dy(vx)+dx(vy)) + dy(uy)*dy(vy))
                             + lambda*(dx(ux)+dy(uy))*(dx(vx)+dy(vy))
                             - sigma*(ux*vx+uy*vy))
             + on(1,ux=0.0,uy=0.0);

/* Bilinear form associated to the rhs of the eigenvalue problem */
varf b([ux,uy],[vx,vy]) = int2d(Th)(ux*vx+uy*vy);

/* Matrices associated to both biinear forms */
matrix A = elas(Vh2,Vh2,solver=Crout,factorize=1);
matrix B = elas(Vh2,Vh2,solver=Crout,factorize=1);

/* Calculation of eigenvalues and eigenvectors ;
   Beware of the misleading syntax:
   passing the first component of eV will store all its components */
ier = EigenValue(A,B,sym=true,sigma=sigma,value=ev,
                                                     vector=eVx,tol=1.e-10,maxit=0,ncv=0);
```

LISTING 7. *Sample resolution of the eigenvalue problem (1.16) using* `FreeFem++`.

## 1.3. The Stokes system

Let us now turn to the numerical resolution of the Stokes equations from fluid mechanics. The situation is that depicted on Fig. 8: a fluid with kinematic viscosity $\nu = 5e^{-3}$ is driven through a channel $\Omega$; it is entering via a portion $\Gamma_{\text{in}}$ of the boundary $\partial\Omega$ with known velocity profile

$$u(x) = u_{\text{in}}(x) \text{ on } \Gamma_{\text{in}},$$

and it exits through a disjoint region $\Gamma_{\text{out}}$, without being subject to any particular stress. The physical state of the fluid is described in terms of its velocity $u : \Omega \to \mathbb{R}^2$ and pressure $p : \Omega \to \mathbb{R}$. The couple $(u, p)$

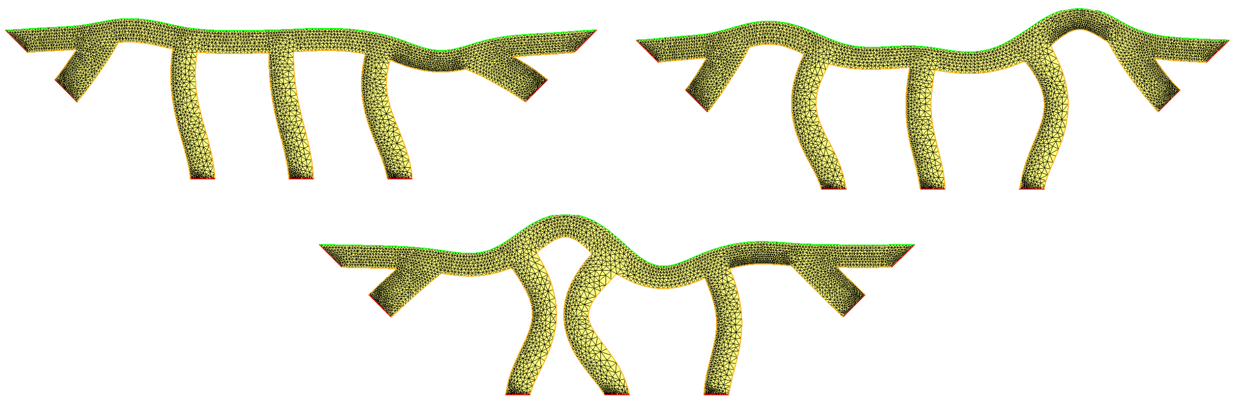


FIGURE 6. *(From left to right, top to bottom) Deformed configurations associated to the first three eigenmodes $u_1, u_2, u_3$ of the bridge considered in Section 1.2.*

satisfies the Stokes system:

$$\left\{\begin{array}{ll} -\text{div}(\sigma(u,p)) = 0 & \text{in } \Omega, \\ \text{div}(u) = 0 & \text{in } \Omega, \\ u = u_{\text{in}} & \text{on } \Gamma_{\text{in}}, \\ u = 0 & \text{on } \Gamma, \\ \sigma(u,p)n = 0 & \text{on } \Gamma_{\text{out}}, \end{array}\right. \tag{1.18}$$

where

$$\sigma(u,p) = 2\nu e(u) - p\text{I},$$

is the stress tensor inside the fluid, depending on the rate of strain tensor $e(u)$ given by (1.12).

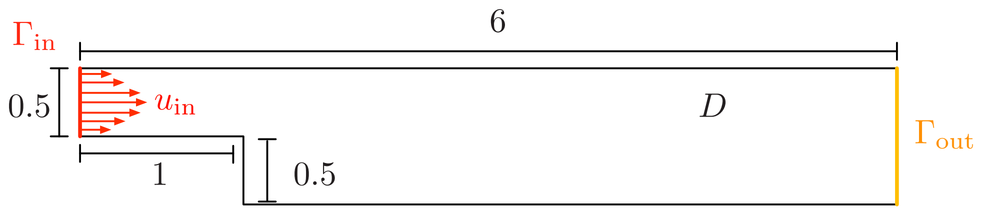


FIGURE 7. *Setting of the fluid mechanics examples of Section 1.3.*

**Question 1.3.1.** Write down the variational formulation for the solution $(u, p)$ to the Stokes equation (1.18).

**Question 1.3.2.** Implement this formulation with `FreeFem++`, using a parabolic incoming velocity profile - i.e. $u_{\text{in}}(x_1, x_2)$ is horizontal; it is a quadratic function of $x_2$ with maximum amplitude 1, which vanishes at both ends of the segment $\Gamma_{\text{in}}$.

Try out different types of Finite Element pairs for discretizing the velocity $u$ and the pressure $p$. Do all combinations work equally well?

*[Hint: Observe that the pressure field $p$ is only defined up to constants in the Stokes system (1.18), as should appear in the choice of finite element spaces involved in the variational formulation of Question 1.3.1. To overcome this difficulty in the numerical context, it is often convenient to solve the following weakly*

*compressible counterpart:*

$$\begin{cases} -\mathrm{div}(\sigma(u,p)) = 0 & \textit{in } \Omega, \\ \mathrm{div}(u) + \varepsilon p = 0 & \textit{in } \Omega, \\ u = u_{\mathrm{in}} & \textit{on } \Gamma_{\mathrm{in}}, \\ u = 0 & \textit{on } \Gamma, \\ \sigma(u,p)n = 0 & \textit{on } \Gamma_{\mathrm{out}}, \end{cases}$$

*where $\varepsilon \ll 1$ is a very small parameter (typically $\varepsilon \approx 1e^{-6}$). ]*

**Question* 1.3.3.** The stationary Navier-Stokes system is a more realistic model for the description of the motion of a fluid:

$$\begin{cases} (u \cdot \nabla)u - \mathrm{div}(\sigma(u,p)) = 0 & \text{in } \Omega, \\ \mathrm{div}(u) = 0 & \text{in } \Omega, \\ u = u_{\mathrm{in}} & \text{on } \Gamma_{\mathrm{in}}, \\ u = 0 & \text{on } \Gamma, \\ \sigma(u,p)n = 0 & \text{on } \Gamma_{\mathrm{out}}, \end{cases} \tag{1.19}$$

where

$$(u \cdot \nabla)v = \left( u_1 \frac{\partial v_1}{\partial x_1} + u_2 \frac{\partial v_1}{\partial x_2}, u_1 \frac{\partial v_2}{\partial x_1} + u_2 \frac{\partial v_2}{\partial x_2} \right).$$

The difficulty in solving numerically (1.19) comes from the presence of a non linear term. The Newton method is one method of choice to overcome this issue:

(i) Write down a (non linear) variational formulation for (1.19) under the form: search for $(u,p) \in V$ such that, for all $(v,q) \in V$,

$$a(u,v) + b(p,v) + b(u,q) + c(u,u,v) = \ell(v), \tag{1.20}$$

where $\ell$ is a linear form, $(u,v) \mapsto a(u,v)$ and $(u,p) \mapsto b(u,p)$ are bilinear forms, $(u,v,w) \mapsto c(u,v,w)$ is trilinear and $V$ is a suitable function space.

(ii) The solution $(u,p)$ to (1.19) is achieved at convergence of the series of iterations $(u^n,p^n)$ defined by:
- $(u^0,p^0)$ is the solution to the Stokes system (1.18), whose variational formulation should match (1.20) up to the trilinear term $c(u,u,v)$;
- For $n = 0, \ldots$ until convergence,

$$(u^{n+1},p^{n+1}) = (u^n,p^n) + (\delta u^n, \delta p^n),$$

where $(\delta u^n, \delta p^n)$ is the solution to the linearization of the variational formulation (1.20) at $(u^n,p^n)$:

$$\forall (v,q) \in V,\ \ a(\delta u^n, v) + b(\delta p^n, v) + b(\delta u^n, q) + c(\delta u^n, u^n, v) + c(u^n, \delta u^n, v) = \ell(v).$$

Implement this idea in `FreeFem++`; a sample of the result is represented on Fig. 8.

What difference do you observe between the velocity profiles in the Stokes and Navier-Stokes descriptions of the considered fluid?

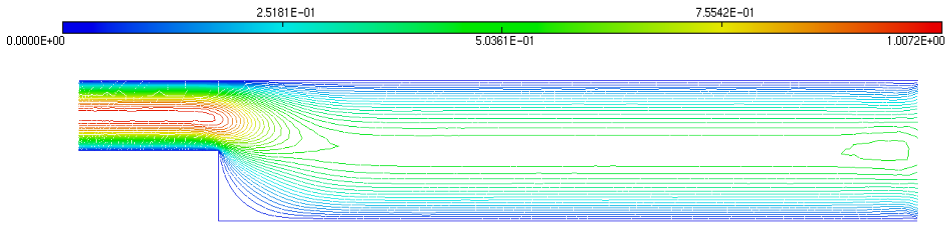
## 2. Topology optimization using density-based methods

This second section focuses on one (possibly quite unefficient with respect to the existing body of literature) implementation of density-based methods for topology optimization.

### 2.1. Generalities about the implementation of density-based methods

Let us first briefly outline the main features of the implementation of density-based topology optimization methods; these are exemplified in the context of linearized elasticity described in Section 1.2, where the minimization of a function $J(\Omega)$ is of interest.

(1) A fixed computational domain $D$ is considered, which is meshed once and for all.



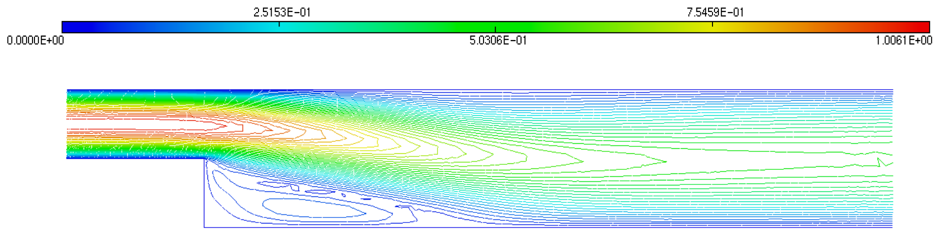


FIGURE 8. *Norm of the velocity field $u$ featured in the (top) Stokes (bottom) Navier-Stokes system in the example of Section 1.3.*

(2) Instead of the 'exact' linearized elasticity system (1.11) used to describe the physical behavior of a shape $\Omega \subset D$, we rather consider the ersatz material approximation (1.13), which is conveniently posed on $D$:

$$\left\{ \begin{array}{rl} -\text{div}(A_{\Omega,\varepsilon} e(u_{\Omega,\varepsilon})) = 0 & \text{in } D, \\ A_{\Omega,\varepsilon} e(u_{\Omega,\varepsilon}) n = g & \text{on } \Gamma_N, \\ A_{\Omega,\varepsilon} e(u_{\Omega,\varepsilon}) n = 0 & \text{on } \Gamma, \\ u_{\Omega,\varepsilon} = 0 & \text{on } \Gamma_D, \end{array} \right. \tag{2.1}$$

where the tensor $A_{\Omega,\varepsilon}$ is defined by:

$$A_{\Omega,\varepsilon} = \chi_\Omega A + (1-\chi_\Omega)\varepsilon A, \tag{2.2}$$

bringing into play the characteristic function $\chi_\Omega$ of $\Omega$:

$$\forall x \in D, \;\; \chi_\Omega(x) = \left\{ \begin{array}{cl} 1 & \text{if } x \in \Omega, \\ 0 & \text{if } x \in D \setminus \overline{\Omega}, \end{array} \right.$$

and the 'ersatz' parameter $\varepsilon \ll 1$.

