Overview of Numerical Approaches

Finite Element Formulations for High-Temperature Superconductors

C. Geuzaine, J. Dular and B. Vanderheyden

University of Liège, Institut Montefiore B28, 4000 Liège, Belgium

Third International School on Numerical Modelling for Applied Superconductivity

Saas-Fee, Switzerland, 6-10 June 2022

Some background

- I am a professor at the University of Liège in Belgium, where I lead a team of about 15 people in the Montefiore Institute (Electrical Engineering and Computer Science Department), at the intersection of applied math, electromagnetism and scientific computing

Some background

- I am a professor at the University of Liège in Belgium, where I lead a team of about 15 people in the Montefiore Institute (Electrical Engineering and Computer Science Department), at the intersection of applied math, electromagnetism and scientific computing
- Our research interests include modelling, analysis, algorithm development, and simulation for problems arising in various areas of engineering and science
  - Current applications: electromagnetics, geophysics, biomedical problems

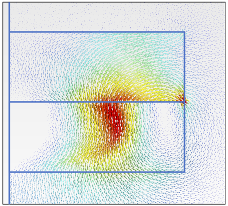
Some background

- I am a professor at the University of Liège in Belgium, where I lead a team of about 15 people in the Montefiore Institute (Electrical Engineering and Computer Science Department), at the intersection of applied math, electromagnetism and scientific computing
- Our research interests include modelling, analysis, algorithm development, and simulation for problems arising in various areas of engineering and science
  - Current applications: electromagnetics, geophysics, biomedical problems
- We write quite a lot of codes, several released as open source software: Gmsh, GetDP, ...

Some background

- I am a professor at the University of Liège in Belgium, where I lead a team of about 15 people in the Montefiore Institute (Electrical Engineering and Computer Science Department), at the intersection of applied math, electromagnetism and scientific computing
- Our research interests include modelling, analysis, algorithm development, and simulation for problems arising in various areas of engineering and science
  - Current applications: electromagnetics, geophysics, biomedical problems
- We write quite a lot of codes, several released as open source software: Gmsh, GetDP, ...
- Our toolkit for modelling superconductors: Life-HTS

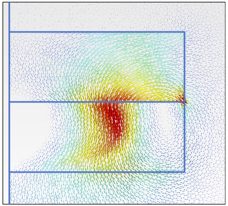
Life-HTS



http://www.life-hts.uliege.be

- Life-HTS: Liège University finite element models for High-Temperature Superconductors
- Numerical models for systems that contain both superconducting and ferromagnetic materials

Life-HTS



http://www.life-hts.uliege.be

- Life-HTS: Liège University finite element models for High-Temperature Superconductors
- Numerical models for systems that contain both superconducting and ferromagnetic materials

More specifically:

- Transient analysis for calculating field maps, magnetization, eddy currents, losses, . . .
- Stable schemes for dealing with nonlinear constitutive laws
- Includes formulations (e.g. $h(\text{-}\phi)\text{-}a$) for combining ferromagnetic and superconducting materials

University of Liège

Sart Tilman Campus

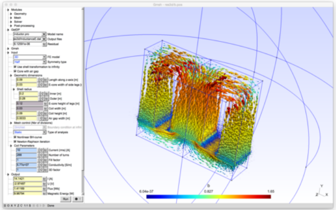
Montefiore Institute

The city of Liège

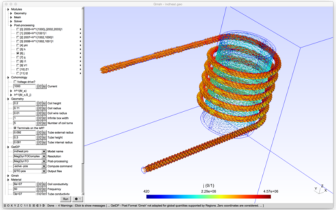
Life-HTS – Under the hood

Life-HTS is based on ONELAB (Open Numerical Engineering LABoratory), an interface to

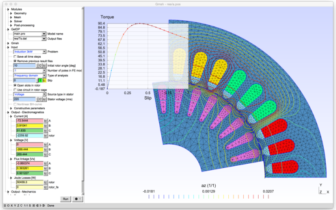
- the mesh generator Gmsh (https://gmsh.info)
- the finite element solver GetDP (https://getdp.info)



transformer



induction heating



rotating machine

Open-source, available for Windows, macOS, Linux, iOS, Android

Download from https://onelab.info

Life-HTS – Under the hood

Some numbers:

- Gmsh and GetDP started in 1996, ONELAB in 2010
- About 500k lines of C++ code
- Released under the GNU GPL v2+ (free and open source)

Life-HTS – Under the hood

Some numbers:

- Gmsh and GetDP started in 1996, ONELAB in 2010
- About 500k lines of C++ code
- Released under the GNU GPL v2+ (free and open source)
- 3 main core developers; about 100 with ≥ 1 commit
- About 2,000 registered users on the development site https://gitlab.onelab.info

Life-HTS – Under the hood

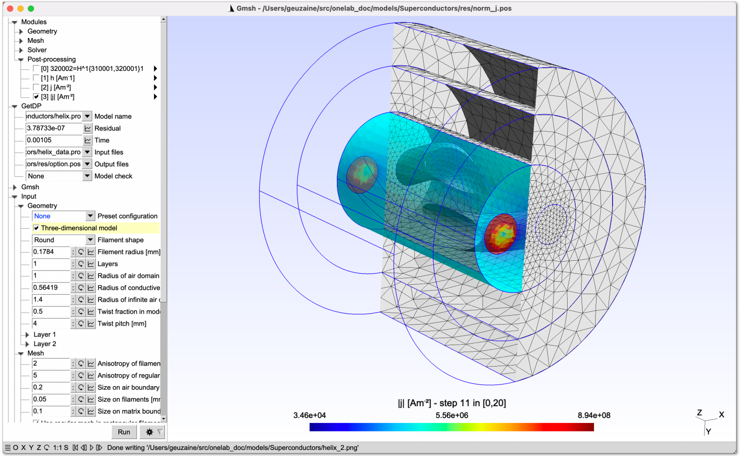
Some numbers:

- Gmsh and GetDP started in 1996, ONELAB in 2010
- About 500k lines of C++ code
- Released under the GNU GPL v2+ (free and open source)
- 3 main core developers; about 100 with ≥ 1 commit
- About 2,000 registered users on the development site https://gitlab.onelab.info
- About 20,000 downloads per month (70% Windows)
- About 1,000 citations per year on Google Scholar; Gmsh has become one of the most popular open source finite element mesh generators

Hands-on: a first example

2D and 3D model of twisted HTS wires

Launch Gmsh, then open `models/Superconductors/helix.pro`



A Sketch of the Finite Element Method

A simple 1D boundary value problem

- Solve

$$-\frac{d}{dx}\left(a(x)\,\frac{du}{dx}\right)+b(x)\,u=f,\quad 0\leq x\leq 1,$$

with

$$a(x)=1+x,\quad b(x)=\frac{1}{1+x},\quad f(x)=\frac{2}{1+x},$$

and boundary conditions $u(0)=0$ and $u(1)=1$.

A simple 1D boundary value problem

- Solve

$$-\frac{d}{dx}\left(a(x)\,\frac{du}{dx}\right)+b(x)\,u=f,\quad 0\leq x\leq 1,$$

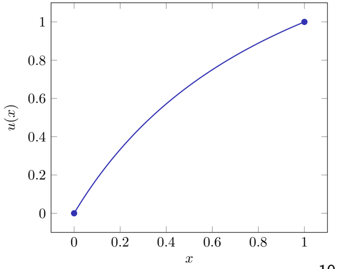
with

$$a(x)=1+x,\quad b(x)=\frac{1}{1+x},\quad f(x)=\frac{2}{1+x},$$

and boundary conditions $u(0)=0$ and $u(1)=1$.

- Solution

$$u(x)=\frac{2x}{1+x}$$



Finite Element Method: step 1

- Approximate $u(x)$ in a finite dimensional space

$$u_m(x) \quad = \quad \phi_0(x) + \sum_{\ell=1}^{m} \gamma_\ell \, \phi_\ell(x),$$

with $\phi_0(x) = x$ such that $\phi_0(0) = 0$ and $\phi_0(1) = 1$, whereas

$$\phi_\ell(0) = 0, \quad \phi_\ell(1) = 0, \quad \ell = 1, \ldots, m.$$

The linearly independent functions $\phi_\ell(x)$, $\ell > 0$ span an approximation space, $\mathcal{H}_m^0$, of dimension m.

Finite Element Method: step 2

- Define the residual

$$r(x) = -\frac{d}{dx}\left(a(x)\,\frac{du_m}{dx}\right) + b(x)\,u_m - f(x),$$

and require $r(x)$ to be orthogonal to $\mathcal{H}_m^0$, i.e.

$$(r, \phi_k) = 0, \quad k = 1, \ldots, m,$$

where $(u, v) = \int_0^1 u(x)v(x)dx$.

Finite Element Method: step 2

- Define the residual

$$r(x) = -\frac{d}{dx}\left(a(x)\,\frac{du_m}{dx}\right) + b(x)\,u_m - f(x),$$

and require $r(x)$ to be orthogonal to $\mathcal{H}_m^0$, i.e.

$$(r, \phi_k) = 0, \quad k = 1, \ldots, m,$$

where $(u, v) = \int_0^1 u(x)v(x)dx$. This gives, for $k = 1, \ldots, m$:

$$\sum_{\ell=0}^{m} \gamma_\ell \left(-\frac{d}{dx}\left(a(x)\,\frac{d\phi_\ell}{dx}\right), \phi_k\right) + \left(b(x)\,\phi_\ell, \phi_k\right) = \left(f(x), \phi_k\right),$$

with $\gamma_0 = 1$.

Finite Element Method: steps 3 and 4

- Integrate by part to relax the differentiability requirements on ϕ_k and seek for a weak solution,

$$\sum_{\ell=1}^{m} a_{k,\ell}\,\gamma_\ell = (f(x), \phi_k) - a_{k,0}, \quad k = 1, \ldots, m,$$

where

$$a_{k,\ell} = \left(a(x)\,\frac{d\phi_\ell}{dx}, \frac{d\phi_k}{dx}\right) + (b(x)\,\phi_\ell, \phi_k)\,.$$

Finite Element Method: steps 3 and 4

- Integrate by part to relax the differentiability requirements on ϕ_k and seek for a weak solution,

$$\sum_{\ell=1}^{m} a_{k,\ell}\, \gamma_\ell = (f(x), \phi_k) - a_{k,0}, \quad k = 1, \ldots, m,$$

where

$$a_{k,\ell} = \left(a(x)\, \frac{d\phi_\ell}{dx}, \frac{d\phi_k}{dx} \right) + (b(x)\, \phi_\ell, \phi_k)\,.$$

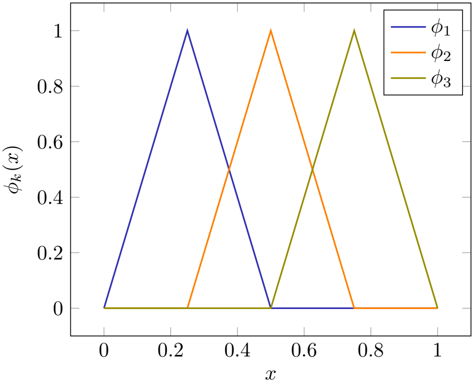
- Choose functions ϕ_k with a restricted support. The resulting matrix elements $a_{k,\ell}$ vanish for most (k,ℓ) pairs.

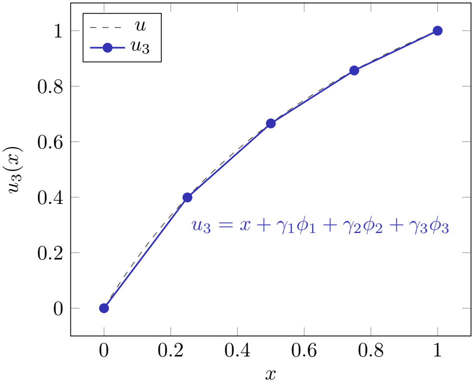
  A sparse system is obtained, which saves computational cost.

