



Finite Element Method (FEM)

GeePs'N Talks special session

T. Cherrière, A. El Gode, T. Gauthey

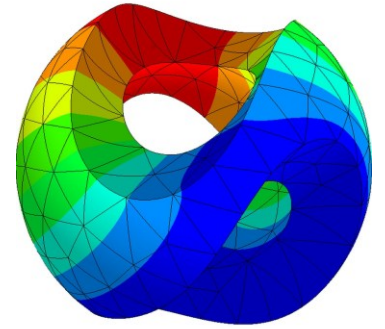
Preliminaries

- Interactive course, using NGSolve (Python FEM toolbox)
- Go to website : https://github.com/tcherrie/tutorial_fem

And click on the yellow button



- The code should run in your browser without installation required.
- If strange bugs: **reload the webpage** (virtual memory overflow)
- GeePs clusters in backup
- For local installation: ask after the tutorial



Outlines

- 1) Lengthy introduction
 - Function spaces & interpolation
 - Integral formulation
 - Linear system
- 2) Academic Poisson problem
 - Variational formulation
 - Boundary conditions
- 3) Non-linear Magnetostatics (2D)
 - Realistic problem
 - Newton method
- 4) 3D Magnetostatics
 - Iterative solver
 - Gauge

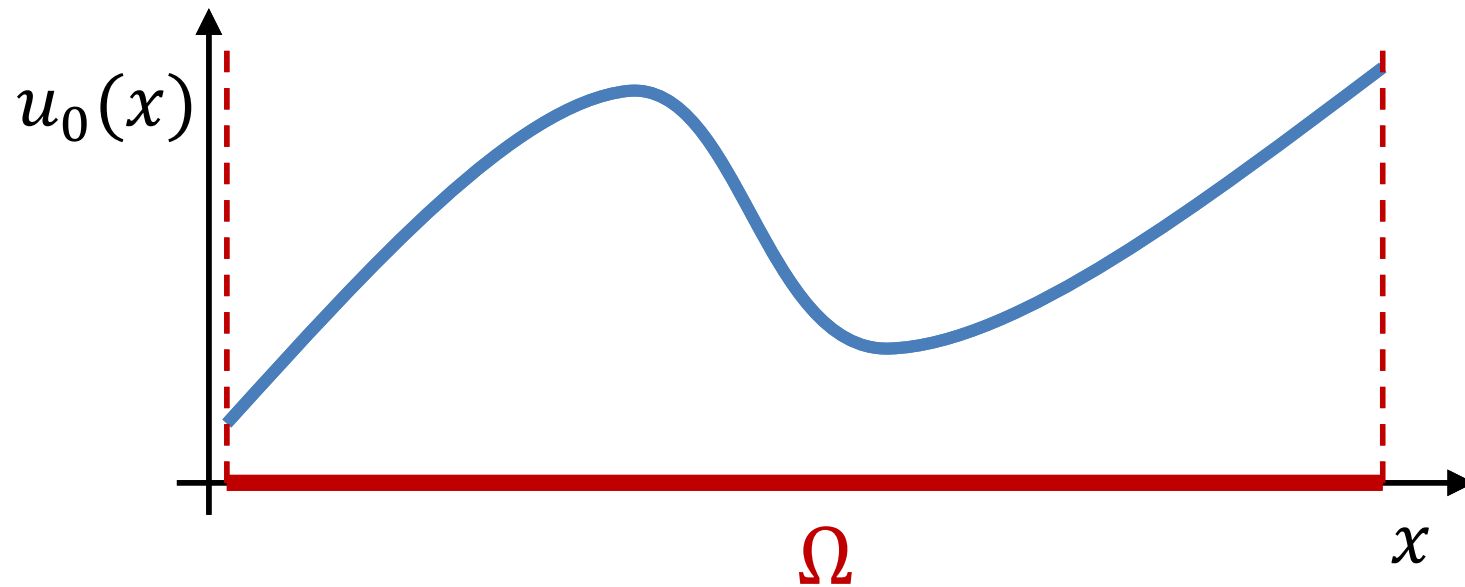
Not addressed in this tutorial: time-harmonic and time-dependent problem.

1) LENGTHY INTRODUCTION

Idea of the method, not boring I promise (hope)

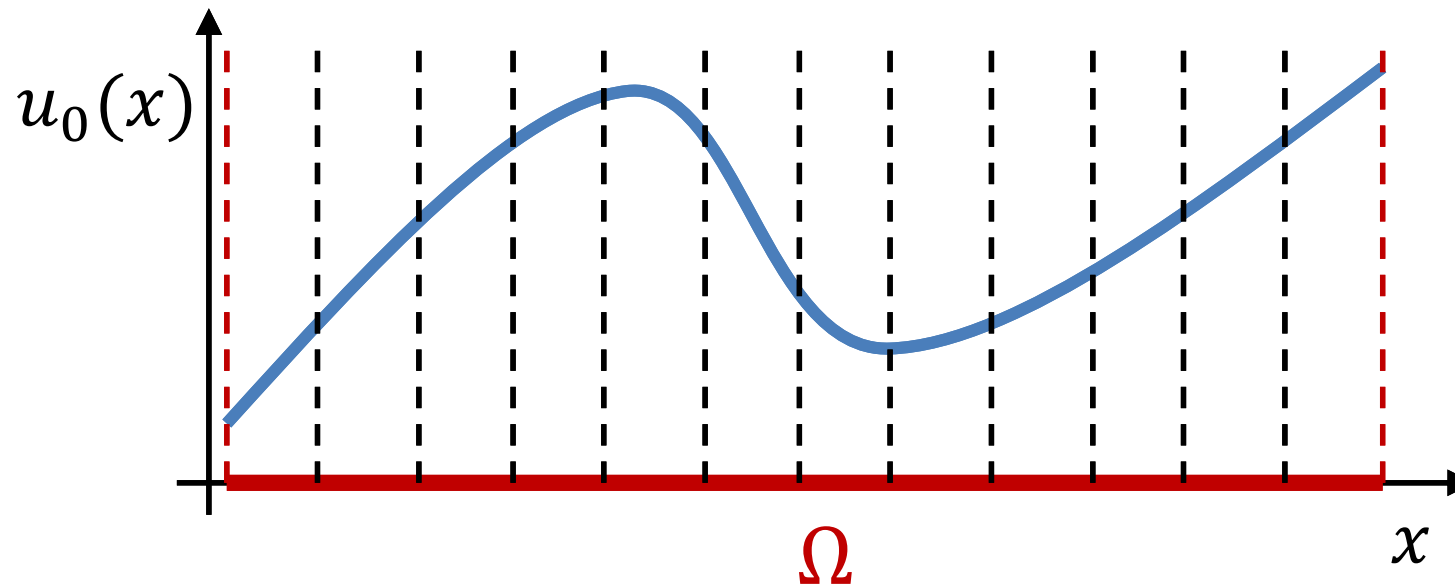
Idea of FEM

How to approximate a function on a finite-dimensional space?



Idea of FEM

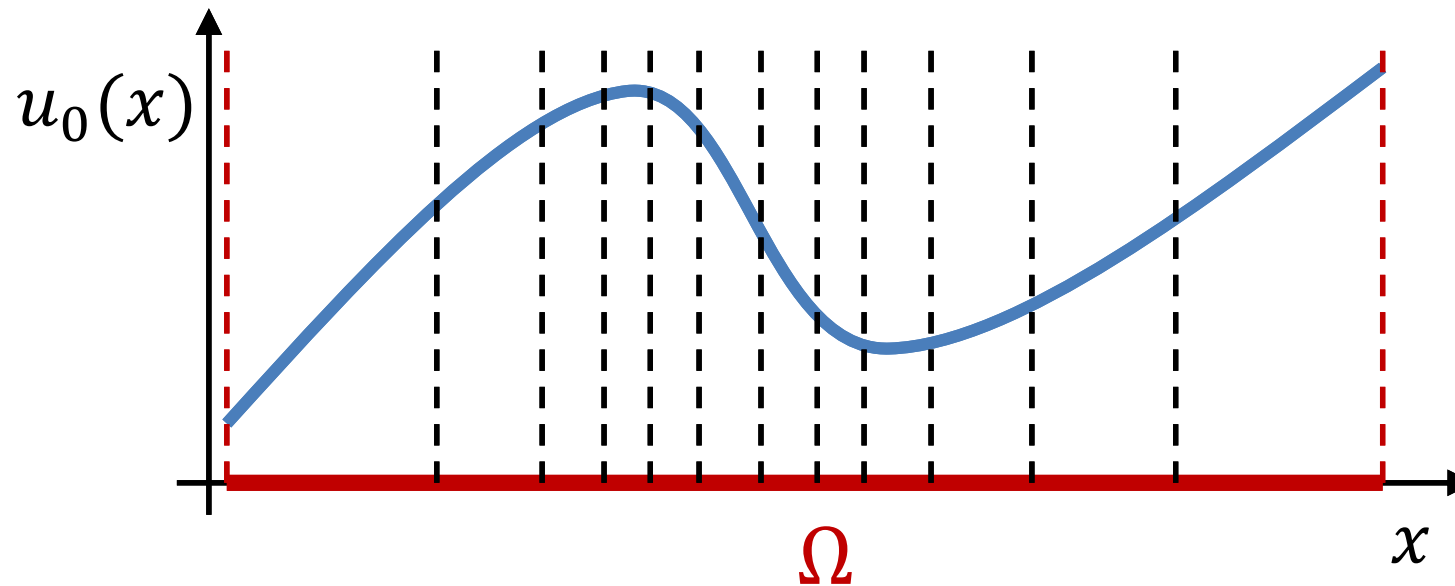
How to approximate a function on a finite-dimensional space?



