An introduction to optimal design

Charles Dapogny

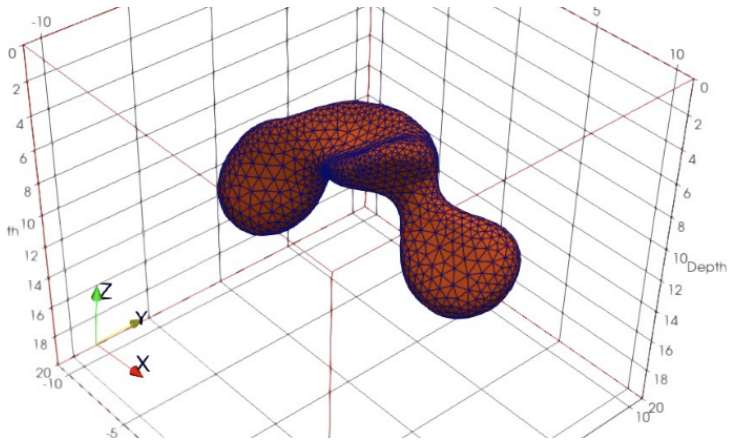
Laboratoire Jacques-Louis Lions, Sorbonne Université, Paris, France

13th April, 2026

Foreword (I)

- Optimal design is the search for the "best" shape, subject to physical requirements.
- This ambition traces back to early human history.
- Lately, optimal design techniques have been arousing much enthusiasm in engineering...
- ... and in many other applied disciplines!
- Its modern formulation brings into play:
  - Partial differential equations,
  - Optimization,
  - Scientific computing.

The myth of foundation of Carthage (Credits: J.-C. Golvin).

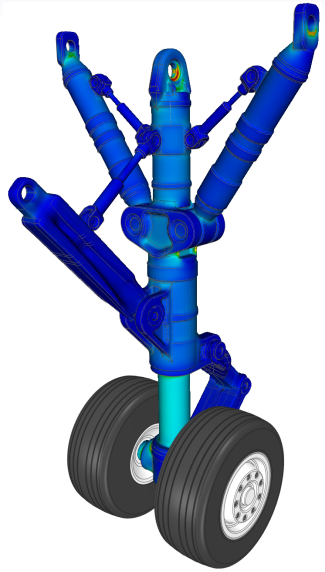


Reconstruction of the shape of a magmatic chamber.

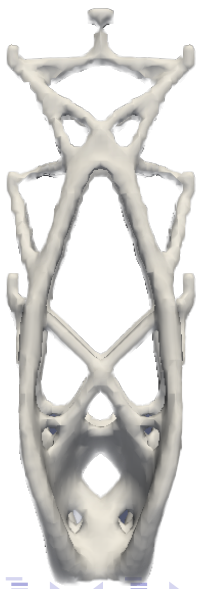
Optimal design in structural mechanics...

Structural mechanics is one of the historical applications contexts of optimal design:

- Mechanical engineering: Optimization of the internal components of cars (engine housing, transmission tree), motors, planes, etc.
- Civil engineering: Optimization of buildings (vaults, columns, etc.), large-scale structures (bridges).
- Optimization of fabrication processes: casting, milling, 3d printing,...
- Optimization of lifespan: resistance to plasticity, fatigue and fracture.



Optimization of a landing gear (courtesy of Ansys).

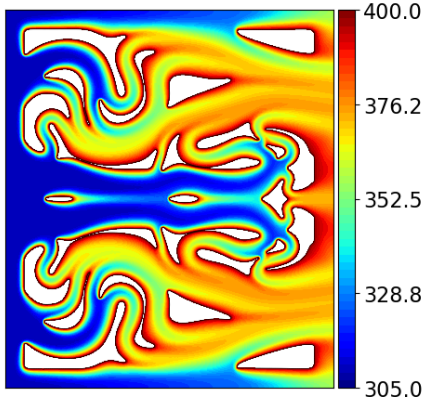


Optimization of an electric mast

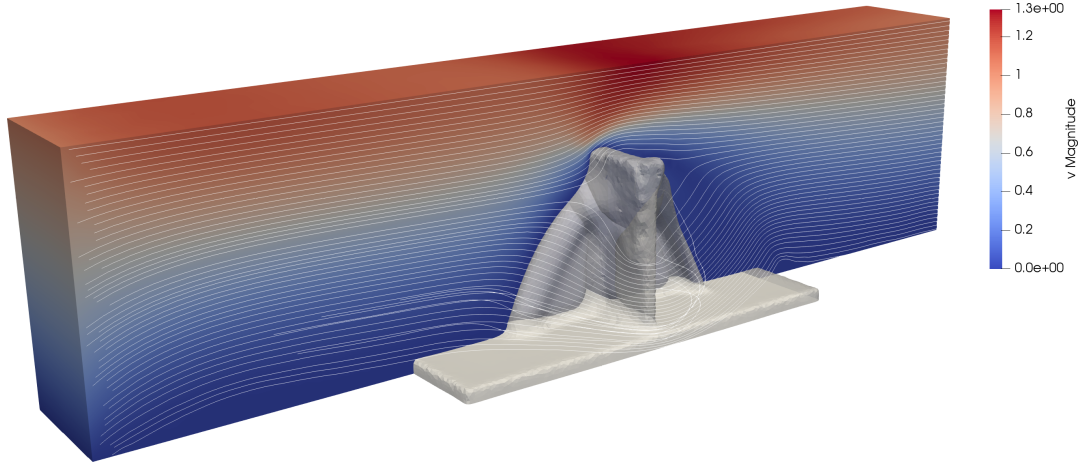
... And a wide variety of applications beyond

Optimal design has recently become a topical issue in such diverse fields as:

- Fluid mechanics: external aerodynamics, fluid transport, mixing devices, etc.

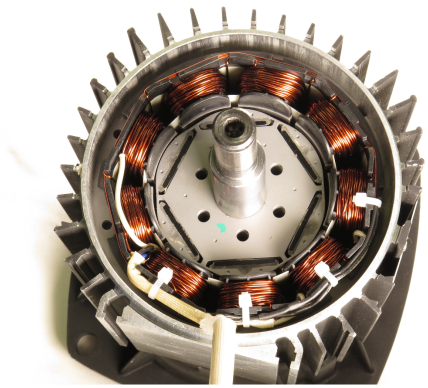


Optimized 2d section of a heat exchanger.

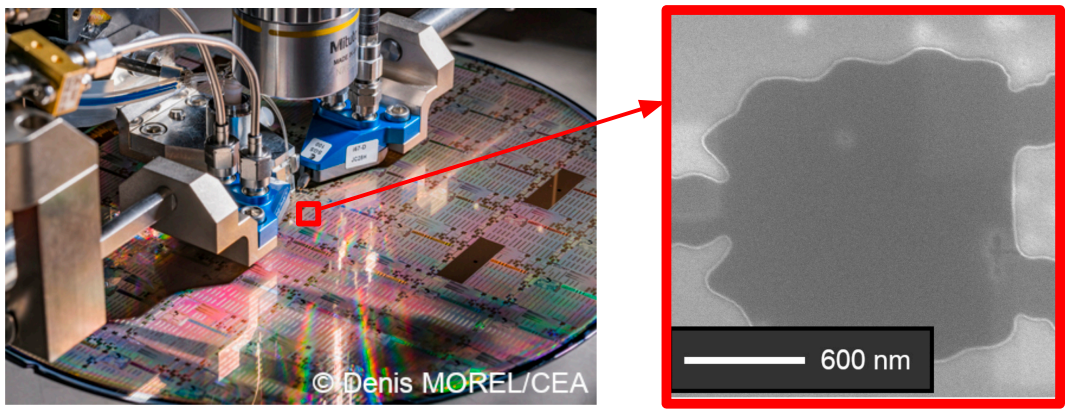


Optimized shape of a solid obstacle to a fluid flow.

- Electromagnetism: electric machines, current sensors, photonic crystals...



Section of a motor, (Hanning Elektro-Werke & Co KG)



Optimization of a nanophotonic device.

- Quantum chemistry, with the theory of Maximum Probability Domains.

Foreword (II)

- This course is about parametric optimization:

  "The design is represented by parameters, lying in a fixed (finite- or infinite-dimensional) vector space."

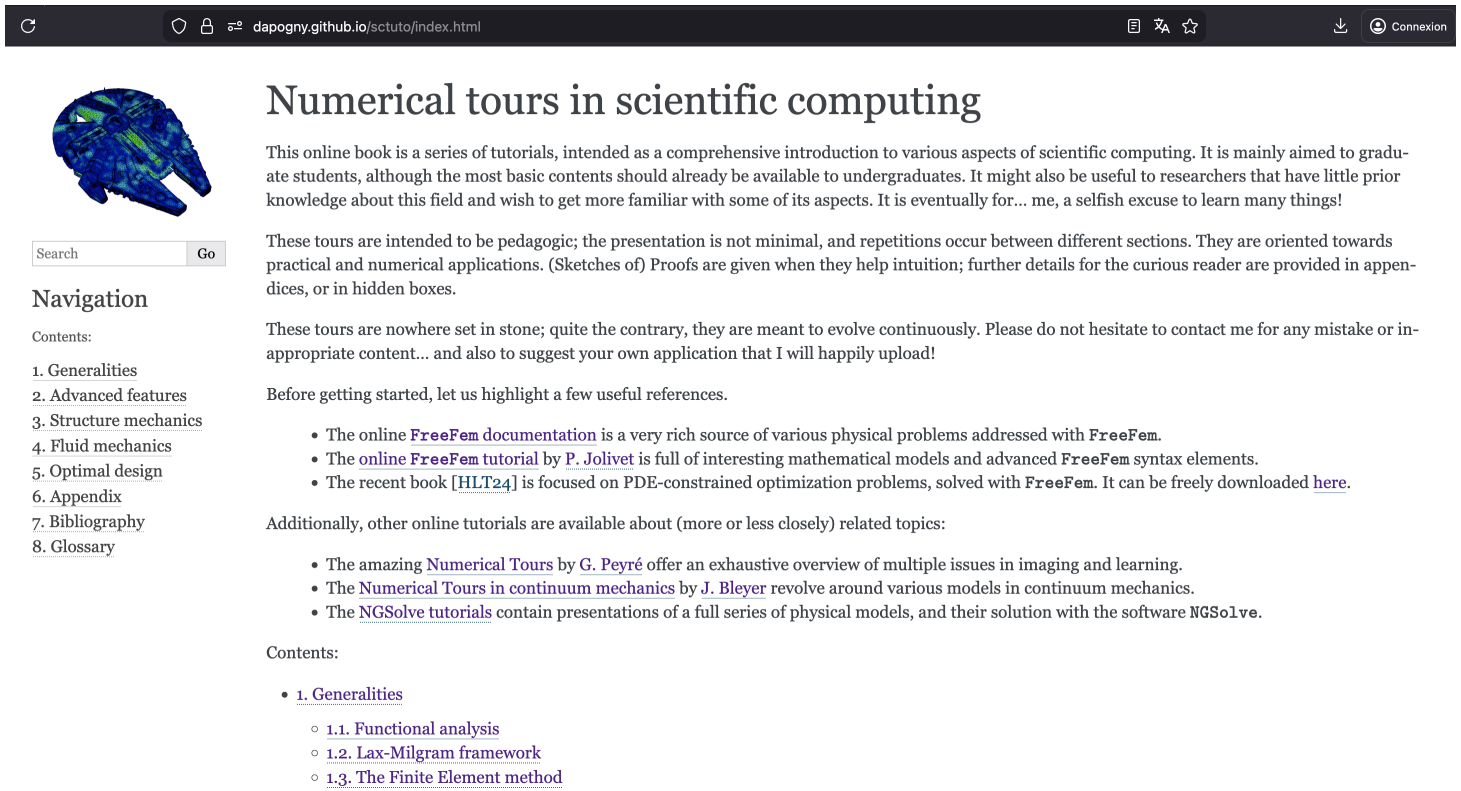
- This simplified optimal design setting thus assumes an a priori description of the design.

- The techniques of this course paves the way to advanced and more technically intricate settings:
  - Shape optimization (course of I. Ftouhi), where the boundary of the design is optimized;
  - Topology optimization (course of D. Seck), where even the topology of the design (its number of holes in 2d) is optimized.

A word of advertising

A webpage (in construction) about scientific computing, gathering tutorials, codes...

`https://dapogny.github.io/sctuto/index.html`



Overview of the lectures

- `Course 1` A few basic from scientific computing.
  - The Finite Element Method;
  - An introduction to `FreeFem`;
  - Numerical optimization.
- `Course 2` Theoretical ingredients for parametric optimization.
  - Generalities about parametric optimization;
  - Differentiation by the adjoint method.
- `Course 3` Numerical methods for parametric optimization.
  - Numerical implementation recipes;
  - Density-based topology optimization.
- `Practical session` A `FreeFem` implementation of the SIMP method.