(3) The essence of density-based methods is to give a meaning to the quantities $A_{\Omega,\varepsilon}$ and $u_{\Omega,\varepsilon}$ featured in (2.1) and (2.2) in the more general situation where $\chi_\Omega$ is replaced by a *density function* $h : D \to [0,1]$ - i.e. $h$ may take 'grayscale' values between 'black' (1, indicating the presence of material) and 'white' (0, indicating ersatz material). To this end, one possibility is to trade (2.2) for the tensor:

$$A_{h,\varepsilon} := \zeta(h) A + (1-\zeta(h))\varepsilon A, \tag{2.3}$$

where $\zeta : \mathbb{R} \to \mathbb{R}$ is an *interpolation function* which endows material properties to intermediate values of the density. The most common choice in the practice of the SIMP method is a power law

$$\zeta(h) = h^p; \;\; \text{(typically, } p = 3),$$

which has the effect of penalizing the presence of grayscale values for $h$; see for instance [5, 7] for other interpolation profiles, enjoying different properties.

One then defines $u_{h,\varepsilon}$ as the solution to (2.1) where the tensor $A_{\Omega,\varepsilon}$ is replaced by $A_{h,\varepsilon}$, and the considered objective function of the domain $J(\Omega)$ is thereby given a density-based counterpart $J(h)$.

(4) For various purposes, it is desirable to *filter* the density $h$ before it is used into the interpolation scheme (2.3). Here are the main two types of filters we shall be considering in the course of this exercise session (see [21] for multiple other possibilities):

- *Smoothing filter* [10]*:* $h$ is replaced by $L_\alpha h = q$, the solution in $H^1(D)$ to the following equation:

$$\begin{cases} -\alpha^2 \Delta q + q = h & \text{in } D, \\ \frac{\partial q}{\partial n} = 0 & \text{on } \partial D, \end{cases}$$

  for a small parameter $\alpha > 0$ (chosen of the order of the mesh size in practice). Doing so is a means to ensure that the effective density $q = L_\alpha h$ has a minimum regularity, and does not show too steep variations between the values 0 and 1, which could cause numerical instabilities.

- *Heaviside filter:* $h$ is replaced in (2.3) by $H_\beta h$, where

$$H_\beta h = 1 - e^{-\beta h} + e^{-\beta} h.$$

  The rationale behind this filter is that $H_\beta$ gets closer and closer to the usual Heaviside function as $\beta \to \infty$; thus, the intermediate values of the effective density will be attracted to either 0 or 1; see [17] about this idea.
  One possible drawback of the above filter is that it is strongly biased towards the value 1. The following alternative, taken from [23], does not suffer from such an inconvenience:

$$\widetilde{H_{\beta,\eta}} h = \frac{\tanh(\beta\eta) + \tanh(\beta(h - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1-\eta))},$$

  where $\eta \in (0, 1)$ tunes the bias between 0 and 1 of the filter as $\beta \to \infty$.

Of course, both types of filters may be combined (i.e. composed); it is important to note that the use of filters implies that the mappings $h \mapsto L_\alpha h$ and $h \mapsto H_\beta h$ (or $h \mapsto \widetilde{H_{\beta,\eta}} h$) have to be differentiated and that the corresponding derivatives will appear in the derivative of the optimized criterion.

## 2.2. **Topology optimization of a heat lens**

The purpose of this example is to optimize the shape of a *heat lens* - a device meant to insulate a user-defined region from an incoming heat flux. This test-case is inspired from the thermal lens problem discussed in [13], Example 5.4.2.

The considered 2d setting is that depicted on Fig. 9. The physics at play when the domain is occupied with a density $h$ is described by the following steady-state heat conduction equation:

$$\begin{cases} -\text{div}(\zeta(h)\nabla u_h) = 0 & \text{in } D, \\ \zeta(h)\frac{\partial u_h}{\partial n} = g_{\text{in}} & \text{on } \Gamma_{\text{in}}, \\ \zeta(h)\frac{\partial u_h}{\partial n} = g_{\text{out}} & \text{on } \Gamma_{\text{out}}, \\ \zeta(h)\frac{\partial u_h}{\partial n} = 0 & \text{on } \partial D \setminus (\Gamma_{\text{in}} \cup \Gamma_{\text{out}}), \end{cases} \tag{2.4}$$

where the incoming and outgoing fluxes equal respectively $g_{\text{in}} = -1.0$ and $g_{\text{out}} = 1.0$, and $\zeta(h)$ is an interpolation profile between a poor conductor with conductivity $\gamma_0 = 1$ (the phase where $h(x) = 0$) and a good conductor with conductivity $\gamma_1 = 10$ (the phase where $h(x) = 1$).

By acting on the repartition $h$ between weak and good conductors within $D$, our aim is to nullify the horizontal heat flux $\gamma_0 \nabla u_h \cdot e_1$ inside a fixed region $\omega \Subset D$ occupied with the weak conductor. In practical applications, good conductors are more expensive than weak ones, and so a constraint on the volume $\text{Vol}(h) = \int_D h\,dx$ of the phase filled with good conductor has to be added. The considered topology optimization problem then reads:

$$\min_h \{F(h) + \ell \text{Vol}(h)\}, \text{ where } F(h) = \int_D \chi_\omega \gamma_0^2 \left|\frac{\partial u_h}{\partial x_1}\right|^2 dx, \tag{2.5}$$

and $\ell$ is a fixed weight (or Lagrange multiplier); e.g. $\ell = 10$ for this example.

**Question 2.2.1.** Write down the variational formulation for (2.4), and implement it, for instance in the case where the density $h^0$ identically takes the value 0.5, except inside $\omega$ where it is set to 0.

*[Hint: Take care of the fact that only homogeneous Neumann boundary conditions are applied in (2.4), so that the field $u_h$ is defined up to a constant. How do you take this fact into account in `FreeFem++`?]*

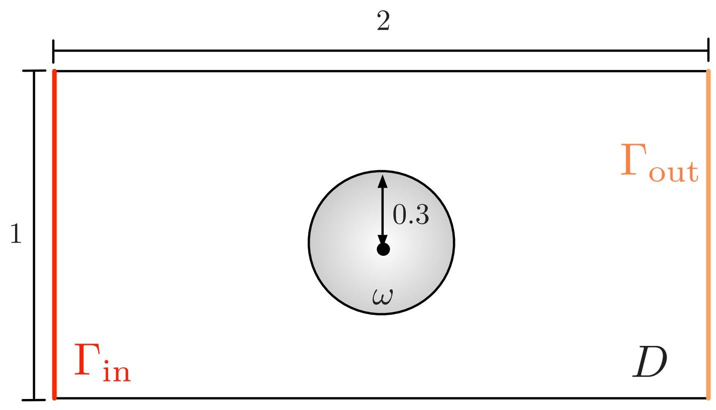


FIGURE 9. *Setting of the thermal lens test-case of Section 2.2.*

**Question 2.2.2.** Calculate the derivative of the objective function $F(h)$ given in (2.5), for instance by using the method of Céa.

**Question 2.2.3.** Implement a basic, steepest-descent density-based topology optimization algorithm to solve the problem (2.5). The device of your algorithm notably implies several choices as regards:

- The interpolation profile $\zeta(h)$ between the conductivity values $\gamma_0$ and $\gamma_1$ of the poor and good conductors, respectively;
- The use of one or several of the density filters described in Section 2.1;
- The use of a change of inner products to infer a descent direction from the derivative calculated in Question 2.2.2 (see Section 3.1 below, where this feature is explained in the context of geometric optimization methods);
- The particular gradient-based optimization algorithm based on the result of Question 2.2.2 (use of a line search procedure for finding a good descent step, etc.).

A proposition of correction lies in the file `Solutions/DensityThermal/simpheat.edp`; see Fig. 10 for an illustration of the result.

**Question* 2.2.4.** Of great physical interest for applications are also devices achieving the opposite physical behavior to that of thermal lenses, namely so-called *field concentrators*, whose aim is to *maximize* the horizontal heat flux $F(h)$ through the region $\omega$, while still imposing a constraint on the volume of the phase filled with good conductor.

Adapt the previous questions to this new setting.

**Question* 2.2.5.** In the practice of the SIMP method, people generally use different (still gradient-based) optimization algorithms from the steepest-descent strategy advocated in Question 2.2.3. Here are two examples of popular strategies that you may try to implement:

- The Optimality Criteria (OC) method, which allows to impose exactly a desired volume constraint (instead of a mere penalization of the objective in (2.5)) by relying on the optimality conditions of constrained optimization programs; see for instance [8] §1.2.2 about this strategy.
- The Method of Moving Asymptotes (MMA) [22], a method from constrained optimization relying on convex approximations of the objective and constraint functions which is especially tailored to density-based topology optimization; see also [8] §1.2.3.

You may also consider to try out different optimization algorithms such as the nonlinear conjugate gradient method; see [19], Section 8.1 about the implementation of such algorithms in `FreeFem++`

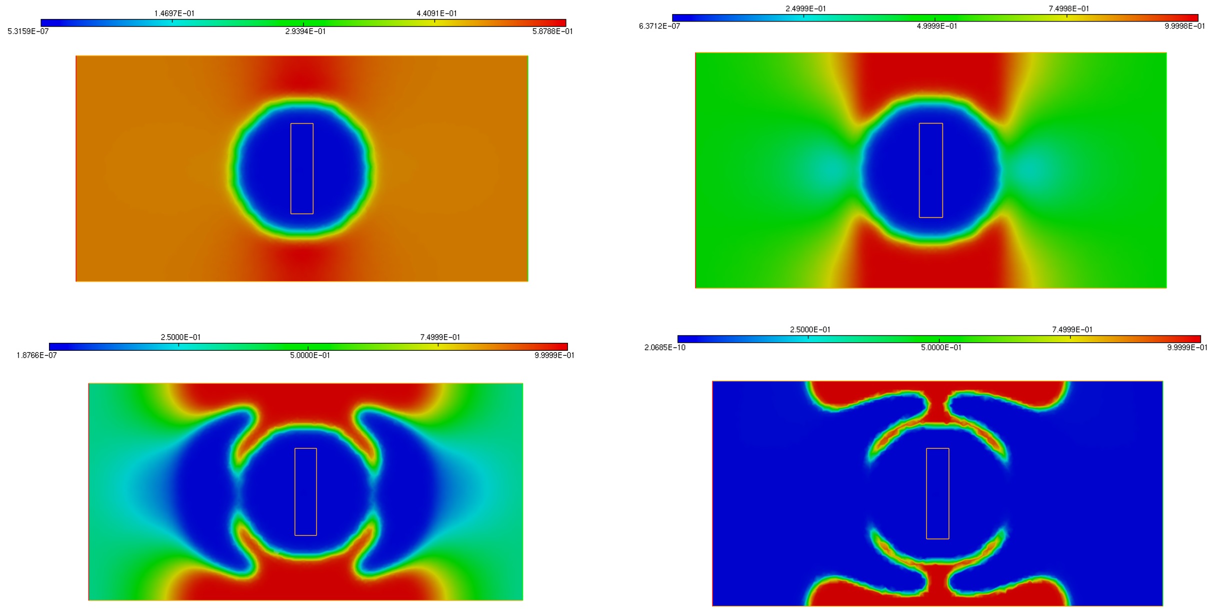


FIGURE 10. *Iterations* 0, 10, 75 *and* 200 *of the heat lens optimization test-case of Section 2.2.*

## 2.3. **The classical cantilever benchmark in linearized elasticity**

We now turn to the first instance of the perhaps most classical example in shape and topology optimization, namely the so-called *cantilever* example. The physical setting is illustrated in Fig. 11: a beam occupies a rectangular hold-all domain $D$ with size $2 \times 1$; $D$ is clamped on its left-hand side $\Gamma_D$ and vertical traction loads $g = (0, -1)$ are applied on a part $\Gamma_N$ of their right-hand side. Body forces within $D$ are neglected.