Discretization of geometric space Ω
(uniform mesh)

Idea of FEM

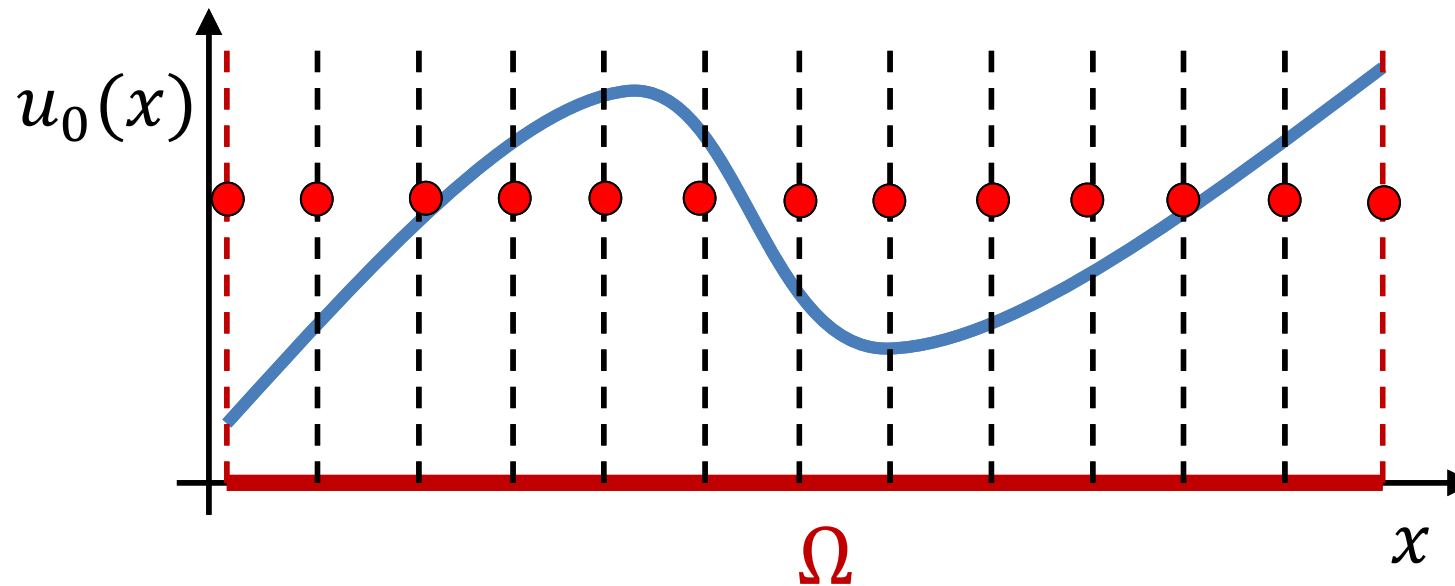
How to approximate a function on a finite-dimensional space?



Discretization of geometric space Ω
(irregular *mesh*)

Idea of FEM

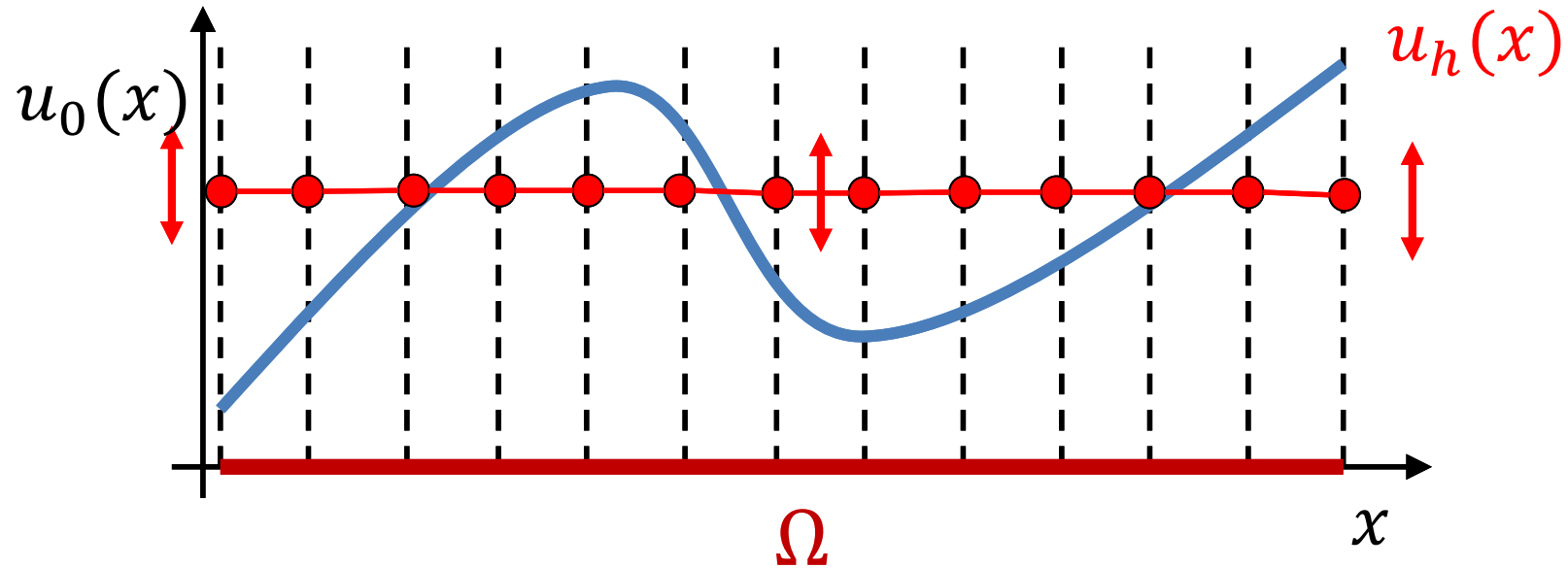
How to approximate a function on a finite-dimensional space?



Degrees of freedom (DoFs)
(unknowns of the problem)

Idea of FEM

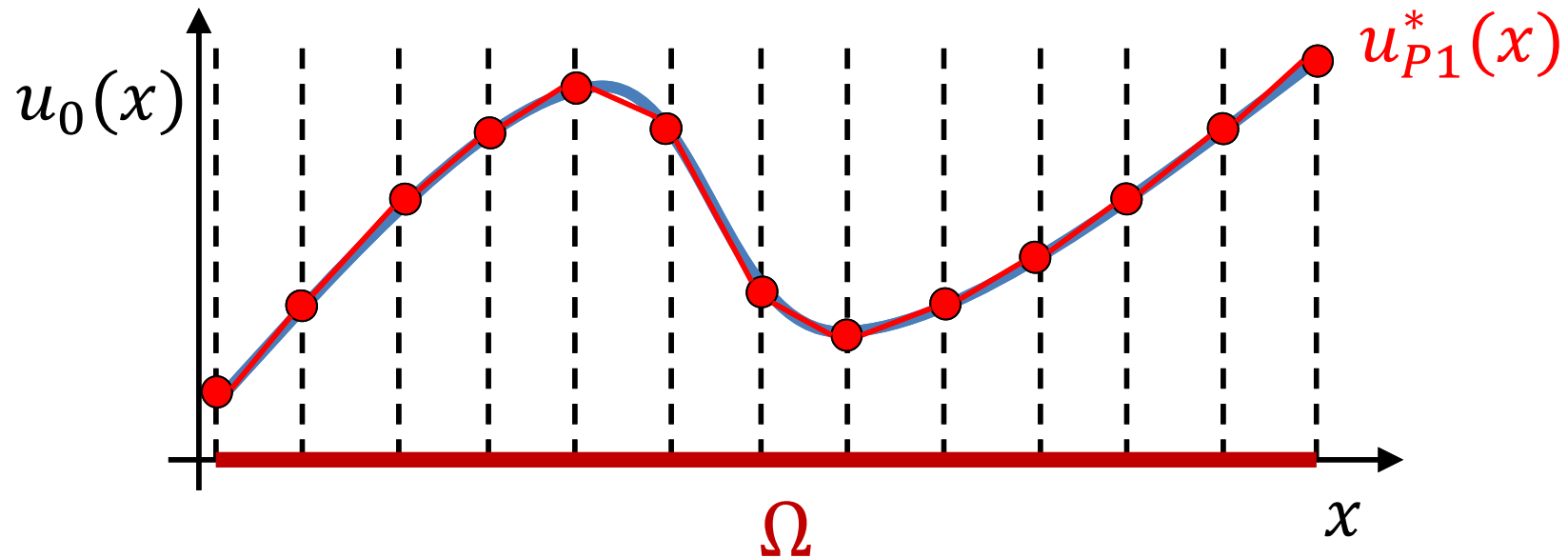
How to approximate a function on a finite-dimensional space?



Interpolation defined from the DoFs

Idea of FEM

How to approximate a function on a finite-dimensional space?



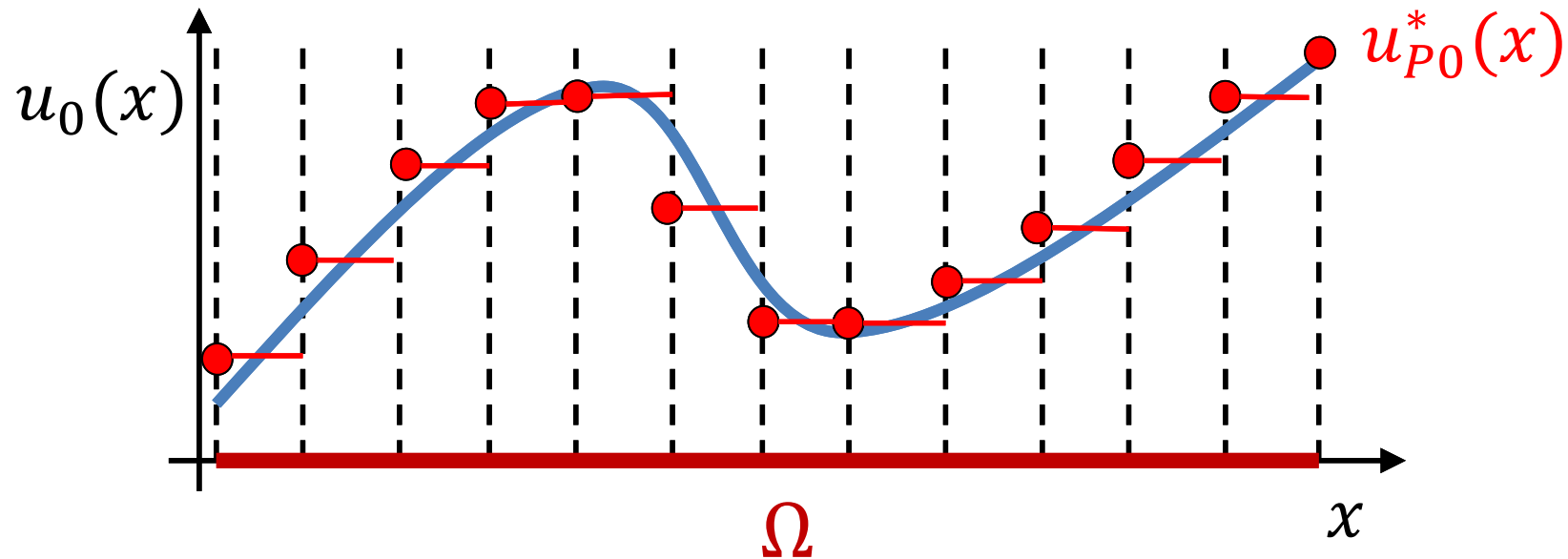
Interpolation defined from the DoFs

Best possible linear interpolation

Idea of FEM

How to approximate a function on a finite-dimensional space?