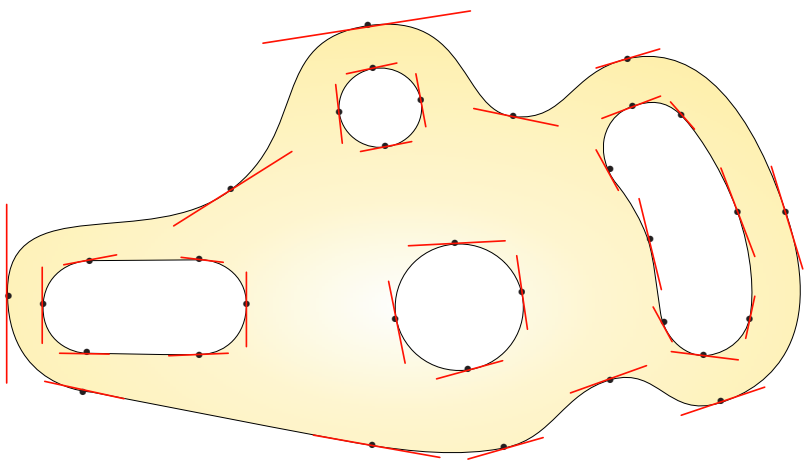
Part I

A few basic ingredient

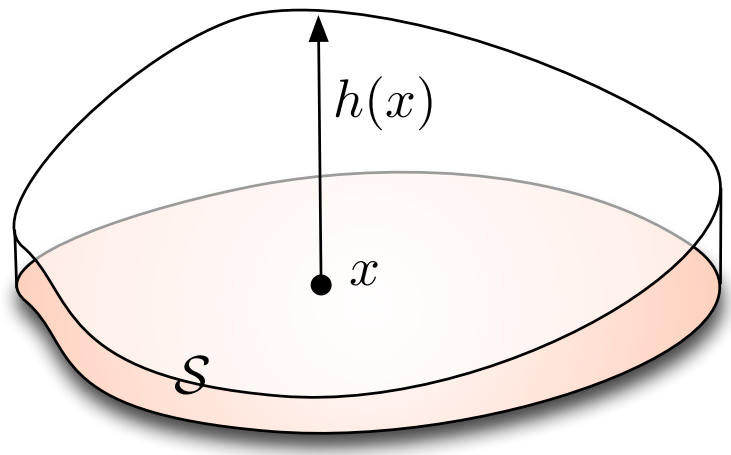
Parametric optimization (I)

The designs are described by a parameter h in a fixed vector space H, which may be:

- *Finite-dimensional:* $h = \{h_i\}_{i=1,\dots,N}$ is a collection of characteristic lengths, curvature radii...
- *Infinite-dimensional:* h is an height or thickness function.



Description of a mechanical part via the control points of a CAD model.



Parametrization of a plate with cross-section $\mathcal{S}$ by the thickness function $h : \mathcal{S} \to \mathbb{R}$.

Parametric optimization (II)

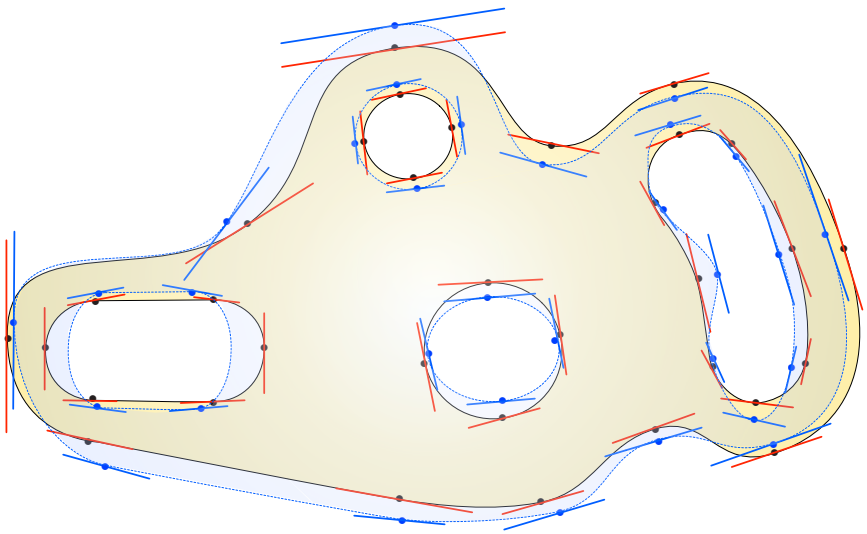
The optimal design problem is formulated in terms of the parameter h:

$$\min_{h \in H} \; J(h) \;\; \text{s.t.} \;\; G(h) = 0,$$

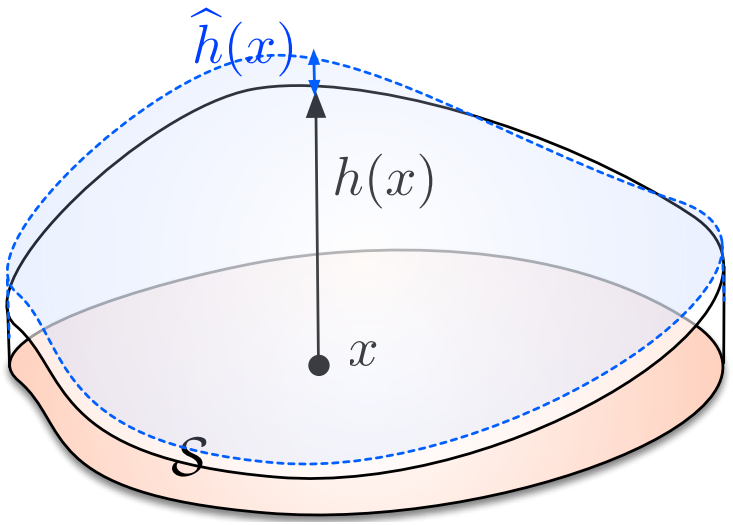
where:

- $J : H \to \mathbb{R}$ and $G : H \to \mathbb{R}^p$ are objective and (equality) constraint functions:
- Since H is a vector space, "small" variations of a design h are defined by:

$$h \longmapsto h + \widehat{h}, \text{ where } \widehat{h} \in H \text{ is "small".}$$



Small variations of the control points of a CAD model.



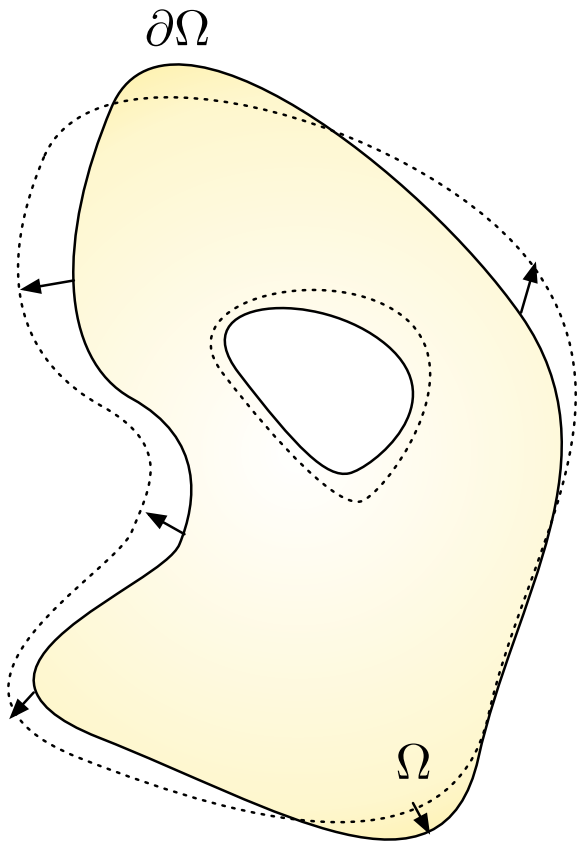
Small variation of the thickness of a plate.

Geometric optimization

- The design is a shape, with fixed topology (i.e. number of holes in 2d).
- The whole boundary $\partial\Omega$ of shapes Ω is the optimization variable.
- Shape optimization problems read:

$$\min_{\Omega \subset \mathbb{R}^d} \; J(\Omega) \text{ s.t. } \quad G(\Omega) = 0,$$

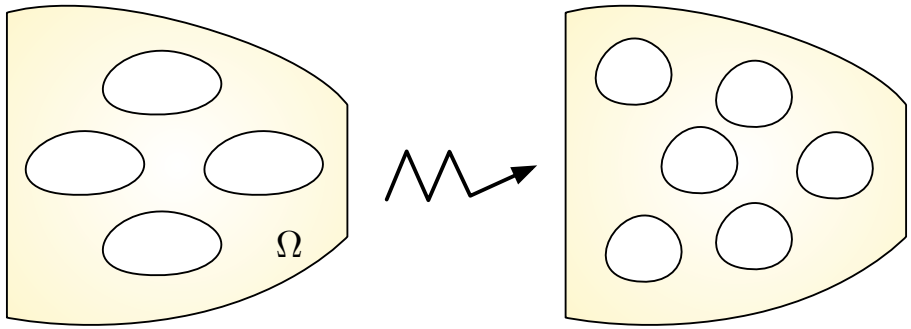
  where the objective and constraint functionals $J(\Omega)$ and $G(\Omega)$ depend on the domain.
- This setting allows for much freedom, since no a priori knowledge about the relevant features of the optimized design is required.



Optimization of Ω via "free" perturbations of the boundary $\partial\Omega$.

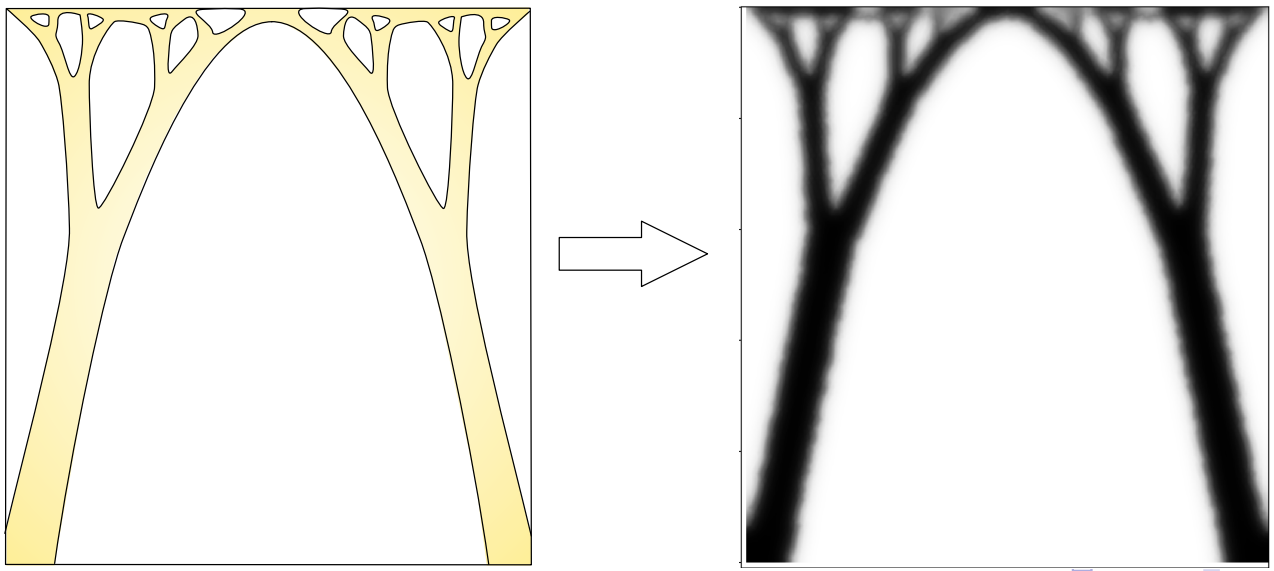
Topology optimization

- The topology of the optimized shape is also subject to optimization.



- It is often preferred not to represent the design via the boundary of a shape, but to employ descriptions that allow for an easier account of topological changes.

Example: The shape Ω is replaced by a density function $h : D \to [0, 1]$.



Part I

A few basic ingredient

Parametric optimization problems (I)

This course focuses on parametric optimization problems, of the form:

$$\min_{h \in \mathcal{U}_{\mathrm{ad}}} J(h) \text{ s.t. } G(h) = 0.$$

Here,

- The design variable h is sought within a subset $\mathcal{U}_{\mathrm{ad}}$ of a fixed vector space H.
- $J(h)$ is an objective function.
- $G(h) = (G_1(h), \ldots, G_p(h))$ is a collection of p (equality) constraints.
- $J(h)$ and $G(h)$ may depend on h via a state u_h, solution in a functional space V to an h-dependent boundary-value problem:

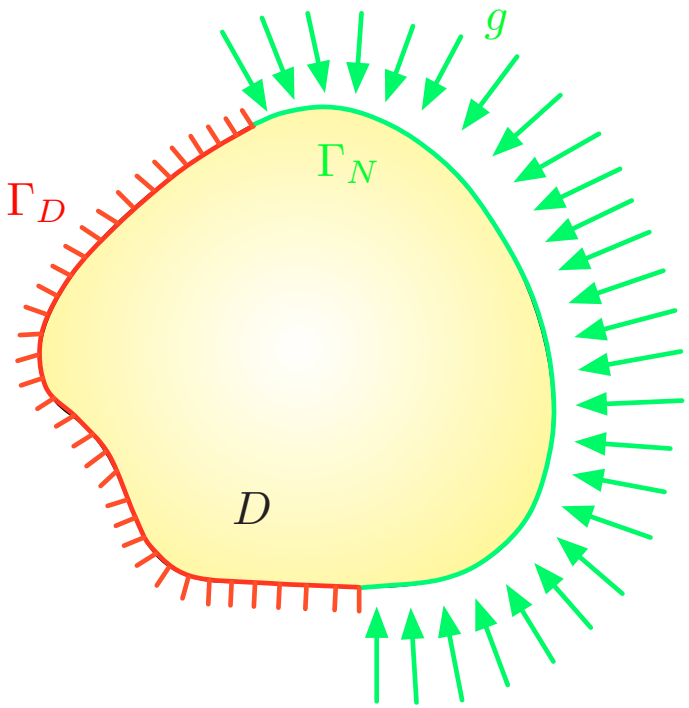
$$\text{Search for } u_h \in V \text{ s.t. for all } v \in V, \quad a_h(u_h, v) = \ell_h(v).$$

Example 1: Optimization of a thermal chamber

- A fixed thermal chamber $D \subset \mathbb{R}^d$ $(d = 2, 3)$ is filled by a material with conductivity $h : D \to \mathbb{R}$.
- The temperature u_h is the solution in $H^1(D)$ to the "state", conductivity equation:

$$\left\{ \begin{array}{rcll} -\mathrm{div}(h \nabla u_h) & = & f & \text{in } D, \\ u_h & = & 0 & \text{on } \Gamma_D, \\ h \frac{\partial u_h}{\partial n} & = & g & \text{on } \Gamma_N, \end{array} \right.$$

where $f \in L^2(D)$ and $g \in L^2(\Gamma_N)$ are sources.



