

Optimization and control of magnetic high fields

Romain Hild, Christophe Prud'homme, Christophe Trophime

30/05/2018

CANUM 2018

Universté de Strasbourg - IRMA

Introduction

Magnetic Fields (T: Tesla)

Static Field:

- Grenoble: 36T, soon 43T,
- Tallahassee, USA: 45T

Applications

- Semiconductors and nanophysics,
- Chemistry, Biochemistry,

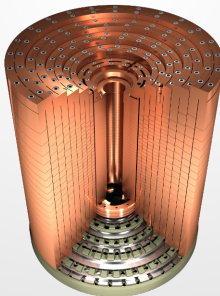
Pulsed Field:

- Toulouse: 99T,
- Los Alamos, USA: 100T

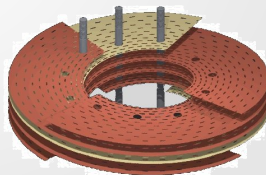
- Magneto-science, Levitation,
- Applied Superconductivity



(a) Exterior

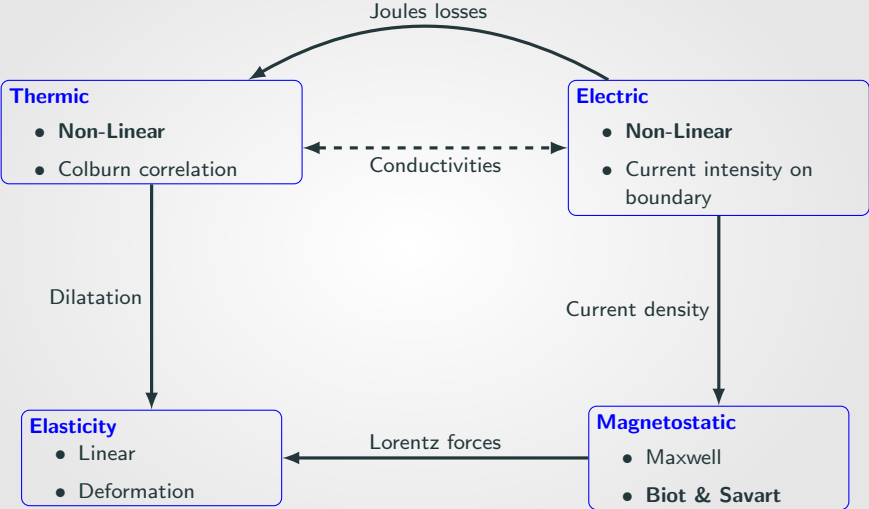


(b) Interior



(c) Bitter

Involved physics



Optimisation

- Use experimental measurements to **identify unknown parameters**, such as heat exchange coefficient or conductivities
- Parametrize the form of the magnet to reach a **better homogeneity** of the magnetic field in the region of interest

Output and control quantity of interest

- Use different experimental and geometric parameters to **control the magnetic field** to create a magnetic structure to confine plasma
- Find parameters such as the magnetic field allow the **levitation** of an object inside the region of interest

Real time computation and visualization of the magnetic field

- Allow the **real time visualization** of the effect of different parameters on the magnetic field



Reduced Order Method

The Reduced Basis Method

Weak formulation of the model : $a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu})$

FEM Approximation

$$\underbrace{X^{\mathcal{N}} = \text{span}\{\phi_1, \dots, \phi_{\mathcal{N}}\}}_{\text{FEM approximation space}} \rightarrow \underbrace{u^{\mathcal{N}}(\boldsymbol{\mu}) = \sum_{i=1}^{\mathcal{N}} u_i^{\mathcal{N}} \phi_i}_{\mathcal{N} \approx 10^6} \rightarrow \underbrace{A^{\mathcal{N}}(\boldsymbol{\mu}) u^{\mathcal{N}}(\boldsymbol{\mu}) = F^{\mathcal{N}}(\boldsymbol{\mu})}_{\mathcal{N} \times \mathcal{N} \text{ system expensive to solve}}$$