*Given an interpolation, how can we determine the **optimal** DoF values?*

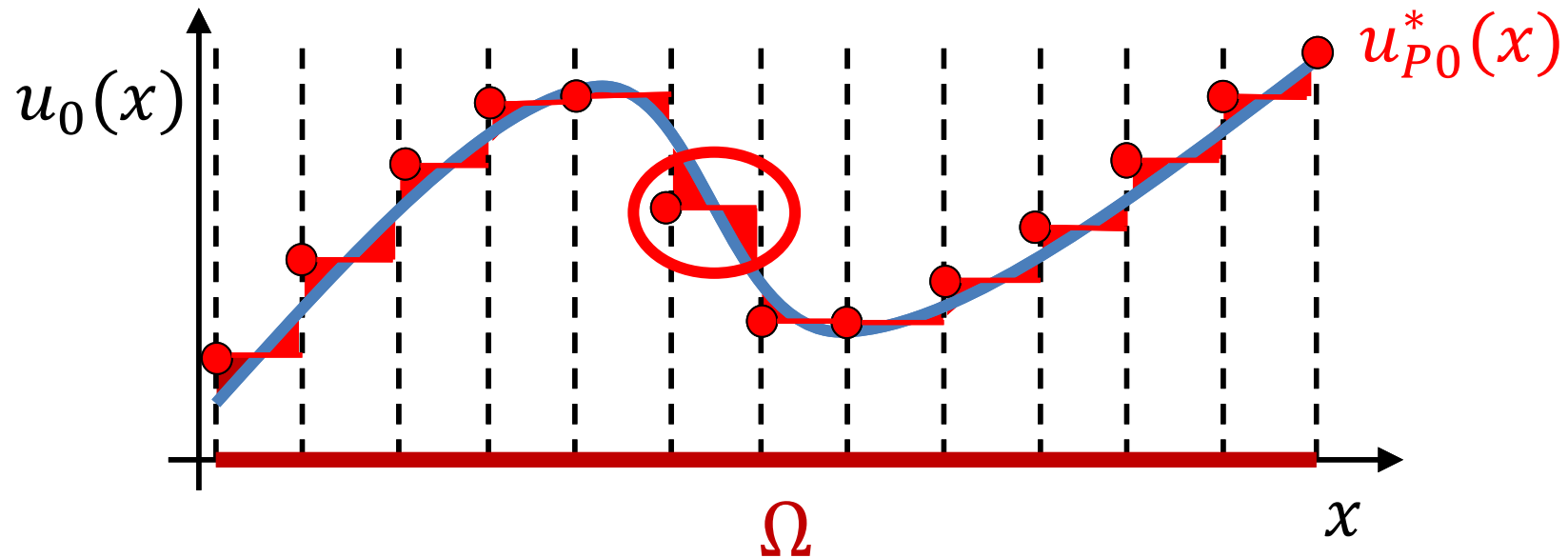


Interpolation defined from the DoFs

Best possible **constant** interpolation

Mean squared error minimization

Illustration



Interpolation defined from the DoFs

Best possible constant interpolation

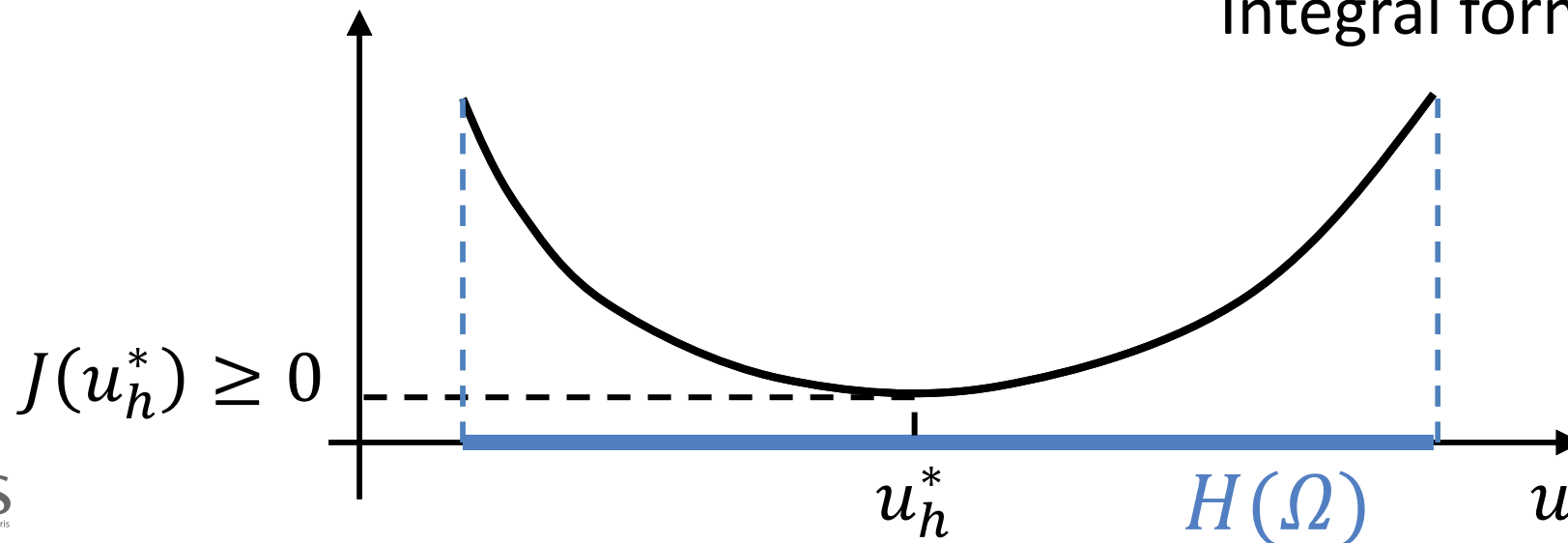
Mean squared error minimization

Mathematical formulation

$$u_h^* = \arg \min_{u_h \in H(\Omega)} J(u_h) = \frac{1}{2} \int_{\Omega} \underbrace{(u_h(x) - u_0(x))^2}_{\text{Squared error}} dx$$

Admissible function space
(continuous or discretized)

Integral formulation

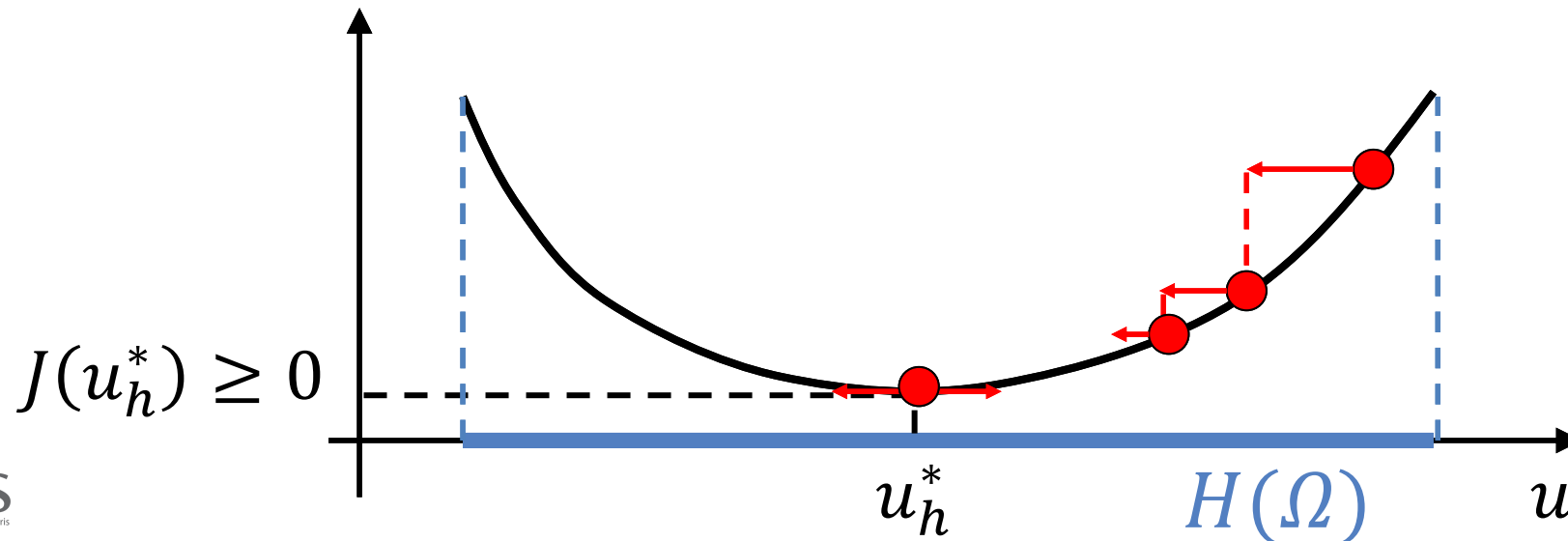


Mean squared error minimization

Algorithms

- Naïve idea: **gradient descent** : $u_h^{k+1} = u_h^k - \alpha J'(u_h^k)$
- Better idea: **convex problem** \Rightarrow unique minimum satisfying $J'(u_h^*) = 0$

How to compute the derivative J' ?