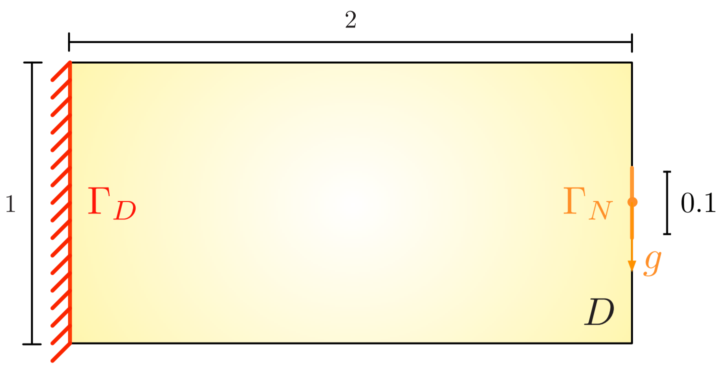


FIGURE 11. *Setting of the cantilever test case of Section 2.3.*

In the context of density-based topology optimization, the problem at stake consists in searching for the repartition of an elastic material with Hooke's tensor $A$ defined by (1.8) and (1.9), and of the corresponding ersatz material $\varepsilon A$, with $\varepsilon = 1e^{-3}$ within $D$, as described in Section 2.1.

**Question 2.3.1.** Write down the linearized elasticity system for the displacement $u_h \in H^1_{\Gamma_D}(D)^2$ characterizing the behavior of $D$ when the repartition between both phases is characterized by the density function $h : D \to (0, 1)$. Write down the corresponding variational formulation for $u_h$.

Our aim is to minimize the *compliance* of the structure in the above situation; the latter may be equivalently understood as the elastic energy stored in the structure $D$, or as the work of external loads acting on $D$:

$$C(h) = \int_D A_h e(u_h) : e(u_h)\, dx = \int_{\Gamma_N} g \cdot u_h \, ds.$$

A constraint on the volume $\text{Vol}(h)$ of the material phase is added, so that we solve the problem

$$\min_h \{C(h) + \ell \text{Vol}(h)\} \tag{2.6}$$

for a fixed Lagrange multiplier $\ell$ (in the application of this section, you may use $\ell = 10$).

**Question 2.3.2.** Calculate the derivative of the mapping $h \mapsto C(h)$, for instance by using the method of Céa.

**Question 2.3.3.** Implement a steepest-descent algorithm for the numerical resolution of (2.6), in the spirit of that of Question 2.2.3.

Again, you may be interested in try out different mesh sizes, different filters, and different initial designs. What do you observe?

The formulation (2.6) of the considered optimization problem is a little unsatisfactory, since it does not account for a constraint on the volume of used material, so to speak, but rather for a penalization of the minimized objective (the compliance) by this volume constraint. In particular, a little tuning of the Lagrange multiplier $\ell$ is in order so as to 'guess' which value will yield an optimized design with the desired volume.

**Question 2.3.4.** The purpose of this question is to implement an augmented Lagrangian algorithm (see e.g. [20], Chap. 17) dedicated to the resolution of the equality constrained optimization program

$$\min_h C(h) \text{ s.t. } \text{Vol}(h) = V_T, \tag{2.7}$$

where $V_T$ is a user-defined desired volume target.

Roughly speaking, the augmented Lagrangian algorithm transforms the constrained optimization problem (2.7) into a sequence of unconstrained problems of the form (2.6) (hereafter indexed with the superscript $^n$), in which the weighting parameter is consistently updated.

More precisely, let us define the augmented Lagrangian functional $\mathcal{L}(h, \ell, b)$ by:

$$\mathcal{L}(h, \ell, b) = J(h) - \ell(\text{Vol}(h) - V_T) + \frac{b}{2}(\text{Vol}(h) - V_T)^2;$$

intuitively, $\ell$ is intended as a closer and closer estimate of the Lagrange multiplier for the equality constraint in the optimality conditions of (2.7) while $b$ is a penalty parameter, initialized with a 'low' value, increasing so as to impose steadily this constraint in (2.7).

The augmented Lagrangian algorithm then alternates between the update of the coefficients $\ell^n$ and $b^n$ and the search for a minimizer $h^n$ to $h \mapsto \mathcal{L}(h, \ell^n, b^n)$ for fixed values $\ell^n$ and $b^n$. In practice, since the latter search is computationally expensive, the following practical version is used:

- **Initialization:** Start with an initial design $h^0$ and initial values $\ell^0$ and $b^0$.
- **For** $n = 0, \ldots$ **until convergence:**
  - Find a descent direction $\delta h^n$ for $h \mapsto \mathcal{L}(h, \ell^n, b^n)$;
  - Find a small enough time-step $\tau^n$ so that $\mathcal{L}(h^n + \tau^n \delta h^n, \ell^n, b^n) < \mathcal{L}(h^n, \ell^n, b^n)$ and set $h^{n+1} = h^n + \tau^n \delta h^n$;
  - Update the coefficients $\ell^n$ and $b^n$ according to the rule:

$$\ell^{n+1} = \ell^n - b^n(\text{Vol}(h^n) - V_T), \text{ and } b^{n+1} = \begin{cases} \alpha b^n & \text{if } b^n < b_{\max}, \\ b^n & \text{otherwise,} \end{cases}$$

    where $\alpha > 1$ and $b_{\max}$ are user-defined parameters.

Implement the augmented Lagrangian algorithm in the context of the cantilever test-case.

A proposition of correction lies in the files `Solutions/DensityCantilever/simpcanti.edp` and `Solutions/DensityCantilever/simpcantiauglag.edp`; see Fig. 12 for an illustration of the result.

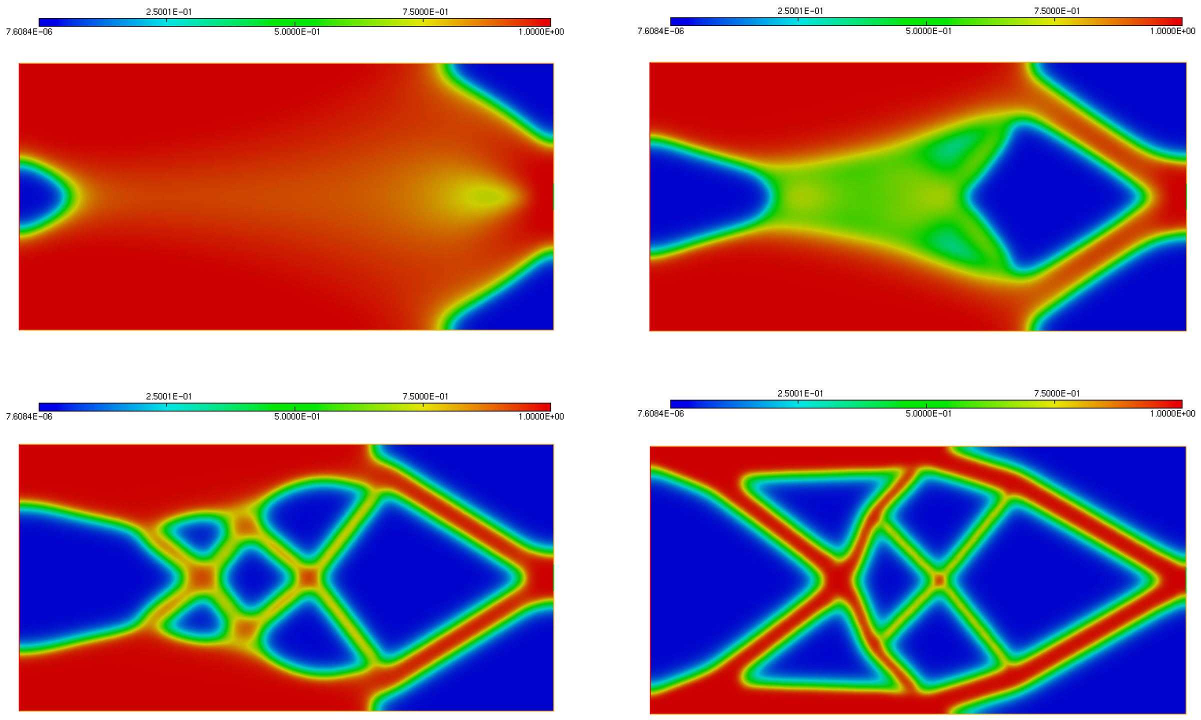


Figure 12. *(From left to right, top to bottom) Iterations* 20, 35, 50 *and* 200 *in the topology optimization example of the cantilever beam of Section 2.3.*

### 2.4. **Topology optimization of a compliant mechanism**

One very important application field of topology optimization is concerned with the design of efficient *actuators*, or *compliant mechanisms*, namely mechanisms adopting a prescribed behavior (in terms of their elastic displacement, for example) when they are submitted to a given input force.

In this section, we consider the optimal design of a displacement inverter, as illustrated on Fig. 13. In a square-shaped hold-all domain $D$, the considered structures $\Omega \subset D$ are clamped near the upper and lower regions $\Gamma_D$ of their left-hand side, and a known force $g = (100, 0)$ is applied on the middle $\Gamma_N$ of their left-hand side. A region $\Gamma_S$ of their right-hand side is connected to a spring with stiffness constant $k_s = 0.1$, which opposes the motion of the latter part. In this setting, the displacement $u_\Omega \in H^1_{\Gamma_D}(\Omega)^2$ of a shape $\Omega \subset D$ satisfies the following linearized elasticity system:

$$\left\{ \begin{array}{cl} -\text{div}(Ae(u)) = 0 & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_D, \\ Ae(u)n = g & \text{on } \Gamma_N, \\ Ae(u)n + k_s u = 0 & \text{on } \Gamma_S, \\ Ae(u)n = 0 & \text{on } \Gamma. \end{array} \right. \tag{2.8}$$

In this context, we aim to maximize the negative horizontal displacement of $\Omega$ on the output area $\Gamma_S$, while minimizing the positive horizontal displacement in the input region $\Gamma_N$. To achieve this goal, we minimize the functional $A(\Omega)$ defined by:

$$A(\Omega) = \alpha_{\text{in}} \int_{\Gamma_N} u_\Omega \cdot e_1 \, ds + \alpha_{\text{out}} \int_{\Gamma_S} u_\Omega \cdot e_1 \, ds,$$

under a constraint on the volume of shapes. Here $\alpha_{\text{in}}$ and $\alpha_{\text{out}}$ are two weighting constants (for instance, $\alpha_{\text{in}} = 1$ and $\alpha_{\text{out}} = 2$).

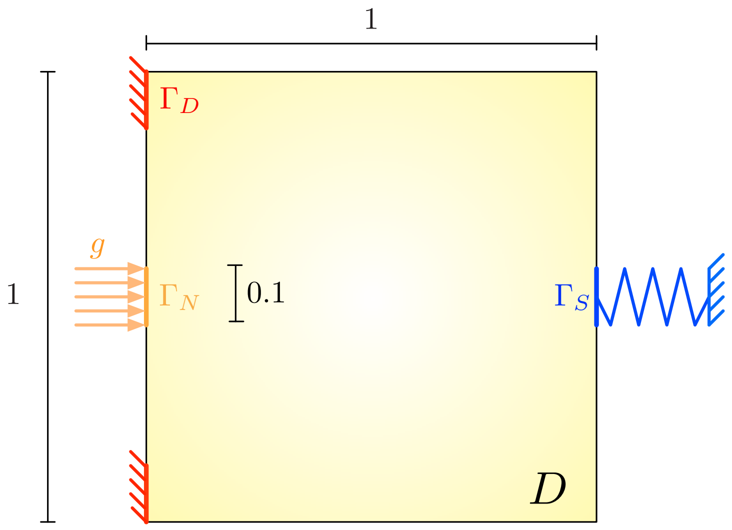


FIGURE 13. *Mechanical setting of the displacement inverter mechanism test-case of Section 2.4.*

**Question 2.4.1.** Write down the density-based formulation corresponding to the shape optimization problem

$$\min_{\Omega} A(\Omega) \text{ s.t. } \operatorname{Vol}(\Omega) = V_T,$$

where $V_T$ is a user-defined target volume.

**Question 2.4.2.** Calculate the derivative of the objective function $A(h)$ involved in this density-based topology optimization problem, for instance by using the method of Céa.

**Question 2.4.3.** Implement a steepest-descent algorithm for the resolution of this topology optimization problem, taking stock of the knowledge obtained from the work in Sections 2.2 and 2.3.

One possible correction is proposed in the file `Solutions/DensityInverter/simpinvert.edp`; see Figs. 14 and 15 for an illustration of the result.