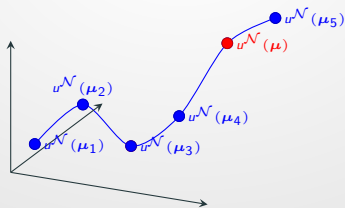
RB Approximation : $u^N(\boldsymbol{\mu}) \approx u^{\mathcal{N}}(\boldsymbol{\mu})$: linear combination of FEM solution

$$\underbrace{W^N = \text{span}\{u^{\mathcal{N}}(\boldsymbol{\mu}_1), \dots, u^{\mathcal{N}}(\boldsymbol{\mu}_N)\}}_{\text{RB approximation space}} \rightarrow \underbrace{u^N(\boldsymbol{\mu}) = \sum_{i=1}^N u_i^N(\boldsymbol{\mu}) \underline{u}^{\mathcal{N}}(\boldsymbol{\mu}_i)}_{10 \leq N \leq 100} \rightarrow \underbrace{A^N(\boldsymbol{\mu}) u^N(\boldsymbol{\mu}) = F^N(\boldsymbol{\mu})}_{N \times N \text{ system cheaper to solve}}$$

The reduced basis method

Ingredients

- Set of parameters : \mathcal{D}^μ
- FEM ‘truth’ approximation
 - $X^{\mathcal{N}}$: finite element approximation space of dimension $\mathcal{N} \gg 1$
 - $u^{\mathcal{N}}(\mu) \in X^{\mathcal{N}}$ is solution of $a(u^{\mathcal{N}}(\mu), v; \mu) = f(v; \mu) \quad \forall v \in X^{\mathcal{N}}$
- RB approximation
 - Sample : $S_N = \{\mu_1 \in \mathcal{D}^\mu, \dots, \mu_N \in \mathcal{D}^\mu\}$
 - Approximation space : $W_N = \text{span}\{u^{\mathcal{N}}(\mu_1), \dots, u^{\mathcal{N}}(\mu_N)\}$ with $N \ll \mathcal{N}$
 - Galerkin projection on W_N to determine RB coefficients



$$u^N(\boldsymbol{\mu}) = \sum_{i=1}^N u_i^N(\boldsymbol{\mu}) \underline{u}^{\mathcal{N}}(\boldsymbol{\mu}_i)$$

$N \times N$ system to solve :

$$\sum_{i=1}^N a(\underline{u}^{\mathcal{N}}(\boldsymbol{\mu}_i), v_k; \boldsymbol{\mu}) u_i^N(\boldsymbol{\mu}) = f(v_k; \boldsymbol{\mu}), \quad 1 \leq k \leq N$$

Computational opportunities

If the parameter dependence can be expressed as an affine decomposition :

$$a(u, v; \boldsymbol{\mu}) = \sum_q^{Q_a} \theta_a^q(\boldsymbol{\mu}) a_q(u, v) \quad \text{and} \quad f(v; \boldsymbol{\mu}) = \sum_q^{Q_f} \theta_f^q(\boldsymbol{\mu}) f_q(v)$$
$$\Rightarrow \sum_{i=1}^N \left[\sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu}) \underbrace{a^q(\underline{u}^{\mathcal{N}}(\boldsymbol{\mu}_i), \underline{u}^{\mathcal{N}}(\boldsymbol{\mu}_j))}_{\text{precomputed}} \right] u_i^N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu}) \underbrace{f^q(\underline{u}^{\mathcal{N}}(\boldsymbol{\mu}_j))}_{\text{precomputed}}$$

Strategy : Offline-Online decomposition

- Computationally intensive initial preprocessing
- Greatly reduced marginal cost

Some Problematics

Multi-Physics Problems

- Monolithic Resolution
- Use Block Affine Decomposition

Non Affine Problems

- Empirical Interpolation Method (EIM) : approximate an expression with an affine decomposition
- Discrete EIM (DEIM) : approximate a discrete operator (matrix or vector) with an affine decomposition

Non Linear Problems

- Use of EIM/DEIM
- Iterative methods : Picard / Newton

What if the problem is non-linear and non-affine ?

a and f depend on non affinely parametrized $\{w_i(u, \mathbf{x}, \boldsymbol{\mu})\}_i$:

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}; \{w_i(u, \mathbf{x}; \boldsymbol{\mu})\}_i) = f(v; \boldsymbol{\mu}; \{w_i(u, \mathbf{x}; \boldsymbol{\mu})\}_i) \quad \forall v \in X$$

⇒ Affine decomposition doesn't exist !