Mean squared error minimization

Directional derivative

- **Directional derivative** in the direction v :

$$\underbrace{J'(u; v) = J'(u)(v) = \langle J'(u), v \rangle}_{\text{Directional derivative}} = \lim_{t \rightarrow 0} \frac{J(u + tv) - J(u)}{t} \in \mathbb{R}$$

Different notations exist ; all highlighting that v (« test function ») plays a different role than u (point where the derivative is computed).

- We can define a *linear application* $J'(u): v \mapsto J'(u; v) \in L(H, \mathbb{R})$

Exercise : compute the directional derivative of the MSE

$$J(u) = \frac{1}{2} \int_{\Omega} (u(x) - u_0(x))^2 \, dx$$

Mean squared error minimization

Directional derivative & algorithms

$$J'(u; v) = \int_{\Omega} (u - u_0) v \, dx$$

For $d \propto -(u - u_0)$, $J'(u; d) \leq 0$

$\Rightarrow \boxed{d = -(u - u_0)}$ is a ***descent direction***

Gradient descent

$$u_h^{k+1} = u_h^k + \alpha \, d(u_h^k)$$

We can also find $u_h^* \in H(\Omega)$ such that

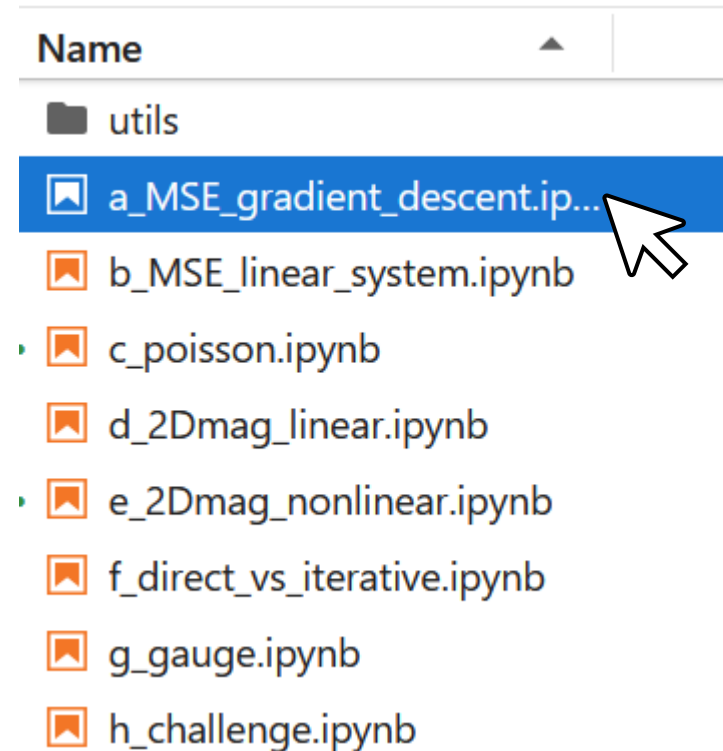
$$\forall v \in H(\Omega), \quad J'(u_h^*; v) = 0$$

Linear system

To assemble and solve!

Application 1 : gradient descent on MSE

Jupyter Notebook « a_MSE_gradient_descent »



Try out different interpolations:

- Function spaces

$$- L^2(\Omega) = \{v: \Omega \rightarrow \mathbb{R}, \int_{\Omega} v(x) dx < \infty\}$$

(discretized by element-wise **discontinuous** functions)

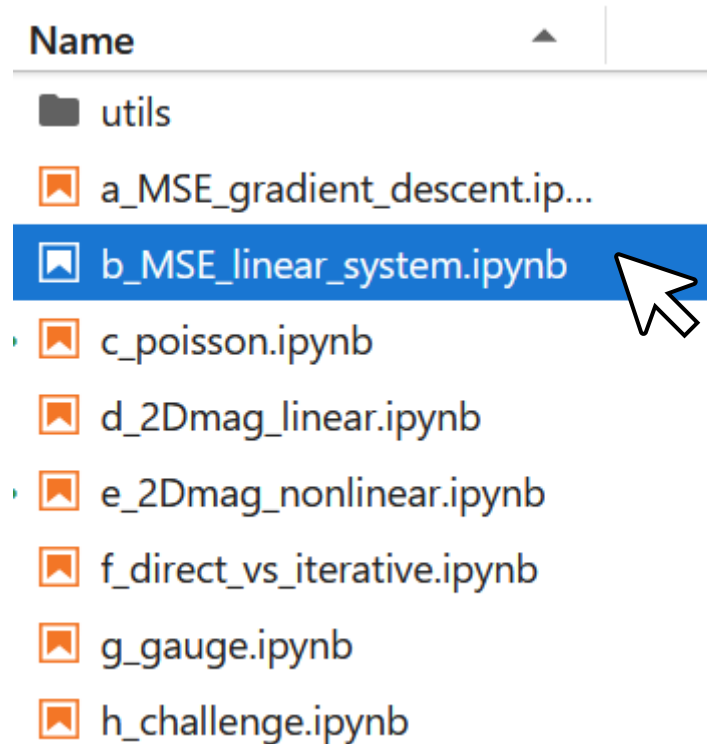
$$- H^1(\Omega) = \{v \in L^2(\Omega), \nabla v \in L^2(\Omega)\}$$

(discretized by nodal **continuous** functions)

- Polynomial degrees / order

Application 2 : linear system assembly

Jupyter Notebook « b_MSE_linear_system »



- Gradient descent is generally inefficient and sometimes inapplicable
- From the optimality condition, one can assemble a linear system. So

$$\forall v \in H(\Omega), \quad \int_{\Omega} u v = \int_{\Omega} u_0 v$$

Becomes

$$Ku = f$$

2) ACADEMIC POISSON PROBLEM

Now let's solve partial differential equations

Variational formulation

General method

- The finite element method is based on variational formulations
- **Main objective of the session:** obtain a variational formulation from the strong equations.
- **Methodology**
 1. Choice of relevant variables → *not trivial...* see literature!
 2. Choice of the function space H → *often easy*
 3. **Projection of the equation on H** → ***often easy***

Variational formulation

Choice of relevant variables

- We consider electrostatics

- Maxwell equations

$$\begin{cases} \nabla \cdot \mathbf{D} = \rho & \text{(Maxwell–Gauss)} \\ \nabla \times \mathbf{E} = 0 & \text{(Maxwell–Faraday)} \end{cases}$$

- Material constitutive law

$$\mathbf{D} = \epsilon_0 \epsilon_r \mathbf{E}$$

With $\epsilon_0 = 8.85 \times 10^{-12} \text{ F/m}$, ϵ_r depending on material

Material	Dielectric constant ϵ_r
Vacuum	1
Air	1,0006
Reinforced concrete	1,51
Teflon	2,1
Paper	3,85
Silicon dioxide	3,9
FR-4	4
Mica	5,6 - 8
Marble	8,3
Silicon	11,7
Calcium titanate	150

[What is electric permittivity? - Electrical e-Library.com](http://www.electrical-e-library.com)

Variational formulation

Choice of relevant variables

- Many formulations are possible. We usually use scalar electric potential :

$$\mathbf{E} = -\nabla u$$

$\Rightarrow \nabla \times \mathbf{E} = 0$ is automatically verified (curl of grad is always 0) ; but u is now *defined up to a constant* that should be fixed.

- From the other equations we obtain

Poisson equation

$$-\nabla \cdot (\epsilon_0 \epsilon_r \nabla u) = \rho$$



Variational formulation

Formal projection

- We consider a geometric space Ω and a function space $H(\Omega)$, detailed later.

1. Multiplication by any test function $v \in H(\Omega)$ and **integration** over Ω :

$$-\int_{\Omega} \nabla \cdot (\epsilon_0 \epsilon_r \nabla u) v \, dx = \int_{\Omega} \rho v \, dx$$

2. Integration by part; using the following formulae :

Leibniz:

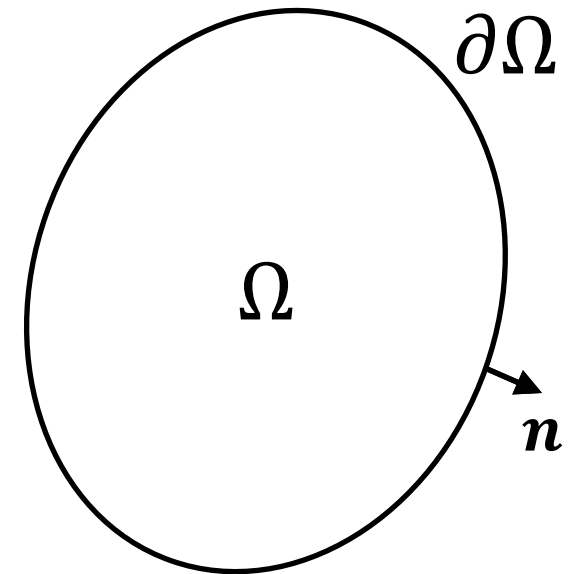
$$b \nabla \cdot \mathbf{A} = \nabla \cdot (b \mathbf{A}) - \mathbf{A} \cdot \nabla b$$

Green-Ostrogradski :

$$\int_{\Omega} \nabla \cdot \mathbf{A} = \int_{\partial\Omega} \mathbf{A} \cdot \mathbf{n}$$

Boundary of Ω

Outward normal to $\partial\Omega$



Variational formulation

Formal projection

- We obtain:

$$\underbrace{\int_{\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \nabla v \, dx}_{\text{Bilinear form}} - \underbrace{\int_{\partial\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \mathbf{n} \, v \, ds}_{\text{Boundary term}} = \underbrace{\int_{\Omega} v \, \rho}_{\text{Linear form}}$$

Boundary value problem (BVP) with a boundary term on the normal component of electrical displacement:

$$\epsilon_0 \epsilon_r \nabla u \cdot \mathbf{n} = \mathbf{D} \cdot \mathbf{n}$$

Homogeneous to a surface charge density ρ_s .