- The set $\mathcal{U}_{\mathrm{ad}}$ of design variables is that of admissible conductivity functions:

$$\mathcal{U}_{\mathrm{ad}} = \left\{ h \in L^\infty(D), \ \alpha \leq h(x) \leq \beta \text{ a.e. } x \in D \right\} \subset L^\infty(D),$$

where $0 < \alpha < \beta$ are fixed bounds.

- We wish to minimize the mean temperature within D under a volume constraint:

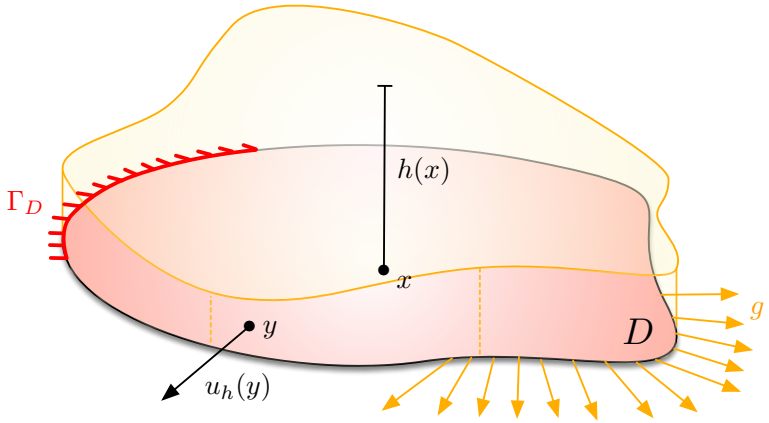
$$J(h) = \frac{1}{|D|} \int_D u_h \, \mathrm{d}x \text{ and } G(h) = \int_D h \, \mathrm{d}x - V_T.$$

Example 2: Optimization of the thickness of an elastic plate

- We optimize the thickness $h : D \to \mathbb{R}$ of an elastic plate with fixed cross-section $D \subset \mathbb{R}^d$.
- Its in-plane displacement $u_h \in H^1(D)^d$ satisfies:

$$\begin{cases} -\mathrm{div}(hAe(u_h)) = f & \text{in } D, \\ u_h = 0 & \text{on } \Gamma_D, \\ hAe(u_h)n = g & \text{on } \Gamma_N, \end{cases}$$

where $f \in L^2(D)^d$ and $g \in L^2(\Gamma_N)^d$ are body forces and traction loads.



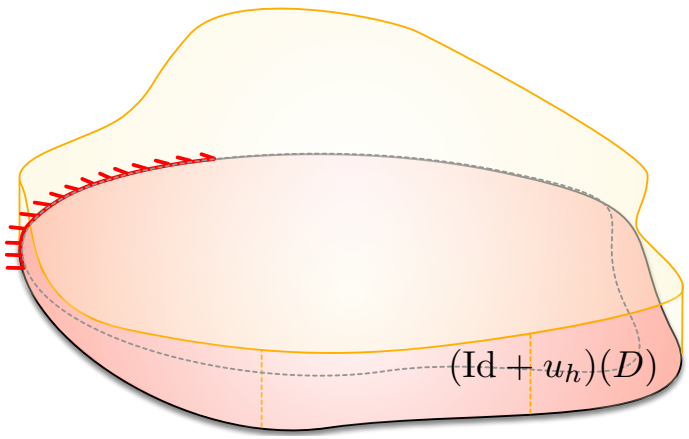
Reference configuration of the plate.

- We minimize the compliance of the plate

$$J(h) = \int_D hAe(u_h) : e(u_h)\,\mathrm{d}x,$$

under a volume constraint:

$$G(h) = \int_D h\,\mathrm{d}x - V_T.$$



Deformed configuration of the plate.

Parametric optimization problems (II)

The numerical treatment of such optimal design problems rests on:

1. The simulation of physical boundary-value problems on "complex" domains.

   ☞ Course 1.

2. The calculation of derivatives of functionals of the form

   $$J(h) = j(u_h), \text{ where } u_h \text{ is the solution to a PDE depending on } h.$$

   ☞ Course 2.

3. The implementation of these derivatives in an optimization algorithm.

   ☞ Courses 1 & 3.

Part II

Refresher about the Finite Element Method

The Finite Element Method

- The Finite Element Method is one method of choice for the numerical solution of partial differential equations.
- It builds on the variational formulations of boundary-value problems.
- Simple and efficient open-source softwares exist, that allow to solve complex problems in a few lines of code: [FreeFem], [Firedrake], [Fenics], [NGSolve]...
- We briefly introduce the main concepts, and the `FreeFem` environment.
- For more exhaustive presentations, consult the references [All], [Ci] or [ErnGue].

Variational formulations (I)

- As a model problem, we consider the Laplace equation in a domain $D \subset \mathbb{R}^d$:

$$\text{Search for } u \in H_0^1(D) \text{ s.t. } \begin{cases} -\Delta u = f & \text{in } D, \\ u = 0 & \text{on } \partial D, \end{cases}$$

  where $f \in L^2(D)$ is a given source.

- The associated variational formulation reads:

$$\boxed{\text{Search for } u \in V \text{ s.t. } \quad \forall v \in V, \ \ a(u,v) = \ell(v),}$$

  where

  - The Hilbert space V is the Sobolev space $H_0^1(D)$;
  - $a(\cdot,\cdot)$ is the coercive bilinear form on V given by: $a(u,v) = \displaystyle\int_D \nabla u \cdot \nabla v \, \mathrm{d}x$;
  - $\ell(\cdot)$ is the linear form on V defined by: $\ell(v) = \displaystyle\int_D fv \, \mathrm{d}x$.

- The above variational problem has a unique solution $u \in V$ owing to the Lax-Milgram theorem.

Variational formulations (II)

- The Finite Element Method consists in searching for an approximation u_h to h inside a finite-dimensional subspace $V_h \subset V$.

- The exact variational problem is replaced by:

$$\boxed{\text{Search for } u_h \in V_h \text{ s.t. } \quad \forall v_h \in V_h, \quad a(u_h, v_h) = \ell(v_h),}$$

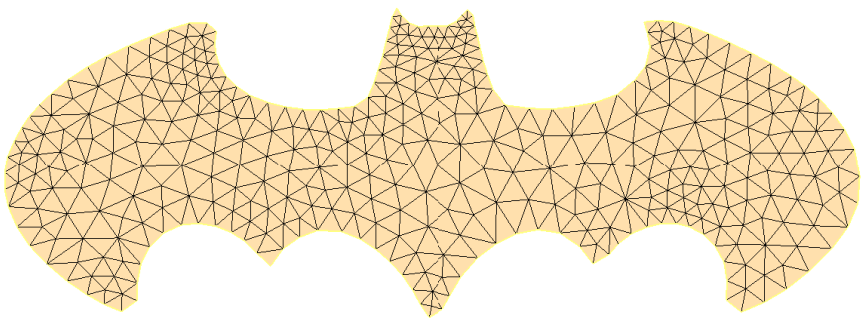
  which is also well-posed owing to the Lax-Milgram theorem.

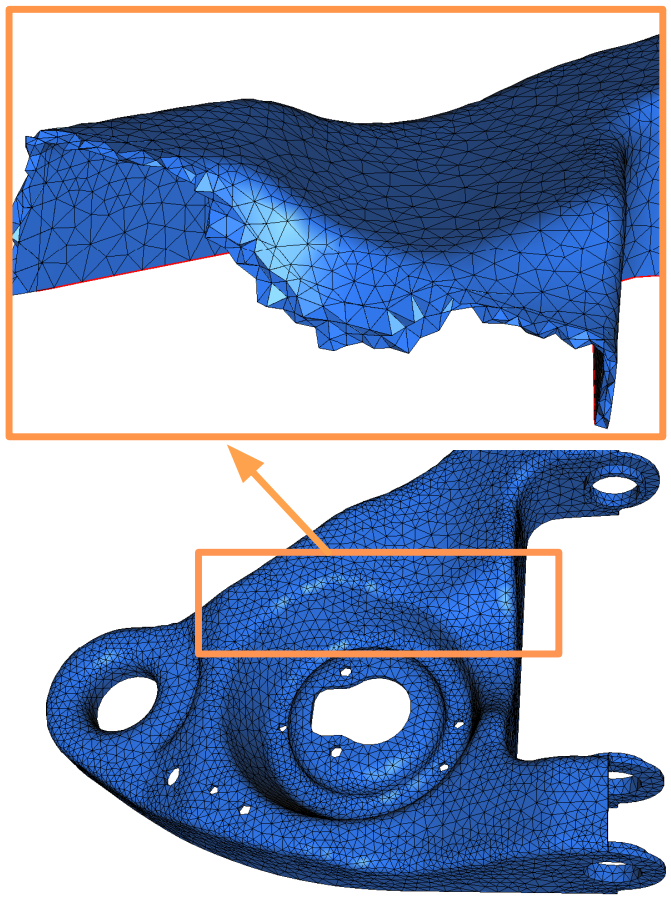
- The subscript h refers to the sharpness of the approximation: as $h \to 0$, it is expected that $V_h \approx V$ and $u_h \approx u$.

- The Finite Element space V_h is constructed from a mesh of the domain D.

Meshing the physical domain (I)

In practice, the domain D is discretized by means of a mesh $\mathcal{T}$, i.e. a covering by simplices (triangles in 2d, tetrahedra in 3d).





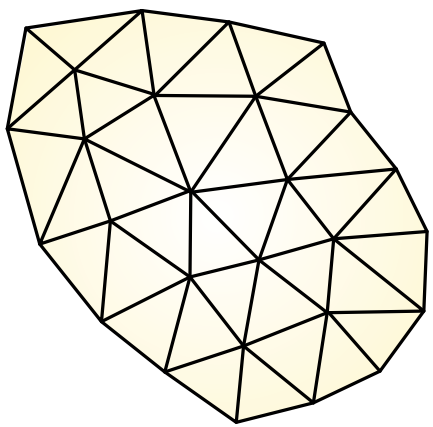
Meshing the physical domain (II)

A mesh $\mathcal{T}$ is defined by the datum of:

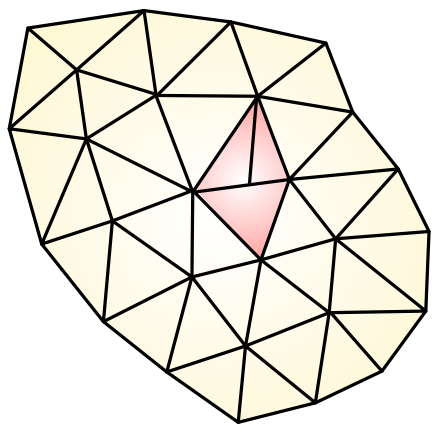
- A set of vertices $\{a_i\}_{i=1,\dots,N_V}$;
- A set of (closed) simplices $\{T_j\}_{j=1,\dots,N_T}$, with vertices in $\{a_i\}$.

We also require that the mesh $\mathcal{T}$ be:

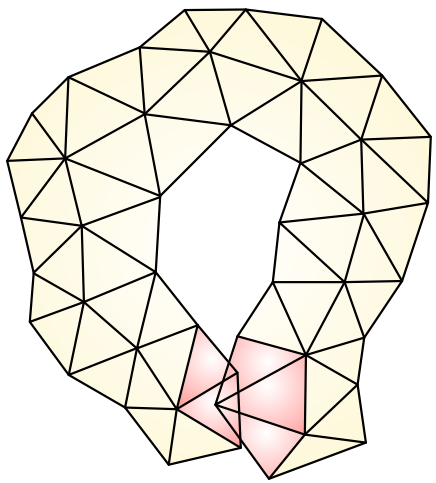
- Valid: For all simplices T_i, T_j with $i \neq j$, $\mathring{T}_i \cap \mathring{T}_j = \emptyset$.
- Conforming: For distinct simplices T_i, T_j, $i \neq j$, the intersection $T_i \cap T_j$ is either a vertex, or an edge (or a triangle in 3d) of $\mathcal{T}$.



Valid, conforming mesh



Non conforming mesh



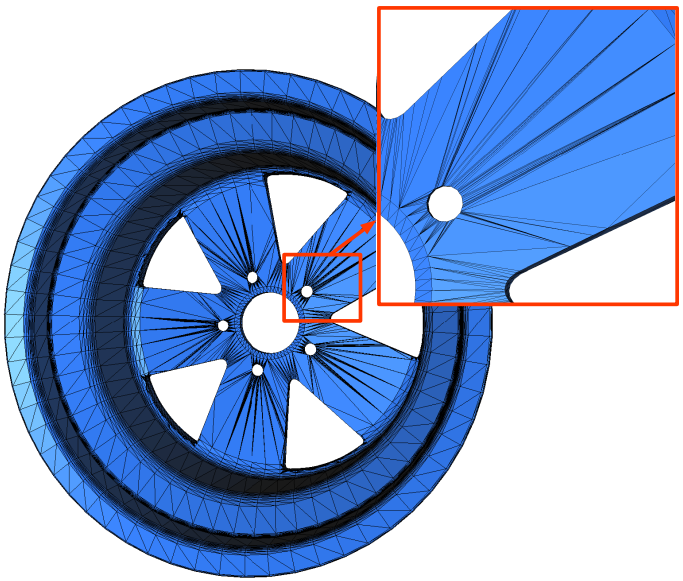
Invalid mesh

Meshing the physical domain (III)

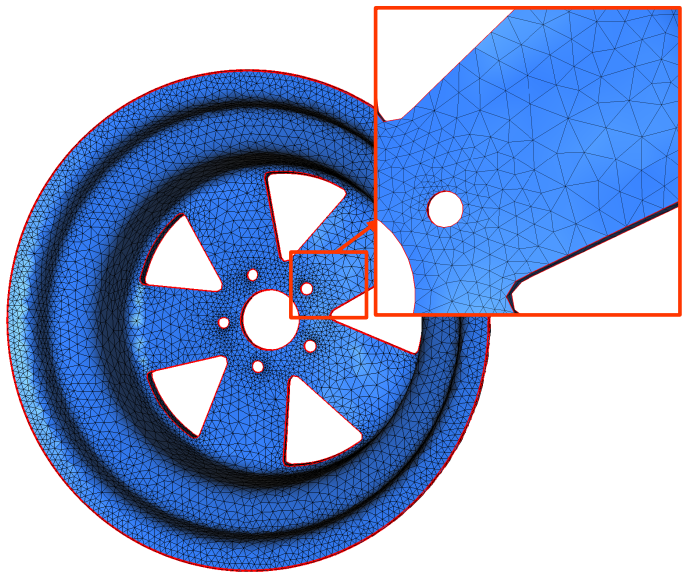
- In applications, it is crucial that $\mathcal{T}$ have good quality, i.e. that its elements be close to equilateral.
- The quality of a simplex T can be evaluated e.g. by the function:

$$Q(T) = \alpha \frac{\mathrm{Vol}(T)}{\left(\sum_{j=1}^{d(d+1)/2} |e_j|^2 \right)^{\frac{d}{2}}}, \quad e_j = \text{ edges of } T,$$

where $\alpha \in \mathbb{R}$ is such that $Q(T) = 1$ if T is equilateral and $Q(T) = 0$ if T is flat.