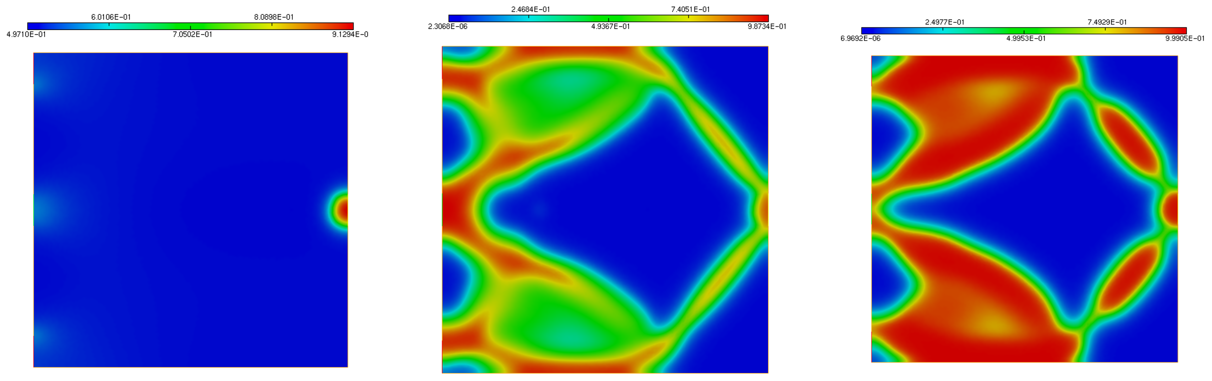


FIGURE 14. *(From left to right) Iterations* 0, 50 *and* 200 *in the topology optimization example of the inverter mechanism of Section 2.4.*

## 3. GEOMETRIC OPTIMIZATION USING MESH MOVEMENT

This section describes the implementation of a typical geometric optimization method.

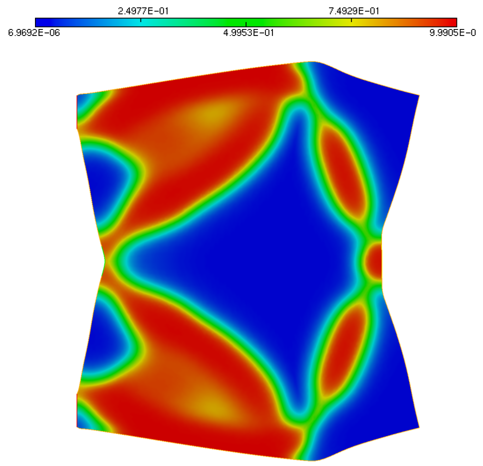


FIGURE 15. *Deformed configuration of the optimized inverter mechanism of Section 2.4.*

## 3.1. **Basics about geometric optimization methods**

We first sketch the main features of geometric optimization methods, still in the structural mechanics context of Section 1.2: an objective function $J(\Omega)$ is minimized (without constraints, for simplicity), which depends on the domain $\Omega$ via the solution $u_\Omega$ to the linearized elasticity system (1.11).

(1) The optimized shape $\Omega$ is explicitly meshed at each iteration of the process. More precisely, a sequence of shapes $\Omega^n$, $n = 0, \dots$ is produced, and each $\Omega^n$ is equipped with a triangular mesh $\mathcal{T}^n$. Therefore, the resolution of a PDE of the form (1.11) on the optimized shape may be achieved straightforwardly by using the finite element method, along the lines of Section 1.2.1.

(2) A *descent direction* for $J(\Omega)$ is found as a vector field $\theta : \Omega \to \mathbb{R}^2$ such that, equivalently

$$J((\mathrm{Id} + \tau\theta)(\Omega)) < J(\Omega) \text{ for } \tau > 0 \text{ small enough, or } J'(\Omega)(\theta) < 0.$$

(3) A naive attempt to calculate such a descent direction relies on the general structure of the shape derivative of $J(\Omega)$:

$$J'(\Omega)(\theta) = \int_\Gamma v_\Omega \, \theta \cdot n \, ds, \tag{3.1}$$

where $\Gamma$ is the optimized region of the boundary of $\Omega$, and $v_\Omega : \Gamma \to \mathbb{R}$ is a scalar function depending on $u_\Omega$ and the optimized criterion $J(\Omega)$ (for instance via an adjoint state). Indeed, from (3.1), a descent direction for $J(\Omega)$ is revealed as:

$$\theta = -v_\Omega n. \tag{3.2}$$

In other terms, $\theta$ is the (negative) gradient associated to the differential $\theta \mapsto J'(\Omega)(\theta)$ via the $L^2(\partial\Omega)$ inner product. Unfortunately, this choice often proves awkward for a number of reasons, including the lack of regularity of the field $v_\Omega$ (which may cause spurious oscillations of the boundary $\Gamma$ in the course of the optimization process).

(4) A common trick consists in changing inner products when it comes to finding a descent direction from $J'(\Omega)(\theta)$. More precisely, a Hilbert space $\mathcal{H}$ of admissible deformations is chosen, as well as a coercive bilinear form $a(\cdot,\cdot)$ on $\mathcal{H}$; the following problem is then solved:

$$\text{Find } \widetilde{v} \in \mathcal{H} \text{ s.t. } \forall w \in \mathcal{H}, \;\; a(\widetilde{v}, w) = J'(\Omega)(w),$$

and it is easily seen that $\theta = -\widetilde{v}$ is a descent direction for $J(\Omega)$. Several choices are possible as for the pair $(\mathcal{H}, a(\cdot,\cdot))$; for instance (see also [11, 15]):

- The choice $\mathcal{H} = H^1(\Omega)^2$ and $a(w,z) = \int_\Omega (\nabla w : \nabla z + w \cdot z)\, dx$ yields a shape gradient which is defined on $\Omega$ as a whole, and which is more regular than the naive choice (3.2);

- The choice $\mathcal{H} = H^1(\Omega)^2$ and $a(w,z) = \int_\Omega (Ae(w) : e(z) + wz)\, dx$ yields a shape gradient which 'resembles' an elastic displacement;

other requirements on the sought descent direction (e.g. that it should vanish on some fixed region of $\partial\Omega$) may be encoded likewise in the choice of $\mathcal{H}$ and $a$.

(5) Once a descent direction $\theta^n$ is obtained at some stage $\Omega^n$, the new shape $\Omega^{n+1}$ is obtained by:

$$\Omega^{n+1} := (\mathrm{Id} + \tau^n \theta^n)(\Omega^n)$$

for a suitable choice of a time step $\tau^n$ (resulting e.g. from a line search strategy).

From the numerical viewpoint, the mesh $\mathcal{T}^{n+1}$ of $\Omega^{n+1}$ may be obtained from that $\mathcal{T}^n$ of $\Omega^n$ by moving each vertex $x_i^n$, $i = 1, ..., N$ along the vector field $\theta^n$, formally:

$$x_i^{n+1} = x_i^n + \tau^n \theta^n(x_i^n), \tag{3.3}$$

while the connectivity of the mesh (the connections between vertices) remain unchanged.

(6) In practice, the greatest stake of geometric optimization methods has to do with the procedure used to update the shape. Indeed, such a basic rule as (3.3) is bound to result in an invalid mesh, i.e. a mesh containing inverted, or overlapping triangles.

For this reason, one needs to devise more 'clever' mesh update techniques, which alleviate this issue, or at least postpone its emergence. This may involve

- A periodic remeshing of the shape, i.e. an improvement of the quality of the computational mesh thanks to mesh operations. In `FreeFem++`, such an operation is possible via the `adaptmesh` command which we have already encountered in Section 1.1; see Listing 4.
- The use of a descent direction produced by an 'adapted' inner product; see the point (4) above.

## 3.2. The cantilever example

We revisit the 2d cantilever example of Section 2.3 with geometric shape optimization methods; see the educational article [4] for a similar example. The considered shapes $\Omega$ are clamped on a region $\Gamma_D$ of their boundary, and a vertical load $g = (0, -1) \in L^2(\Gamma_N)^d$ is applied on a disjoint subset $\Gamma_N \subset \partial\Omega$. Neither $\Gamma_D$ nor $\Gamma_N$ is subject to optimization, which leaves the remaining, 'traction-free' region of $\partial\Omega$ as the unique region subject to optimization; see Fig. 11 for an illustration.

In these circumstances, the elastic displacement $u_\Omega$ of $\Omega$ belongs to the space $H^1_{\Gamma_D}(\Omega)^2$, where we recall that $H^1_{\Gamma_D}(\Omega)$ is defined by (1.10); it is the unique solution in this space to the system of linear elasticity

$$\begin{cases} -\mathrm{div}(Ae(u_\Omega)) = 0 & \text{in } \Omega, \\ Ae(u_\Omega)n = g & \text{on } \Gamma_N, \\ Ae(u_\Omega)n = 0 & \text{on } \Gamma, \\ u_\Omega = 0 & \text{on } \Gamma_D. \end{cases}$$

As in Section 2.3, our aim is to minimize a weighted sum of the compliance of the beam and its volume:

$$\min_\Omega \{C(\Omega) + \ell \mathrm{Vol}(\Omega)\}, \text{ where } C(\Omega) = \int_\Omega Ae(u_\Omega) : e(u_\Omega)\, dx, \;\; \mathrm{Vol}(\Omega) = \int_\Omega dx, \tag{3.4}$$

and $\ell$ is a fixed Lagrange multiplier (in this example, one may take $\ell = 5$).

**Question 3.2.1.** By using either a rigorous calculation or the formal method of Céa, prove that the shape derivative of the compliance $C(\Omega)$ reads:

$$\forall \theta, \;\; C'(\Omega)(\theta) = -\int_\Gamma Ae(u_\Omega) : e(u_\Omega)\, \theta \cdot n\, ds.$$

*[Hint: Remember that the optimized region $\Gamma \subset \partial\Omega$ bears homogeneous Neumann boundary conditions.]*

**Question 3.2.2.** Implement a steepest-descent algorithm for the resolution of the program (3.4). You may start from the initial geometry constructed in the supplied file `Subjects/GeomCanti/cantigeom_ini.edp`.

In particular, you may want to pay close attention to:

- The needed remeshing needed to maintain a good quality of the computational mesh;
- The strategy for deriving a descent direction from the knowledge of the shape derivative calculated in Question 3.2.1; in particular, it is interesting to investigate different possibilities as regards the choice of an inner product; see Point (4) in Section 3.1.

You may also want to try out different initial designs, different values for the Lagrange multiplier $\ell$, etc.

*[Remark: you may notice a parasitic behavior of the descent direction near the corners marking the interface between the regions $\Gamma_D$ and $\Gamma$; this is because the shape gradient is ill-defined at these points, whose movements ought better be prevented.]*

**Question 3.2.3.** As was observed in Section 2.3, the formulation (3.4) of the considered optimization problem is a little clumsy, since the weighting parameter $\ell$ has no physical meaning, and is in particular difficult to relate to the volume of the resulting optimized design. It is therefore tempting to try and solve the constrained optimization program:

$$\min_{\Omega} C(\Omega) \text{ s.t. } \mathrm{Vol}(\Omega) = V_T,$$

where $V_T$ is a volume target.

Implement an augmented Lagrangian algorithm to achieve this purpose, by relying on the sketch provided in Question 2.3.4, which applies mutatis mutandis to the present situation.

A proposition of correction lies in the file `Solutions/GeomCanti/cantigeom.edp` and Fig. 16 depicts a glimpse of the result.

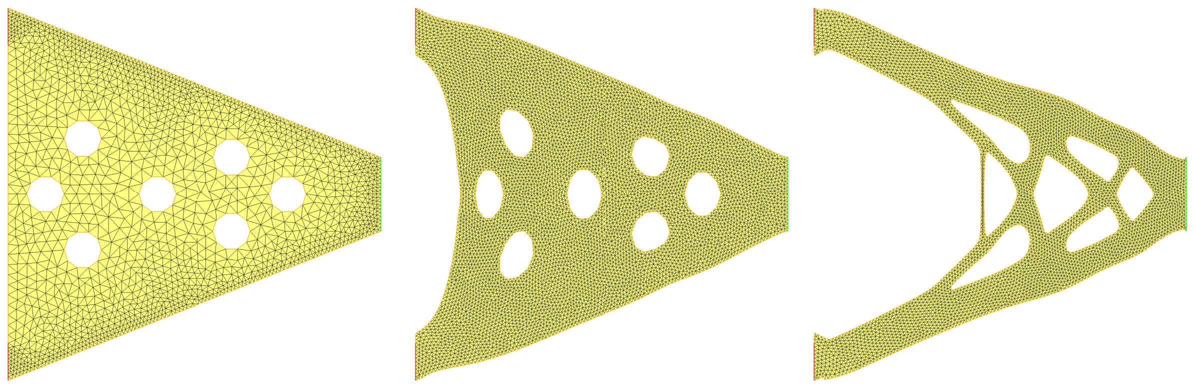


FIGURE 16. *Iterations 0, 40 and 200 of the cantilever geometric optimization test-case of Section 3.2.*

## 3.3. Geometric optimization for an eigenvalue problem

As we have already seen in Section 1.2.3, the excitation of an eigenmode of a structure may dramatically jeopardize its safety; hence, it is sometimes desirable to reinforce a given structure (with as little additionnal material as possible) in such a way to increase its fundamental frequency.