Variational formulation

Flashback to the function space

- We should find $u \in H(\Omega)$, such that

$$\forall v \in H, \quad \int_{\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \mathbf{n} \, v \, ds = \int_{\Omega} v \rho \, dx$$

Function space

To have well defined integrals, $H(\Omega) = \{u \in L^2(\Omega), \nabla u \in L^2(\Omega)\} = H^1(\Omega)$

Boundary conditions

Natural boundary conditions

- We should find $u \in H(\Omega)$, such that

$$\forall v \in H, \quad \int_{\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \mathbf{n} \, v \, ds = \int_{\Omega} v \rho \, dx$$

- **Natural boundary conditions** : we rewrite the boundary term
 - **Neumann** : $\epsilon_0 \epsilon_r \nabla u \cdot \mathbf{n} = g \rightarrow$ boundary term becomes a linear form
 - **Robin** : $\epsilon_0 \epsilon_r \nabla u \cdot \mathbf{n} = g - \alpha u \rightarrow$ boundary term becomes linear + bilinear form

Since these boundary conditions are part of the variational form, they are « *weakly* » imposed.

Boundary conditions

Natural boundary conditions

- We should find $u \in H(\Omega)$, such that

$$\forall v \in H, \quad \int_{\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \mathbf{n} \, v \, ds = \int_{\Omega} v \rho \, dx$$

- **Neumann** (special case of Robin) :

$$\forall v \in H, \quad \int_{\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \nabla v \, dx = \int_{\Omega} v \rho \, dx + \int_{\partial\Omega} g \, v \, ds$$

- **Robin** :

$$\forall v \in H, \quad \int_{\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \nabla v \, dx + \int_{\partial\Omega} \alpha u \, v \, ds = \int_{\Omega} v \rho \, dx + \int_{\partial\Omega} g \, v \, ds$$

Boundary conditions

Essential boundary conditions

- We should find $u \in H(\Omega)$, such that

$$\forall v \in H, \quad \int_{\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \mathbf{n} \, v \, ds = \int_{\Omega} v \rho \, dx$$

- **Essential boundary conditions:** not appearing directly in the boundary term; therefore should be imposed (exactly!) in the **function space**

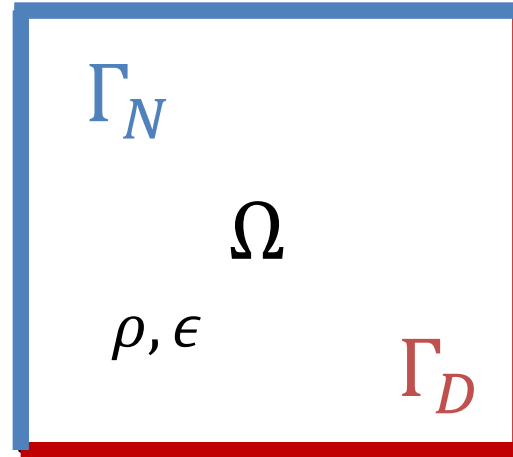
- **Dirichlet** : $u = u_d$ on the boundary
- **Periodicity / anti-periodicity** : $u_1 = \pm u_2$

Sets the boundary term to 0

Application 3 : Poisson problem

Jupyter Notebook « c_Poisson »

Name
utils
a_MSE_gradient_descent.ip...
b_MSE_linear_system.ipynb
• c_poisson.ipynb
d_2Dmag_linear.ipynb
• e_2Dmag_nonlinear.ipynb
f_direct_vs_iterative.ipynb
g_gauge.ipynb
h_challenge.ipynb



$$\text{Find } u \in H(\Omega), \forall v \in H, \\ \int_{\Omega} \epsilon_0 \epsilon_r \nabla u \cdot \nabla v \, dx = \int_{\Omega} v \rho \, dx$$

- Homogeneous Dirichlet : $u|_{\partial\Omega} = 0$
 $\Rightarrow \vec{E}$ orthogonal to the boundary (gradient is orthogonal to the isolines of u).
 $\Rightarrow \partial\Omega$ is an anti-symmetry plane
Can also truncate infinity ($u(\infty) = 0$)
- Homogeneous Neumann : $\vec{D} \cdot \vec{n} = 0$
 $\Rightarrow \vec{D}$ tangential to the boundary
 $\Rightarrow \partial\Omega$ is a symmetry plane

3) 2D MAGNETOSTATICS

Non-linearity and Newton method

Magnetostatics

Equations

- We give the equations

$$\mathbf{B} = \nabla \times \mathbf{a}$$

Magnetic vector potential (unknown)

$$\nabla \times \mathbf{H} = \mathbf{j}$$

Maxwell Ampère

$$\mathbf{H} = \nu(|\mathbf{B}|^2)\mathbf{B}$$

Constitutive law of iron

$$\mathbf{a} = \mathbf{0} \text{ on } \partial\Omega$$

Dirichlet boundary condition (boundary term $\rightarrow 0$)

What is the strong formulation?

Magnetostatics

Strong form

- The b-conform strong equation reads

$$\nabla \times (\nu(|\nabla \times \mathbf{a}|^2) \nabla \times \mathbf{a}) = \mathbf{j}$$

Or

$$\mathbf{curl}(\nu(|\mathbf{curl} \mathbf{a}|^2) \mathbf{curl} (\mathbf{a})) = \mathbf{j}$$

What is the variational formulation?

Donner formules de Leibniz + Green Ostrogradsky

Magnetostatics

Variational formulation

- We find $\mathbf{a} \in H_0(\mathbf{curl}; \Omega) = \{\mathbf{a} \in L^2(\Omega), \mathbf{curl}(\mathbf{a}) \in L^2(\Omega), \mathbf{a} \times \mathbf{n} = 0 \text{ on } \partial\Omega\}$

$$\forall \mathbf{v} \in H_0(\mathbf{curl}; \Omega), \quad \int_{\Omega} \mathbf{curl} \mathbf{v} \cdot (\nu(|\mathbf{curl} \mathbf{a}|^2) \mathbf{curl} \mathbf{a}) = \int_{\Omega} \mathbf{v} \cdot \mathbf{j}$$

- In 2D, we have

$$\mathbf{j} = \begin{bmatrix} 0 \\ 0 \\ j_z \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} 0 \\ 0 \\ a_z \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} 0 \\ 0 \\ v_z \end{bmatrix}, \quad \mathbf{curl}(\mathbf{a}) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{grad}(a_z)$$

So that we can rewrite the equation w.r.t the z -components only.

Application 4 : 2D linear magnetostatics

Jupyter Notebook « d_nonlinear.ipynb »

Name

utils

a_MSE_gradient_descent.ip...

b_MSE_linear_system.ipynb

c_poisson.ipynb

d_2Dmag_linear.ipynb

e_2Dmag_nonlinear.ipynb

f_direct_vs_iterative.ipynb

g_gauge.ipynb

h_challenge.ipynb

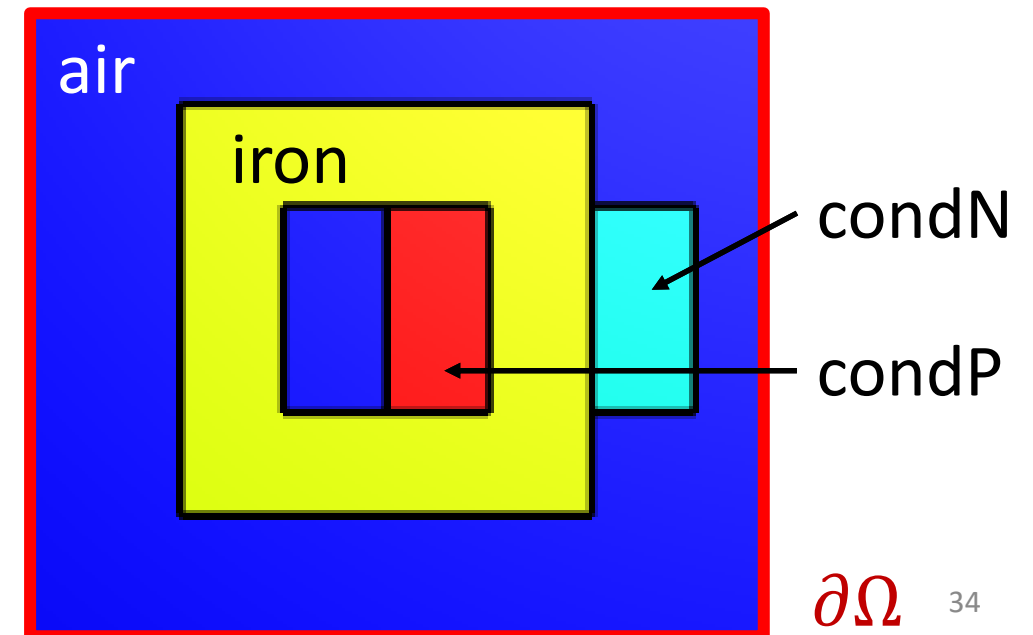
For now, we assume iron is linear :

- $\mu_{iron} = 1000\mu_0 \Rightarrow \nu_{iron} = \frac{1}{1000\mu_0}$
- $J = 10 \text{ A/mm}^2$

- Find

$$a_z \in H_0^1(\Omega) = \{a \in L^2(\Omega), \nabla a \in L^2(\Omega), a = 0 \text{ on } \partial\Omega\}$$

$$\forall v \in H_0^1(\Omega), \quad \int_{\Omega} \mathbf{curl} v \cdot \nu \mathbf{curl} a_z = \int_{\Omega} v j_z$$