Bad quality mesh, with nearly flat elements



Good quality mesh, with almost regular elements

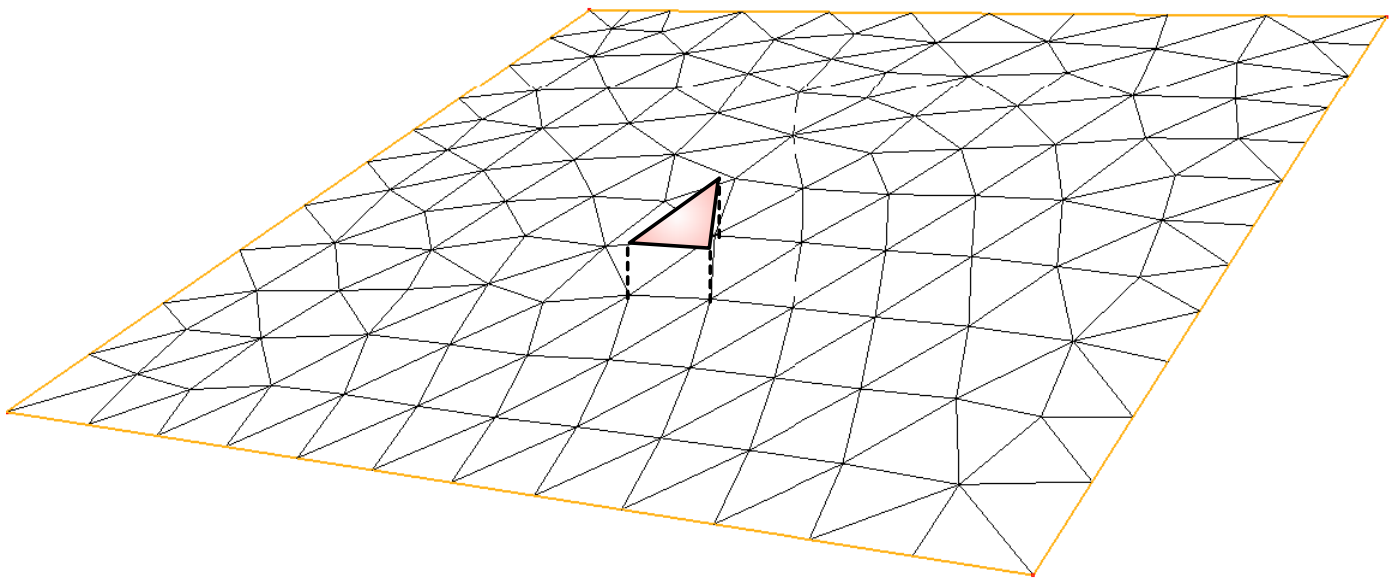
Construction of the finite element space V_h (I)

- The mesh $\mathcal{T}_h$ is labelled by the maximum size h of its elements.
- The Finite Element space V_h and a basis $\{\varphi_1, ..., \varphi_{N_h}\}$ are defined according to $\mathcal{T}_h$.

Example: the $\mathbb{P}_0$ Finite element method

- N_h is the number N_T of simplices $T_1, ..., T_{N_h}$ in the mesh;
- For $i = 1, ..., N_h$, φ_i is constant on each simplex $T \in \mathcal{T}_h$ and

$$\varphi_i(x) = 1 \text{ on } T_i \text{ and } \varphi_i(x) = 0 \text{ for } x \notin T_i.$$

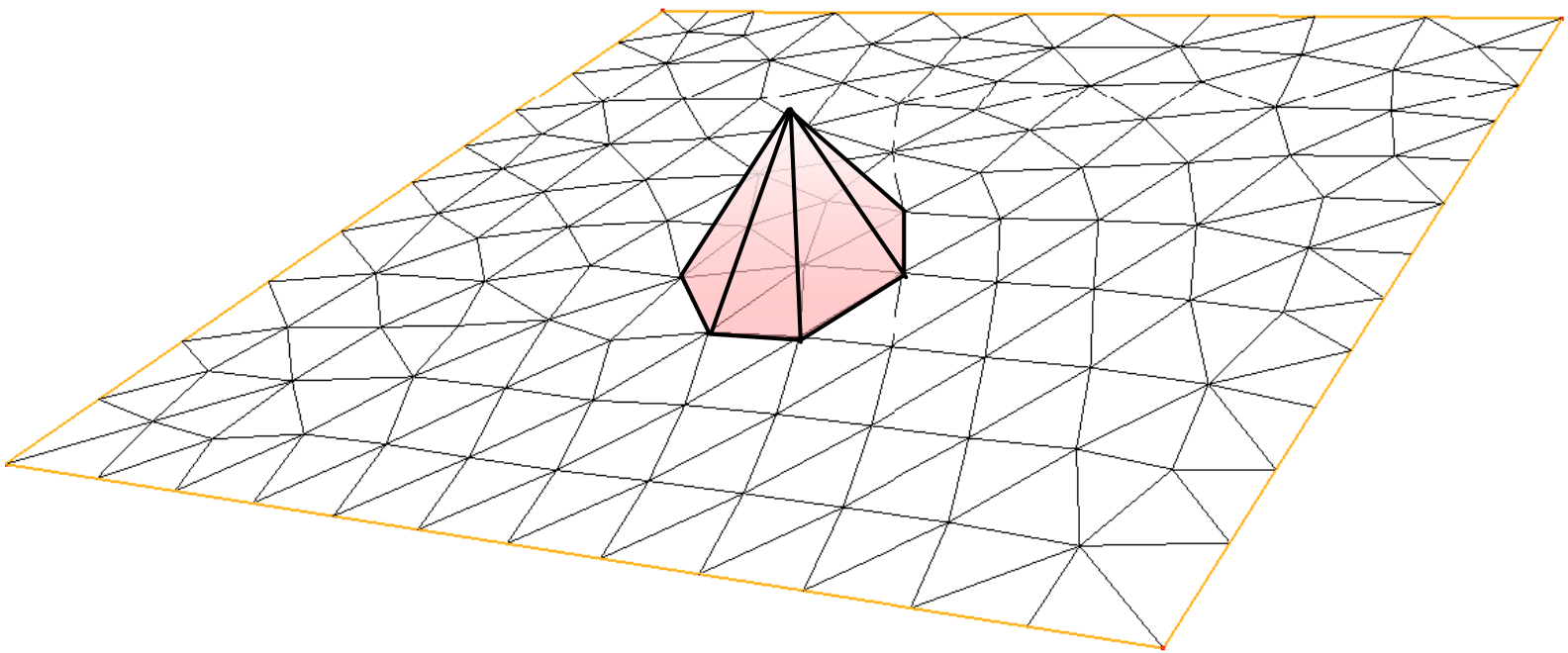


Construction of the finite element space V_h (II)

Example: the $\mathbb{P}_1$ Finite element method

- N_h is the number N_V of vertices $a_1, ..., a_{N_h}$ of the mesh;
- For $i = 1, ..., N_h$, φ_i is affine in restriction to each element $T \in \mathcal{T}_h$,

$$\varphi_i(a_i) = 1 \text{ and } \varphi_i(a_j) = 0 \text{ for } j \neq i.$$



Reduction to an algebraic system

Introducing the (sought) decomposition of the (sought) function u_h on this basis:

$$u_h = \sum_{j=1}^{N_h} u_j \varphi_j,$$

the variational problem becomes an $N_h \times N_h$ linear system:

$$KU = F,$$

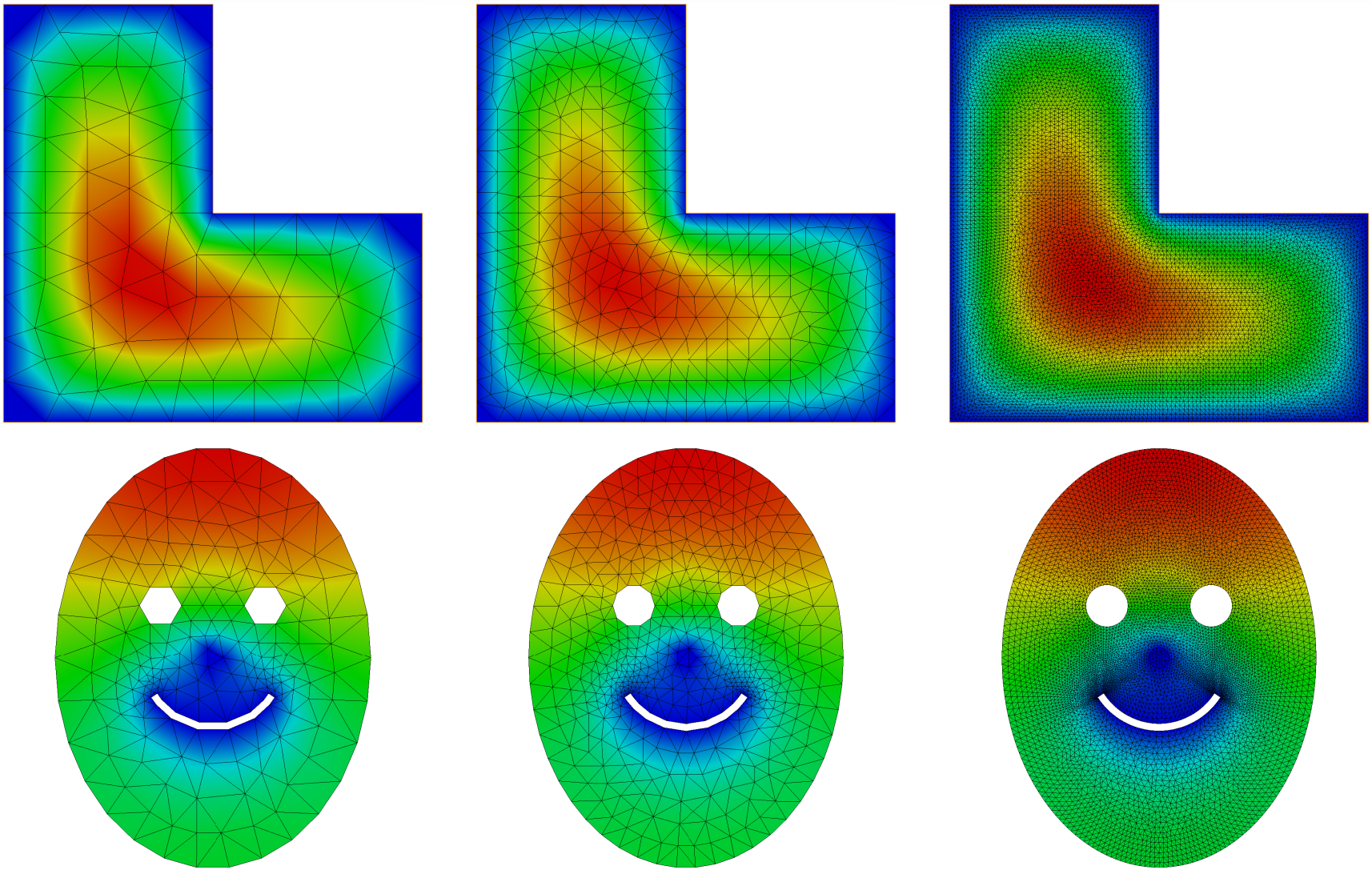
where

- $U = \begin{pmatrix} u_1 \\ \vdots \\ u_{N_h} \end{pmatrix} \in \mathbb{R}^{N_h}$ is the vector of unknowns,
- $K \in \mathbb{R}^{N_h \times N_h}$ is the stiffness matrix, defined by its entries:

$$K_{ij} = a(\varphi_j, \varphi_i), \quad i, j, = 1, \ldots, N_h;$$

- $F \in \mathbb{R}^{N_h}$ is the right-hand side vector: $F_i = \ell(\varphi_i)$.

A few numerical solutions



Solution of the Laplace equation by the Finite Element Method on several domains D, using various meshes $\mathcal{T}_h$.

- In practice, the discrete finite element system

$$KU = F$$

is a large $N_h \times N_h$ linear system, which is often sparse.

Example: Using $\mathbb{P}_1$ Finite Elements,

If a_i and a_j are not neighbours, $a(\varphi_j, \varphi_i) = 0 \Rightarrow$ "Most" entries K_{ij} equal 0.

- In realistic examples, its resolution is achieved thanks to iterative methods, such as the Conjugate Gradient algorithm, GMRES, etc.

- The numerical efficiency of these methods depends on the condition number of the matrix K, which is, in turn, related to the quality of the computational mesh.

Final remarks about the finite element method

The Finite Element paradigm extends (with some work!) to various frameworks:

- Mixed variational formulations, like in the case of the Stokes equations;
- Eigenvalue problems;
- Non linear PDE, such as the Navier-Stokes equations, or the non linear elasticity system.