Solution : build an affine approximation using Empirical Interpolation Method

$$w_i(u, \mathbf{x}, \boldsymbol{\mu}) \approx w_{i, M_i}(u, \mathbf{x}; \boldsymbol{\mu}) = \sum_{m=1}^{M_i} \beta_{i,m}^{M_i}(u; \boldsymbol{\mu}) q_{i,m}(\mathbf{x})$$

What if the problem is non-linear and non-affine ?

a and f depend on non affinely parametrized $\{w_i(u, \mathbf{x}, \boldsymbol{\mu})\}_i$:

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}; \{w_i(u, \mathbf{x}, \boldsymbol{\mu})\}_i) = f(v; \boldsymbol{\mu}; \{w_i(u, \mathbf{x}, \boldsymbol{\mu})\}_i) \quad \forall v \in X$$

⇒ Affine decomposition doesn't exist !

Solution : build an affine approximation using Empirical Interpolation Method

$$w_i(u, \mathbf{x}, \boldsymbol{\mu}) \approx w_{i, M_i}(u, \mathbf{x}; \boldsymbol{\mu}) = \sum_{m=1}^{M_i} \beta_{i, m}^{M_i}(u; \boldsymbol{\mu}) q_{i, m}(\mathbf{x})$$

$$\underbrace{\sum_{q=1}^{Q_a^{eim}} \sum_{m=1}^{M_q^a} \gamma_{a, m}^q(u; \boldsymbol{\mu}) a_m^q(u, v)}_{\text{non-affine part of } a} + \underbrace{\sum_{l=1}^{Q_a^{aff}} \theta_a^l(\boldsymbol{\mu}) a^l(u, v)}_{\text{affine part of } a} = \underbrace{\sum_{q=1}^{Q_f^{eim}} \sum_{m=1}^{M_q^f} \gamma_{f, m}^q(u; \boldsymbol{\mu}) f_m^q(v)}_{\text{non-affine part of } f} + \underbrace{\sum_{l=1}^{Q_f^{aff}} \theta_f^l(\boldsymbol{\mu}) f^l(v)}_{\text{affine part of } f}$$

The Discrete Empirical Interpolation Method (DEIM)

- Build an affine approximation of a tensor $T(\mu)$

$$T_M(\mu) = \sum_{m=0}^M \beta^m(\mu) \Phi^m \quad (1)$$

where

- $\beta^m(\mu)$ are scalar functions, $\beta^m : \mathcal{D} \rightarrow \mathbf{R}$
 - Φ^m are tensors independent of μ
-
- Offline/Online strategy very close to EIM algorithm.
-
- Can automatically provide the affine decomposition of a parametrized operator
 - Geometric Parameters
 - Complex Operator, ex : stabilization parameter

DEIM : Computation of $\beta^m(\mu)$

At any rank M , the vector of coefficient $\Theta_M(\mu) = (\beta^1(\mu), \dots, \beta^M(\mu))^T$ is the solution of the $M \times M$ system

$$B_{\mathcal{I}_M} \Theta_M(\mu) = T_{\mathcal{I}_M}(\mu) \quad (2)$$

where

- $(B_{\mathcal{I}_M})_{k,l} = (\Phi_l)_{i_k}$: the columns of $B_{\mathcal{I}_M}$ are composed by the vectors Φ_l in which you only keep the entry for the index $i_k \in \mathcal{I}_M$
- $(T_{\mathcal{I}_M}(\mu))_k = (T(\mu))_{i_k}$: $T_{\mathcal{I}_M}(\mu)$ is the evaluation of the tensor $T(\mu)$ in which you only keep the entry for the index $i_k \in \mathcal{I}_M$

$B_{\mathcal{I}_M}$ is updated during the greedy algorithm when a new couple (Φ_M, i_m) is selected.

The computation of $T_{\mathcal{I}_M}(\mu)$ has to be done online. It can not depend of the finite element dimension \mathcal{N} !