$\partial\Omega$

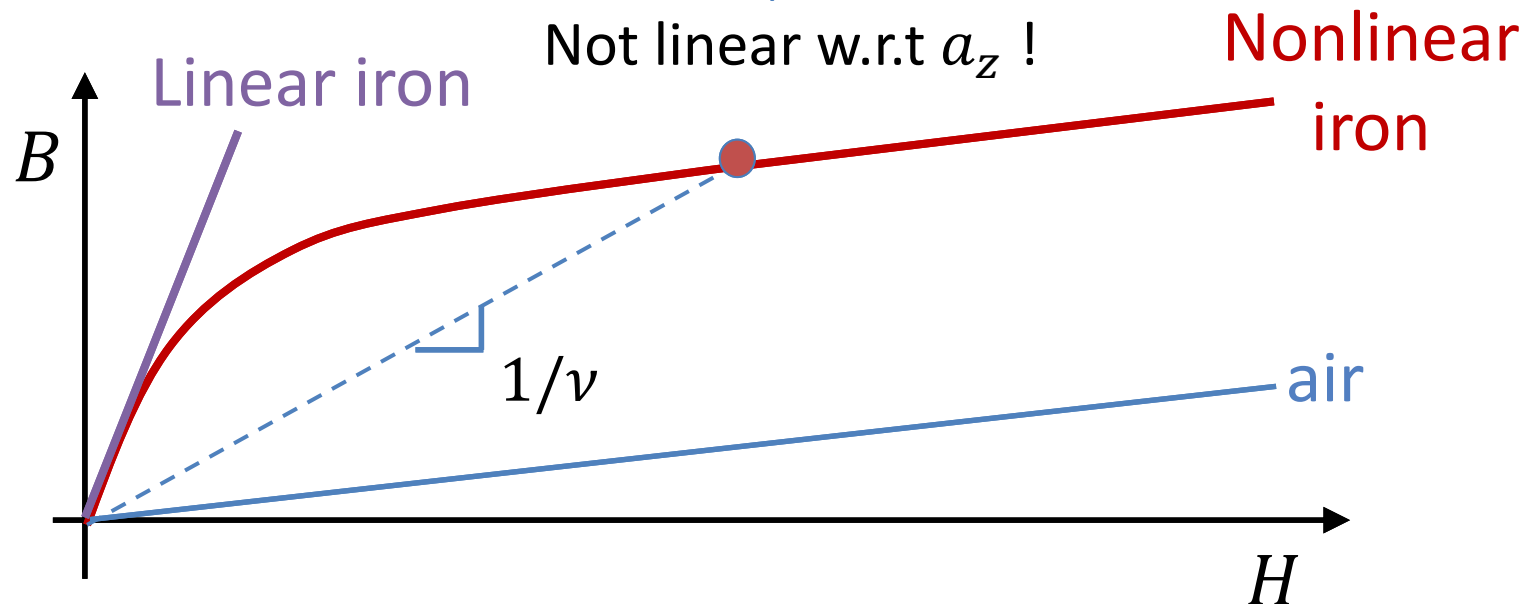
Nonlinearity

2D Variational formulation

- We should find $a_z \in H_0^1(\Omega) = \{a \in L^2(\Omega), \nabla a \in L^2(\Omega), a = 0 \text{ on } \partial\Omega\}$

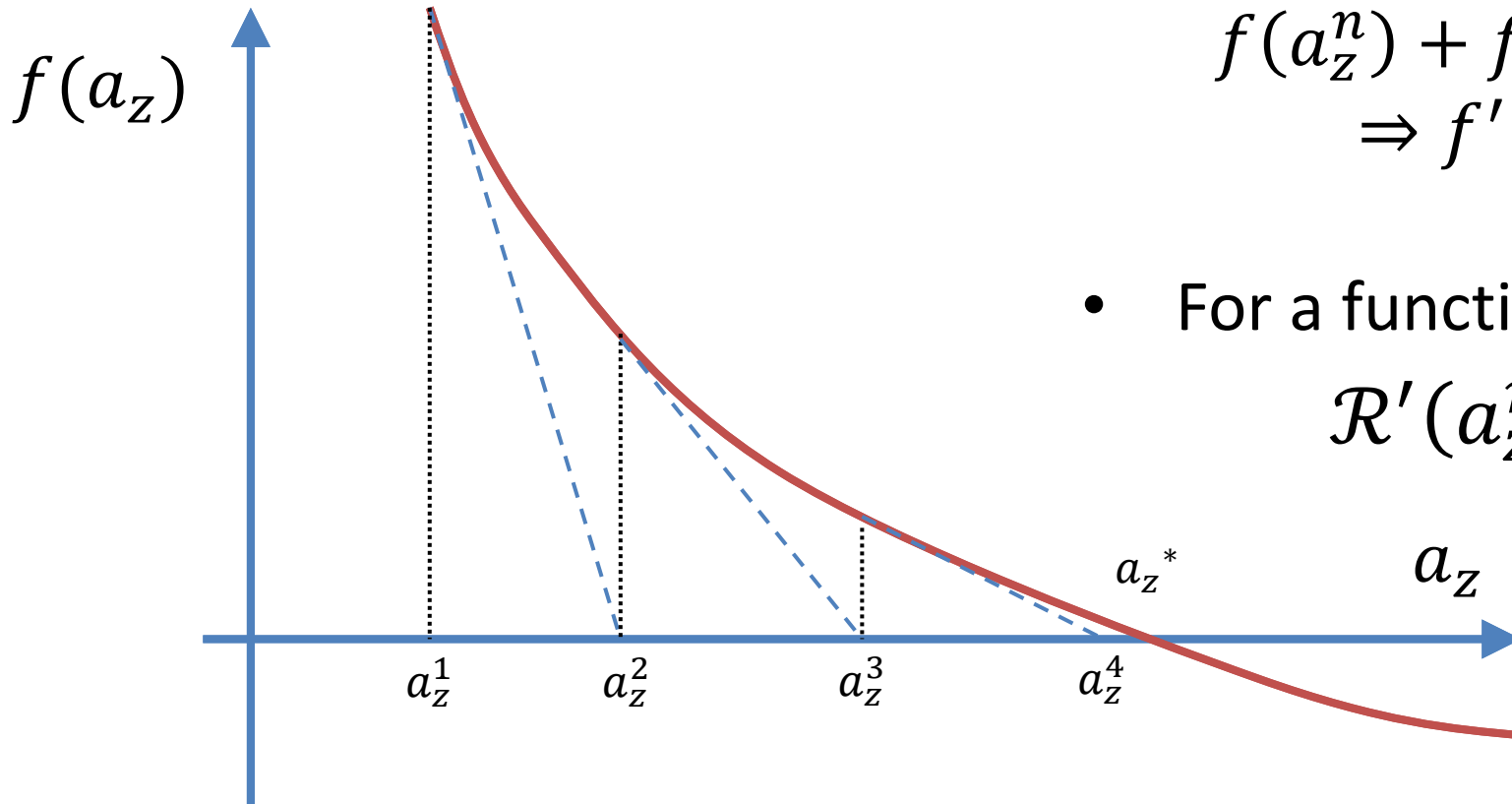
$$\forall v \in H_0^1(\Omega), \quad \underbrace{\int_{\Omega} \mathbf{curl} v \cdot v (|\mathbf{curl} a_z|^2) \mathbf{curl} a_z}_{\text{Not linear w.r.t } a_z!} = \int_{\Omega} v j_z$$

BH curves:



Nonlinearity

Newton method



- For a usual function:

$$f(a_z^n) + f'(a_z^n) \overbrace{(a_z^{n+1} - a_z^n)}^{\delta a} = 0$$
$$\Rightarrow f'(a_z^n) \delta a = -f(a_z^n)$$

- For a functional \mathcal{R} : find δa , such that

$$\mathcal{R}'(a_z^n; \delta a) = -\mathcal{R}(a_z^n)$$

Nonlinearity

Newton method

- We should now define the functional \mathcal{R} to cancel.
- **Residual**

$$\mathcal{R}(a_z, v) = \int_{\Omega} \mathbf{curl} v \cdot v(|\mathbf{curl} a_z|^2) \mathbf{curl} a_z - \int_{\Omega} v j_z$$

- **Directional derivative:**

$$\begin{aligned} & \mathcal{R}'(a_z, v; \delta a) \\ &= \int_{\Omega} \mathbf{curl} v \cdot v(|\mathbf{curl} a_z|^2) \mathbf{curl} \delta a \\ &+ 2 \int_{\Omega} \mathbf{curl} v \cdot (v'(|\mathbf{curl} a_z|^2) \mathbf{curl} a_z \cdot \mathbf{curl} \delta a) \mathbf{curl} a_z \end{aligned}$$

Nonlinearity

Newton method

- Typical algorithm:

1) Solve the linearized problem

$$\forall v \in H_0^1(\Omega), \quad \mathcal{R}'(a_z, v; \delta a) = -\mathcal{R}(a_z, v)$$

2) Adapt step size α and update

$$a_z^{n+1} = a_z^n + \alpha \delta a$$

3) Stop criterion (several possibilities...)

$|\mathcal{R}(a_z^n, \delta a)| \leq \text{tol}$, or $|\mathcal{R}(a_z^n, v_i)| \leq \text{tol}$... and always $n > n_{max}$

Application 5 : non-linear 2D magnetostatics

Jupyter Notebook « e_nonlinear_2D_magnetostatics »

Name	
utils	
a_MSE_gradient_descent.ip...	
b_MSE_linear_system.ipynb	
c_poisson.ipynb	
d_2Dmag_linear.ipynb	
e_2Dmag_nonlinear.ipynb	
f_direct_vs_iterative.ipynb	
g_gauge.ipynb	
h_challenge.ipynb	

- **Residual**

$$\mathcal{R}(a_z, v) = \int_{\Omega} \mathbf{curl} v \cdot v (|\mathbf{curl} a_z|^2) \mathbf{curl} a_z - \int_{\Omega} v j_z$$

- **Directional derivative:**

$$\begin{aligned} \mathcal{R}'(a_z, v; \delta a) &= \int_{\Omega} \mathbf{curl} v \cdot v (|\mathbf{curl} a_z|^2) \mathbf{curl} \delta a \\ &+ 2 \int_{\Omega} \mathbf{curl} v \cdot (v'(|\mathbf{curl} a_z|^2) \mathbf{curl} a_z \cdot \mathbf{curl} \delta a) \mathbf{curl} a_z \end{aligned}$$

- 1) Find $\delta a \in H_0^1(\Omega)$, such that $\forall v \in H_0^1(\Omega)$, $\mathcal{R}'(a_z, v; \delta a) = -\mathcal{R}(a_z, v)$
- 2) Update $a_z \leftarrow a_z + \alpha \delta a$
- 3) Stop criterion