Numerical example

Function space: use piece-wise linear nodal functions (here, $m = 3$)

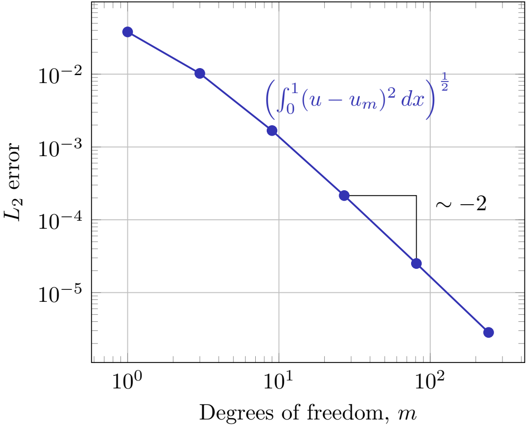
Approximate solution:





Numerical example

Convergence when the mesh is refined:



Finite Element Method: summary

- Need a function space for the approximations u_m,

$$u_m(x) = \phi_0(x) + \sum_{\ell=1}^{m} \gamma_\ell \, \phi_\ell(x), \quad \text{with boundary conditions}$$

- Impose $(r, \phi_k) = 0$ in weak form for all ϕ_k, to get the linear system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

with

$$\boldsymbol{A}_{k,\ell} = \left(a\frac{d\phi_\ell}{dx}, \frac{d\phi_k}{dx} \right) + (b\phi_\ell, \phi_k)\,, \quad \boldsymbol{x}_\ell = \gamma_\ell, \quad \text{and} \quad \boldsymbol{b}_k = (f, \phi_k).$$

Finite Element Method: summary

- Need a function space for the approximations u_m,

$$u_m(x) = \phi_0(x) + \sum_{\ell=1}^{m} \gamma_\ell \, \phi_\ell(x), \quad \text{with boundary conditions}$$

- Impose $(r, \phi_k) = 0$ in weak form for all ϕ_k, to get the linear system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

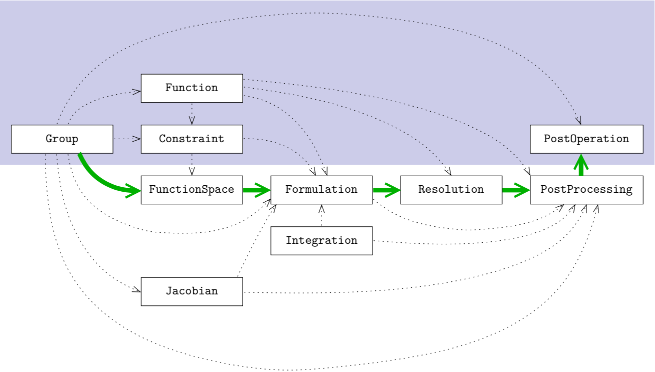
with

$$\boldsymbol{A}_{k,\ell} = \left(a\frac{d\phi_\ell}{dx}, \frac{d\phi_k}{dx} \right) + (b\phi_\ell, \phi_k)\,, \quad \boldsymbol{x}_\ell = \gamma_\ell, \quad \text{and} \quad \boldsymbol{b}_k = (f, \phi_k).$$

In Life-HTS, a problem is described by specifying the function space and the weak form equations

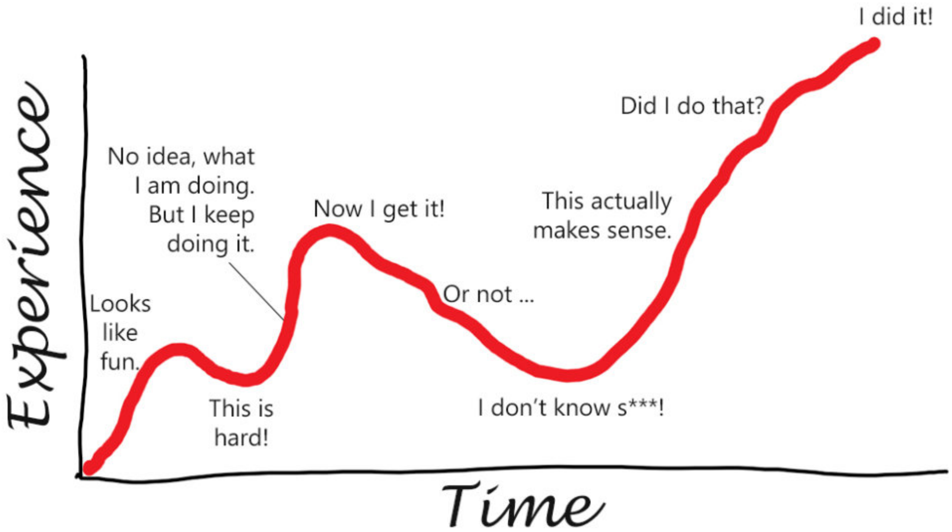
Finite Element Method with Life-HTS

- In practice, a text script (`.pro` file) contains the GetDP problem definition structure
- A finite element mesh is required as input, built by Gmsh from a geometrical description (script or CAD file)



See https://onelab.info/slides/onelab.pdf for details

Learning curve



www.theexcitedwriter.com

Finite Element Formulations for High-Temperature Superconductors

With technical details related to the Life-HTS implementation

Simple finite element formulations
The a-v-formulation
The h-ϕ-formulation

Resolution techniques
Time integration
Linearization methods
Comparison of the formulations

Mixed finite element formulations
The h(-ϕ)-a-formulation
The t-a-formulation

Illustrations

Summary

References

Introduction

Objective: Present and analyze various finite element formulations for modelling HTS and their implementation in Life-HTS. We will follow the GetDP philosophy:

- we will focus on building the weak form,
- and exploit the flexible function space possibilities, specifically for global variables.

$\Rightarrow$ We will cover some technical details.

Introduction

Objective: Present and analyze various finite element formulations for modelling HTS and their implementation in Life-HTS. We will follow the GetDP philosophy:

- we will focus on building the weak form,
- and exploit the flexible function space possibilities, specifically for global variables.

⇒ We will cover some technical details.

Important remark: One does not have to deal with these details for running existing templates.

Details are however fundamental for investigating new models and/or understanding the code.

General framework: magneto-quasistatics

- We aim to solve Maxwell's equations in the magneto-quasistatic ("magnetodynamic") approximation

$$\boxed{\textbf{curl}\ \boldsymbol{h} = \boldsymbol{j}, \qquad \textbf{curl}\ \boldsymbol{e} = -\partial_t \boldsymbol{b}, \qquad \text{div}\ \boldsymbol{b} = 0,}$$

  with
  - $\boldsymbol{h}$ the magnetic field (A/m),
  - $\boldsymbol{j}$ the current density (A/m^2),
  - $\boldsymbol{e}$ the electric field (V/m), and
  - $\boldsymbol{b}$ the magnetic flux density (T),

  while the displacement current $\partial_t \boldsymbol{d}$ is neglected

General framework: magneto-quasistatics

- We aim to solve Maxwell's equations in the magneto-quasistatic ("magnetodynamic") approximation

$$\boxed{\textbf{curl}\ \boldsymbol{h} = \boldsymbol{j}, \qquad \textbf{curl}\ \boldsymbol{e} = -\partial_t \boldsymbol{b}, \qquad \text{div}\ \boldsymbol{b} = 0,}$$

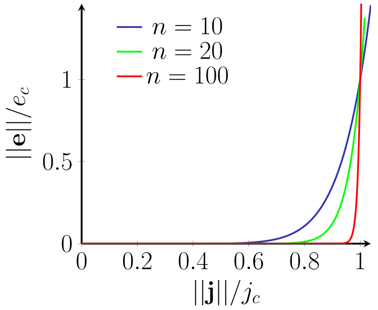
  with
  - $\boldsymbol{h}$ the magnetic field (A/m),
  - $\boldsymbol{j}$ the current density (A/m^2),
  - $\boldsymbol{e}$ the electric field (V/m), and
  - $\boldsymbol{b}$ the magnetic flux density (T),

  while the displacement current $\partial_t \boldsymbol{d}$ is neglected
- Boundary conditions and constitutive laws relating $\boldsymbol{b}$ to $\boldsymbol{h}$ and $\boldsymbol{e}$ to $\boldsymbol{j}$ are needed to obtain a well-posed problem

Constitutive laws

1. High-temperature superconductors (HTS):

$$\boldsymbol{e} = \rho(\|\boldsymbol{j}\|)\,\boldsymbol{j} \quad \text{and} \quad \boldsymbol{b} = \mu_0\,\boldsymbol{h},$$



where the electrical resistivity is given as

$$\rho(\|\boldsymbol{j}\|) = \frac{e_c}{j_c}\left(\frac{\|\boldsymbol{j}\|}{j_c}\right)^{n-1},$$

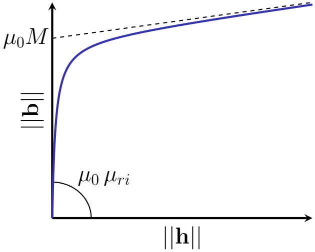
with $e_c = 10^{-4}$ V/m,
j_c, the critical current density,
$n \in [10, 1000]$

[C.J.G. Plummer and J. E. Evetts, IEEE TAS **23** (1987) 1179]
[E. Zeldov et al., Appl. Phys. Lett. **56** (1990) 680]

Constitutive laws

2. Ferromagnetic materials (FM):

$$\boxed{\boldsymbol{b} = \mu(\boldsymbol{h})\,\boldsymbol{h} \quad \text{and} \quad \boldsymbol{j} = \mathbf{0}.}$$



Typical values (supra50):

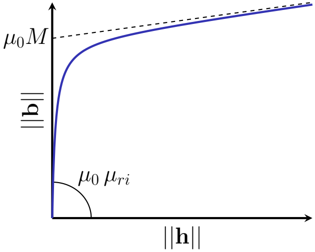
- initial relative permeability $\mu_{ri} = 1700$
- saturation magnetization $\mu_0 M = 1.3$ T

Eddy currents are neglected

Constitutive laws

2. Ferromagnetic materials (FM):

$$\boxed{\boldsymbol{b} = \mu(\boldsymbol{h})\,\boldsymbol{h} \quad \text{and} \quad \boldsymbol{j} = \boldsymbol{0}.}$$



Typical values (supra50):

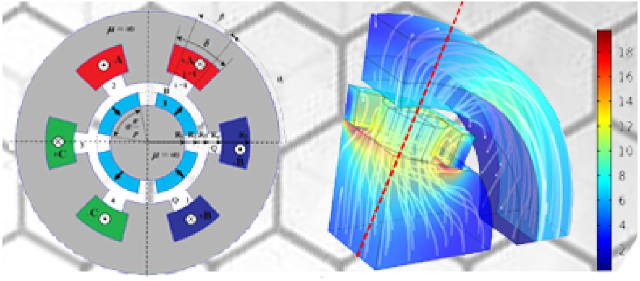
- initial relative permeability $\mu_{ri} = 1700$
- saturation magnetization
  $\mu_0 M = 1.3$ T

Eddy currents are neglected

3. Air:

$$\boxed{\boldsymbol{b} = \mu_0\,\boldsymbol{h} \quad \text{and} \quad \boldsymbol{j} = \boldsymbol{0}.}$$

Constitutive laws, extensions



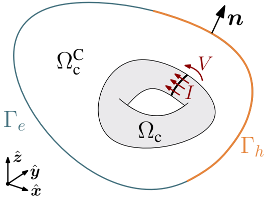
One can also consider

- normal conductors and coils,
- permanent magnets,
- ferromagnetic materials with hysteresis (e.g. [K. Jacques, thesis (2018)])
- type-I superconductors (need a London length)

Boundary conditions and global variables

Domain Ω decomposed into:

- Ω_c, the conducting domain ($\Omega_c = \cup_{i=1}^{N} \Omega_{c_i}$),
- Ω_c^C, the complementary non-conducting domain.