The present example is inspired by [8]: here, shapes $\Omega$ are 2d beams clamped on their left-hand side $\Gamma_D$, which have to be optimized so as to increase their lower eigenfrequency. So as to prevent trivial optimization results (such as the void domain!) incompatible with the anticipated use of the structure, we artificially assume that the beam has inhomogeneous density $\rho$ (and that the outer frame of the beam is much denser than the interior region).

More rigorously, the optimization problem reads

$$\min_{\Omega} \left\{ -\lambda_1(\Omega) + \ell \mathrm{Vol}(\Omega) \right\}, \tag{3.5}$$

where $\ell$ is a fixed Lagrange multiplier (here, $\ell = 1e^{-5}$) and $\lambda_1(\Omega)$ is the lowest eigenvalue of the linearized elasticity operator. In other terms, $\lambda_1(\Omega)$ is the lowest positive value $\lambda$ such that there exists a function $u_\Omega \in H^1_{\Gamma_D}(\Omega)^2$ which does not vanish everywhere, satisfying

$$\begin{cases} -\mathrm{div}(Ae(u_\Omega)) = \lambda \rho u_\Omega & \text{in } \Omega, \\ u_\Omega = 0 & \text{on } \Gamma_D, \\ Ae(u_\Omega)n = 0 & \text{on } \Gamma. \end{cases} \tag{3.6}$$

As usual, $u_\Omega$ is normalized by the condition

$$\int_\Omega \rho u_\Omega^2 \, dx = 1. \tag{3.7}$$

**Question 3.3.1.** By using, for instance, the formal method of Céa, and admitting that the fundamental mode, defined by (3.6) and (3.7), is single, prove that the shape derivative of $\lambda_1(\Omega)$ is

$$\forall \theta, \ \lambda_1'(\Omega)(\theta) = \int_\Gamma \left( Ae(u_\Omega) : e(u_\Omega) - \rho u_\Omega^2 \right) \, \theta \cdot n \, ds.$$

**Question 3.3.2.** Implement a steepest-descent algorithm for the resolution of the optimization problem (3.5). You may use the domain constructed in the supplied file `Subjects/GeomEigen/eigengeom_ini.edp` as initial design; the density function $\rho$ featured in (3.6) is defined in the same file.

A proposition of correction lies in the files `Solutions/GeomEigen/eigengeom.edp` and Fig. 24 depicts one possible result.

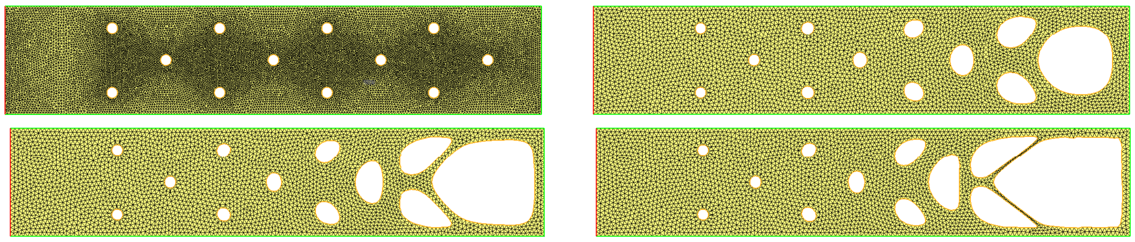


FIGURE 17. *Iterations* 0, 50, 70 *and* 200 *of the eigenvalue optimization test-case of Section 3.3.*

**Question* 3.3.3.** You may be interested in using a similar strategy to reinforce the bridge of Fig. 3 (left), i.e. to increase the value of its fundamental frequency.

## 3.4. An example in fluid mechanics

In this section, we slip into a fairly different context, that of fluid mechanics. A Newtonian fluid with kinematic viscosity $\nu$ is entering a pipe $\Omega$ via an inlet $\Gamma_{\text{in}}$ with known (parabolic) velocity profile $u_{\text{in}}$. The fluid is conveyed by the pipe, from where it exits via the outlet $\Gamma_{\text{out}}$; see Fig. 18 for a schematics of the situation. As in Section 1.3, the physical state of the fluid is described by its velocity $u_\Omega \in H^1(\Omega)^2$ and pressure $p_\Omega \in L^2(\Omega)/\mathbb{R}$, which are the solutions to the Stokes system:

$$\left\{ \begin{array}{cl} -\text{div}(\sigma(u,p)) = 0 & \text{in } \Omega, \\ \text{div}(u) = 0 & \text{in } \Omega, \\ u = u_{\text{in}} & \text{on } \Gamma_{\text{in}}, \\ u = 0 & \text{on } \Gamma, \\ \sigma(u,p)n = 0 & \text{on } \Gamma_{\text{out}}, \end{array} \right. \tag{3.8}$$

where the viscous stress tensor reads:

$$\sigma(u,p) = 2\nu e(u) - p\text{I}.$$

In this setting, we are interested in minimizing the energy dissipated by the pipe under viscous effects,

$$D(\Omega) = 2\nu \int_\Omega e(u_\Omega) : e(u_\Omega) \, dx,$$

under a constraint on the volume $\text{Vol}(\Omega)$ of $\Omega$. As in previous examples, this constraint is simply aggregated to the objective function, so that the problem rewrites as the unconstrained minimization problem

$$\min_\Omega \left\{ D(\Omega) + \ell \text{Vol}(\Omega) \right\}, \tag{3.9}$$

where $\ell$ is a fixed Lagrange multiplier (for this example, you may take $\ell = 5$).

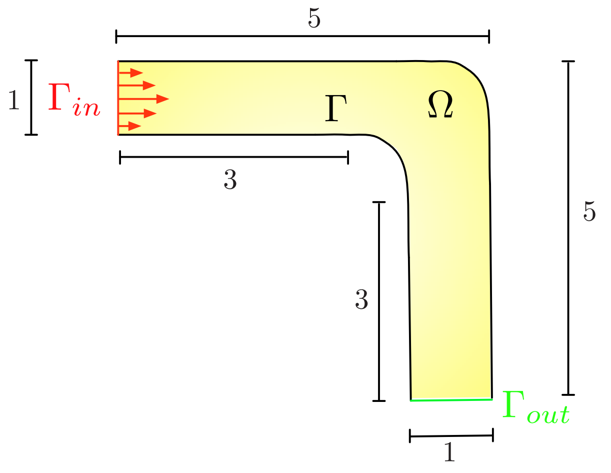


FIGURE 18. *Setting of the pipe geometric optimization test-case of Section 3.4.*

**Question 3.4.1.** By using either a rigorous calculation, or the method of Céa, prove that the shape derivative of the energy dissipation $D(\Omega)$ reads:

$$D'(\Omega)(\theta) = -2\nu \int_\Gamma e(u_\Omega) : e(u_\Omega) \theta \cdot n \, ds.$$

*[Hint: In the practice of the method of Céa, beware of the fact that the optimized boundary in this case is equipped with homogeneous Dirichlet boundary conditions.]*

**Question 3.4.2.** Implement a steepest-descent geometric optimization algorithm for the numerical resolution of (3.9). To this end, you may use the initial design constructed in the supplied file `Subjects/GeomFluids/pipe_ini.edp`, as well as the input velocity profile $u_{\text{in}}$ defined in there.

A proposition of correction is given in the file `Solutions/GeomFluids/pipe.edp`; see Fig. 19 for a sample result.

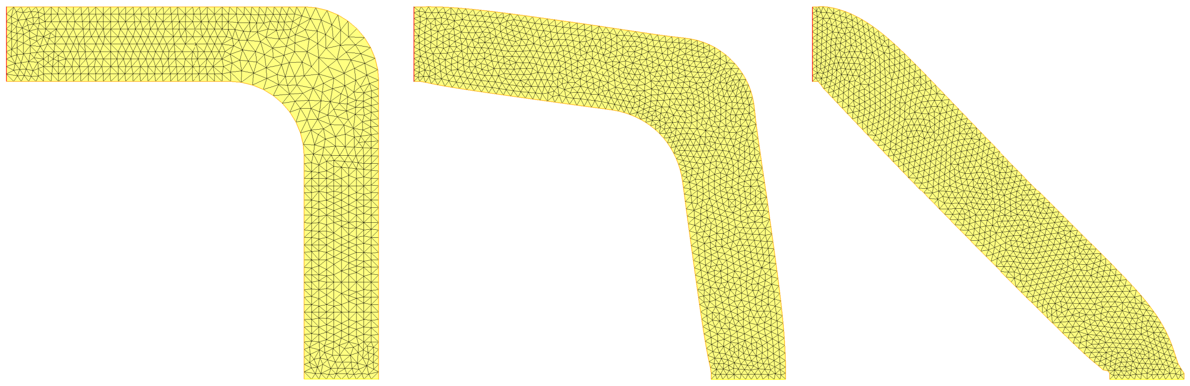


FIGURE 19. *Iterations* 0, 40 *and* 200 *in the pipe optimization test-case of Section 3.4.*

**Question 3.4.3.** You may try another example with the exact same algorithm, such as the double branch structure which consists in solving the exact same optimization program (3.9), in the physical situation of Fig. 20.

**Remark 3.1.** *The interested reader by the context of fluid mechanics may consult the educational article* [14] *in which several similar (and more elaborated) examples are discussed.*

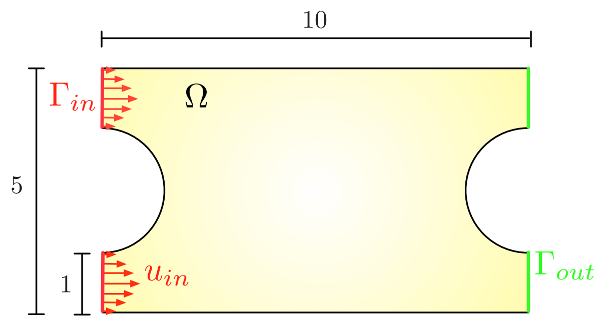


FIGURE 20. *Setting of the double branch geometric optimization test-case of Section 3.4.*

## 4. Shape and topology optimization using the level set method

We finally discuss the implementation of shape and topology optimization algorithms relying on the level set method.

### 4.1. **Practice of the level set method in the context of linearized elasticity**

Let us start our encounter with level set methods with a short summary of their main features, in the context of the minimization of a function $J(\Omega)$ of the domain involving the solution $u_\Omega$ to the linearized elasticity equations on $\Omega$:

(1) A fixed computational domain $D$ is defined, and it is equipped with a fixed mesh $\mathcal{T}$.
(2) Any considered shape $\Omega \subset D$ is represented by means of a level set function $\phi : D \to \mathbb{R}$ such that:

$$\left\{ \begin{array}{ll} \phi(x) < 0 & \text{if } x \in \Omega, \\ \phi(x) = 0 & \text{if } x \in \partial\Omega, \\ \phi(x) > 0 & \text{if } x \in D \setminus \overline{\Omega}; \end{array} \right. \tag{4.1}$$

see Fig. 4 for an illustration.
(3) In practice, only the level set function $\phi : D \to \mathbb{R}$ is known (not the domain $\Omega$); it is discretized, for instance, as a $\mathbb{P}_1$ finite element function on $\mathcal{T}$.
(4) The calculation of geometric quantities attached to $\Omega$, such as the normal vector field $n(x)$ or the mean curvature $\kappa(x)$ may be achieved by discretizing the continuous formulas

$$n(x) = \frac{\nabla\phi(x)}{|\nabla\phi(x)|}, \text{ and } \kappa(x) = \text{div}\left(\frac{\nabla\phi(x)}{|\nabla\phi(x)|}\right).$$

The various supplied files named `tools.idp` contains two macros for calculating these quantities, whose use is exemplified in Listing 8.