4) 3D MAGNETOSTATICS

Iterative solver, edge elements and gauge

Preliminaries

Iterative solver VS direct solver

	Direct solver	Iterative solver
Time complexity	Max $O(bn^2)$ b = bandwith	Each iteration is $O(n_{nz})$ with n_{nz} the number of nonzero terms
Provide matrix decomposition	yes	no
Exact	yes	no
Sensitive to matrix bandwith	yes	no
Sensitive to condition number	no	Yes (need $O(\sqrt{c})$ iterations)
Typical use	1D, 2D	3D

Application 6 : iterative vs direct solver

Poisson in 2D vs 3D

Name

utils

a_MSE_gradient_descent.ip...

b_MSE_linear_system.ipynb

c_poisson.ipynb

d_2Dmag_linear.ipynb

e_2Dmag_nonlinear.ipynb

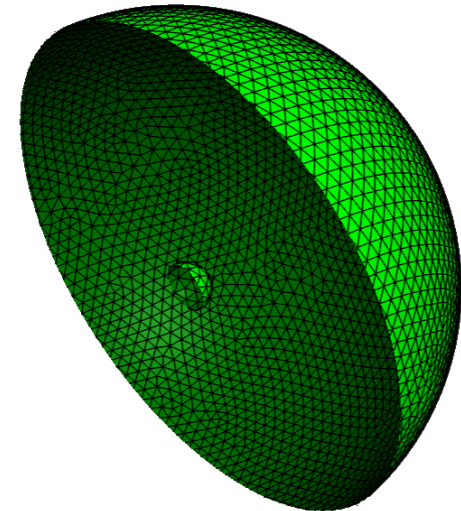
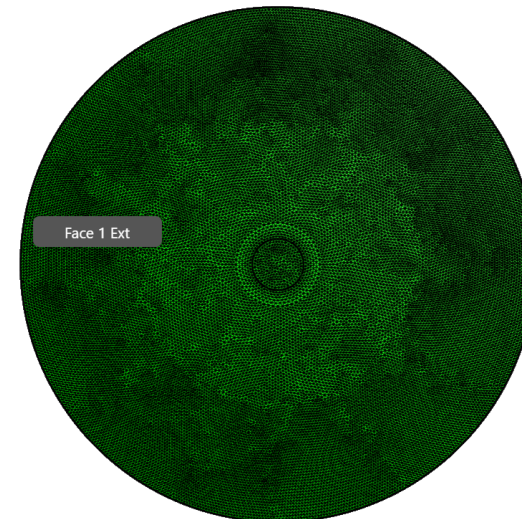
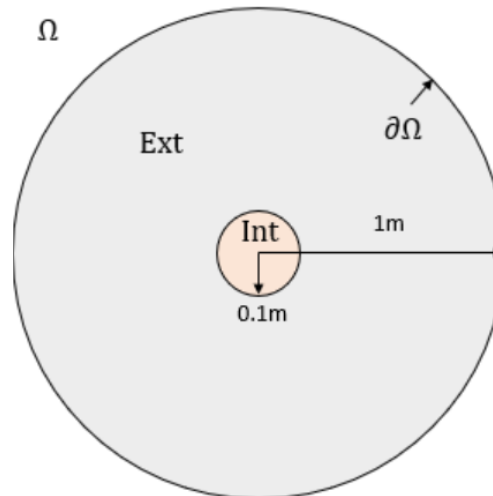
f_direct_vs_iterative.ipynb

g_gauge.ipynb

h_challenge.ipynb

- Poisson equation on a disk and a ball with the same number of DoFs

$$\forall v \in H_0^1(\Omega), \quad \int_{\Omega} \nabla v \cdot \nabla u \, dx = \int_{int} 1 \, v \, dx$$



3D Magnetostatics : variational formulation

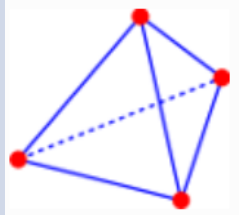
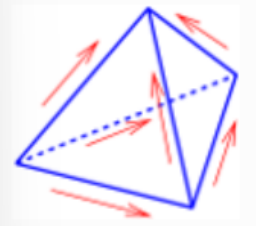
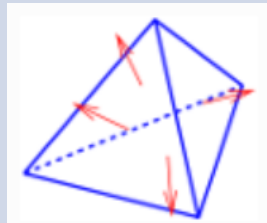
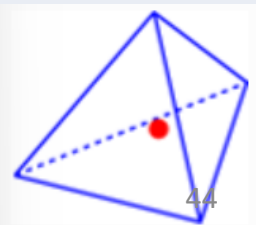
Function space

Find $\mathbf{a} \in H_0(\mathbf{curl}; \Omega) = \{\mathbf{a} \in L^2(\Omega), \mathbf{curl}(\mathbf{a}) \in L^2(\Omega), \mathbf{a} \times \mathbf{n} = 0 \text{ on } \partial\Omega\}$

$$\forall \mathbf{v} \in H_0(\mathbf{curl}; \Omega), \quad \int_{\Omega} \mathbf{curl} \mathbf{v} \cdot \mathbf{curl} \mathbf{a} = \int_{\Omega} \mathbf{v} \cdot \mathbf{j}$$

What is $H(\mathbf{curl}; \Omega)$ function space?

Function spaces

	Space	Continuity	Differential geometry	Natural DoF	Typical Elements	Illustration
<div>grad</div> <div>curl</div> <div>div</div>	H^1	Full	0-form	Field value	Lagrange	
	$H(\text{curl})$	Tangential	1-form	Edge circulation \int	Nédélec	
	$H(\text{div})$	Normal	2-form	Facet flux \iint	Raviart-Thomas	
	L^2	None	3-form	Cell integral \iiint or average value	P_0	

3D Magnetostatics : variational formulation

Gauge

Find $\mathbf{a} \in H_0(\mathbf{curl}; \Omega) = \{\mathbf{a} \in L^2(\Omega), \mathbf{curl}(\mathbf{a}) \in L^2(\Omega), \mathbf{a} \times \mathbf{n} = 0 \text{ on } \partial\Omega\}$

$$\forall \mathbf{v} \in H_0(\mathbf{curl}; \Omega), \quad \int_{\Omega} \mathbf{curl} \mathbf{v} \cdot \mathbf{v} \mathbf{curl} \mathbf{a} = \int_{\Omega} \mathbf{v} \cdot \mathbf{j}$$

Is the solution uniquely defined?

No ! Assuming \mathbf{a} is solution , then $\tilde{\mathbf{a}} = \mathbf{a} + \mathbf{grad} u$ is also solution, for u any differentiable scalar field, since $\mathbf{curl} \mathbf{grad}(\cdot) = \mathbf{0}$

3D Magnetostatics

Gauge

There are many different ways to obtain uniqueness:

- Add a small « mass » term

$$\int_{\Omega} \mathbf{curl} \, \mathbf{v} \cdot \nu \, \mathbf{curl} \, \mathbf{a} + \int_{\Omega} \epsilon \, \mathbf{v} \cdot \mathbf{a} = \int_{\Omega} \mathbf{v} \cdot \mathbf{j}$$

- Add an equation (Coulomb gauge) : $\text{div}(\mathbf{a}) = 0$

Weak form : find $\mathbf{a}, \lambda \in H_0(\mathbf{curl}; \Omega) \times H_0^1(\Omega)$,

$$\begin{cases} \forall \mathbf{v} \in H_0(\mathbf{curl}; \Omega), & \int_{\Omega} \mathbf{curl} \, \mathbf{v} \cdot \nu \, \mathbf{curl} \, \mathbf{a} + \int_{\Omega} \nabla \lambda \cdot \mathbf{v} = \int_{\Omega} \mathbf{v} \cdot \mathbf{j} \\ \forall \mu \in H_0^1(\Omega), & \int_{\Omega} \nabla \mu \cdot \mathbf{a} = 0 \end{cases}$$

Variational formulation

Gauge

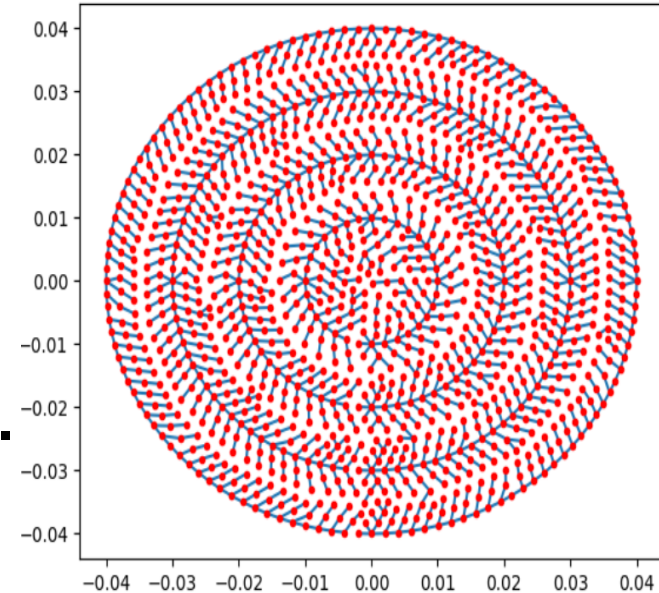
- Solve with an **iterative solver** and a compatible right-hand side

1) Find $\mathbf{T} \in H(\text{curl}, \Omega)$, s.t. $\int_{\Omega} \text{curl } \mathbf{v} \cdot \text{curl } \mathbf{T} = \int_{\Omega} \mathbf{v} \cdot \mathbf{j} \quad \forall \mathbf{v} \in H(\text{curl}, \Omega)$

2) Find $\mathbf{a} \in H(\text{curl}, \Omega)$,

$$\int_{\Omega} \text{curl } \mathbf{v} \cdot \mathbf{v} \text{ curl } \mathbf{a} = \int_{\Omega} \mathbf{v} \cdot \text{curl } \mathbf{T}$$

- Remove the redundant DoF (tree-cotree gauge).