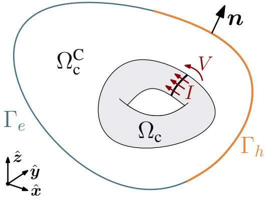
Boundary conditions:

1. Local conditions. On domain boundary $\partial\Omega = \Gamma$:
   - $\boldsymbol{h} \times \boldsymbol{n} = \bar{\boldsymbol{h}} \times \boldsymbol{n}$, imposed on Γ_h,
   - $\boldsymbol{e} \times \boldsymbol{n} = \bar{\boldsymbol{e}} \times \boldsymbol{n}$ (or $\boldsymbol{b} \cdot \boldsymbol{n} = \bar{\boldsymbol{b}} \cdot \boldsymbol{n}$), imposed on Γ_e ($= \Gamma \backslash \Gamma_h$).

Boundary conditions and global variables

Domain Ω decomposed into:

- Ω_c, the conducting domain ($\Omega_c = \cup_{i=1}^{N} \Omega_{c_i}$),
- Ω_c^C, the complementary non-conducting domain.



Boundary conditions:

1. Local conditions. On domain boundary $\partial\Omega = \Gamma$:
   - $\boldsymbol{h} \times \boldsymbol{n} = \bar{\boldsymbol{h}} \times \boldsymbol{n}$, imposed on Γ_h,
   - $\boldsymbol{e} \times \boldsymbol{n} = \bar{\boldsymbol{e}} \times \boldsymbol{n}$ (or $\boldsymbol{b} \cdot \boldsymbol{n} = \bar{\boldsymbol{b}} \cdot \boldsymbol{n}$), imposed on Γ_e $(= \Gamma \backslash \Gamma_h)$.
2. Global conditions. Either the applied current I_i, or voltage V_i is imposed (or a relation between them, not covered here) on each separate conducting region Ω_{c_i},
   - $I_i = \bar{I}_i$, imposed for $i \in C_I$, a subset of $C = \{1, \ldots, N\}$,
   - $V_i = \bar{V}_i$, imposed for $i \in C_V$, the complementary subset.

Summary

- Equations in Ω:

$$\text{div}\, \boldsymbol{b} = 0, \quad \textbf{curl}\, \boldsymbol{h} = \boldsymbol{j}, \quad \textbf{curl}\, \boldsymbol{e} = -\partial_t \boldsymbol{b}.$$

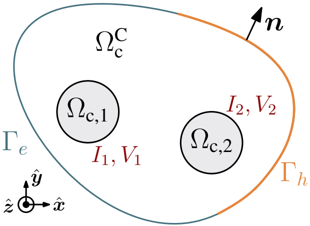
- Constitutive laws:

$$\boldsymbol{e} = \rho \boldsymbol{j}, \quad \boldsymbol{b} = \mu \boldsymbol{h}.$$

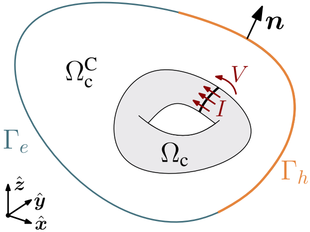
- Boundary conditions:

$$(\boldsymbol{h} - \bar{\boldsymbol{h}}) \times \boldsymbol{n}|_{\Gamma_h} = \boldsymbol{0}, \quad (\boldsymbol{e} - \bar{\boldsymbol{e}}) \times \boldsymbol{n}|_{\Gamma_e} = \boldsymbol{0},$$
$$I_i = \bar{I}_i \text{ for } i \in C_I, \quad V_i = \bar{V}_i \text{ for } i \in C_V.$$

2D:



3D:



Finite element formulations

Two classes of formulations:

- h-conform, e.g. h-ϕ-formulation,
  - enforces the continuity of the tangential component of $\boldsymbol{h}$,
  - involves $\boldsymbol{e} = \rho\, \boldsymbol{j}$ and $\boldsymbol{b} = \mu \boldsymbol{h}$,
  - much used for HTS modelling.

Finite element formulations

Two classes of formulations:

- h-conform, e.g. h-ϕ-formulation,
  - enforces the continuity of the tangential component of $\boldsymbol{h}$,
  - involves $\boldsymbol{e} = \rho\, \boldsymbol{j}$ and $\boldsymbol{b} = \mu \boldsymbol{h}$,
  - much used for HTS modelling.
- b-conform, e.g. a-v-formulation,
  - enforces the continuity of the normal component of $\boldsymbol{b}$,
  - involves $\boldsymbol{j} = \sigma \boldsymbol{e}$ and $\boldsymbol{h} = \nu \boldsymbol{b}$, $(\sigma = \rho^{-1}, \nu = \mu^{-1})$
  - much used in electric rotating machine design.

Nonlinear constitutive laws involved in opposite ways $\Rightarrow$ very different numerical behaviors are expected… and observed.

Differential forms

We discretize the fields as differential k-forms. The exterior derivative d applied on a k-form gives a $k+1$-form:

- 0-form, H^1, e.g. ϕ (scalar magnetic potential), v (scalar electric potential):
  - continuous scalar fields (conform),
  - generated by **nodal** functions ψ_n, value (point evaluation) at node $\tilde{n} = \delta_{n\tilde{n}}$,
  - exterior derivative is **grad** .

Differential forms

We discretize the fields as differential k-forms. The exterior derivative d applied on a k-form gives a $k+1$-form:

- 0-form, H^1, e.g. ϕ (scalar magnetic potential), v (scalar electric potential):
  - continuous scalar fields (conform),
  - generated by **nodal** functions ψ_n, value (point evaluation) at node $\tilde{n} = \delta_{n\tilde{n}}$,
  - exterior derivative is **grad** .
- 1-form, $H(\mathbf{curl})$, e.g. $\boldsymbol{h}$, $\boldsymbol{e}$, $\boldsymbol{a}$ (magnetic vector potential), $\boldsymbol{t}$ (electric vector potential):
  - vector fields with continuous tangential trace (curl-conform),
  - generated by **edge** functions $\boldsymbol{\psi}_e$, circulation (line integral) along edge $\tilde{e}$ $= \delta_{e\tilde{e}}$,
  - exterior derivative is **curl** .

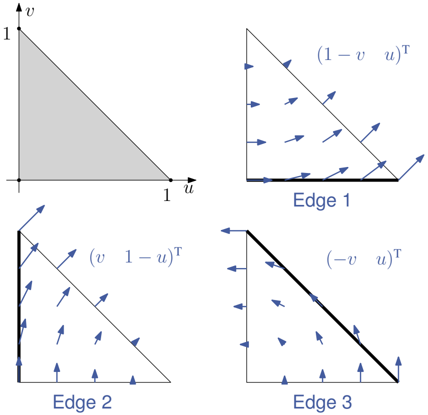
Differential forms

We discretize the fields as differential k-forms. The exterior derivative d applied on a k-form gives a $k+1$-form:

- 0-form, H^1, e.g. ϕ (scalar magnetic potential), v (scalar electric potential):
  - continuous scalar fields (conform),
  - generated by **nodal** functions ψ_n, value (point evaluation) at node $\tilde{n} = \delta_{n\tilde{n}}$,
  - exterior derivative is **grad** .
- 1-form, $H(\textbf{curl})$, e.g. $\boldsymbol{h}$, $\boldsymbol{e}$, $\boldsymbol{a}$ (magnetic vector potential), $\boldsymbol{t}$ (electric vector potential):
  - vector fields with continuous tangential trace (curl-conform),
  - generated by **edge** functions $\boldsymbol{\psi}_e$, circulation (line integral) along edge $\tilde{e}$ $= \delta_{e\tilde{e}}$,
  - exterior derivative is **curl** .
- 2-form, $H(\textbf{div})$, e.g. $\boldsymbol{b}$, $\boldsymbol{j}$:
  - vector fields with continuous normal trace (div-conform),
  - generated by **facet** functions $\boldsymbol{\psi}_f$, flux (surface integral) through facet $\tilde{f}$ $= \delta_{f\tilde{f}}$,
  - exterior derivative is div .

Differential forms: illustration

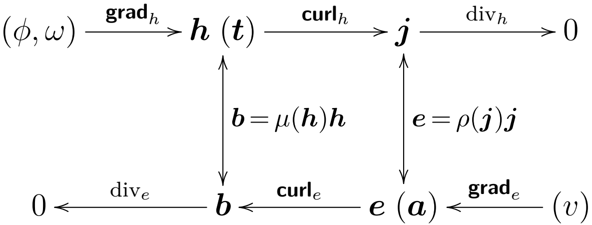
Lowest order edge functions (1-form) for a triangular finite element:



Their **curl** (2-form) are constant.

Differential forms: Tonti diagram

- We can summarize it all on a Tonti diagram:



Differential forms: Tonti diagram

- We can summarize it all on a Tonti diagram:

$$
\begin{array}{ccccccc}
(\phi,\omega) & \xrightarrow{\textbf{grad}_h} & \boldsymbol{h}\ (\boldsymbol{t}) & \xrightarrow{\textbf{curl}_h} & \boldsymbol{j} & \xrightarrow{\mathrm{div}_h} & 0 \\
 & & \updownarrow \ \boldsymbol{b} = \mu(\boldsymbol{h})\boldsymbol{h} & & \updownarrow \ \boldsymbol{e} = \rho(\boldsymbol{j})\boldsymbol{j} & & \\
0 & \xleftarrow{\mathrm{div}_e} & \boldsymbol{b} & \xleftarrow{\textbf{curl}_e} & \boldsymbol{e}\ (\boldsymbol{a}) & \xleftarrow{\textbf{grad}_e} & (v)
\end{array}
$$

- $\boldsymbol{h}$-conform formulations ($\boldsymbol{h}$, $\boldsymbol{h}$-ϕ, $\boldsymbol{t}$-ω, ...) satisfy the top exactly
- $\boldsymbol{b}$-conform formulations ($\boldsymbol{a}$, $\boldsymbol{a}$-v, ...) satisfy the bottom exactly

Simple finite element formulations
The a-v-formulation
The h-ϕ-formulation

Resolution techniques
Time integration
Linearization methods
Comparison of the formulations

Mixed finite element formulations
The h(-ϕ)-a-formulation
The t-a-formulation

Illustrations

Summary

References

Derivation of the a-v-formulation

Introduce the vector potential $\boldsymbol{a}$, and the electric potential v:

$$\boldsymbol{b} = \textbf{curl}\ \boldsymbol{a}, \qquad \boldsymbol{e} = -\partial_t \boldsymbol{a} - \textbf{grad}\ v.$$

Derivation of the a-v-formulation

Introduce the vector potential $\boldsymbol{a}$, and the electric potential v:

$$\boldsymbol{b} = \textbf{curl}\ \boldsymbol{a}, \qquad \boldsymbol{e} = -\partial_t \boldsymbol{a} - \textbf{grad}\ v.$$

Define $\boldsymbol{a}$ in Ω and v in Ω_c (discontinuous across electrodes):

- $\boldsymbol{a}$ as a 1-form and v as a 0-form,
- satisfying the local BC $(\boldsymbol{e} - \bar{\boldsymbol{e}}) \times \boldsymbol{n}|_{\Gamma_e} = \boldsymbol{0}$,
- and global BC $V_i = \bar{V}_i$ for $i \in C_V$ (i.e. the circulation of $-\textbf{grad}\ v$ around conducting domain Ω_{c_i} is equal to $\bar{V}_i$).