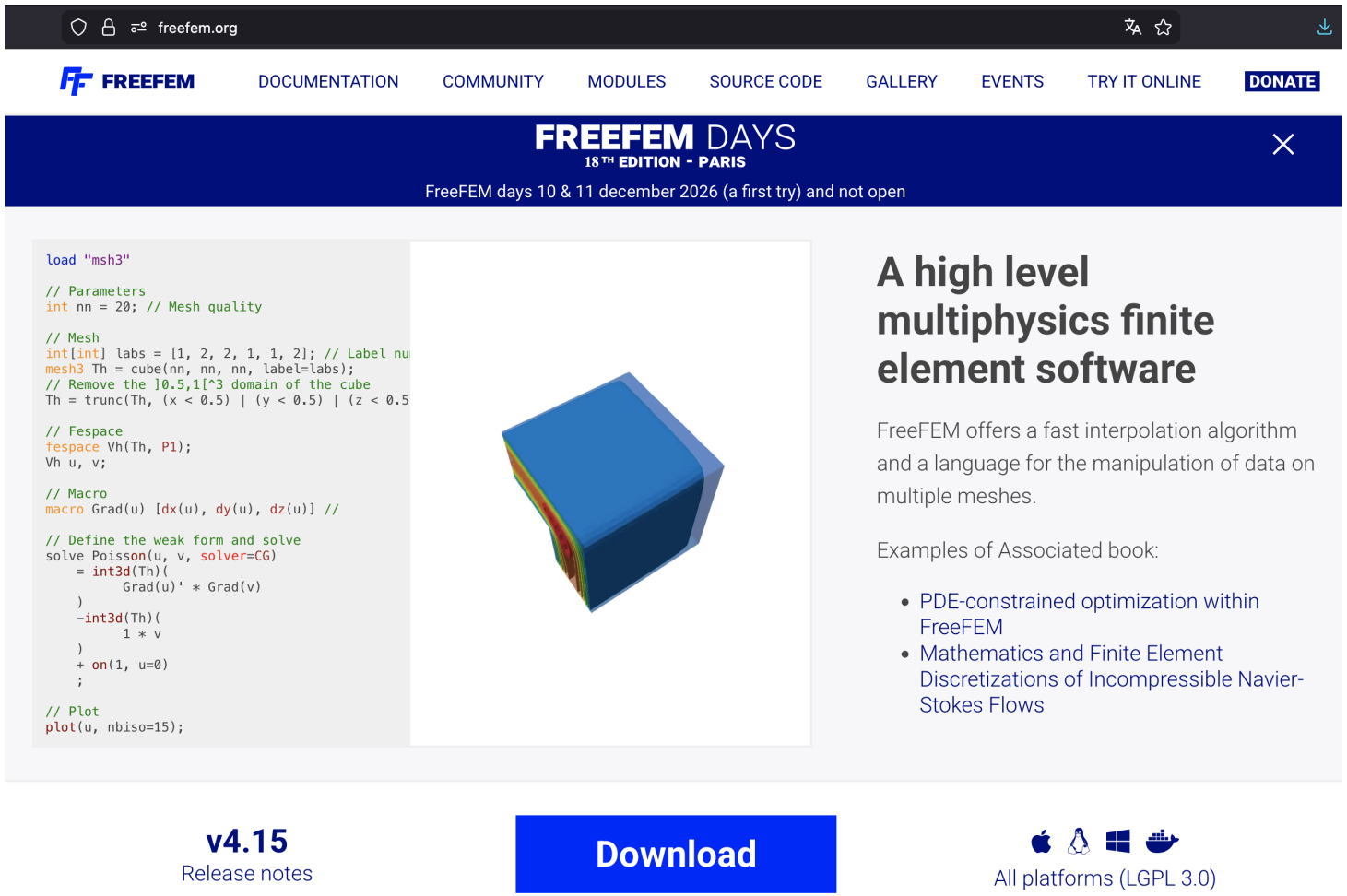
Part II

Refresher about the Finite Element Method

A convenient Finite Element environment: FreeFem

FreeFem is a user-friendly software for solving PDE with the Finite Element Method.

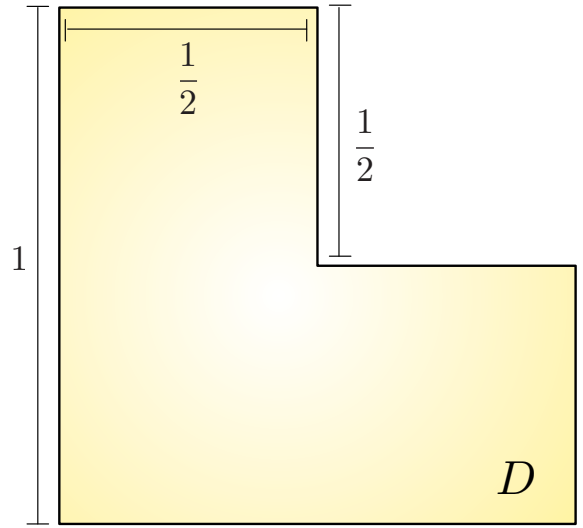
https://https://freefem.org/



A worked example with FreeFem (I)

- We solve the Laplace equation in an L-shaped domain.

$$\begin{cases} -\Delta u = f & \text{in } D, \\ u = 0 & \text{on } \partial D. \end{cases}$$



- The variational formulation of the problem reads:

$$\text{Search for } u \in H_0^1(D) \text{ s.t. for all } v \in H_0^1(D), \quad \int_D \nabla u \cdot \nabla v \, \mathrm{d}x = \int_D f v \, \mathrm{d}x.$$

- After introduction of a mesh $\mathcal{T}_h$ of D and a Finite Element space V_h, the discrete problem reads:

$$\boxed{\text{Search for } u_h \in V_h \text{ s.t. for all } v_h \in V_h, \quad \int_D \nabla u_h \cdot \nabla v_h \, \mathrm{d}x = \int_D f v_h \, \mathrm{d}x.}$$

A worked example with FreeFem (II)

The domain D is defined by

1. Providing a parametrization of its boundary as input with the keyword `border`;
2. Creating a mesh of D with the command `buildmesh()`.

```
/* 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");
```

A worked example with FreeFem (III)

- The Finite Element space V_h is defined thanks to the keyword `fespace`.
- The source term f is an analytical function.

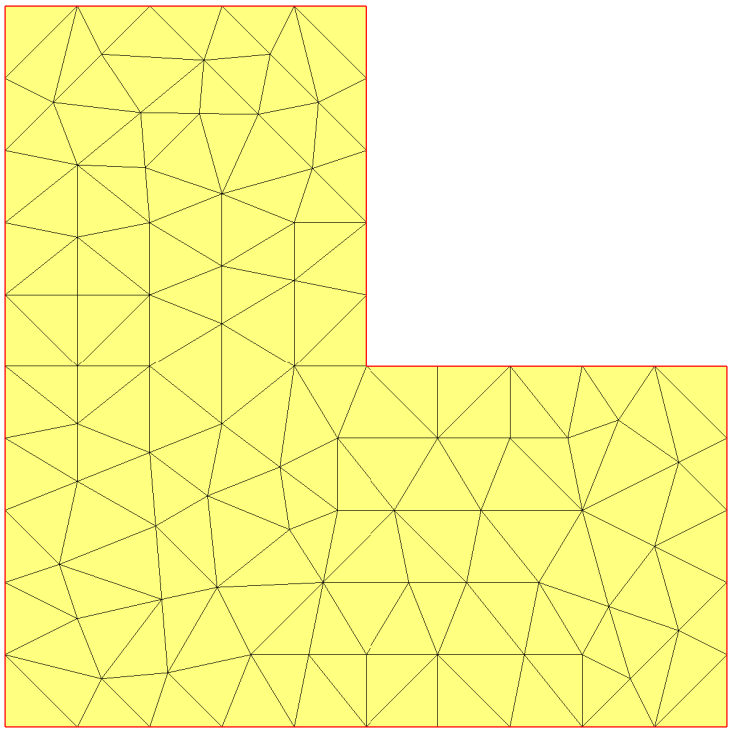
```
/* Definition of Finite Element spaces and functions */
fespace Vh(Th,P1); // or P2
Vh u,v; // u,v are declared as elements in Vh

/* Source term; the spatial coordinates are
                                    undisclosed arguments */
func real f() {
  return(1.0);
}
```

A worked example with FreeFem (IV)

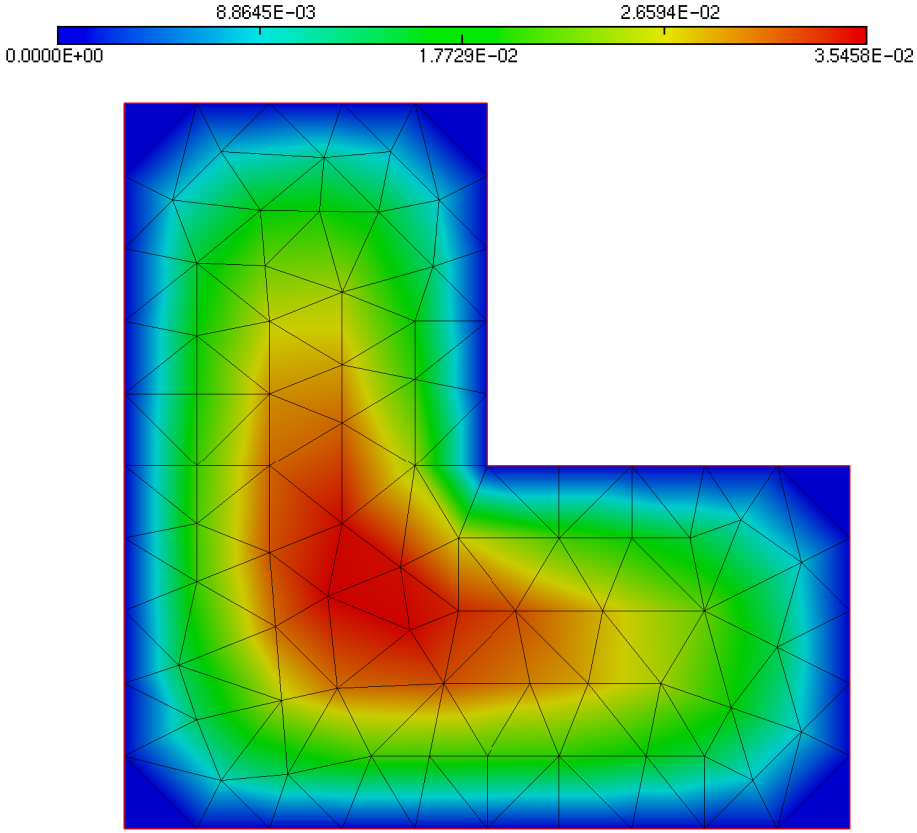
- The Finite Element problem is declared via its variational formulation, using the keyword problem.
- It is then solved from the input of its name as command.

```
/* Laplace equation (solver = Conjugate Gradient) */
problem laplace(u,v,solver=CG) =
            int2d(Th)(dx(u)*dx(v)+dy(u)*dy(v))
          - int2d(Th)(f()*v)
          + on(1,u=0.0);   // Homogeneous Dirichlet B.C.

/* Resolution of the problem */
laplace;

/* Display and save the result */
savemesh(Th,"Lshape.mesh");
savesol("Lshape.sol",Th,u);
```

Results



Mesh of the domain D

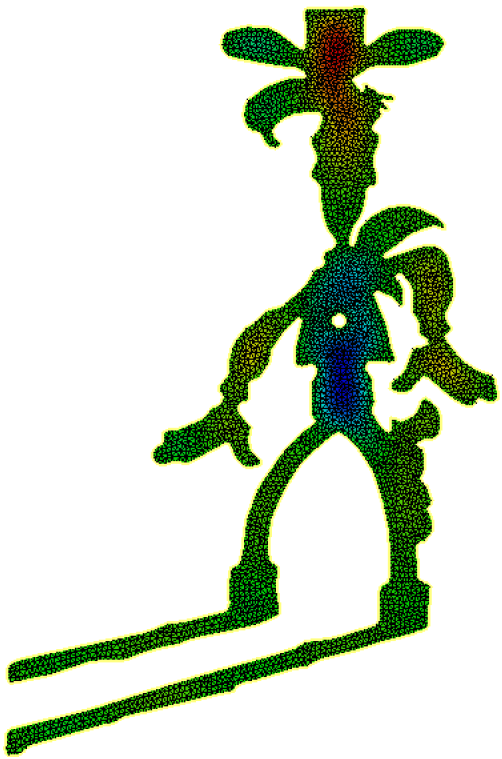


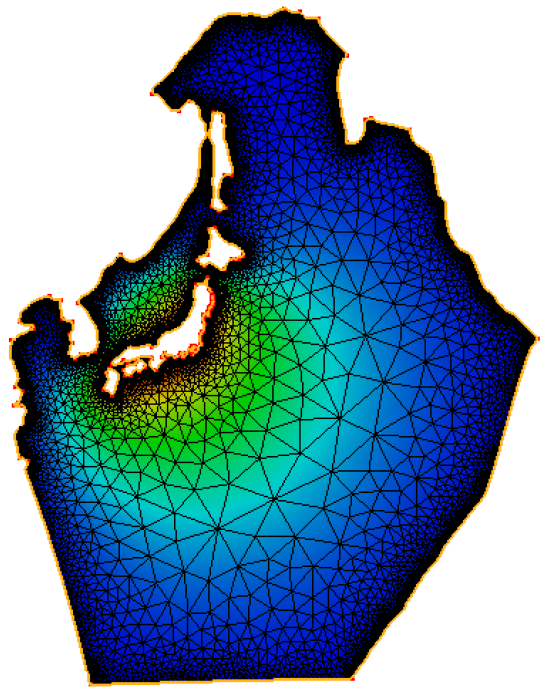
Solution of the Laplace equation

Try it yourselves!

Try to adapt these recipes to solve Finite Element problems involving:

- Different computational domains D;
- Finer meshes, different Finite Element spaces;
- Different right-hand sides functions;
- Different physical problems (heterogeneous conductivity, time dependence...).





Solutions of the Laplace equation on different domains, with different boundary conditions and right-hand sides.