```
/* Include packages */
include "tools.idp"

/* Finite Element spaces */
fespace Vh(Th,P1);
fespace Vh0(Th,P0);

/* Finite Element functions used internally in the macros */
Vh nx,ny,v,vx,vy,kappa;
Vh0 nx0,ny0,ngrphi,kappa0;

/* [nx,ny] = P1 normal vector field to the domain defined by phi */
Vh phi;
```

```
normalvec(phi);

/* kappa = P1 curvature of the domain defined by phi*/
curvature;
```

LISTING 8. *Calculation of the normal vector and the mean curvature of a domain from its level set function in* `FreeFem++` *using the macros in* `tools.idp`*.*

(5) The evaluation of the objective function $J(\Omega)$ (or the shape derivative $J'(\Omega)$) is a little more difficult than in the context of geometric methods of Section 3.1: no mesh of $\Omega$ is available for the practice of the finite element method, only the fixed mesh $\mathcal{T}$ of $D$. To overcome this issue, the level set function $\phi$ for $\Omega$ is used to calculate an *approximate* displacement $u_\varepsilon$ for $u_\Omega$ on the *whole* mesh $\mathcal{T}$, by relying on the ersatz material method of Section 1.2.2. More precisely, the system

$$\left\{ \begin{array}{rl} -\text{div}(A_\varepsilon e(u_\varepsilon)) = 0 & \text{in } D, \\ A_\varepsilon e(u_\varepsilon)n = g & \text{on } \Gamma_N, \\ A_\varepsilon e(u_\varepsilon)n = 0 & \text{on } \partial D \setminus (\overline{\Gamma_N} \cup \overline{\Gamma_D}), \\ u_\varepsilon = 0 & \text{on } \Gamma_D, \end{array} \right. \quad \text{where } A_\varepsilon(x) = \left\{ \begin{array}{ll} A & \text{if } x \in \Omega \Leftrightarrow \phi(x) < 0; \\ \varepsilon A & \text{if } x \in D \setminus \Omega \Leftrightarrow \phi(x) \geq 0, \end{array} \right. \tag{4.2}$$

is solved on $D$ for a small value $\varepsilon \ll 1$ (typically, $\varepsilon = 1.e^{-3}$).

The supplied files `tools.idp` contain the macro `volfrac`, which was already used in Section 1.2.2 to calculate the approximate tensor $A_\varepsilon$ in (4.2); see also Listing 6.

(6) A second important issue about the level set method has to do with how the shape $\Omega$ is updated from the knowledge of the shape derivative $J'(\Omega)(\theta)$ and of the inferred descent direction $\theta$ (see also Section 3.1, Point (4)). In the framework of the level set method, the motion of a domain $\Omega(t)$ according to the velocity field $\theta(x)$ with normal component $v(x) = \theta(x) \cdot n(x)$ is achieved by solving the following Hamilton-Jacobi equation for a level set function $\phi(t,x)$:

$$\frac{\partial \phi}{\partial t}(t,x) + v(x)|\nabla \phi(t,x)| = 0, \;\; t > 0, \;\; x \in D. \tag{4.3}$$

In practice, an advection equation, counterpart to (4.3), is solved, by using the set of commands in Listing 9, relyting on macros defined in the supplied files `tools.idp`.

```
/* Include packages */
load "distance"
include "tools.idp"

/* Finite Element spaces and functions */
fespace Vh(Th,P1);
Vh phio,phi,phitmp,nx,ny;

real step = 0.1;

/* Solve the level set HJ equation with
  - initial data phio,
  - vector velocity [nx,ny],
   - final time step
  and perform reinitialization as a distance function */
advectRedist(phio,phi,nx,ny,step);
```

LISTING 9. *Advection and redistancing of a level set function in* `FreeFem++` *using the macros in* `tools.idp`*.*

## 4.2. **Optimal shape of a wheeled bridge**

### 4.2.1. *Implementation of the 'classical' level set method*

Our first example in the context of the level set method is concerned with the optimal design of a wheeled bridge, as depicted on Fig. 21: the structure $\Omega$ to be optimized is filled with a linearly elastic material with Hooke's law $A$ given by (1.8); $\Omega$ is clamped on its bottom right-hand side $\Gamma_D$, and its vertical displacement is prevented on its bottom left-hand side $\Gamma_S$ (so that the structure may only slide horizontally in this area). A vertical load $g = (0, -1)$ is applied on the central region $\Gamma_N$ of the bottom part, and only the remaining free part $\Gamma := \partial\Omega \setminus \left(\overline{\Gamma_D} \cup \overline{\Gamma_S} \cup \overline{\Gamma_N}\right)$ of the boundary of shapes is subject to optimization.

The displacement $u_\Omega = (u_{\Omega,1}, u_{\Omega,2}) : \Omega \to \mathbb{R}^2$ of the structure is therefore the solution to the following linearized elasticity system:

$$\left\{\begin{array}{rl} -\text{div}(Ae(u_\Omega)) = 0 & \text{in } \Omega, \\ Ae(u_\Omega)n = g & \text{on } \Gamma_N, \\ Ae(u_\Omega)n = 0 & \text{on } \Gamma, \\ u_\Omega = 0 & \text{on } \Gamma_D, \\ u_{\Omega,2} = 0 & \text{on } \Gamma_S. \end{array}\right. \tag{4.4}$$

In this context, our purpose is to minimize the compliance of the domain

$$C(\Omega) := \int_\Omega Ae(u_\Omega) : e(u_\Omega)\, dx = \int_{\Gamma_N} g \cdot u_\Omega \, ds, \tag{4.5}$$

under a constraint on the volume $\text{Vol}(\Omega) = \int_\Omega dx$ of shapes. A simplified version of this program arises under the form of the unconstrained minimization of a weighted sum of both functionals:

$$\min_\Omega \left\{ C(\Omega) + \ell \text{Vol}(\Omega) \right\}, \tag{4.6}$$

where $\ell$ is a fixed Lagrange multiplier (here, the value $\ell = 0.2$ may be used).

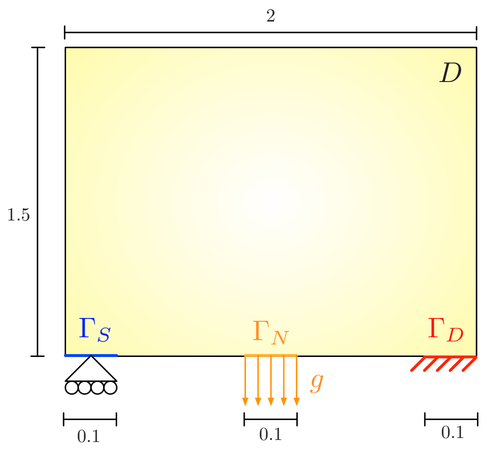


FIGURE 21. *Setting of the wheeled bridge test-case of Section 4.2.*

**Question 4.2.1.** Establish the variational formulation for the state system (4.4).

**Question 4.2.2.** Calculate the shape derivative of the compliance functional $C(\Omega)$ in (4.5); you may either resort to a rigorous calculation to this end, or to the fast method of Céa.

**Question 4.2.3.** Implement the level set framework sketched in Section 4.1 to deal with the above wheeled bridge optimization example. You may take advantage of the functions or macros present in the file `LSBridge/tools.idp` in your implementation.

Experience the effect of using different initial shapes (with different numbers of holes), different mesh sizes, as well as different inner products for calculating a descent direction from the knowledge of $J'(\Omega)$ (see Section 3.1, Point (4)), etc...

A proposition of correction is given in the file `LSBridge/LSBridge.edp` and a few iterations of the optimization process are represented on Fig. 22.

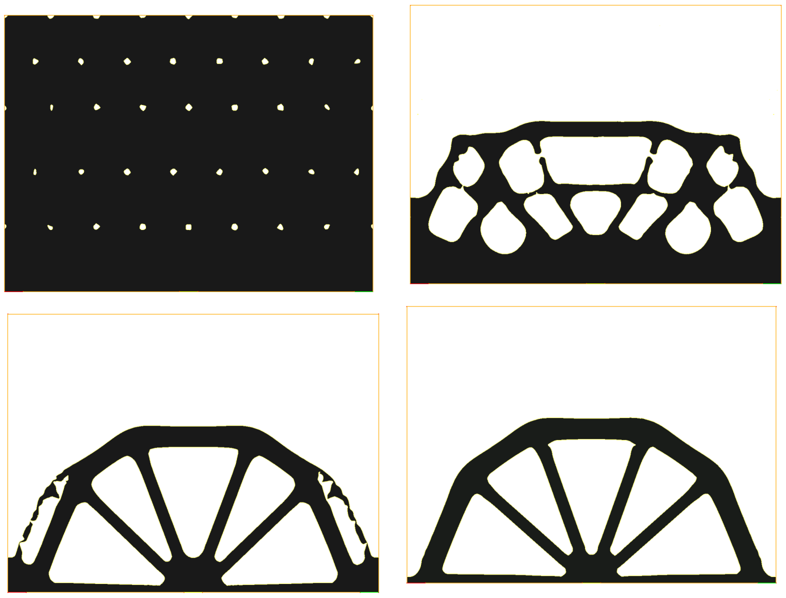


FIGURE 22. *Iterations* 0, 20, 100 *and* 200 *of the wheeled bridge optimization test-case of Section 4.2.*

**Question 4.2.4.** In some applications, it may be of interest to put also a constraint on the perimeter

$$\mathrm{Per}(\Omega) = \int_{\partial\Omega} ds$$

of shapes. By elaborating on your numerical program, solve now the problem

$$\min_{\Omega} \left\{ C(\Omega) + \ell_1 \mathrm{Vol}(\Omega) + \ell_2 \mathrm{Per}(\Omega) \right\}, \tag{4.7}$$

where $\ell_1$ and $\ell_2$ are two fixed Lagrange multipliers (in this example, you may take $\ell_1 = 0.2$ and $\ell_2 = 0.02$). You may use the macros and functions contained in the file `Subjects/LSBridge/tools.idp` during your implementation.

A proposition of correction is given in the file `Solutions/LSBridgePer/LSBridgePer.edp` and a few iterations are represented on Fig. 23.

**Question 4.2.5.** Adapt your optimization algorithm to implement the augmented Lagrangian strategy outlined in Question 2.3.4 in order to deal with the true constrained optimization programs

$$\min_{\Omega} C(\Omega) \text{ s.t. } \mathrm{Vol}(\Omega) = V_T,$$

and

$$\min_{\Omega} C(\Omega) \text{ s.t. } \left\{ \begin{array}{l} \mathrm{Vol}(\Omega) = V_T, \\ \mathrm{Per}(\Omega) = P_T, \end{array} \right.$$

where $V_T$ and $P_T$ are user-defined targets for the volume and perimeter constraints, instead of the (penalized) unconstrained formulations (4.6) and (4.7).

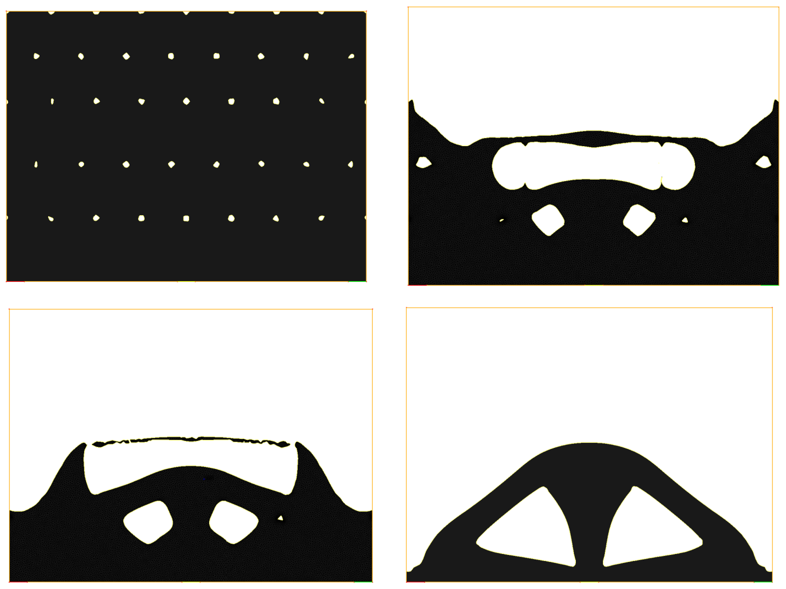


FIGURE 23. *Iterations 0, 50, 70 and 200 of the wheeled bridge optimization test-case of Section 4.2 under volume and perimeter constraints.*

4.2.2. *Optimisation of the shape of a bridge using topological derivatives*

As you may have experienced, it is sometimes difficult to find a suitable initial shape: on the one hand, considering an initial shape with many holes may precipitate the algorithm into local minima with poor structural performance; on the other hand, initializing the method with few holes is equally awkward since the algorithm is not able to nucleate holes inside shape (it only proceeds by moving the boundary of shapes).