This strongly satisfies

$$\text{div}\ \boldsymbol{b} = 0, \quad \textbf{curl}\ \boldsymbol{e} = -\partial_t \boldsymbol{b}, \quad (\boldsymbol{e} - \bar{\boldsymbol{e}}) \times \boldsymbol{n}|_{\Gamma_e} = \boldsymbol{0}, \quad V_i = \bar{V}_i \text{ for } i \in C_V.$$

Derivation of the a-v-formulation

Introduce the vector potential $\boldsymbol{a}$, and the electric potential v:

$$\boldsymbol{b} = \textbf{curl}\ \boldsymbol{a}, \qquad \boldsymbol{e} = -\partial_t \boldsymbol{a} - \textbf{grad}\ v.$$

Define $\boldsymbol{a}$ in Ω and v in Ω_c (discontinuous across electrodes):

- $\boldsymbol{a}$ as a 1-form and v as a 0-form,
- satisfying the local BC $(\boldsymbol{e} - \bar{\boldsymbol{e}}) \times \boldsymbol{n}|_{\Gamma_e} = \boldsymbol{0}$,
- and global BC $V_i = \bar{V}_i$ for $i \in C_V$ (i.e. the circulation of $-\textbf{grad}\ v$ around conducting domain Ω_{c_i} is equal to $\bar{V}_i$).

This strongly satisfies

$$\text{div}\ \boldsymbol{b} = 0, \quad \textbf{curl}\ \boldsymbol{e} = -\partial_t \boldsymbol{b}, \quad (\boldsymbol{e} - \bar{\boldsymbol{e}}) \times \boldsymbol{n}|_{\Gamma_e} = \boldsymbol{0}, \quad V_i = \bar{V}_i \text{ for } i \in C_V.$$

What remains is:

$$\textbf{curl}\ \boldsymbol{h} = \boldsymbol{j}, \quad \boldsymbol{j} = \sigma \boldsymbol{e}, \quad \boldsymbol{h} = \nu \boldsymbol{b}, \quad (\boldsymbol{h} - \bar{\boldsymbol{h}}) \times \boldsymbol{n}|_{\Gamma_h} = \boldsymbol{0}, \quad I_i = \bar{I}_i \text{ for } i \in C_I.$$

Choosing $\boldsymbol{a}$ and v

We still have freedom on the choice of $\boldsymbol{a}$ and v. Indeed, for any scalar field ϕ, the substitution

$$\boldsymbol{a} \to \boldsymbol{a} + \int_0^t \textbf{grad}\,\phi\, dt$$
$$v \to v - \phi$$

lets the physical solution, $\boldsymbol{b}$ and $\boldsymbol{e}$, unchanged.

We present here one possibility for gauging $\boldsymbol{a}$ and v in:

(1) 2D case with in-plane $\boldsymbol{b}$, (2) 3D case.

Choosing $\boldsymbol{a}$ and v

We still have freedom on the choice of $\boldsymbol{a}$ and v. Indeed, for any scalar field ϕ, the substitution

$$\boldsymbol{a} \to \boldsymbol{a} + \int_0^t \textbf{grad}\ \phi\ dt$$
$$v \to v - \phi$$

lets the physical solution, $\boldsymbol{b}$ and $\boldsymbol{e}$, unchanged.

We present here one possibility for gauging $\boldsymbol{a}$ and v in:

(1) 2D case with in-plane $\boldsymbol{b}$, (2) 3D case.

In both cases, **one** global shape function $v_{d,i}$ in each Ω_{c_i} is sufficient for representing a unit voltage in Ω_{c_i}, s.t. we have:

$$\textbf{grad}\ v = \sum_{i=1}^{N} V_i\ \textbf{grad}\ v_{d,i}.$$

Choosing $\boldsymbol{a}$ and v, cont'd

$$\boldsymbol{b} = \textbf{curl}\ \boldsymbol{a}, \qquad \boldsymbol{e} = -\partial_t \boldsymbol{a} - \textbf{grad}\ v, \qquad \textbf{grad}\ v = \sum_{i=1}^{N} V_i\, \textbf{grad}\ v_{d,i}$$

1. 2D with in-plane $\boldsymbol{b}$:

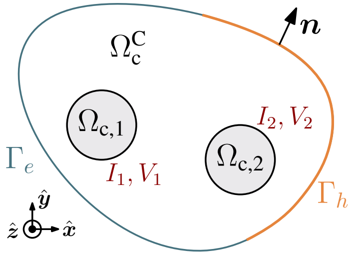
- We choose $\boldsymbol{a}$ along $\hat{\boldsymbol{z}}$,

$$\boldsymbol{a} = \sum_{n \in \Omega} a_n\ \psi_n \hat{\boldsymbol{z}},$$

  with ψ_n the nodal function of node n.
  NB: It is a Coulomb gauge, as div $\boldsymbol{a} = 0$

- $\textbf{grad}\ v_{d,i}$ is along $\hat{\boldsymbol{z}}$ and constant $(= 1)$ in each Ω_{c_i}. (V is a voltage per unit length.)

- Remaining constant fixed by BC.



Life-HTS a in 2D, with in-plane b

$$\boldsymbol{a} = \sum_{n \in \Omega} a_n \, \psi_n \hat{\boldsymbol{z}},$$

```
FunctionSpace {
  // Perpendicular edge functions (1-form field in the out-of-plane direction)
  { Name a_space_2D; Type Form1P;
    BasisFunction {
      { Name psin; NameOfCoef an; Function BF_PerpendicularEdge;
        Support Omega_a_AndBnd; Entity NodesOf[All]; }
    }
    Constraint {
      { NameOfCoef an; EntityType NodesOf; NameOfConstraint a; }
    }
  }
}
```

Life-HTS grad v in 2D, with in-plane b

$$\textbf{grad}\ v = \sum_{i=1}^{N} V_i\, \textbf{grad}\ v_{d,i} = \sum_{i=1}^{N} V_i\, \hat{\boldsymbol{z}}_i$$

```
FunctionSpace {
  { Name grad_v_space_2D; Type Form1P;
    BasisFunction {
      // Constant per region and along z. Corresponds to a voltage per unit length
      { Name zi; NameOfCoef Vi; Function BF_RegionZ;
        Support Region[OmegaC]; Entity Region[OmegaC]; }
    }
    GlobalQuantity {
      // Associated global quantities to be used in the formulation
      { Name V; Type AliasOf; NameOfCoef Vi; }
      { Name I; Type AssociatedWith; NameOfCoef Vi; }
    }
    Constraint {
      { NameOfCoef V; EntityType Region; NameOfConstraint Voltage; }
      { NameOfCoef I; EntityType Region; NameOfConstraint Current; }
    }
  }
}
```

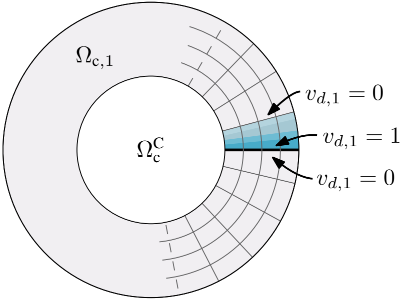
Choosing $\boldsymbol{a}$ and v

2. 3D:

- In Ω_c, define $v_{d,i}$ to be zero everywhere except on a transition layer in Ω_{c_i}: layer of one element, on one side of the electrodes, in each Ω_{c_i} (v has no longer a physical interpretation),

$$\textbf{grad } v = \sum_{i=1}^{N} V_i \textbf{ grad } v_{d,i}.$$

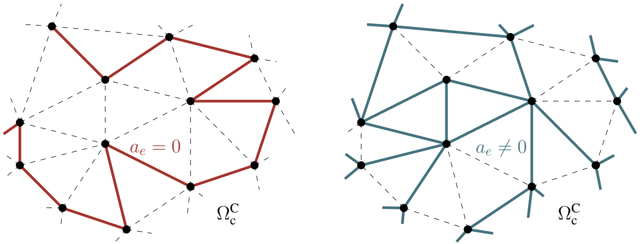
- $\boldsymbol{a}$ is generated by edge functions.
- In Ω_c, $\boldsymbol{a}$ is unique, e.g. outside the transition layer $\boldsymbol{e} = -\partial_t \boldsymbol{a}$ (reduced vector potential).
- In $\Omega_\mathrm{c}^\mathrm{C}$, $\boldsymbol{a}$ is made unique with a co-tree gauge. . .



Co-tree gauge for $\boldsymbol{a}$ in $\Omega_{\mathrm{c}}^{\mathrm{C}}$ in 3D

- In $\Omega_{\mathrm{c}}^{\mathrm{C}}$, only **curl** $\boldsymbol{a} = \boldsymbol{b}$ has a physical meaning. One degree of freedom (DoF) per facet is sufficient (and necessary), instead of one DoF per edge.
- The support entities of the 1-form $\boldsymbol{a}$ are the edges.
- To associate a unique edge to each facet: consider only edges in a **co-tree**, i.e. the complementary of a tree:

$$\boldsymbol{a} = \sum_{e \in \Omega_{\mathrm{c}} \cup (\textbf{co-tree} \text{ in } \Omega_{\mathrm{c}}^{\mathrm{C}})} a_e \, \boldsymbol{\psi}_e .$$



NB: Be careful on the conducting domain boundary $\partial\Omega_{\mathrm{c}}$, no gauge there because $\boldsymbol{a}$ is already unique.

Life-HTS a in 3D

$$\boldsymbol{a} = \sum_{e \in \Omega_c \cup (\text{co-tree in } \Omega_c^C)} a_e \, \boldsymbol{\psi}_e$$

```
FunctionSpace {
  { Name a_space_3D; Type Form1;
    BasisFunction {
      // Usual edge functions everywhere (decomposed to handle BndOmegaC) correctly
      { Name psie ; NameOfCoef ae ; Function BF_Edge ;
        Support Omega_a_AndBnd ; Entity EdgesOf[ All, Not BndOmegaC ] ; }
      { Name psie2 ; NameOfCoef ae2 ; Function BF_Edge ;
        Support Omega_a_AndBnd ; Entity EdgesOf[ BndOmegaC ] ; }
    }
    Constraint {
      { NameOfCoef ae; EntityType EdgesOf; NameOfConstraint a; }
      { NameOfCoef ae2; EntityType EdgesOf; NameOfConstraint a; }
      { NameOfCoef ae; EntityType EdgesOfTreeIn; EntitySubType StartingOn;
        NameOfConstraint GaugeCondition; }
    }
  }
}
Constraint {
  { Name GaugeCondition ; Type Assign ;
    Case {
      // Zero on edges of a tree in Omega_CC, containing a complete tree on Surf_a_noGauge
      {Region Omega_a_OmegaCC ; SubRegion Surf_a_noGauge; Value 0.; }
    }
  }
}
```

Life-HTS v in 3D

$$\textbf{grad}\ v = \sum_{i=1}^{N} V_i\ \textbf{grad}\ v_{d,i}$$

```
FunctionSpace{
  { Name grad_v_space_3D; Type Form1;
    BasisFunction {
      // Global unit voltage shape function. Support limited to only one side of the electrodes
      { Name vi; NameOfCoef Vi; Function BF_GradGroupOfNodes;
        Support ElementsOf[OmegaC, OnPositiveSideOf Electrodes];
        Entity GroupsOfNodesOf[Electrodes]; }
    }
    GlobalQuantity {
      // Associated global quantities to be used in the formulation.
      { Name V; Type AliasOf; NameOfCoef Vi; }
      { Name I; Type AssociatedWith; NameOfCoef Vi; }
    }
    Constraint {
      { NameOfCoef V;
        EntityType GroupsOfNodesOf; NameOfConstraint Voltage; }
      { NameOfCoef I;
        EntityType GroupsOfNodesOf; NameOfConstraint Current; }
    }
  }
}
```

Choosing a and v, other possibilities

Various alternatives can also be considered in 3D:

- Distributed support for v, via a preliminary FE resolution [S. Schöps, et al., COMPEL (2013)]
- Coulomb gauge in $\Omega_{\mathrm{c}}^{\mathrm{C}}$ via a Lagrange multiplier [Creusé, et al., Computers & Mathematics with Applications, 77(6), 1563-1582 (2019)]