Part III

Numerical optimization

- The considered optimal design problems are of the form:

$$\min_{h \in H} J(h) \quad \text{s.t.} \quad G(h) = 0.$$

In this formulation,

  - The design variable h lies in a possibly infinite-dimensional vector space H;
  - The evaluations of the objective and constraint functions $J(h)$ and $G(h)$ are costly (as they depend on the solution of h-dependent PDEs).

.

- Their numerical solution proceeds by iterations, of the form

$$h^{n+1} = h^n + \tau^n \widehat{h}^n,$$

where

  - The descent direction $\widehat{h}^n$ is obtained from the derivatives of J and G.
  - The pseudo-time step τ^n is "small enough".

Local and global minimizers (I)

Let X be a normed vector space, and let $J : X \to \mathbb{R}$; we consider the unconstrained minimization problem:

$$\min_{u \in X} J(u). \tag{UC}$$

Definition (Global minimizer).

A point $u \in X$ is a global minimizer for (UC) if:

$$\forall v \in X, \;\; J(u) \leq J(v).$$

Definition (Local minimizer).

A point $u \in X$ is a local minimizer for (UC) if there exists an open neighborhood $U \subset X$ containing u such that:

$$\forall v \in U, \;\; J(u) \leq J(v).$$

Local and global minimizers (II)

Let X be a normed vector space, and let $J : X \to \mathbb{R}$ and $G : X \to \mathbb{R}^p$; we consider the equality-constrained minimization problem:

$$\min_{u \in X} J(u) \text{ s.t. } G(u) = 0. \tag{EC}$$

Definition (Global minimizer).

A point $u \in X$ is a global minimizer for (EC) if:

$$\forall v \in X \text{ s.t. } G(v) = 0, \ J(u) \leq J(v).$$

Definition (Local minimizer).

A point $u \in X$ is a local minimizer for (EC) if there exists an open neighborhood $U \subset X$ containing u such that:

$$\forall v \in U \text{ s.t. } G(v) = 0, \ J(u) \leq J(v).$$

Derivatives and gradients (I)

Definition (Fréchet derivative).

Let $(X, ||\cdot||_X)$ be a Banach space. A real-valued function $F : X \to \mathbb{R}$ is differentiable at $u \in X$ if there exists a linear, continuous mapping $F'(u) : X \to \mathbb{R}$ such that:

$$F(u+v) = F(u) + F'(u)(v) + o(||v||), \text{ where } \frac{o(||v||_X)}{||v||_X} \xrightarrow{v \to 0} 0.$$

The linear mapping $F'(u) \in X^*$ is the Fréchet derivative of F at u.

Definition (Gradient).

If in addition X is a Hilbert space $(H, \langle \cdot, \cdot \rangle_H)$, the Riesz representation theorem allows to identify the derivative $F'(u)$ with an element $\nabla F(u) \in H$:

$$\forall v \in H, \quad F'(u)(v) = \langle \nabla F(u), v \rangle_H;$$

$\nabla F(u)$ is called the gradient of F at u.

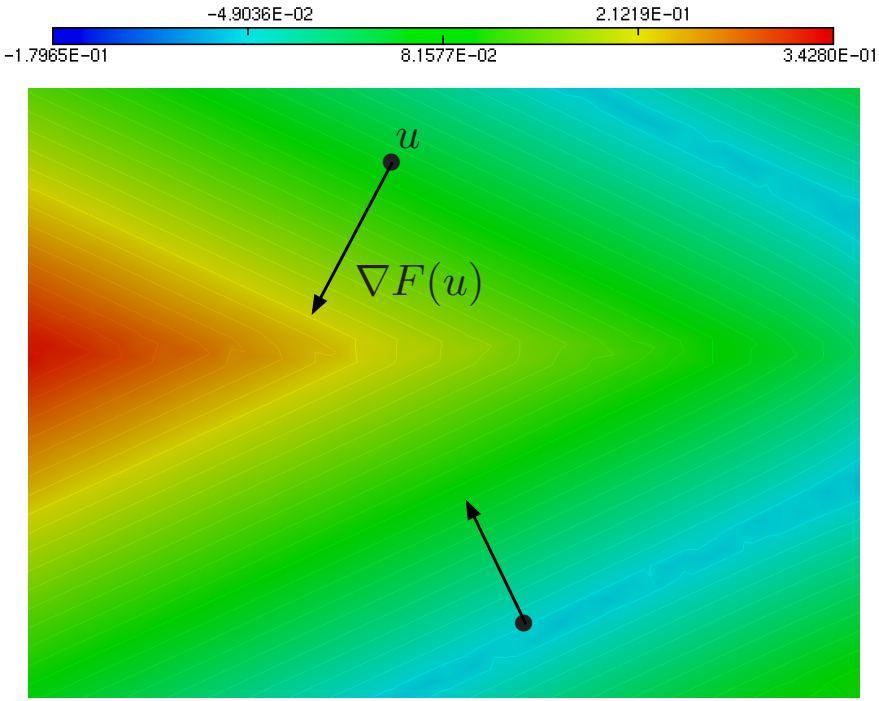
Derivatives and gradients (II)

Physical interpretation: If F is differentiable at $u \in H$, it holds, for "small" $\tau > 0$:

$$\begin{array}{rcl} \forall \widehat{u} \in H,\ ||\widehat{u}||_H \leq 1, \quad F(u+\tau\widehat{u}) & \approx & F(u) + \tau\langle \nabla F(u), \widehat{u} \rangle_H, \\ & \leq & F(u) + \tau||\nabla F(u)||_H, \end{array}$$

where equality holds if and only if $\widehat{u} = \frac{\nabla F(u)}{||\nabla F(u)||_H}$ (Cauchy-Schwarz inequality).

$\Rightarrow \nabla F(u)$ (resp. $-\nabla F(u)$) is the best ascent (resp. descent) direction for F from u.



Some isolines of a function $F : \mathbb{R}^2 \to \mathbb{R}$ and the gradient $\nabla F(u) \in \mathbb{R}^2$ at some point $u \in \mathbb{R}^2$.

The gradient algorithm (I)

In a Hilbert space H, we consider the unconstrained minimization problem:

$$\min_{h \in H} J(h),$$

where $J(h)$ is a differentiable function.

Initialization: Start from an initial design h^0.

For $n = 0, \ldots$ **convergence:**

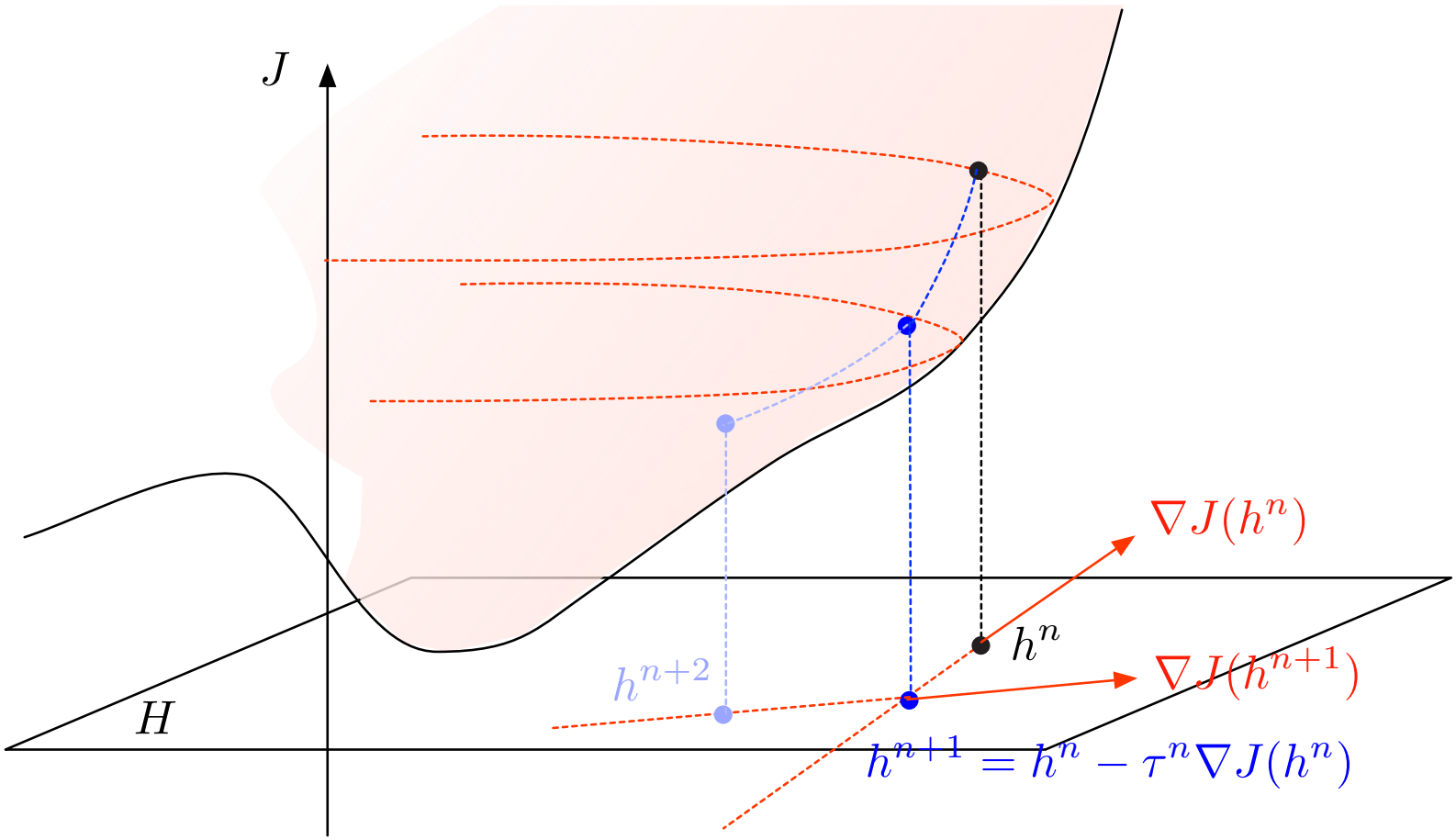
1. Calculate the derivative $J'(h^n)$ of J at h^n and the gradient $\nabla J(h^n) \in H$; infer a descent direction $\widehat{h^n} = -\nabla J(h^n)$.
2. Select a suitably small time step $\tau^n > 0$ such that:

$$J(h^n + \tau^n \widehat{h^n}) < J(h^n).$$

3. The next iterate is $h^{n+1} = h^n + \tau^n \widehat{h^n}$.

Return: h^n.

The gradient algorithm (II)



The gradient algorithm proceeds by successive steps in the negative direction of the gradient of $J(h)$.

Remark: The gradient algorithm converges to a local minimizer of the problem.

The augmented Lagrangian algorithm (I) [NoWri]

- Let us now consider a problem featuring one (for simplicity) equality constraint:

$$\min_{h \in H} J(h) \text{ s.t. } G(h) = 0, \qquad \text{(EC)}$$

where $J : H \to \mathbb{R}$ and $G : H \to \mathbb{R}$ are differentiable.

- One possibility is to replace this problem with an unconstrained one:

$$\min_{h \in H} J(h) + \ell G(h),$$

in which $J(h)$ is penalized by the constraint $G(h)$, using a fixed weight $\ell > 0$.

- In practice, the "suitable" value ℓ^* for ℓ, i.e. that driving the optimization process to the desired level of constraint $G(h) = 0$, is estimated after trial and errors.

- This unknown value ℓ^* can be interpreted as the Lagrange multiplier associated to the constraint $G(h) = 0$ at the obtained local minimum.

The augmented Lagrangian algorithm (II)

- For $\ell \in \mathbb{R}$ and $b > 0$, let us define the augmented Lagrangian:

$$\mathcal{L}(h, \ell, b) := \underbrace{J(h) + \ell G(h)}_{\approx \text{ Lagrangian of the constrained problem}} + \frac{b}{2} G(h)^2,$$

  where

  - ℓ represents the Lagrange multiplier for the constraint $G(h) = 0$;
  - b is a weight for the quadratic penalization of the constraint function $G(h)$.

- The augmented Lagrangian algorithm reduces the resolution of (EC) to that of a series of unconstrained problems:

$$\min_{h \in H} \ \mathcal{L}(h, \ell^n, b^n), \quad n = 0, \ldots,$$

  in which ℓ^n and b^n are updated so that $G(h) = 0$ holds at convergence.

The augmented Lagrangian algorithm (III)

Initialization: Initial design h^0, initial parameters ℓ^0, b^0.

For $n = 0, \ldots$ **convergence:**

1. Solve the unconstrained optimization problem:

$$\min_{h \in H} \; J(h) + \ell^n G(h) + \frac{b^n}{2} G(h)^2,$$

   starting from h^n to obtain h^{n+1}.

2. Update the optimization parameters via:

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

As such, the algorithm is very slow, as it consists in a series of complete resolutions of unconstrained problems.

The augmented Lagrangian algorithm (IV)

A "pragmatic" version involves fewer evaluations of $J(h)$, $G(h)$ and $J'(h)$, $G'(h)$.

Initialization: Initial design h^0, initial parameters ℓ^0, b^0.

For $n = 0, \ldots$ **convergence:**

1. Calculate a descent direction $\widehat{h^n}$ for the functional:

$$h \mapsto \mathcal{L}(h, \ell^n, b^n) := J(h) + \ell^n G(h) + \frac{b^n}{2} G(h)^2.$$

2. Select a suitably small time step τ^n so that:

$$\mathcal{L}(h^n + \tau^n \widehat{h^n}, \ell^n, b^n) < \mathcal{L}(h^n, \ell^n, b^n).$$

3. Update the design: $h^{n+1} = h^n + \tau^n \widehat{h^n}$.

4. Update ℓ^n and b^n via:

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

Part III

Numerical optimization

The L^2 image denoising model (I)

- Let D be a square in $\mathbb{R}^2$, and let $u_T : D \to [0,1]$ be a noisy, black-and white image.