Applications

Problem

Find $V = \sum_{n=1}^N V_n(\boldsymbol{\mu})\xi_i^V$ and $T = \sum_{n=1}^N T_n(\boldsymbol{\mu})\xi_i^T$ such that:

$$\left\{ \begin{array}{ll} -\nabla(\sigma_i(T)\nabla V) = 0 & \text{in } \Omega_i, i = 1, \dots, P_{mat} \\ -\nabla(k_i(T)\nabla T) = \mathbf{j} \cdot \mathbf{E} & \text{in } \Omega_i, i = 1, \dots, P_{mat} \\ V = g_{VD} & \text{on } \Gamma_{VD} \\ -\sigma_i(T)\nabla V \cdot \mathbf{n} = g_{VN} & \text{on } \Gamma_{VN} \\ -k_i(T)\nabla T \cdot \mathbf{n} = g_{TN} & \text{on } \Gamma_{TN} \\ -k_i(T)\nabla T \cdot \mathbf{n} + G_{TR}^1 T = g_{TR}^2 & \text{on } \Gamma_{TN} \end{array} \right.$$

All the configuration is made through a json file.

Empirical Interpolation Method

We need to use EIM for the conductivities that depends on T and for the right hand side:

$$\sigma(T) \approx \sum_{m_\sigma=1}^{M_\sigma} \beta_{m_\sigma}^\sigma(\boldsymbol{\mu}) \mathbf{q}_{m_\sigma}^\sigma, \quad k(T) \approx \sum_{m_k=1}^{M_k} \beta_{m_k}^k(\boldsymbol{\mu}) \mathbf{q}_{m_k}^k, \quad \mathbf{j} \cdot \mathbf{E} \approx \sum_{m_f=1}^{M_f} \beta_{m_f}^f(\boldsymbol{\mu}) \mathbf{q}_{m_f}^f$$

In the case of experimental parameters, non geometric (σ, k, h, \dots):

$$\mathbf{B}(\mathbf{x}; \boldsymbol{\mu}) = \frac{\mu_0}{4\pi} \int_{\Omega_{cond}} \frac{\mathbf{j}(\mathbf{r}; \boldsymbol{\mu}) \times (\mathbf{x} - \mathbf{r})}{|\mathbf{x} - \mathbf{r}|^3} d\mathbf{r} \quad \forall \mathbf{x} \in \Omega_{mgn}$$

Using reduced basis for \mathbf{j} :

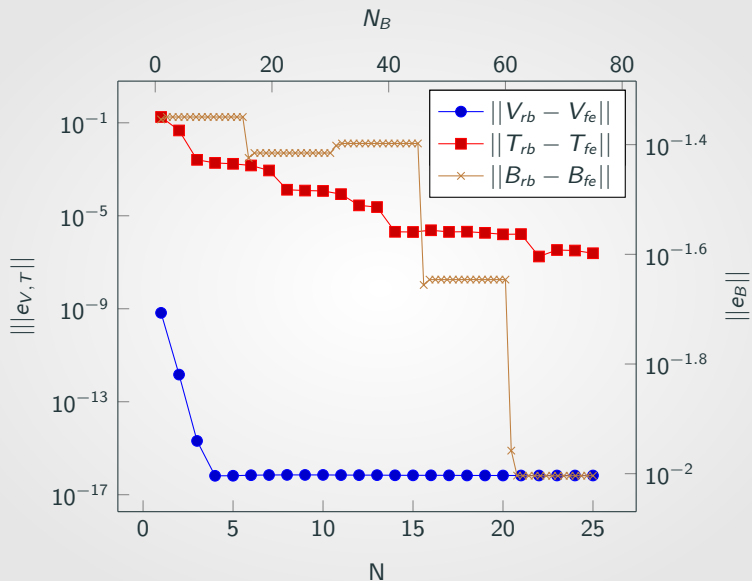
$$\mathbf{j}(\mathbf{r}; \boldsymbol{\mu}) = -\sigma(T)\nabla V = - \sum_{m_\sigma=1}^{M_\sigma} \beta_{m_\sigma}^\sigma(\boldsymbol{\mu}) \mathbf{q}_{m_\sigma}^\sigma(\mathbf{r}) \sum_{i=1}^N V_i^N(\boldsymbol{\mu}) \nabla \xi_i^V(\mathbf{r})$$

leads to:

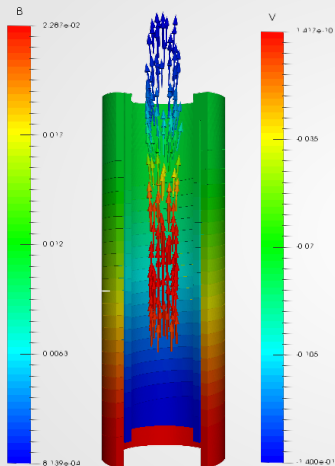
$$\mathbf{B}(\mathbf{x}; \boldsymbol{\mu}) = \sum_{i=1}^N \sum_{m_\sigma=1}^{M_\sigma} V_i^N(\boldsymbol{\mu}) \beta_{m_\sigma}^\sigma(\boldsymbol{\mu}) \underbrace{\frac{\mu_0}{4\pi} \int_{\Omega_{cond}} \frac{-\mathbf{q}_{m_\sigma}(\mathbf{r}) \nabla \xi_i^V(\mathbf{r}) \times (\mathbf{x} - \mathbf{r})}{|\mathbf{x} - \mathbf{r}|^3} d\mathbf{r}}_{\xi_{i,m_\sigma}^{\mathbf{B}}(\mathbf{x}) \text{ precomputed}}$$

We can use the same parallel algorithm to compute $\xi_{i,m_\sigma}^{\mathbf{B}}(\mathbf{x})$.

Results



Results



- Parameters: σ, k, V_D, h, T_w
- Linear case: $\sigma(T) = \sigma(T_0)$ and $k(T) = k(T_0)$

- Dimensions:

	Ω_{cond}	Ω_{mgn}
dofs	1350000	3000

- Times:

	FE	RB
V	45s	0.05s
B	100s	0.0005s

In the case of geometric parameter, we need to compute:

$$\begin{aligned}\mathbf{B}(\mathbf{x}; \boldsymbol{\mu}) &= \frac{\mu_0}{4\pi} \int_{\phi_{\boldsymbol{\mu}}(\Omega_{cond})} \frac{-\sigma \nabla V \times (\mathbf{x} - \mathbf{r})}{|\mathbf{x} - \mathbf{r}|^3} d\mathbf{r} \quad \forall \mathbf{x} \in \Omega_{mgn} \\ &= \frac{\mu_0}{4\pi} \int_{\Omega_{cond}^{ref}} \frac{-\sigma \nabla V \cdot \mathbf{J}_{\phi_{\boldsymbol{\mu}}}^{-1} \times (\mathbf{x} - \phi_{\boldsymbol{\mu}}(\mathbf{r}))}{|\mathbf{x} - \phi_{\boldsymbol{\mu}}(\mathbf{r})|^3} d\mathbf{r} \quad \forall \mathbf{x} \in \Omega_{mgn}\end{aligned}$$

How to handle the term $|\mathbf{x} - \phi_{\boldsymbol{\mu}}(\mathbf{r})|^3$?

- We could use EIM for the term $\phi_{\boldsymbol{\mu}}(\mathbf{r})$, but the absolute value would prevent to extract the coefficients for the affine decomposition
- And we need to compute it for every $\mathbf{x} \in \Omega_{mgn}$

In the case of geometric parameter, we need to compute:

$$\begin{aligned}\mathbf{B}(\mathbf{x}; \boldsymbol{\mu}) &= \frac{\mu_0}{4\pi} \int_{\phi_{\boldsymbol{\mu}}(\Omega_{cond})} \frac{-\sigma \nabla V \times (\mathbf{x} - \mathbf{r})}{|\mathbf{x} - \mathbf{r}|^3} d\mathbf{r} \quad \forall \mathbf{x} \in \Omega_{mgn} \\ &= \frac{\mu_0}{4\pi} \int_{\Omega_{cond}^{ref}} \frac{-\sigma \nabla V \cdot \mathbf{J}_{\phi_{\boldsymbol{\mu}}}^{-1} \times (\mathbf{x} - \phi_{\boldsymbol{\mu}}(\mathbf{r}))}{|\mathbf{x} - \phi_{\boldsymbol{\mu}}(\mathbf{r})|^3} d\mathbf{r} \quad \forall \mathbf{x} \in \Omega_{mgn}\end{aligned}$$

How to handle the term $|\mathbf{x} - \phi_{\boldsymbol{\mu}}(\mathbf{r})|^3$?