Derivation of the a-v-formulation, cont'd

What remains is:

$$\underbrace{\textbf{curl}\ \boldsymbol{h} = \boldsymbol{j}, \quad \boldsymbol{j} = \sigma \boldsymbol{e}, \quad \overbrace{\boldsymbol{h} = \nu \boldsymbol{b}}_{}}_{\Rightarrow\ \textbf{curl}\ (\nu\,\textbf{curl}\ \boldsymbol{a}) = -\sigma\,(\partial_t \boldsymbol{a} + \textbf{grad}\ v)\ \textcircled{\star}} \overbrace{, \quad (\boldsymbol{h} - \bar{\boldsymbol{h}}) \times \boldsymbol{n}|_{\Gamma_h} = \boldsymbol{0}}^{\textcircled{\diamond}}, \quad \underbrace{I_i = \bar{I}_i \text{ for } i \in C_I}_{\textcircled{\ddagger}}$$

Derivation of the a-v-formulation, cont'd

What remains is:

$$\underbrace{\textbf{curl}\, \boldsymbol{h} = \boldsymbol{j}, \quad \boldsymbol{j} = \sigma \boldsymbol{e}, \quad \boldsymbol{h} = \nu \boldsymbol{b}}_{\Rightarrow\ \textbf{curl}\,(\nu\,\textbf{curl}\,\boldsymbol{a}) = -\sigma\,(\partial_t \boldsymbol{a} + \textbf{grad}\, v)\ \circledast}, \quad \overbrace{(\boldsymbol{h} - \bar{\boldsymbol{h}}) \times \boldsymbol{n}|_{\Gamma_h} = \boldsymbol{0}}^{\diamond}, \quad \underbrace{I_i = \bar{I}_i \text{ for } i \in C_I}_{\ddagger}$$

- Multiply $\circledast$ by a test function $\boldsymbol{a}'$, in the same space than $\boldsymbol{a}$ but with homogeneous BC, and integrate over Ω,

$$\left(\textbf{curl}\,(\nu\,\textbf{curl}\,\boldsymbol{a})\,,\boldsymbol{a}'\right)_\Omega + \left(\sigma\,(\partial_t \boldsymbol{a} + \textbf{grad}\, v)\,,\boldsymbol{a}'\right)_{\Omega_c} = 0$$

$$\Rightarrow \quad (\nu\,\textbf{curl}\,\boldsymbol{a}\,,\textbf{curl}\,\boldsymbol{a}')_\Omega - (\underbrace{\nu\,\textbf{curl}\,\boldsymbol{a} \times \boldsymbol{n}}_{\text{Neumann BC } \diamond}\,,\ \boldsymbol{a}')_{\Gamma_h}$$

$$+ (\sigma\,\partial_t \boldsymbol{a}\,,\boldsymbol{a}')_{\Omega_c} + (\sigma\,\textbf{grad}\, v\,,\boldsymbol{a}')_{\Omega_c} = 0$$

Derivation of the a-v-formulation, cont'd

What remains is:

$$\underbrace{\textbf{curl}\, \boldsymbol{h} = \boldsymbol{j}, \quad \boldsymbol{j} = \sigma \boldsymbol{e}, \quad \boldsymbol{h} = \nu \boldsymbol{b}}_{\Rightarrow\ \textbf{curl}\,(\nu\, \textbf{curl}\, \boldsymbol{a}) = -\sigma\,(\partial_t \boldsymbol{a} + \textbf{grad}\, v)\ \circledast}, \quad \overbrace{\boldsymbol{h} = \nu \boldsymbol{b}, \quad (\boldsymbol{h} - \bar{\boldsymbol{h}}) \times \boldsymbol{n}|_{\Gamma_h} = \boldsymbol{0}}^{\diamond}, \quad \underbrace{I_i = \bar{I}_i \text{ for } i \in C_I}_{\ddagger}$$

- Multiply $\circledast$ by a test function $\textbf{grad}\, v'$, and integrate over Ω_c,

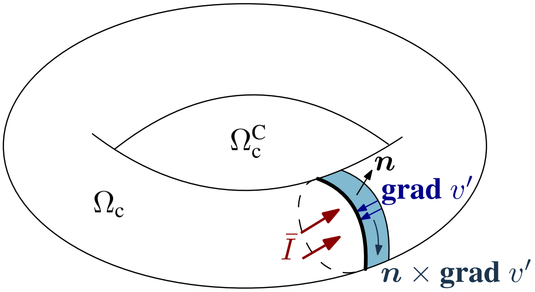
$$(\textbf{curl}\,(\nu\, \textbf{curl}\, \boldsymbol{a}), \textbf{grad}\, v')_{\Omega_c} + (\sigma\, \partial_t \boldsymbol{a}, \textbf{grad}\, v')_{\Omega_c} + (\sigma\, \textbf{grad}\, v, \textbf{grad}\, v')_{\Omega_c} = 0$$

$$\Rightarrow \quad - \underbrace{(\nu\, \textbf{curl}\, \boldsymbol{a} \times \boldsymbol{n}, \textbf{grad}\, v')_{\partial\Omega_c}}_{\ddagger \dots} + (\sigma\, \partial_t \boldsymbol{a}, \textbf{grad}\, v')_{\Omega_c} + (\sigma\, \textbf{grad}\, v, \textbf{grad}\, v')_{\Omega_c} = 0$$

Derivation of the a-v-formulation, cont'd

- The surface term simplifies

$$
\begin{aligned}
(\nu\, \mathbf{curl}\, \boldsymbol{a} \times \boldsymbol{n}\ , \mathbf{grad}\ v')_{\partial\Omega_c} &= (\boldsymbol{h} \times \boldsymbol{n}\ , \mathbf{grad}\ v')_{\partial\Omega_c} \\
&= (\boldsymbol{h}\ , \boldsymbol{n} \times \mathbf{grad}\ v')_{\partial\Omega_c} \\
&= (\boldsymbol{h}\ , \boldsymbol{n} \times \mathbf{grad}\ v')_{\partial(\text{transition layer})} \\
&= I\ V' = \bar{I}\ V' \qquad (\text{Ampère's law} + ⦶).
\end{aligned}
$$



a-v-formulation

Finally, the a-v-formulation amounts to find $\boldsymbol{a}$ and v in the chosen function spaces such that, $\forall \boldsymbol{a}'$ and v',

$$\begin{aligned}(\nu \,\textbf{curl}\, \boldsymbol{a} \,, \textbf{curl}\, \boldsymbol{a}')_{\Omega} - \left(\bar{\boldsymbol{h}} \times \boldsymbol{n}_{\Omega} \,, \boldsymbol{a}'\right)_{\Gamma_h} \\ + (\sigma\, \partial_t \boldsymbol{a} \,, \boldsymbol{a}')_{\Omega_c} + (\sigma \,\textbf{grad}\, v \,, \boldsymbol{a}')_{\Omega_c} = 0,\end{aligned}$$

$$(\sigma\, \partial_t \boldsymbol{a} \,, \textbf{grad}\, v')_{\Omega_c} + (\sigma \,\textbf{grad}\, v \,, \textbf{grad}\, v')_{\Omega_c} = \sum_{i=1}^{N} I_i \mathcal{V}_i(v'),$$

with $I_i = \bar{I}_i$ for $i \in C_I$, and $\mathcal{V}_i(v') = V_i'$ (i.e. the DoF associated with the unit voltage function $v_{d,i}$).

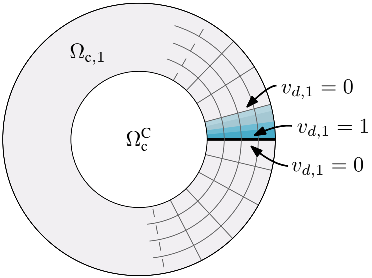
a-v-formulation – Interpretation

When the test function $v' = v_{d,i}$ is chosen ($\mathcal{V}_i(v_{d,i}) = 1$), the second equation reads

$$\left(\sigma\left(\partial_t \boldsymbol{a} + \textbf{grad}\ v\right), \textbf{grad}\ v_{d,i}\right)_{\Omega_c} = I_i$$

$$\Rightarrow \quad \left(\sigma\, \boldsymbol{e}, -\textbf{grad}\ v_{d,i}\right)_{\Omega_c} = I_i.$$

"Flux of $\sigma \boldsymbol{e}$ $(= \boldsymbol{j})$ averaged over a transition layer = total current".



NB: The flux of $\sigma \boldsymbol{e}$ depends on the chosen cross-section as $\sigma \boldsymbol{e}$ is not a 2-form (as $\boldsymbol{j}$ should be). Conservation of current is weakly satisfied.

Simple finite element formulations
The a-v-formulation
The h-ϕ-formulation

Resolution techniques
Time integration
Linearization methods
Comparison of the formulations

Mixed finite element formulations
The $h(\text{-}\phi)$-a-formulation
The t-a-formulation

Illustrations

Summary

References

Derivation of the h-ϕ-formulation

Choose $\boldsymbol{h}$ such that

- it is a 1-form,
- $(\boldsymbol{h} - \bar{\boldsymbol{h}}) \times \boldsymbol{n}|_{\Gamma_h} = \boldsymbol{0}$,
- **curl** $\boldsymbol{h} = \boldsymbol{0}$ in Ω_c^C (this is the **key** point),
- and express $\boldsymbol{j}$ directly as $\boldsymbol{j} =$ **curl** $\boldsymbol{h}$ in Ω_c, with $\boldsymbol{h}$ generated by edge functions.

Derivation of the h-ϕ-formulation

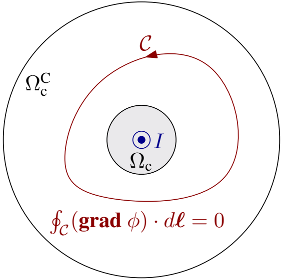
Choose $\boldsymbol{h}$ such that

- it is a 1-form,
- $(\boldsymbol{h} - \bar{\boldsymbol{h}}) \times \boldsymbol{n}|_{\Gamma_h} = \boldsymbol{0}$,
- **curl** $\boldsymbol{h} = \boldsymbol{0}$ in Ω_c^C (this is the **key** point),
- and express $\boldsymbol{j}$ directly as $\boldsymbol{j} =$ **curl** $\boldsymbol{h}$ in Ω_c, with $\boldsymbol{h}$ generated by edge functions.

What are the functions $\boldsymbol{h}$ that satisfy **curl** $\boldsymbol{h} = \boldsymbol{0}$ in Ω_c^C?

$\Rightarrow$ Surely gradients of scalar functions!

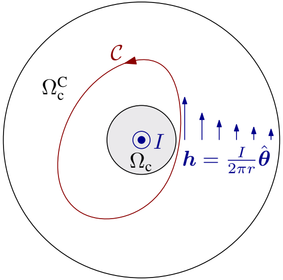
- If $\boldsymbol{h} =$ **grad** ϕ, then **curl** $\boldsymbol{h} = \boldsymbol{0}$, $\forall \phi$.
- However, choosing only $\boldsymbol{h} =$ **grad** ϕ does not allow to represent a net current intensity (necessary if Ω_c^C is multiply connected).
- We need additional functions...