Noisy photo of a cosmonaut

Denoised image [Ma]

- We search for a denoised version u of u_T by solving the unconstrained problem:

$$\min_{u \in H^1(D,[0,1])} J(u), \text{ where } J(u) = \underbrace{\frac{1}{2}\int_D |u - u_T|^2 \, \mathrm{d}x}_{\text{Data fitting term}} + \frac{\lambda}{2} \underbrace{\int_D |\nabla u|^2 \, \mathrm{d}x}_{\text{Regularization term}} ;$$

  - The data-fitting term imposes that u "resemble" u_T;
  - The regularizing term penalizes "parasitic" large intensity variations.

Gradient-based resolution of the L^2 denoising model (I)

The gradient-based resolution of this problem hinges on the derivative of $J(u)$.

Proposition.

The functional $J(u)$ is differentiable at any $u \in H^1(D, [0,1])$, with derivative:

$$\forall h \in H^1(D), \quad J'(u)(h) = \int_D (u - u_T)h \, \mathrm{d}x + \lambda \int_D \nabla u \cdot \nabla h \, \mathrm{d}x.$$

Proof: For an arbitrary function $h \in H^1(D)$, it holds:

$$\begin{aligned} J(u+h) - J(h) &= \int_D (u - u_T)h \, \mathrm{d}x + \lambda \int_D \nabla u \cdot \nabla h \, \mathrm{d}x \\ &\quad + \frac{1}{2} \int_D h^2 \, \mathrm{d}x + \frac{\lambda}{2} \int_D |\nabla h|^2 \, \mathrm{d}x \\ &= \int_D (u - u_T)h \, \mathrm{d}x + \lambda \int_D \nabla u \cdot \nabla h \, \mathrm{d}x + \mathrm{o}(||h||_{H^1(D)}), \end{aligned}$$

which yields the result, by definition of the Fréchet derivative. □

Gradient-based resolution of the L^2 denoising model (II)

- The identification of the gradient $g \in H^1(D)$ of $J(u)$ from the derivative $J'(u)$ arises as a variational problem:

  Search for $g \in H^1(D)$ s.t. $\forall v \in H^1(D)$,

$$\underbrace{\alpha^2 \int_D \nabla g \cdot \nabla v \, \mathrm{d}x + \int_D g v \, \mathrm{d}x}_{\approx H^1(D) \text{ inner product}} = J'(u)(v). \quad \text{(G)}$$

- In principle, any inner product on $H^1(D)$ can be used at the left-hand side.
- In practice, however, this results in different gradients.
- In (G), α acts as a regularization length-scale for the information in $J'(u)$.
- This issue is crucial in the practice of gradient flows, see [Neu] and `Course 3`.

The algorithm

- The computational domain D is discretized by a triangular mesh $\mathcal{T}$.
- The images u_T and u^n are defined as $\mathbb{P}_1$ Finite Element functions on $\mathcal{T}$, $n = 0, \ldots$

Initialization: Mesh $\mathcal{T}$ of D, image u_T and initial guess u^0 (e.g. $u^0 = u_T$).

For $n = 0, \ldots$ **convergence:**

1. Calculate a gradient g^n of J at u^n by solving the Finite Element problem:

$$\forall v \in H^1(D), \quad \alpha^2 \int_D \nabla g^n \cdot \nabla v \, \mathrm{d}x + \int_D g^n v \, \mathrm{d}x = \int_D (u^n - u_T) v \, \mathrm{d}x + \lambda \int_D \nabla u^n \cdot \nabla v \, \mathrm{d}x.$$

2. Select a suitably small time step $\tau^n > 0$ so that:

$$J(u^n - \tau^n g^n) < J(u^n).$$

3. Update the image: $u^{n+1} = u^n - \tau^n g^n$.

Sketch of a FreeFem implementation (I)

```
/* Define mesh and images */
mesh Th;
fespace Vh(Th,P1);
Vh u,uT;

/* Load Th and uT */
[...]

/* Macro for calculating the energy functional */
macro J(uu) ( 0.5*int2d(Th)((uu-uT)^2)
              + lm*int2d(Th)(dx(uu)^2+dy(uu)^2) ) // EOM

/* Calculation of the gradient of J as a P1 function */
macro dJ(uu) {
  solve gradJ(g1,v) =
  int2d(Th)( alpha^2*(dx(g1)*dx(v)+dy(g1)*dy(v)) + g1*v )
 - int2d(Th)( (uu-uT)*v + lm*(dx(uu)*dx(v)+dy(uu)*dy(v)) );
} // EOM
```

Sketch of a FreeFem implementation (II)

```
/* Main loop */
for (int n=1; n<=MAXIT; n++) {
  /* Calculation of gradient */
  dJ(u);

  /* Update of the optimized function */
  gmax = max(-g1[].min,g1[].max);
  newu = u - coef/gmax*g1;

  /* Thresholding between 0 and 1 */
  thres(newu);

  /* Evaluation of the new objective */
  newobj = J(newu);

  /* Decision: accept if objective has decreased */
  if ( newobj < obj ) [...]
  else  [...]
}
```

Results

Original black-and-white image.

Denoised image after application of the L^2 regularization model.

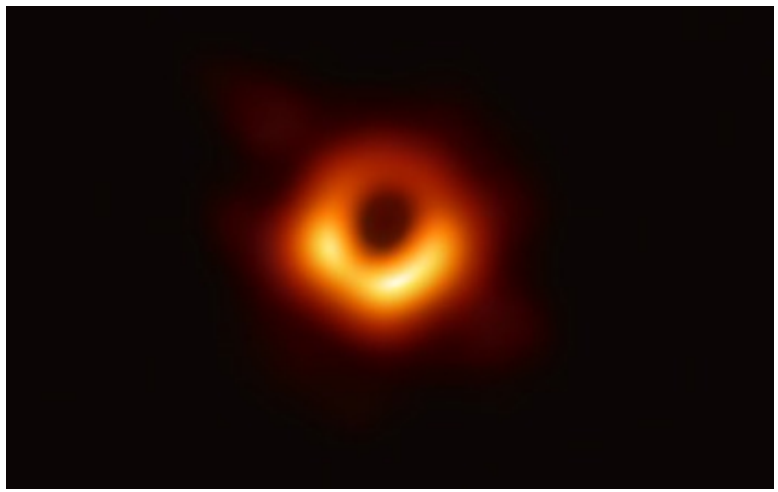
The Rudin-Osher-Fatemi model

- This L^2 denoising model yields undesirably blurred functions u.
- This effect is induced by the penalization by the L^2 norm of the gradient of u.
- The Rudin-Osher-Fatemi model considers the alternative problem:

$$\min_{u \in H^1(D,[0,1])} J(u), \text{ where } J(u) = \frac{1}{2} \int_D |u - u_T|^2 \, \mathrm{d}x \; + \; \lambda \int_D |\nabla u| \, \mathrm{d}x,$$

in which the use of the L^1 norm is meant to promote sparsity of the solution.

- This historical model has been used to denoise the first images of black holes [Wi]!



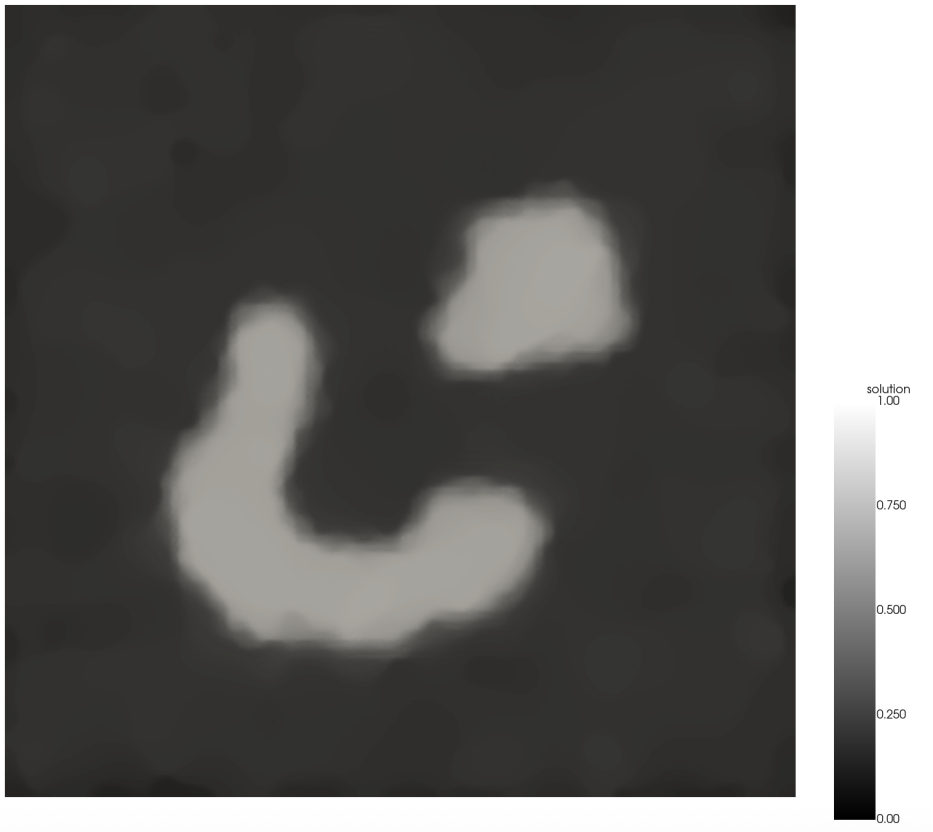
Credit: Event Horizon Telescope collaboration et al.

L. I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: nonlinear phenomena, 60(1-4), (1992). pp. 259–268.

Results

☞ Devise a gradient method for minimizing the Rudin-Osher-Fatemi functional.

Original black-and-white image.



Denoised image after application of the Rudin-Osher-Fatemi model.

Overview of the next lectures

- `Course 2` Theoretical ingredients for parametric optimization.
  - Generalities about parametric optimization;
  - Differentiation by the adjoint method.

- `Course 3` Numerical methods for parametric optimization.
  - Numerical implementation recipes;
  - Density-based topology optimization.

- `Practical session` A `FreeFem` implementation of the SIMP method.

Thank you for your attention!

Technical appendix

The Lax Milgram theorem

In a Hilbert space H, let $a : H \times H \to \mathbb{R}$ be a bilinear form and $\ell : H \to \mathbb{R}$ be a linear form such that:

- a is continuous, i.e. there exists $M > 0$ such that:
$$\forall u, v \in H, \ |a(u, v)| \leq M||u||_H||v||_H.$$
- a is coercive, i.e. there exists $\alpha > 0$ such that:
$$\forall u \in H, \ \alpha||u||_H^2 \leq a(u, u).$$
- ℓ is continuous (i.e. ℓ belongs to the dual space H^*):
$$||\ell||_{H^*} := \sup_{\substack{v \in H \\ v \neq 0}} \frac{|\ell(v)|}{||v||_H} < \infty.$$

Theorem (Lax-Milgram).

Under the above hypotheses, the variational problem

$$\text{Search for } u \in H \text{ s.t. for all } v \in H, \ a(u, v) = \ell(v)$$

has a unique solution $u \in H$, which depends continuously on ℓ:

$$||u||_H \leq \frac{M}{\alpha}||\ell||_{H^*}.$$

Fréchet and Gateaux derivatives

Several notions of derivative are available for a function $F : U \to V$ between two normed vector spaces $(U, ||\cdot||_U)$ and $(V, ||\cdot||_V)$.

Definition (Fréchet differentiability).

- *A function $F : U \to V$ is called Fréchet differentiable at some point $x \in U$ if there exists a linear, continuous mapping $L_x : U \to V$ such that:*

$$F(x+v) = F(x) + L_x(v) + \mathrm{o}(||v||_U), \text{ where } \frac{||\mathrm{o}(||v||_U)||_V}{||v||_U} \xrightarrow{v \to 0} 0.$$

- *The mapping $v \mapsto L_x(v)$ is denoted by $v \mapsto F'(x)(v)$, or $\mathrm{d}_x F(v)$ and is called the Fréchet derivative of F at x.*

- *The function $F : U \to V$ is called Gateaux differentiable at $x \in U$ if for any direction $v \in U$, the following limit exists:*

$$\lim_{\substack{t \to 0 \\ t > 0}} \frac{F(x+tv) - F(x)}{t}.$$

Remark: The notion of Fréchet differentiability is stronger than that of Gateaux differentiability, which is a generalization of directional differentiability.

Fréchet derivatives: the "chain rule"

The chain rule is a *fundamental result*, which supplies the Fréchet derivative of the composite $G \circ F$ of two functions

$$F : U \to V \text{ and } G : V \to W$$

between three normed vector spaces $(U, ||\cdot||_U)$, $(V, ||\cdot||_V)$ and $(W, ||\cdot||_W)$.

Theorem (Chain rule).

Let $x \in U$ be a point such that:

- *F is Fréchet differentiable at x;*
- *G is Fréchet differentiable at $F(x) \in V$.*

Then, the composite function $G \circ F : U \to W$ is Fréchet differentiable at x, and its Fréchet derivative $v \mapsto (G \circ F)'(x)(v)$ is the linear mapping defined by:

$$\forall v \in U, \ (G \circ F)'(x)(v) = G'(F(x))(F'(x)(v)).$$

The implicit function theorem

The implicit function theorem is a key result, ensuring the existence and smoothness of a solution $u = u_\theta$ to a parametrized, non linear equation of the form:

$$\mathcal{F}(\theta, u) = 0,$$

where u is the unknown and θ is a "parameter"; see [La], Chap. I, Th. 5.9.

Theorem (Implicit function theorem).

Let Θ, E, F be Banach spaces, $\mathcal{V} \subset \Theta$, $U \subset E$ be open sets. and $\mathcal{F} : \mathcal{V} \times U \to F$ be a function of class $\mathcal{C}^p$ for $p \geq 1$. Let $(\theta_0, u_0) \in \mathcal{V} \times U$ be such that $\mathcal{F}(\theta_0, u_0) = 0$ and assume that:

$$\text{The derivative } \frac{\partial \mathcal{F}}{\partial u}(\theta_0, u_0) : E \to F \text{ is a linear isomorphism.}$$

Then there exist open subsets $\mathcal{V}' \subset \mathcal{V}$ of θ_0 in Θ and $U' \subset U$ of u_0 in E, and a mapping $g : \mathcal{V}' \to U'$ of class $\mathcal{C}^p$ satisfying the properties:

1. $g(\theta_0) = u_0$,
2. *For all $\theta \in \mathcal{V}'$, the equation $\mathcal{F}(\theta, u) = 0$ has a unique solution $u \in U'$, given by $u = g(\theta)$.*

First-order necessary optimality conditions (I)

Let H be a Hilbert space, and let $J : H \to \mathbb{R}$ be a differentiable function; we consider the unconstrained minimization problem:

$$\min_{u \in H} J(u). \tag{UC}$$

Definition (Local minimizer).