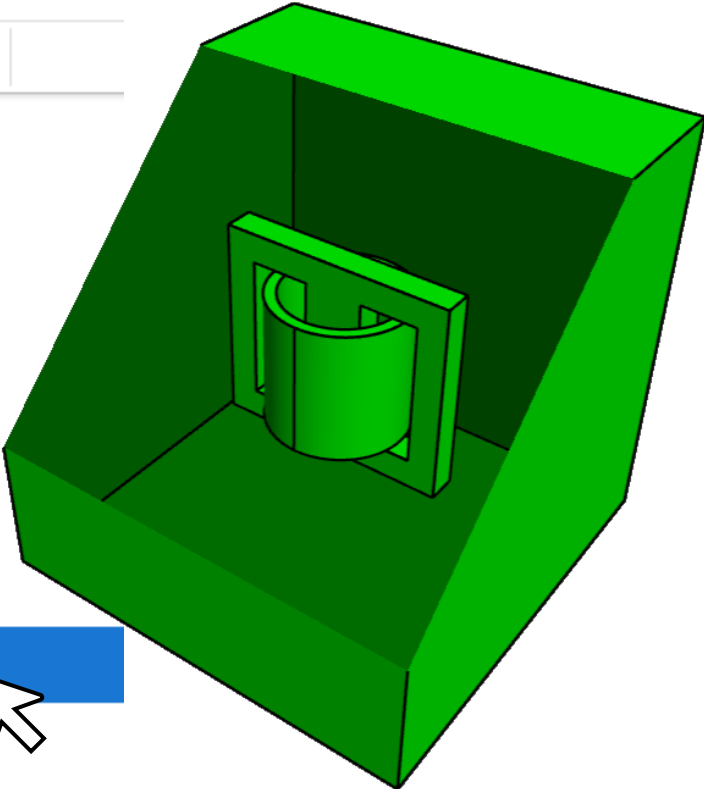
You can implement and compare all of these possibilities!

Application 7 : 3D Magnetostatics

Uniqueness of the solution

Name ▲

- utils
- a_MSE_gradient_descent.ipynb
- b_MSE_linear_system.ipynb
- c_poisson.ipynb
- d_2Dmag_linear.ipynb
- e_2Dmag_nonlinear.ipynb
- f_direct_vs_iterative.ipynb
- g_gauge.ipynb**
- h_challenge.ipynb



- **Small mass term**

$$\int_{\Omega} \text{curl } \mathbf{v} \cdot \nu \text{ curl } \mathbf{a} + \int_{\Omega} \epsilon \mathbf{v} \cdot \mathbf{a} = \int_{\Omega} \mathbf{v} \cdot \mathbf{j}$$

- **Coulomb gauge:** solve simultaneously

$$\begin{cases} \forall \mathbf{v} \in H_0(\text{curl}; \Omega), & \int_{\Omega} \text{curl } \mathbf{v} \cdot \nu \text{ curl } \mathbf{a} + \int_{\Omega} \nabla \lambda \cdot \mathbf{v} = \int_{\Omega} \mathbf{v} \cdot \mathbf{j} \\ \forall \mu \in H_0^1(\Omega), & \int_{\Omega} \nabla \mu \cdot \mathbf{a} = 0 \end{cases}$$

- **Compatible RHS**

1) Find $\mathbf{T} \in H(\text{curl}, \Omega)$, s. t.

$$\forall \mathbf{v} \in H_0(\text{curl}; \Omega), \int_{\Omega} \text{curl } \mathbf{v} \cdot \text{curl } \mathbf{j} = \int_{\Omega} \mathbf{v} \cdot \mathbf{j}$$

2) Find $\mathbf{a} \in H(\text{curl}, \Omega)$, s. t.

$$\int_{\Omega} \text{curl } \mathbf{v} \cdot \nu \text{ curl } \mathbf{a} = \int_{\Omega} \mathbf{v} \cdot \text{curl } \mathbf{T}$$

- **Tree-Cotree gauging**

3D Magnetostatics : symmetries

Boundary conditions

- Homogeneous Neumann :

$$\forall v \in H(\mathbf{curl}, \Omega) \int_{\partial\Omega} (\mathbf{h} \times \mathbf{n}) \cdot \mathbf{v} = 0 \Rightarrow \mathbf{h} \times \mathbf{n} = 0$$

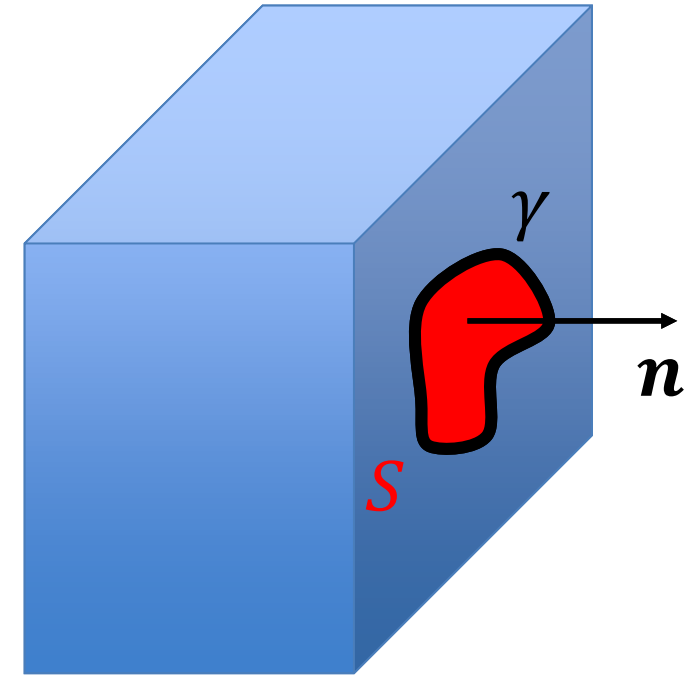
\Rightarrow magnetic field orthogonal to the boundary
= **Symmetry**

- Homogeneous Dirichlet : $\mathbf{a} \times \mathbf{n} = 0$

\Rightarrow vector potential orthogonal to the boundary

$$\Rightarrow \forall S \in \partial\Omega, \phi_{out} = \iint_{S_\gamma \in \partial\Omega} \mathbf{B} \cdot d\mathbf{S} = \oint_{\gamma=\partial S} \mathbf{a} \cdot d\mathbf{l} = 0$$

\Rightarrow flux density tangential to the boundary
= **Anti-symmetry**



Application 8 : synthesis

Challenge : implement the fastest 3D magnetostatic solver for the inductance problem

Name ▲

utils

• a_MSE_gradient_descent.ipynb

• b_MSE_linear_system.ipynb

• c_poisson.ipynb

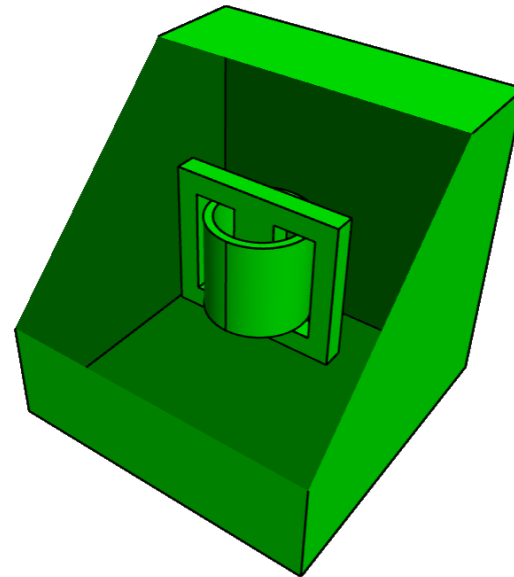
• d_2Dmag_linear.ipynb

• e_2Dmag_nonlinear.ipynb

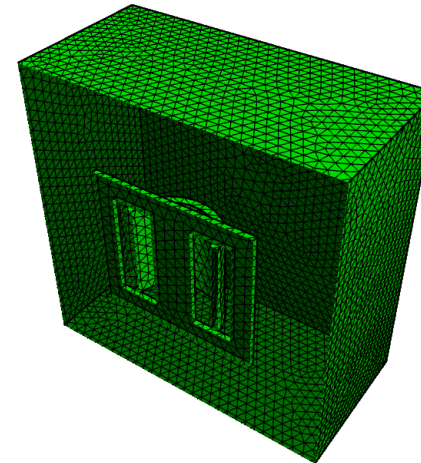
• f_direct_vs_iterative.ipynb

• g_gauge.ipynb

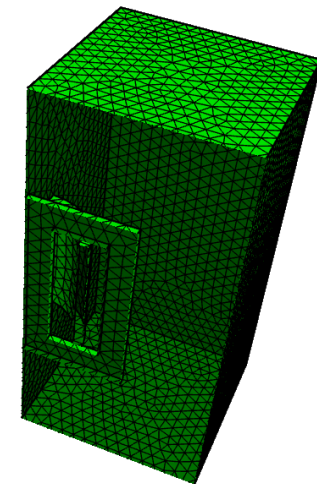
• h_challenge.ipynb



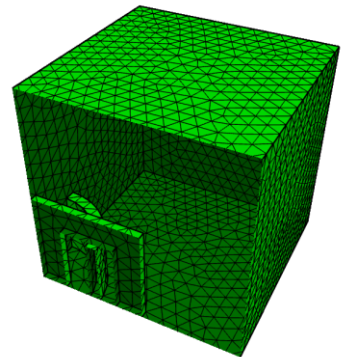
Full model



1/2



1/4



1/8

À discuter le 10/12 en office hours !

Outlook

The journey is not over...

- How to mesh / remesh? → GMSH, Netgen...
- How to control the error ? → adaptive mesh refinement
- Advanced solvers (multifrontal, multigrid, etc.)
- Harmonic / Time dependant problems...
- Multiphysics / coupled problems
- What can we put over FEM? → interface tracking, topology optimization...
- Other methods(BEM, FIT, IGA, MoM, hybrid methods...)

References

- J. Schöberl, An Interactive Introduction to the Finite Element Method (<https://jschoeberl.github.io/iFEM/intro.html>)
- A. Ern, Finite Elements I: Approximation and interpolation <https://hal.science/hal-03226049v1>
- Z. Ren, “Influence of the R.H.S. on the Convergence Behaviour of the Curl-Curl Equation,” *IEEE Trans. Magn.*, vol. 32, no. 3, pp. 655–658, 1996.



Thank you for your attention !

GeePs'N Talks special session

T. Cherrière, A. El Gode, T. Gauthey