A quite convenient remedy to this drawback relies on a different means for differentiating a functional of the domain, that of *topological derivative*. This notion appraises how the optimized functional, say $J(\Omega)$, behaves when a small hole, centered at a given point $x_0 \in \Omega$, with radius $r > 0$ is nucleated inside $\Omega$, i.e. when $\Omega$ is replaced by

$$\Omega_{x_0,r} := \Omega \setminus \overline{B(x_0,r)}.$$

**Definition 4.1.** *A functional of the domain $J(\Omega)$ has a topological derivative at a point $x_0 \in \Omega$ if there exists a real value $g^T_\Omega(x_0)$ such that the following asymptotic expansion holds in the neighborhood of $r = 0$:*

$$J(\Omega_{x_0,r}) = J(\Omega) + r^d g^T_\Omega(x_0) + o(r^d), \text{ where } \frac{|o(r^d)|}{r^d} \xrightarrow{r\to 0} 0.$$

Hence, at some point $x_0$ where $g^T_\Omega(x_0) < 0$, a hole with small radius $r > 0$ may be nucleated inside $\Omega$, leading to a new shape $\Omega_{x_0,r}$ with a lesser value of $J(\Omega)$.

When the considered shape functional $J(\Omega)$ is the compliance, the topological derivative reads (in two space dimensions):

$$\forall x \in \Omega, \ g^T_\Omega(x) = \frac{\pi(\lambda+2\mu)}{2\mu(\lambda+\mu)}\left(4\mu Ae(u_\Omega) : e(u_\Omega) + (\lambda-\mu)\mathrm{tr}(Ae(u_\Omega))\mathrm{tr}(e(u_\Omega))\right)(x).$$

**Question 4.2.6.** What is the value of the topological derivative of the volume functional $\mathrm{Vol}(\Omega)$ at some given point $x \in \Omega$?

As proposed for instance in [3, 12], topological derivatives may be combined with shape derivatives in the framework of the level set method for shape and topology optimization. Indeed, in the iterative optimization process of a function $J(\Omega)$, one may alternate between stages where the shape derivative $J'(\Omega)$ is calculated and used (exactly as in the previous section), and some iterations where the topological derivative $g^T_\Omega$ is calculated everywhere inside $\Omega$, and where a small hole is nucleated around the point $x_0$ where $g_\Omega^T(x_0)$ is the most negative.

**Question 4.2.7.** Redo the bridge example of Section 4.2.1 by starting from a half-domain (without hole) as in Fig. 24 (top, left), and replacing one every, say, five step, the update of the shape based on the shape derivative of the optimized functional with one using its topological derivative.

You may in particular try several strategies as regards the nucleation of holes during steps involving topological derivatives (frequency of these stages, number of holes nucleated, etc.); see the macro `perTopDer` from the file `Subjects/LSBridgeTopDer/tools.idp` about this point.

One possible correction is contained in the file `Solutions/LSBridgeTopDer/LSBridgeTopDer.edp`, and an illustration is provided on Fig. 24.

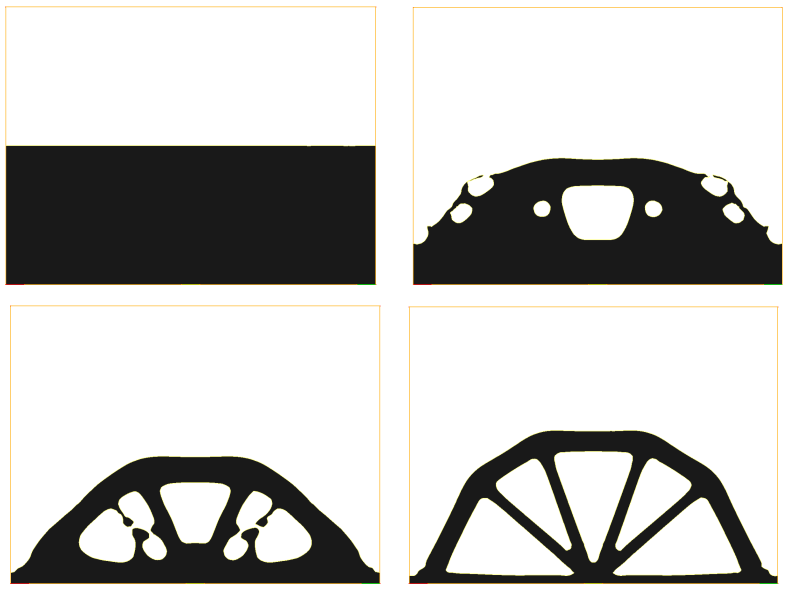


FIGURE 24. *Iterations* 0, 15, 50 *and* 200 *of the wheeled bridge optimization test-case of Section 4.2, using topological derivatives.*

### 4.2.3. *Beyond the bridge*

**Question 4.2.8.** There are several quite classical benchmark test-cases that you may consider by using and elaborating on the previous material; to name a few,

- You may first revisit the classical cantilever example of Sections 2.3 and 3.2 from the viewpoint of the level set method.
- You may consider the double cantilever example - an example of a multiple load case: the setting is presented on Fig. 25 (left). The aim is to minimize the sum

$$J(\Omega) = \sum_{i=1}^{2} C_i(\Omega), \text{ where } C_i(\Omega) := \int_\Omega Ae(u_{\Omega,i}) : e(u_{\Omega,i})\, dx,$$

is the compliance under the load $g_i$ applied on the fixed region $\Gamma_{N,i}$. The function $C_i(\Omega)$ involves the displacement $u_{\Omega,i}$, solution to the linearized elasticity system where only the load $g_i$ is at play; for instance:

$$\begin{cases} -\mathrm{div}(Ae(u_{\Omega,1})) = 0 & \text{in } \Omega, \\ Ae(u_{\Omega,1})n = g_1 & \text{on } \Gamma_{N,1}, \\ Ae(u_{\Omega,1})n = 0 & \text{on } \Gamma \cup \Gamma_{N,2}, \\ u_{\Omega,1} = 0 & \text{on } \Gamma_D, \end{cases}$$

and similarly for $u_{\Omega,2}$. A volume constraint is added either under the form of a fixed penalization of $J(\Omega)$, or by means of an augmented Lagrangian strategy.

- Finally, you may turn to a stress minimization problem, such as the L-shaped beam example depicted on Fig. 25 (right). In there, the minimized functional is a global measure $S(\Omega)$ of the stress $\sigma(u_\Omega) := Ae(u_\Omega)$ in the structure:

$$S(\Omega) = \left( \int_\Omega \chi_\omega(x) ||\sigma(u_\Omega)||^2 \, dx \right)^{\frac{1}{2}},$$

where $\chi_\omega$ is the characteristic function of a subdomain $\omega \subset D$ which avoids the region (in blue in Fig. 25) where the load is applied. In this example again, a constraint on the volume of shapes should be added (either by means of a fixed penalization, or by an augmented Lagrangian strategy).

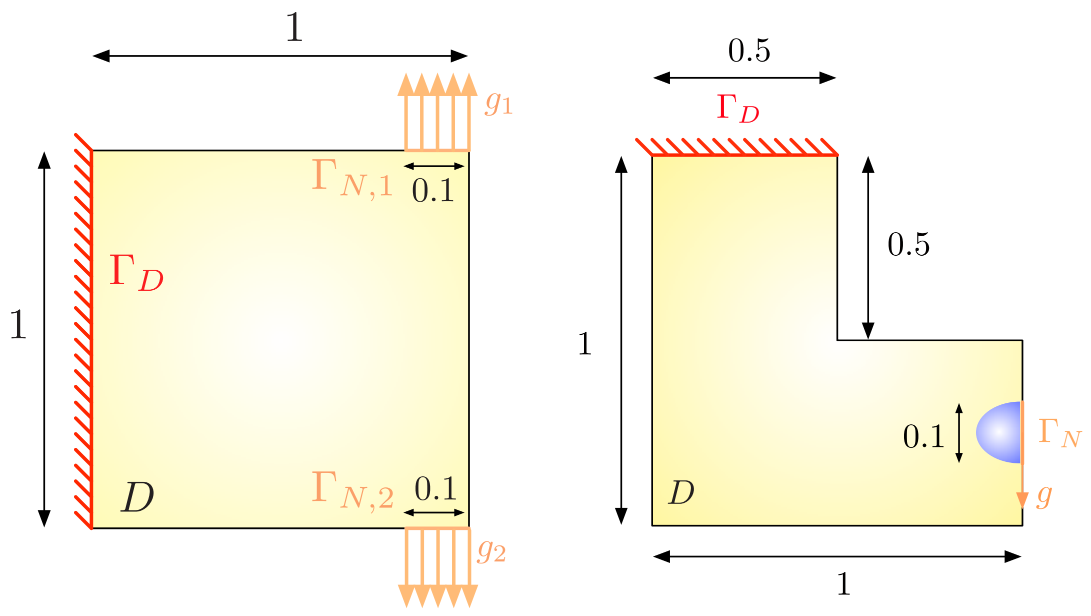


FIGURE 25. *(Left) Setting of the double cantilever test-case; (right) setting of the L-shaped beam example.*

## 4.3. **A level set-based fixed point algorithm relying solely on topological derivatives**

The notion of topological derivative may alternatively be used on its own in a topology algorithm, in combination with a level set representation of shapes, as was initially proposed by S. Amstutz [6].

### 4.3.1. *A description of the algorithm*

Let us present the main stakes of this topology optimization algorithm in the context of linearized elasticity.

(1) This algorithm is fundamentally a two-phase topology optimization algorithm; more precisely, the setting is as follows: a 'hold-all' domain $D$ is equipped with a fixed mesh $\mathcal{T}$. Each shape $\Omega \subset D$ divides the domain $D$ into two complementary phases $\Omega^0 = \Omega$ and $\Omega^1 = D \setminus \overline{\Omega}$, filled with different elastic materials, associated to the respective Hooke's tensors $A_0$ and $A_1$. The total structure $D$ is clamped on a region $\Gamma_D$ of its boundary $\partial D$, and traction loads are applied on the disjoint subset $\Gamma_N \subset \partial D$, so that the displacement $u_\Omega$ of the total structure $D$ belongs to the space $H^1_{\Gamma_D}(D)^2$ and

is the unique solution to the system:

$$\begin{cases} -\mathrm{div}(A_\Omega e(u_\Omega)) = 0 & \text{in } D, \\ A_\Omega e(u_\Omega) n = g & \text{on } \Gamma_N, \\ A_\Omega e(u_\Omega) n = 0 & \text{on } \partial D \setminus (\overline{\Gamma_N} \cup \overline{\Gamma_D}), \\ u_\Omega = 0 & \text{on } \Gamma_D, \end{cases} \quad \text{where } A_\Omega(x) = \begin{cases} A_0 & \text{if } x \in \Omega^0; \\ A_1 & \text{if } x \in \Omega^1. \end{cases} \tag{4.8}$$

Note that the context of optimization of only one material, as investigated in the previous sections, is recovered in this setting by the particular choice of materials $A_0 = A$ and $A_1 = \varepsilon A$, where $\varepsilon \ll 1$ is the parameter involved in the ersatz material approximation, as in Section 1.2.2.

(2) An objective function $J(\Omega)$ is then considered, which depends on $\Omega$ via the solution $u_\Omega$ to (4.8). For instance, $J(\Omega)$ may stand for the compliance $C(\Omega)$ of the total structure $D$:

$$C(\Omega) = \int_D A_\Omega e(u_\Omega) : e(u_\Omega)\, dx. \tag{4.9}$$

(3) The notion of topological derivative for such a function $J(\Omega)$ of the domain takes a slightly different meaning from that of Definition 4.1. More precisely, we now consider perturbations of $\Omega$ of the form:

$$\Omega_{x,r} := \begin{cases} \Omega \setminus \overline{B(x,r)} & \text{if } x \in \Omega, \\ \Omega \cup B(x,r) & \text{if } x \in D \setminus \overline{\Omega}; \end{cases}$$

i.e. $\Omega_{x,r}$ is obtained from $\Omega$ by nucleation of a small ball centered at $x$ if $x \in \Omega$, or by addition of a small ball if $x \notin \Omega$.