A point $u \in H$ is a local minimizer for (UC) if there exists an open neighborhood $V \subset H$ containing u such that:

$$\forall v \in V, \;\; J(u) \leq J(v).$$

Theorem (First-order necessary optimality condition for (UC)).

Let u be a local minimizer for (UC); then:

$$\nabla J(u) = 0.$$

First-order necessary optimality conditions (II)

Proof: Let $h \in H$ be given; by the definition of u, it holds for $t > 0$ small enough:

$$J(u + th) \geq J(u), \text{ and so } \frac{J(u + th) - J(u)}{t} \geq 0.$$

Letting $t \to 0$, the differentiability of J yields:

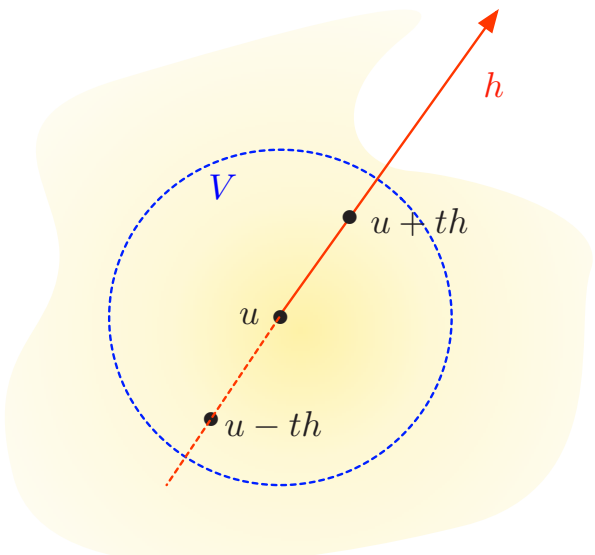
$$J'(u)(h) = \langle \nabla J(u), h \rangle \geq 0.$$

Replacing h by $-h$ in the previous argument yields the converse inequality

$$\langle \nabla J(u), h \rangle \leq 0,$$

which completes the proof.

Remark The proof uses in a crucial way that a local minimizer u of (UC) minimizes $J(v)$ (locally) in any direction $h \in H$.



First-order necessary optimality conditions (III)

Let H be a Hilbert space, and let $J : H \to \mathbb{R}$ and $G : H \to \mathbb{R}^p$ be differentiable functions; we consider the equality-constrained minimization problem:

$$\min_{h \in H} J(h) \text{ s.t. } G(h) = 0. \tag{EC}$$

Definition.

A point $u \in H$ is a local minimizer for (EC) if there exists an open neighborhood $V \subset H$ containing u such that:

$$\forall v \in V \text{ s.t. } G(v) = 0, \ J(u) \leq J(v).$$

Theorem 1 (First-order necessary optimality conditions for (EC)).

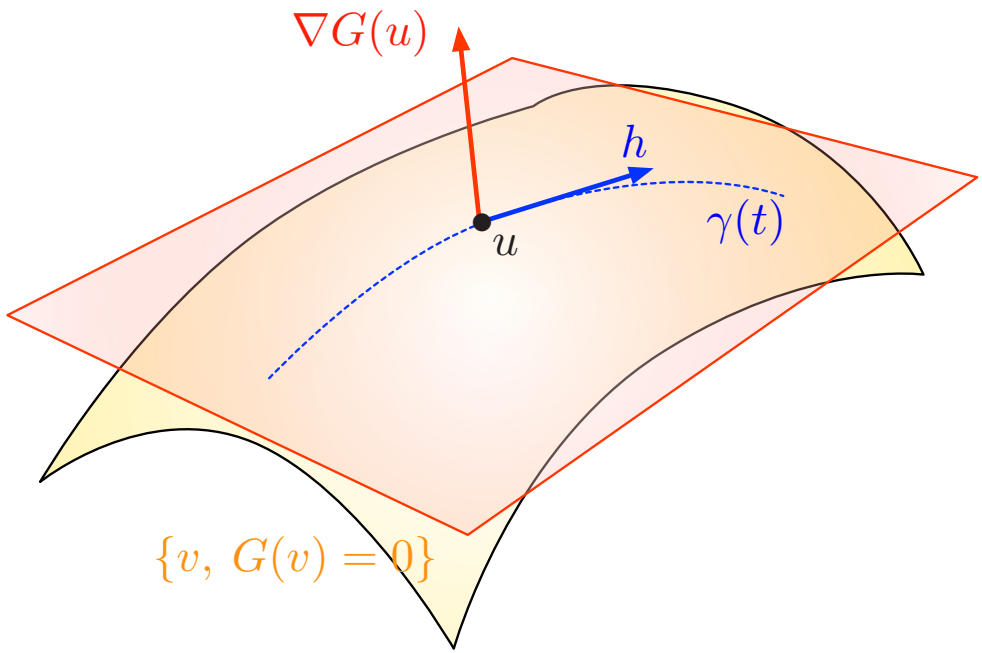
Let u be a local minimizer for (EC), and assume that the gradients $\nabla G_1(u), \ldots, \nabla G_p(u)$ are linearly independent. Then there exist Lagrange multipliers $\lambda_1, \ldots, \lambda_p \in \mathbb{R}$ such that:

$$\nabla J(u) + \sum_{i=1}^{p} \lambda_i \nabla G_i(u) = 0.$$

First-order necessary optimality conditions (IV)

Hint of proof:

- The local optimality of u no longer implies that, for *arbitrary* $h \in H$ and t small enough,
$$J(u + th) \geq J(u).$$
- Such an inequality can only be written with directions h in the admissible space:
$$K(u) := \{h \in H, \text{ there exists } \varepsilon > 0 \text{ and a curve } \gamma : [-\varepsilon, \varepsilon] \to H \text{ s.t. } \gamma(0) = u,\ \gamma'(0) = h \text{ and } G(\gamma(t)) = 0 \text{ for } t > 0\}.$$
- $K(u)$ is a vector space, which rewrites, using the implicit function theorem:
$$K(u) = \bigcap_{i=1}^{p} \{\nabla G_i(u)\}^{\perp}.$$



First-order necessary optimality conditions (II)

- For any $h \in K(u)$, introducing a curve $\gamma(t)$ with the above properties:

$$J(\gamma(t)) \geq J(u), \text{ and so } \frac{J(\gamma(t)) - J(u)}{t} \geq 0.$$

Taking limits, it follows,

$$\langle \nabla J(u), h \rangle \geq 0.$$

Since $K(u)$ is a vector space, the same argument applies to $-h$, and so:

$$\langle \nabla J(u), h \rangle = 0.$$

- Hence, we have proved that

$$\forall h \in K(u), \ \langle \nabla J(u), h \rangle = 0, \text{ that is: } \nabla J(u) \in \left(\bigcap_{i=1}^{p} \{\nabla G_i(u)\}^{\perp} \right)^{\perp}.$$

- Finally, using the general fact that, for arbitrary subsets $A_1, \ldots, A_p \subset H$,

$$(\text{span}\,\{A_i, \ i = 1, \ldots, p\})^{\perp} = \bigcap_{i=1}^{p} A_i^{\perp},$$

the desired result follows.

First-order necessary optimality conditions (III)

Interpretation (when $p = 1$): The above optimality condition implies that:

- Either $\nabla J(u) = 0$, which is the necessary first-order optimality condition for u to be an unconstrained minimizer of $J(v)$.
- Or $\lambda \neq 0$, and so,

$$\nabla G(u) = -\frac{1}{\lambda} \nabla J(u).$$

- "At first order", a direction $h \in H$ such that $J(u + th) < J(u)$ for small $t > 0$, has a non zero coordinate along $\nabla J(u)$:

  $h = \alpha \nabla J(u) + v$ with $v \perp \nabla J(u), \alpha < 0$.

- Hence, h rewrites:.

  $h = \beta \nabla G(u) + w$ with $w \perp \nabla G(u), \beta \neq 0$.

- As a result, $G(u + th) \neq 0$, so that $u + th$ is not an admissible point in (EC).

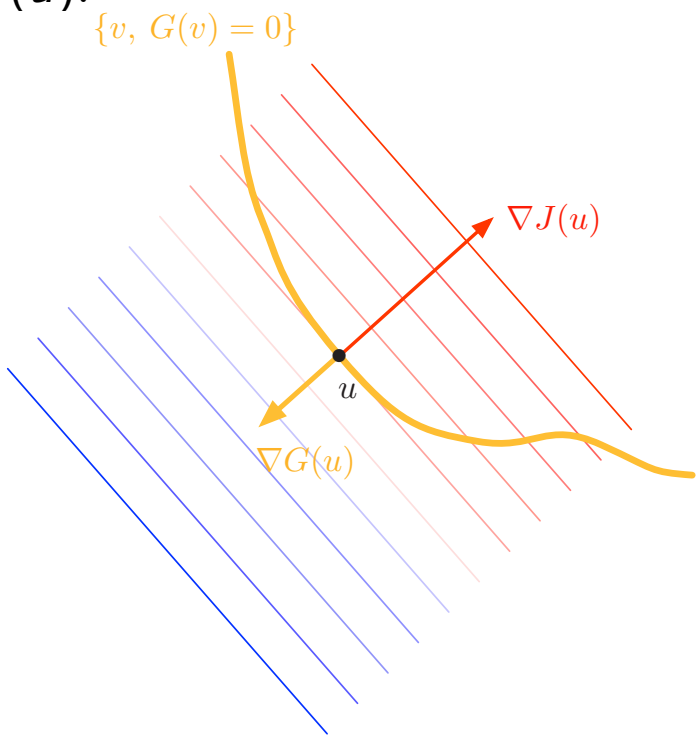


Illustration when $H = \mathbb{R}^2$, $p = 1$ and J is an affine function, whose isolines are depicted. At a local optimum u of (EC), $\nabla J(u)$ and $\nabla G(u)$ are aligned.

First-order necessary optimality conditions (IV)

- These necessary conditions are often expressed with the help of the Lagrangian:

$$\mathcal{L} : H \times \mathbb{R}^p \to \mathbb{R}, \text{ defined by: } \quad \mathcal{L}(u, \lambda) = J(u) + \sum_{i=1}^{p} \lambda_i G_i(u).$$

- Under the assumptions of the theorem, if u is a local minimizer of (EC), then:

$$\text{There exists a Lagrange multiplier } \lambda \in \mathbb{R}^p \text{ s.t. } \nabla_u \mathcal{L}(u, \lambda) = 0,$$

  i.e. (u, λ) is a critical point of $\mathcal{L}(u, \lambda)$.

- If u were a minimizer of $\mathcal{L}(\cdot, \lambda)$, then λ would be the "right" level of penalization of $J(u)$ by $G(u)$ to turn (EC) into an unconstrained problem.

Bibliography

References I

[All] G. Allaire, *Analyse Numérique et Optimisation*, Éditions de l'École Polytechnique, (2012).

[Ci] P.G. Ciarlet, *The finite element method for elliptic problems*. Society for Industrial and Applied Mathematics, (2002).

[ErnGue] A. Ern and J.-L. Guermond, *Theory and Practice of Finite Elements*, Springer, (2004).

[FreyGeo] P.J. Frey and P.L. George, *Mesh Generation : Application to Finite Elements*, Wiley, 2nd Edition, (2008).

[He] F. Hecht, *New development in `FreeFem++`*, Journal of numerical mathematics, 20(3-4), (2012), pp. 1–14.

[La] S. Lang, *Fundamentals of differential geometry*, Springer, (1991).

[Neu] J. Neuberger, *Sobolev gradients and differential equations*, Springer Science & Business Media, (2009).

References II

[NoWri] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer Science, (1999).

[RuOsFa] L. I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: nonlinear phenomena, 60(1-4), (1992). pp. 259–268.

Educational and online resources I

[Allaire2] *Grégoire Allaire's web page*, `http://www.cmap.polytechnique.fr/ allaire/`.

[Bo] B. Bogosel, *Beni Bogosel's webpage*, `http://www.cmap.polytechnique.fr/~beniamin.bogosel`.

[AlPan] G. Allaire and O. Pantz, *Structural Optimization with FreeFem++*, Struct. Multidiscip. Optim., 32, (2006), pp. 173–181.

[BonDa] E. Bonnetier and C. Dapogny, *Web page of the course*, `https://ljk.imag.fr/membres/Charles.Dapogny/coursoptim.html`.

[DaFrOmPri] C. Dapogny, P. Frey, F. Omnes and Y. Privat, *Geometrical shape optimization in Fluid Mechanics using* `FreeFem++`, Struct. Multidisc. Optim, 58, no. 6, (2018), pp. 2761–2788.

[DTU] *Web page of the Topopt group at DTU*, `http://www.topopt.dtu.dk`.

[Fenics] *Web page of the* `Fenics` *project*, `https://fenicsproject.org/`.

Educational and online resources II

- [Firedrake] *Web page of the* `Firedrake` *project*, `https://www.firedrakeproject.org/`.
- [FreeFem] *Web page of the* `FreeFem` *project*, `https://freefem.org/`.
- [Ma] V. Mazet, *Basics of image processing*, University of Strasbourg, 2020-2025, `https://vincmazet.github.io/bip/`.
- [NGSolve] *Web page of the* `NGSolve` *project*, `https://ngsolve.org/`.
- [Sigmund] O. Sigmund, *A 99 line topology optimization code written in MATLAB*, Struct. Multidiscip. Optim., 21, 2, (2001), pp. 120–127.
- [Wi] Wikipedia, *Messier 87*, `https://en.wikipedia.org/wiki/Messier_87`.