- We could use EIM for the term $\phi_{\boldsymbol{\mu}}(\mathbf{r})$, but the absolute value would prevent to extract the coefficients for the affine decomposition
- And we need to compute it for every $\mathbf{x} \in \Omega_{mgn}$

Discrete Empirical Interpolation Method

We approach the magnetic field with:

$$\mathbf{B}(\mathbf{x}; \boldsymbol{\mu}) \approx \sum_{m=1}^{M_B} \beta_m(\boldsymbol{\mu}) \mathbf{Q}_m(\mathbf{x})$$

To compute $\beta(\boldsymbol{\mu})$, we need to solve the following problem:

$$\begin{pmatrix} \xi_1(t_1) & \cdots & \xi_1(t_M) \\ \vdots & \ddots & \vdots \\ \xi_M(t_1) & \cdots & \xi_M(t_M) \end{pmatrix} \begin{pmatrix} \beta_1(\boldsymbol{\mu}) \\ \vdots \\ \beta_M(\boldsymbol{\mu}) \end{pmatrix} = \begin{pmatrix} \mathbf{B}(t_1; \boldsymbol{\mu}) \\ \vdots \\ \mathbf{B}(t_M; \boldsymbol{\mu}) \end{pmatrix}$$

So we need to know $\mathbf{B}(t_i; \boldsymbol{\mu})$, where is the advantage ?

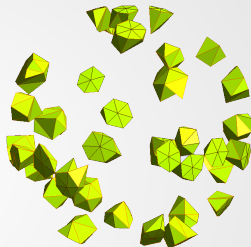
We approach the magnetic field with:

$$\mathbf{B}(\mathbf{x}; \boldsymbol{\mu}) \approx \sum_{m=1}^{M_B} \beta_m(\boldsymbol{\mu}) \mathbf{Q}_m(\mathbf{x})$$

To compute $\beta(\boldsymbol{\mu})$, we need to solve the following problem:

$$\begin{pmatrix} \xi_1(t_1) & \cdots & \xi_1(t_M) \\ \vdots & \ddots & \vdots \\ \xi_M(t_1) & \cdots & \xi_M(t_M) \end{pmatrix} \begin{pmatrix} \beta_1(\boldsymbol{\mu}) \\ \vdots \\ \beta_M(\boldsymbol{\mu}) \end{pmatrix} = \begin{pmatrix} \mathbf{B}(t_1; \boldsymbol{\mu}) \\ \vdots \\ \mathbf{B}(t_M; \boldsymbol{\mu}) \end{pmatrix}$$

So we need to know $\mathbf{B}(t_i; \boldsymbol{\mu})$, where is the advantage ?



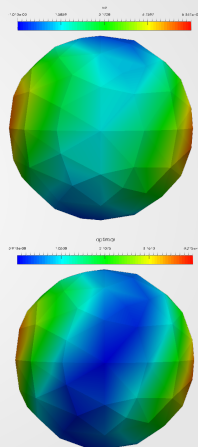
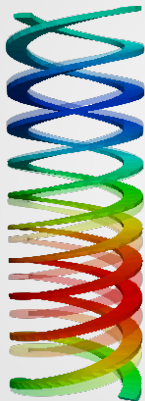
We *only* need to know \mathbf{B} on the interpolation points t_i , $i = 1, \dots, M_B$

Optimization of the homogeneity

We apply ϕ a geometrical transformation to modify the shape of the helices to attain a better homogeneity

$$\phi = \begin{pmatrix} x \cos(\alpha(\mu)) + y \sin(\alpha(\mu)) \\ -x \sin(\alpha(\mu)) + y \cos(\alpha(\mu)) \\ z \end{pmatrix}$$

where α is a Bezier curve which control points are the parameters μ of the optimization.



	initial	optimal
$\max \mathbf{B}_z $	0.41337	0.4051
$\min \mathbf{B}_z $	0.41076	0.4034
$\frac{\max \mathbf{B}_z }{\min \mathbf{B}_z } - 1$	5381.9	3544.9

Conclusion and Perspectives

Conclusions

- Non-Linear ThermoElectric model with or without geometric parameters
- Biot & Savart model with or without geometric parameters
- First implementation of optimization problems with or without geometric parameters

Perspectives

- Add Levitation and ions source applications
- Add Maxwell and Elasticity model
- Use HDG methods

Conclusions

- Non-Linear ThermoElectric model with or without geometric parameters
- Biot & Savart model with or without geometric parameters
- First implementation of optimization problems with or without geometric parameters

Thank you!

Perspectives

- Add Levitation and ions source applications
- Add Maxwell and Elasticity model
- Use HDG methods