**Definition 4.2.** *A function $J(\Omega)$ of the domain has a topological derivative at a particular point $x \in \Omega^0 \cup \Omega^1$ if there exists a real number $g^T_\Omega(x)$ such that the following asymptotic expansion holds:*

$$F(\Omega_{r,x}) = F(\Omega) + s_\Omega(x) g^T_\Omega(x) |B(x,r)| + o(|B(x,r)|), \ \text{where} \ \lim_{r \to 0} \frac{o(|B(x,r)|)}{|B(x,r)|} = 0,$$

*and:*

$$s_\Omega(x) = 1 \text{ when } x \in \Omega^0 \text{ and } s_\Omega(x) = -1 \text{ when } x \in \Omega^1. \tag{4.10}$$

For instance, the following result yields the topological derivative of the two-phase compliance (4.9); see [9, 16]:

**Theorem 4.1.** *For any two-dimensional admissible shape $\Omega \in \mathcal{U}_{ad}$, the function $C(\Omega)$ defined in (4.9) has a topological derivative $g^T_\Omega(x)$ at any point $x \in \Omega^0 \cup \Omega^1$. Its expression reads:*

$$g^T_\Omega(x) = s_\Omega(x)\, \mathbb{P}\sigma(u_\Omega(x)) : e(u_\Omega(x)),$$

*where $s_\Omega(x)$ is defined by (4.10) and $\mathbb{P}$ is the fourth-order Pólya-Szegö polarization tensor, given by:*

$$\forall e \in \mathcal{S}(\mathbb{R}^d), \ \mathbb{P}e = \frac{1}{\rho_2\rho_3 + \tau_1}\left( (1+\rho_2)(\tau_1 - \rho_3)e + \frac{1}{2}(\rho_1 - \rho_2)\frac{\rho_3(\rho_3 - 2\tau_3) + \tau_1\tau_2}{\rho_1\rho_3 + \tau_2}\mathrm{tr}(e) I \right), \tag{4.11}$$

*where*

$$\rho_1 = \frac{1+\nu}{1-\nu}, \ \rho_2 = \frac{3-\nu}{1+\nu}, \ \rho_3 = \frac{E^\star}{E}, \ \tau_1 = \frac{1+\nu^\star}{1+\nu}, \ \tau_2 = \frac{1-\nu^\star}{1-\nu} \quad \text{and} \quad \tau_3 = \frac{\nu^\star(3\nu-4)+1}{\nu(3\nu-4)+1}. \tag{4.12}$$

*and we have posed:*

- *If $x \in \Omega^1$, $E = E_1$, $E^\star = E_0$, $\nu = \nu_1$, and $\nu^\star = \nu_0$,*
- *If $x \in \Omega^0$, $E = E_0$, $E^\star = E_1$, $\nu = \nu_0$, and $\nu^\star = \nu_1$.*

(4) From the numerical point of view, an arbitrary shape $\Omega \subset D$ is represented by means of an associated level set function $\phi : D \to \mathbb{R}$, i.e. (4.1) holds.

(5) The main idea of the algorithm is that one shape $\Omega \subset D$ is (locally) optimal for the functional $J(\Omega)$ provided the following holds:

$$\begin{cases} \forall x \in \Omega, & g^T_\Omega(x) \leq 0, \\ \forall x \in D \setminus \overline{\Omega}, & g^T_\Omega(x) \geq 0. \end{cases}$$

In other terms, $\Omega$ is optimal for $J(\Omega)$ if and only if the topological derivative $g_\Omega^T$ is one level set function for $\Omega$. Noting in addition that if $\phi$ is one level set function for $\Omega$, then so is $c\phi$ for any constant $c > 0$, one sufficient optimality condition for $\phi$ is that:

$$||\phi||_{L^2(D)} = 1, \text{ and } \phi = \frac{1}{||g_\Omega^T||_{L^2(D)}} g_\Omega^T, \text{ with } \Omega = \{x \in D, \ \phi(x) < 0\}. \tag{4.13}$$

(6) A fixed-point algorithm with relaxation is applied to the condition (4.13); elementary calculations yield the sequences of level set functions $\phi^n$ and associated shapes $\Omega^n := \{x \in D, \ \phi^n(x) < 0\}$ defined by:

$$\phi^{n+1} = \frac{1}{\sin a^n} \left(\sin((1-\tau^n)a^n)\phi^n + \sin(\tau^n a^n)\widetilde{g^n}\right), \tag{4.14}$$

where $\widetilde{g^n} = \frac{1}{||g^n||_{L^2(D)}} g^n$ is the normalized version of the gradient $g^n := g_{\Omega^n}^T$, $a^n \in [0, \pi]$ is the angle $a^n = \arccos((\phi^n, g^n)_{L^2(D)})$, and $\tau^n$ is a time step.

### 4.3.2. *Application to the optimization of the shape of an electric mast*

We consider the optimization of the shape and topology of a T-shaped electric mast, as represented on Fig. 26. The considered shapes are clamped on their lower side $\Gamma_D$, and vertical loads $g = (0, -1)$ are applied on the ends $\Gamma_N$ of their arms (accounting for the weight of the electric cables). The remaining part of the boundary $\Gamma := \partial\Omega \setminus (\overline{\Gamma_D} \cup \overline{\Gamma_N})$ is traction-free and it is the only region of $\partial\Omega$ which is subject to optimization.

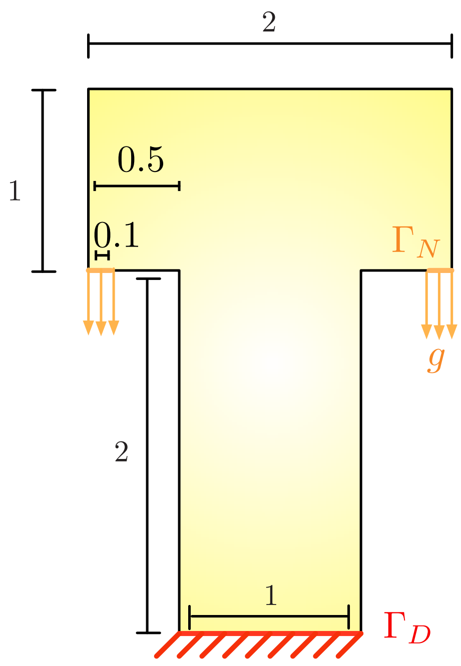


FIGURE 26. *Setting of the mast test-case of Section 4.3.*

The displacement $u_\Omega$ of the structure in this context is then the unique solution to the linearized elasticity system (1.11), and the considered problem is that (4.6) of minimizing a weighted sum of the compliance $C(\Omega)$ of the mast and of its volume $\text{Vol}(\Omega)$ (in this setting, it is possible to use the weighting parameter $\ell = 2$).

**Question* 4.3.1.** Implement the algorithm sketched in Section 4.3.1. Again, you may take advantage of the macros contained in the file `LSMast/tools.idp` to this end.

It is interesting to experiment various initial designs, and various mesh sizes to appraise the sensitivity of the algorithm to these parameters.

A possible correction is given in the file `LSMast/LSMast.edp`, and results are depicted on Fig. 27.

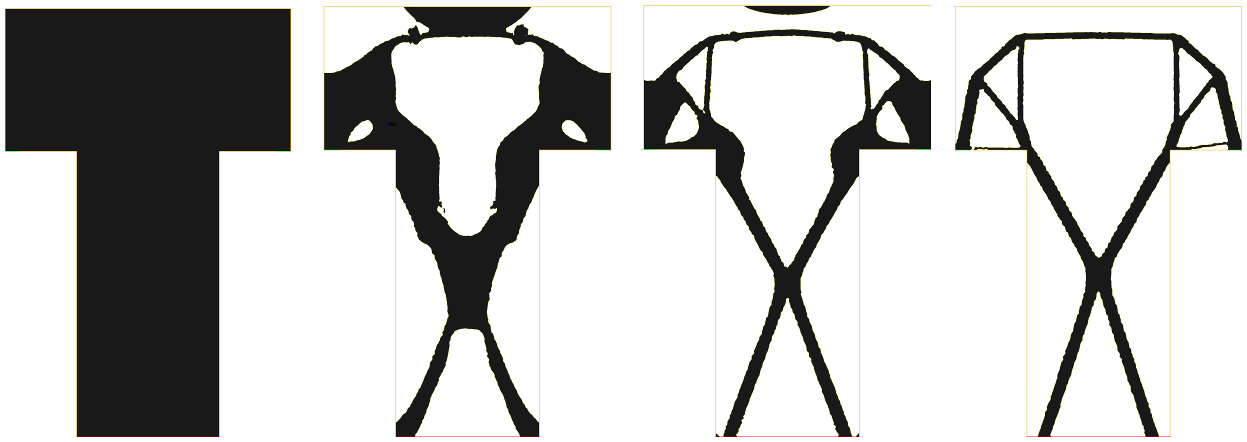


FIGURE 27. *Iterations* 0, 10, 30 *and* 200 *of the mast optimization test-case using the topological gradient algorithm of Section 4.3.*

## REFERENCES

[1] *Tacoma bridge*. https://fr.wikipedia.org/wiki/Pont_du_dtroit_de_Tacoma_(1940). Accessed: 2018-08-27.

[2] G. ALLAIRE, *Analyse numérique et optimisation: Une introduction à la modélisation mathématique et à la simulation numérique*, Editions Ecole Polytechnique, 2005.

[3] G. ALLAIRE, F. DE GOURNAY, F. JOUVE, AND A.-M. TOADER, *Structural optimization using topological and shape sensitivity via a level set method*, Control and cybernetics, 34 (2005), p. 59.

[4] G. ALLAIRE AND O. PANTZ, *Structural optimization with* `FreeFem++`, Structural and Multidisciplinary Optimization, 32 (2006), pp. 173–181.

[5] S. AMSTUTZ, *Connections between topological sensitivity analysis and material interpolation schemes in topology optimization*, Structural and Multidisciplinary Optimization, 43 (2011), pp. 755–765.

[6] S. AMSTUTZ AND H. ANDRÄ, *A new algorithm for topology optimization using a level-set method*, Journal of Computational Physics, 216 (2006), pp. 573–588.

[7] S. AMSTUTZ, C. DAPOGNY, AND À. FERRER, *A consistent relaxation of optimal design problems for coupling shape and topological derivatives*, Numerische Mathematik, (2016), pp. 1–60.

[8] M. P. BENDSOE AND O. SIGMUND, *Topology optimization: theory, methods, and applications*, Springer Science & Business Media, 2013.

[9] M. BONNET AND G. DELGADO, *The topological derivative in anisotropic elasticity*, Quarterly Journal of Mechanics and Applied Mathematics, 66 (2013), pp. 557–586.

[10] B. BOURDIN, *Filters in topology optimization*, International journal for numerical methods in engineering, 50 (2001), pp. 2143–2158.

[11] M. BURGER, *A framework for the construction of level set methods for shape optimization and reconstruction*, Interfaces and Free boundaries, 5 (2003), pp. 301–329.

[12] M. BURGER, B. HACKL, AND W. RING, *Incorporating topological derivatives into level set methods*, Journal of Computational Physics, 194 (2004), pp. 344–362.

[13] A. CHERKAEV, *Variational methods for structural optimization*, vol. 140, Springer Science & Business Media, 2012.

[14] C. DAPOGNY, P. FREY, F. OMNÈS, AND Y. PRIVAT, *Geometrical shape optimization in fluid mechanics using freefem++*, Structural and Multidisciplinary Optimization, (2017), pp. 1–28.

[15] F. DE GOURNAY, *Velocity extension for the level-set method and multiple eigenvalues in shape optimization*, SIAM journal on control and optimization, 45 (2006), pp. 343–367.

[16] S. M. GIUSTI, A. FERRER, AND J. OLIVER, *Topological sensitivity analysis in heterogeneous anisotropic elasticity problem. theoretical and computational aspects*, Computer Methods in Applied Mechanics and Engineering, 311 (2016), pp. 134–150.

[17] J. K. GUEST, J. H. PRÉVOST, AND T. BELYTSCHKO, *Achieving minimum length scale in topology optimization using nodal design variables and projection functions*, International journal for numerical methods in engineering, 61 (2004), pp. 238–254.

[18] F. HECHT, *New development in freefem++*, Journal of numerical mathematics, 20 (2012), pp. 251–266.

[19] F. Hecht, O. Pironneau, A. Le Hyaric, and K. Ohtsuka, *Freefem++ manual*, 2005.
[20] J. Nocedal and S. J. Wright, *Numerical optimization 2nd*, Springer, 2006.
[21] O. Sigmund and K. Maute, *Topology optimization approaches*, Structural and Multidisciplinary Optimization, 48 (2013), pp. 1031–1055.
[22] K. Svanberg, *The method of moving asymptotesa new method for structural optimization*, International journal for numerical methods in engineering, 24 (1987), pp. 359–373.
[23] F. Wang, B. S. Lazarov, and O. Sigmund, *On projection methods, convergence and robust formulations in topology optimization*, Structural and Multidisciplinary Optimization, 43 (2011), pp. 767–784.