Derivation of the h-ϕ-formulation, cont'd

- **One** global shape function $\boldsymbol{c}_i$ for each Ω_{c_i} is enough for representing a unit current intensity in Ω_{c_i}.
- As with the a-v-formulation, we have freedom on the choice of these functions. The only constraint is that

$$\oint_{\mathcal{C}_i} \boldsymbol{c}_j \cdot d\boldsymbol{\ell} = \delta_{ij}.$$



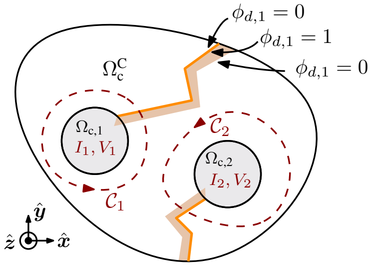
In Ω_c^C, we therefore have

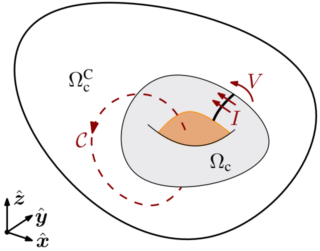
$$\boldsymbol{h} = \textbf{grad}\ \phi + \sum_{i=1}^{N} I_i\, \boldsymbol{c}_i.$$

Choice of the global functions

One possibility for choosing the $\boldsymbol{c}_i$ functions, the cut functions:

- Introduce cuts to make Ω_c^C simply connected.
- Define the $\boldsymbol{c}_i$ on transition layers: layer of one element on one side of the cut, for each cut.
- $\boldsymbol{c}_i = \textbf{grad}\ \phi_{d,i}$, with $\phi_{d,i}$ a discontinuous scalar potential.

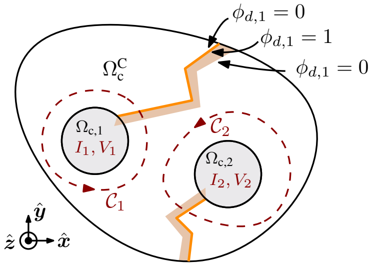


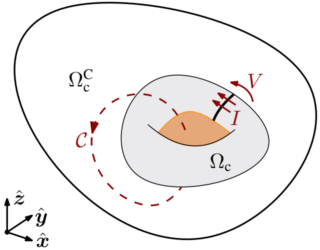


Choice of the global functions

One possibility for choosing the $\boldsymbol{c}_i$ functions, the cut functions:

- Introduce cuts to make Ω_c^C simply connected.
- Define the $\boldsymbol{c}_i$ on transition layers: layer of one element on one side of the cut, for each cut.
- $\boldsymbol{c}_i = \textbf{grad}\ \phi_{d,i}$, with $\phi_{d,i}$ a discontinuous scalar potential.





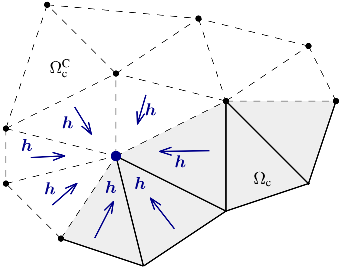
NB: Gmsh has an automatic cohomology solver for generating cuts in complicated geometries (e.g. helix windings)
[M. Pellikka, et al. SIAM Journal on Scientific Computing 35(5), pp. 1195-1214 (2013)]

Summary and shape function supports

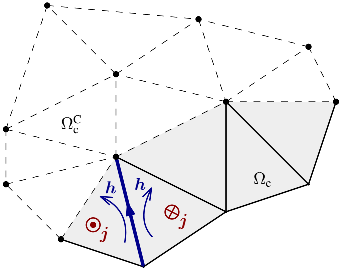
In Ω we have

$$\boldsymbol{h} = \sum_{n\in\Omega_c^C} \phi_n \,\textbf{grad}\, \psi_n + \sum_{e\in\Omega_c\setminus\partial\Omega_c} h_e \,\boldsymbol{\psi}_e + \sum_{i=1}^{N} I_i \,\boldsymbol{c}_i.$$

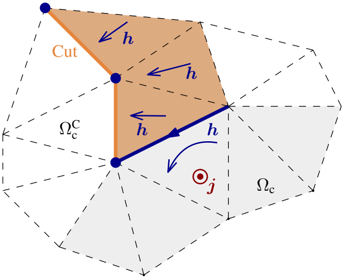
Gradient of node functions.



Classical edge functions.



Global cut function. Net current $\neq 0$.



Note: Gray areas $= \Omega_c$.

Life-HTS h in 2D or 3D

$$\boldsymbol{h} = \sum_{n\in\Omega_c^C} \phi_n \,\textbf{grad}\, \psi_n + \sum_{e\in\Omega_c\setminus\partial\Omega_c} h_e \,\boldsymbol{\psi}_e + \sum_{i=1}^{N} I_i \,\boldsymbol{c}_i.$$

```
FunctionSpace{
  { Name h_space; Type Form1;
    BasisFunction {
      // Nodal functions
      { Name gradpsin; NameOfCoef phin; Function BF_GradNode;
        Support Omega_h_OmegaCC_AndBnd; Entity NodesOf[OmegaCC]; }
      { Name gradpsin; NameOfCoef phin2; Function BF_GroupOfEdges;
        Support Omega_h_OmegaC; Entity GroupsOfEdgesOnNodesOf[BndOmegaC]; }
      // Edge functions
      { Name psie; NameOfCoef he; Function BF_Edge;
        Support Omega_h_OmegaC_AndBnd; Entity EdgesOf[All, Not BndOmegaC]; }
      // Cut functions
      { Name ci; NameOfCoef Ii; Function BF_GradGroupOfNodes;
        Support ElementsOf[Omega_h_OmegaCC, OnPositiveSideOf Cuts];
        Entity GroupsOfNodesOf[Cuts]; }
      { Name ci; NameOfCoef Ii2; Function BF_GroupOfEdges;
        Support Omega_h_OmegaC_AndBnd;
        Entity GroupsOfEdgesOf[Cuts, InSupport TransitionLayerAndBndOmegaC]; }
    }
    GlobalQuantity {
      { Name I ; Type AliasOf        ; NameOfCoef Ii ; }
      { Name V ; Type AssociatedWith ; NameOfCoef Ii ; }
    }
    Constraint {
      { [...] }
      { [...] }
} } }
```

Dealing with global variables, alternatives

Other possibilities can also be considered:

- Winding functions
  [S. Schöps, et al. COMPEL (2013)]
- Large resistivity ($\approx 1\ \Omega$m) in Ω_c^C and integral constraint on the current (simple but much more DoF), leading to a full h-formulation
  [Shen, B., et al., IEEE access, 8 (2020) 100403-100414]

Derivation of the h-ϕ-formulation, cont'd

With the chosen $\boldsymbol{h}$, we strongly satisfy

$$\textbf{curl}\ \boldsymbol{h} = \boldsymbol{j}, \quad (\boldsymbol{h} - \bar{\boldsymbol{h}}) \times \boldsymbol{n}|_{\Gamma_h} = \boldsymbol{0}, \quad I_i = \bar{I}_i \text{ for } i \in C_I.$$

Derivation of the h-ϕ-formulation, cont'd

With the chosen $\boldsymbol{h}$, we strongly satisfy

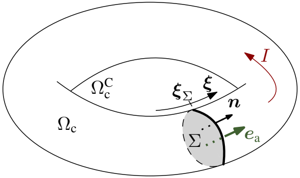
$$\textbf{curl}\ \boldsymbol{h} = \boldsymbol{j}, \quad (\boldsymbol{h} - \bar{\boldsymbol{h}}) \times \boldsymbol{n}|_{\Gamma_h} = \boldsymbol{0}, \quad I_i = \bar{I}_i \text{ for } i \in C_I.$$

What remains is:

$$\text{div}\ \boldsymbol{b} = 0, \quad \textbf{curl}\ \boldsymbol{e} = -\partial_t \boldsymbol{b}, \quad \boldsymbol{e} = \rho\, \boldsymbol{j}, \quad \boldsymbol{b} = \mu \boldsymbol{h},$$
$$(\boldsymbol{e} - \bar{\boldsymbol{e}}) \times \boldsymbol{n}|_{\Gamma_e} = \boldsymbol{0}, \quad V_i = \bar{V}_i \text{ for } i \in C_V.$$

We model an external applied voltage V by a localized $\boldsymbol{e}_\mathrm{a}$ field in a modified Ohm's law:

$$\boldsymbol{e} = \boldsymbol{e}_\mathrm{a} + \rho\, \boldsymbol{j},$$



with $\boldsymbol{e}_\mathrm{a} = V\delta(\boldsymbol{\xi} - \boldsymbol{\xi}_\Sigma)\boldsymbol{n}$ so that we globally have a net E.M.F. ($\delta(\cdot)$ is the Dirac distribution)

Derivation of the h-ϕ-formulation, cont'd

What remains is:

$$\overbrace{\mathbf{curl}\ \boldsymbol{e} = -\partial_t \boldsymbol{b}, \quad \boldsymbol{e} = \boldsymbol{e}_\mathrm{a} + \rho\, \boldsymbol{j}, \quad \boldsymbol{b} = \mu \boldsymbol{h}}^{\Rightarrow\ \mathbf{curl}\ (\rho\, \mathbf{curl}\ \boldsymbol{h}) + \mathbf{curl}\ \boldsymbol{e}_\mathrm{a} = -\partial_t(\mu \boldsymbol{h})\ \circledast}$$

$$\mathrm{div}\ \boldsymbol{b} = 0, \quad \mathbf{curl}\ \boldsymbol{e} = -\partial_t \boldsymbol{b}, \quad \boldsymbol{e} = \boldsymbol{e}_\mathrm{a} + \rho\, \boldsymbol{j}, \quad \boldsymbol{b} = \mu \boldsymbol{h},$$

$$\underbrace{(\boldsymbol{e} - \bar{\boldsymbol{e}}) \times \boldsymbol{n}|_{\Gamma_e} = \boldsymbol{0}}_{\diamond}, \quad \underbrace{V_i = \bar{V}_i \text{ for } i \in C_V}_{\ddagger}.$$

Derivation of the h-ϕ-formulation, cont'd

What remains is:

$$\operatorname{div}\boldsymbol{b}=0,\quad \overbrace{\textbf{curl}\,\boldsymbol{e}=-\partial_t\boldsymbol{b},\quad \boldsymbol{e}=\boldsymbol{e}_\mathrm{a}+\rho\,\boldsymbol{j},\quad \boldsymbol{b}=\mu\boldsymbol{h}}^{\Rightarrow\ \textbf{curl}\,(\rho\,\textbf{curl}\,\boldsymbol{h})+\textbf{curl}\,\boldsymbol{e}_\mathrm{a}=-\partial_t(\mu\boldsymbol{h})\ \circledast},$$

$$\underbrace{(\boldsymbol{e}-\bar{\boldsymbol{e}})\times\boldsymbol{n}|_{\Gamma_e}=\boldsymbol{0}}_{\diamond},\quad \underbrace{V_i=\bar{V}_i\text{ for } i\in C_V}_{\ddagger}.$$

- Multiply $\circledast$ by a test function $\boldsymbol{h}'$, in the same space than $\boldsymbol{h}$ but with homogeneous BC, and integrate over Ω,

$$(\partial_t(\mu\boldsymbol{h})\,,\boldsymbol{h}')_\Omega+(\textbf{curl}\,(\rho\,\textbf{curl}\,\boldsymbol{h})\,,\boldsymbol{h}')_\Omega+(\textbf{curl}\,\boldsymbol{e}_\mathrm{a}\,,\boldsymbol{h}')_\Omega=0,$$

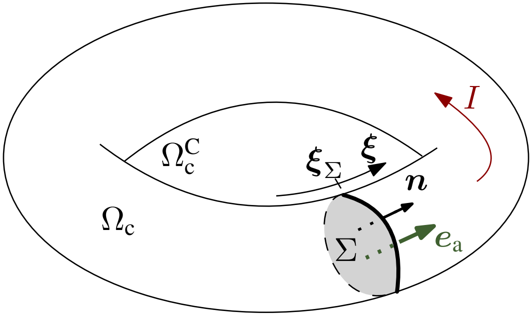
$$\Rightarrow(\partial_t(\mu\boldsymbol{h})\,,\boldsymbol{h}')_\Omega+(\rho\,\textbf{curl}\,\boldsymbol{h}\,,\textbf{curl}\,\boldsymbol{h}')_{\Omega_\mathrm{c}}+\underbrace{(\boldsymbol{e}_\mathrm{a}\,,\textbf{curl}\,\boldsymbol{h}')_{\Omega_\mathrm{c}}}_{\ddagger\ldots}$$

$$-\,(e\,\underbrace{(\boldsymbol{e}_\mathrm{a}+\rho\,\textbf{curl}\,\boldsymbol{h})\times\boldsymbol{n}}_{\text{Neumann BC }\diamond}\,,\ \boldsymbol{h}')_{\Gamma_e}=0$$

Derivation of the h-ϕ-formulation, cont'd

- The third term simplifies

$$
\begin{aligned}
(\boldsymbol{e}_\mathrm{a}\,, \textbf{curl}\; \boldsymbol{h}')_{\Omega_\mathrm{c}} &= V\,(\delta(\boldsymbol{\xi}-\boldsymbol{\xi}_\Sigma)\boldsymbol{n}\,, \textbf{curl}\; \boldsymbol{h}')_{\Omega_\mathrm{c}} \\
&= V\,(\boldsymbol{n}\,, \textbf{curl}\; \boldsymbol{h}')_\Sigma \\
&= V \oint_{\partial\Sigma} \boldsymbol{h}' \cdot d\boldsymbol{\ell} \\
&= V I' = \bar{V} I' \qquad \text{(Ampère's law + ⓘ).}
\end{aligned}
$$



Derivation of the h-ϕ-formulation, cont'd

What about $\operatorname{div} \boldsymbol{b} = 0$?

- Taking $\boldsymbol{h}' = \textbf{grad}\ \phi'$ in the formulation yields

$$
\begin{aligned}
&(\partial_t(\mu \boldsymbol{h}) \,, \textbf{grad}\ \phi')_\Omega + (\textbf{curl}\ (\boldsymbol{e}_\mathrm{a} + \rho\, \textbf{curl}\ \boldsymbol{h}) \,, \textbf{grad}\ \phi')_\Omega = 0, \\
\Rightarrow &- (\operatorname{div}\ (\partial_t(\mu \boldsymbol{h})) \,, \phi')_\Omega + (\partial_t(\mu \boldsymbol{h}) \cdot \boldsymbol{n} \,, \phi')_{\Gamma_e} \\
&- (\bar{\boldsymbol{e}} \times \boldsymbol{n} \,, \textbf{grad}\ \phi')_{\Gamma_e} = 0.
\end{aligned}
$$

One can show that $(\partial_t(\mu \boldsymbol{h}) \cdot \boldsymbol{n} \,, \phi')_{\Gamma_e} = (\boldsymbol{e} \times \boldsymbol{n} \,, \textbf{grad}\ \phi')_{\Gamma_e}$, so with $(\boldsymbol{e} - \bar{\boldsymbol{e}}) \times \boldsymbol{n}|_{\Gamma_e} = \boldsymbol{0}$, what remains is

$$
\partial_t \Big((\operatorname{div}\ (\mu \boldsymbol{h}) \,, \phi')_\Omega \Big) = 0,
$$

such that $\operatorname{div} \boldsymbol{b} = 0$ is (weakly) verified if the initial condition $\boldsymbol{h}_{t_0}$ is such that $(\operatorname{div}\ (\mu\, \boldsymbol{h}_{t_0}) \,, \phi')_\Omega = 0$.

h-ϕ-formulation

Finally, the h-ϕ-formulation amounts to find $\boldsymbol{h}$ in the chosen function space such that, $\forall \boldsymbol{h}'$,

$$\left(\partial_t(\mu \boldsymbol{h}) \ , \boldsymbol{h}'\right)_{\Omega} + \left(\rho \, \textbf{curl}\ \boldsymbol{h} \ , \textbf{curl}\ \boldsymbol{h}'\right)_{\Omega_c} - \left(\bar{\boldsymbol{e}} \times \boldsymbol{n} \ , \boldsymbol{h}'\right)_{\Gamma_e} + \sum_{i=1}^{N} V_i \mathcal{I}_i(\boldsymbol{h}') = 0,$$

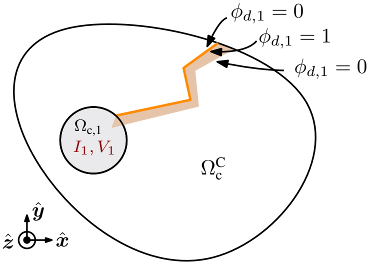
with $V_i = \bar{V}_i$ for $i \in C_V$, and $\mathcal{I}_i(\boldsymbol{h}') = I_i'$ (i.e. the DoF associated with the cut function $\boldsymbol{c}_i$).

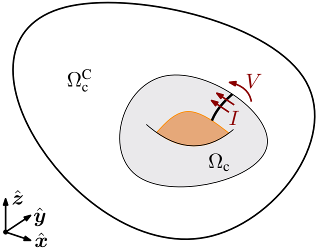
h-ϕ-formulation – Interpretation

When the test function $\boldsymbol{c}_i$ ($\mathcal{I}_i(\boldsymbol{c}_i) = 1$) is chosen, we get the equation:

$$(\partial_t(\mu \boldsymbol{h}) \,, \boldsymbol{c}_i)_{\Omega} + (\rho \, \textbf{curl}\, \boldsymbol{h} \,, \textbf{curl}\, \boldsymbol{c}_i)_{\Omega_c} = -V_i.$$

"Flux change $\mu \boldsymbol{h}$ ($= \boldsymbol{b}$) + circulation of $\rho \boldsymbol{j}$ ($= \boldsymbol{e}$),
both averaged over a transition layer = total voltage".





NB: The flux of $\mu \boldsymbol{h}$ depends on the chosen cut as $\mu \boldsymbol{h}$ is not a 2-form (as $\boldsymbol{b}$ should be).
Same for $\rho \boldsymbol{j}$.

Simple finite element formulations
The a-v-formulation
The h-ϕ-formulation

Resolution techniques
Time integration
Linearization methods
Comparison of the formulations

Mixed finite element formulations
The $h(-\phi)$-a-formulation
The t-a-formulation

Illustrations

Summary

References

Structure of the resolution

- After spatial discretization, we obtain a system of time-dependent, nonlinear ordinary differential equations of the form

$$\boldsymbol{K}(\boldsymbol{x}, t)\,\dot{\boldsymbol{x}}(t) + \boldsymbol{M}(\boldsymbol{x}, t)\,\boldsymbol{x}(t) = \boldsymbol{b}(t)$$

Structure of the resolution

- After spatial discretization, we obtain a system of time-dependent, nonlinear ordinary differential equations of the form

$$\boldsymbol{K}(\boldsymbol{x},t)\,\dot{\boldsymbol{x}}(t) + \boldsymbol{M}(\boldsymbol{x},t)\,\boldsymbol{x}(t) = \boldsymbol{b}(t)$$

- Resolution: two imbricated loops
  - Time-stepping: Implicit Euler with adaptive time steps t_n
  - Iterative solution of the nonlinear system at each time step t_n: Newton-Raphson or fixed point (Picard)

Simple finite element formulations
The a-v-formulation
The h-ϕ-formulation

Resolution techniques
Time integration
Linearization methods
Comparison of the formulations

Mixed finite element formulations
The h(-ϕ)-a-formulation
The t-a-formulation

Illustrations

Summary

References

Implicit Euler

Time derivatives at time step t_n are expressed as:

$$\frac{d\boldsymbol{x}}{dt}(t_n) = \frac{\boldsymbol{x}(t_n) - \boldsymbol{x}(t_{n-1})}{\Delta t},$$

with $\boldsymbol{x}(t_n)$ containing the DoFs and $\boldsymbol{u}(t_{n-1})$ being known from the initial conditions (first step) or from the previous step.

At each step t_n we end up with a system of nonlinear equations of the form

$$\boldsymbol{A}(\boldsymbol{x}(t_n))\,\boldsymbol{x}(t_n) = \boldsymbol{b}(t_n)$$

Implicit Euler

Time derivatives at time step t_n are expressed as:

$$\frac{d\boldsymbol{x}}{dt}(t_n) = \frac{\boldsymbol{x}(t_n) - \boldsymbol{x}(t_{n-1})}{\Delta t},$$

with $\boldsymbol{x}(t_n)$ containing the DoFs and $\boldsymbol{u}(t_{n-1})$ being known from the initial conditions (first step) or from the previous step.

At each step t_n we end up with a system of nonlinear equations of the form

$$\boldsymbol{A}(\boldsymbol{x}(t_n))\,\boldsymbol{x}(t_n) = \boldsymbol{b}(t_n)$$

Other possibilities can be implemented:

- Explicit Euler,
- Crank-Nicholson,
- Higher-order schemes (e.g. BDF)...

⇒ In Life-HTS we just explicitly write the scheme in the GetDP formulation.

Life-HTS implicit Euler in formulation

Example: flux variation term $(\partial_t(\mu \boldsymbol{h}) \,, \boldsymbol{h}')_\Omega$ in h-ϕ-formulation

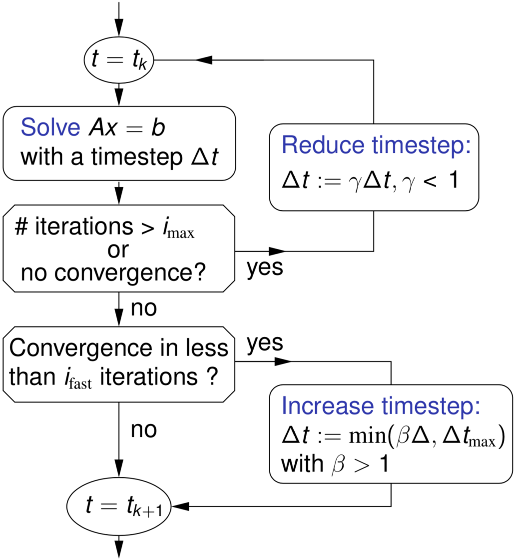
$$\left(\frac{\mu \boldsymbol{h}_n}{\Delta t} \,, \boldsymbol{h}'\right)_\Omega - \left(\frac{\mu \boldsymbol{h}_{n-1}}{\Delta t} \,, \boldsymbol{h}'\right)_\Omega$$

```
Formulation {
  { Name MagDyn_htot; Type FemEquation;
    Quantity {
      { Name h; Type Local; NameOfSpace h_space; }
      { [...] }
    }
    Equation {
      // Flux variation term (on the linear magnetic domain)
      Galerkin { [ mu[] * Dof{h} / $DTime , {h} ];
        In MagnLinDomain; Integration Int; Jacobian Vol; }
      Galerkin { [ - mu[] * {h}[1] / $DTime , {h} ];
        In MagnLinDomain; Integration Int; Jacobian Vol; }
      [...]
} } }
```

Syntax:

- `Dof{h}`: DoF at the current time step n (and iteration),
- `{h}[i]`: saved/known solution of $\boldsymbol{h}$ at time step $n - \mathtt{i}$,
- `{h}`: solution at the previous iteration (see later).

Adaptive time-stepping



Parameters:

- $\gamma = 1/2$
- $\beta = 2$
- $i_{\text{fast}} = i_{\max}/4$
- Fixed-point: $i_{\max} = 400$
- Newton-Raphson
  $i_{\max} = 50$

Life-HTS time-stepping in resolution

```
Resolution {
  { Name MagDyn;
    System { {Name A; NameOfFormulation MagDyn_htot;} }
    Operation {
      [...]
      // Initialize}
      SetTime[ timeStart ]; SetDTime[ dt ]; SetTimeStep[ 0 ];
      // Time loop
      While[$Time < timeFinalSimu && $DTime > 1e-10]{
        SetTime[ $Time + $DTime ]; SetTimeStep[ $TimeStep + 1 ];
        // Customized iterative loop
        Call CustomIterativeLoop;

        // If converged (= less than iter max and not diverged)...
        Test[ $iter < iter_max && ($res / $res0 <= 1e10)]{
          SaveSolution[A];
          Test[ $iter < iter_max / 2 && $DTime < dt_max]{
            Evaluate[ $dt_new = Min[$DTime * 2, dt_max] ];
            SetDTime[$dt_new];
          }
        }
        // ... otherwise, decrease the time step and start again
        {
          RemoveLastSolution[A];
          Evaluate[ $dt_new = $DTime / 2 ];
          SetDTime[$dt_new];
          SetTime[$Time - $DTime]; SetTimeStep[$TimeStep - 1];
        }
      }
    }
  }
}
```

Simple finite element formulations
- The a-v-formulation
- The h-ϕ-formulation

Resolution techniques
- Time integration
- Linearization methods
- Comparison of the formulations

Mixed finite element formulations
- The h(-ϕ)-a-formulation
- The t-a-formulation

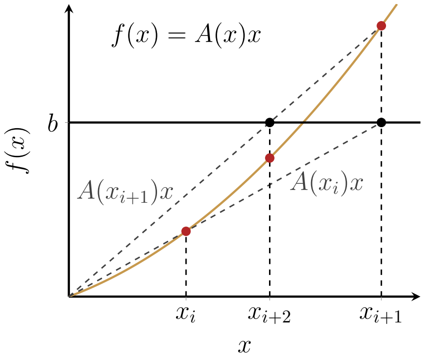
Illustrations

Summary

References

Solving a nonlinear equation: $f(x) = b$

1. Picard iteration method (a fixed point method):

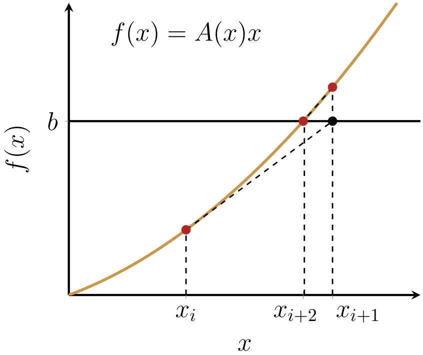


- Write $f(x)$ as $f(x) = A(x)x$.
- Get a first estimate x_0.
- At each iteration i:
    - solve $A(x_{i-1})x = b$,
    - $x_i := x$,
    - $i := i + 1$ and loop.
- Stop when convergence criterion is met.

- May converge for wide range of first estimates x_0.
- Convergence is slow!

Solving a nonlinear equation: $f(x) = b$

2. Newton-Raphson iterative method:



- Get a first estimate x_0.
- At each iteration i, solve for x_i:

$$\frac{df}{dx}(x_{i-1})\,(x_i - x_{i-1}) = f(x_{i-1}).$$

- Stop when convergence criterion is met.

- Quadratic convergence, if the initial est. x_0 is close enough.
- Relaxation factors can also be implemented.
- If x is a vector, $\frac{df}{dx}$ is a matrix (Jacobian matrix)...

Jacobian for isotropic constitutive laws

- Consider a constitutive law of the form

$$\boldsymbol{a}(\boldsymbol{x}) \;=\; g(\|\boldsymbol{x}\|)\,\boldsymbol{x}.$$

  Example: $\boldsymbol{e} = \rho\,\boldsymbol{j}$, or $\boldsymbol{b} = \mu\,\boldsymbol{h}$, ...

- The Newton-Raphson expansion can be cast in the form

$$\boldsymbol{a}(\boldsymbol{x}^i) \approx \boldsymbol{a}(\boldsymbol{x}^{i-1}) + \boldsymbol{J}(\boldsymbol{x}^{i-1}) \cdot \left(\boldsymbol{x}^i - \boldsymbol{x}^{i-1}\right),$$

  where $\boldsymbol{J}$ is the Jacobian matrix (i is the iteration index):

$$(\boldsymbol{J}(\boldsymbol{x}))_{jk} = \frac{\partial a_j}{\partial x_k} = \delta_{jk}\, g(\|\boldsymbol{x}\|) + x_j x_k\, \frac{\frac{dg(\|\boldsymbol{x}\|)}{d\|\boldsymbol{x}\|}}{\|\mathbf{x}\|}.$$

Jacobian for isotropic constitutive laws

- Consider a constitutive law of the form

$$\boldsymbol{a}(\boldsymbol{x}) \;=\; g(\|\boldsymbol{x}\|)\,\boldsymbol{x}.$$

  Example: $\boldsymbol{e} = \rho\,\boldsymbol{j}$, or $\boldsymbol{b} = \mu\,\boldsymbol{h}$, ...

- The Newton-Raphson expansion can be cast in the form

$$\boldsymbol{a}(\boldsymbol{x}^i) \approx \boldsymbol{a}(\boldsymbol{x}^{i-1}) + \boldsymbol{J}(\boldsymbol{x}^{i-1}) \cdot \left(\boldsymbol{x}^i - \boldsymbol{x}^{i-1}\right),$$

  where $\boldsymbol{J}$ is the Jacobian matrix (i is the iteration index):

$$(\boldsymbol{J}(\boldsymbol{x}))_{jk} = \frac{\partial a_j}{\partial x_k} = \delta_{jk}\, g(\|\boldsymbol{x}\|) + x_j x_k \, \frac{\frac{dg(\|\boldsymbol{x}\|)}{d\|\boldsymbol{x}\|}}{\|\mathbf{x}\|}.$$

- Example: $(\rho\,\textbf{curl}\;\boldsymbol{h}\;, \textbf{curl}\;\boldsymbol{h}')_{\Omega_c}$ in h-ϕ-formulation, with $\textbf{curl}\;\boldsymbol{h} = \boldsymbol{j}$:

$$(\rho(\boldsymbol{j}^{i-1})\,\boldsymbol{j}^{i-1}\;, \textbf{curl}\;\boldsymbol{h}')_{\Omega_c} + \left(\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{j}}(\boldsymbol{j}^{i-1})\;\boldsymbol{j}^{i}\;, \textbf{curl}\;\boldsymbol{h}'\right)_{\Omega_c} - \left(\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{j}}(\boldsymbol{j}^{i-1})\;\boldsymbol{j}^{i-1}\;, \textbf{curl}\;\boldsymbol{h}'\right)_{\Omega_c}$$

Worked-out Jacobians in [J. Dular et al. TAS 30 8200113 (2020)]

Life-HTS Picard and Newton-Raphson in formulation

Example: nonlinear term $(\rho\, \textbf{curl}\, \boldsymbol{h}\,, \textbf{curl}\, \boldsymbol{h}')_{\Omega_c}$ in h-ϕ-formulation

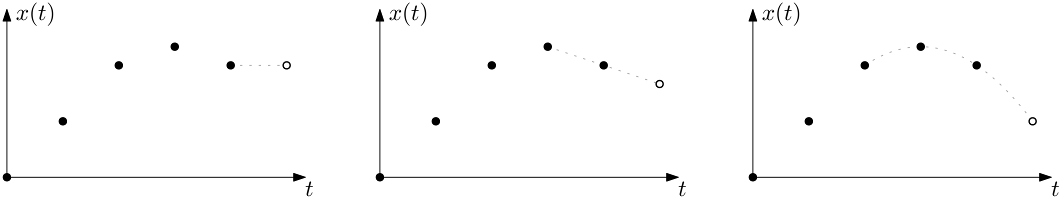
N-R: $$(\rho(\boldsymbol{j}^{i-1})\,\boldsymbol{j}^{i-1}\,, \textbf{curl}\, \boldsymbol{h}')_{\Omega_c} + \left(\frac{\partial e}{\partial \boldsymbol{j}}(\boldsymbol{j}^{i-1})\,\boldsymbol{j}^{i}\,, \textbf{curl}\, \boldsymbol{h}'\right)_{\Omega_c} - \left(\frac{\partial e}{\partial \boldsymbol{j}}(\boldsymbol{j}^{i-1})\,\boldsymbol{j}^{i-1}\,, \textbf{curl}\, \boldsymbol{h}'\right)_{\Omega_c}$$

```
Formulation {
  { Name MagDyn_htot; Type FemEquation;
    Quantity {
      { Name h; Type Local; NameOfSpace h_space; }
      { [...] }
    }
    Equation {
      // (1) Picard
      Galerkin { [ rho[{d h}]] * Dof{d h} , {d h} ];
        In NonLinOmegaC; Integration Int; Jacobian Vol; }

      // (2) Newton-Raphson}
      Galerkin { [ rho[{d h}] * {d h} , {d h} ];
        In NonLinOmegaC; Integration Int; Jacobian Vol; }
      Galerkin { [ dedj[{d h}] * Dof{d h} , {d h} ];
        In NonLinOmegaC; Integration Int; Jacobian Vol; }
      Galerkin { [ - dedj[{d h}] * {d h} , {d h} ];
        In NonLinOmegaC ; Integration Int; Jacobian Vol; }
      [...]
} } }
```

Choosing the first estimate

- We use polynomial extrapolation:



(a) Zeroth-order extrapolation (b) First-order extrapolation (c) Second-order extrapolation

- It can significantly affect the required number of iterations
- Best results:
  - 1st order for the h-ϕ-formulation
  - 2nd order for the a-v-formulation

In the resolution: `SetExtrapolationOrder[ n ];` ($n \in \mathbb{N}$)

Convergence criterion

- The residual $\boldsymbol{b} - \boldsymbol{A}(\boldsymbol{x}_i)\boldsymbol{x}_i$ can be misleading
- In practice we usually choose the electromagnetic power, P, as a (global) convergence indicator:

h-ϕ-formulation

$$P = (\partial_t(\mu\,\boldsymbol{h})\ ,\boldsymbol{h})_{\Omega} + (\rho\,\textbf{curl}\ \boldsymbol{h}\ ,\textbf{curl}\ \boldsymbol{h})_{\Omega_c}$$

a-v-formulation

$$P = (\partial_t(\textbf{curl}\ \boldsymbol{a})\ ,\nu\,\textbf{curl}\ \boldsymbol{a})_{\Omega} + (\sigma\boldsymbol{e}\ ,\boldsymbol{e})_{\Omega_c}$$

with $\boldsymbol{e} = -\partial_t\boldsymbol{a} - \textbf{grad}\ v$

- We stop when $|\Delta P/P|$ is small enough:
  - $\approx 10^{-8}$ with Newton-Raphson
  - $\approx 10^{-4}$ with Picard

Simple finite element formulations
The a-v-formulation
The h-ϕ-formulation

Resolution techniques
Time integration
Linearization methods
Comparison of the formulations

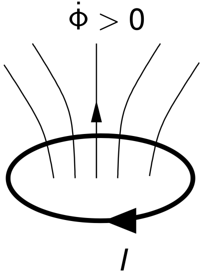
Mixed finite element formulations
The h(-ϕ)-a-formulation
The t-a-formulation

Illustrations

Summary

References

To fix ideas: a superconducting ring

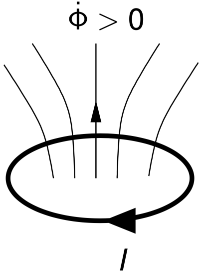


Consider a superconducting ring subjected to a time-varying flux, $\dot{\Phi}$. The ring is modelled as a non-linear lump resistor with

$$R(|I|) = \frac{V_c}{I_c} \left(\frac{|I|}{I_c}\right)^{n-1},$$

where V_c and I_c are characteristic voltage and current, and n is a critical index.

To fix ideas: a superconducting ring



Consider a superconducting ring subjected to a time-varying flux, $\dot{\Phi}$. The ring is modelled as a non-linear lump resistor with

$$R(|I|) = \frac{V_c}{I_c}\left(\frac{|I|}{I_c}\right)^{n-1},$$

where V_c and I_c are characteristic voltage and current, and n is a critical index.

The circuit equation

$$\dot{\Phi} = R(|I|)\, I + L\dot{I}$$

can be solved in one of two ways!

Ring, 1st way: solve for the current I

- Discretize in time: $t_j = j\Delta t, j = 0, 1, 2, \ldots,$
- Consider the implicit Euler method with $\dot{I} \approx (I_j - I_{j-1})/\Delta t$,

$$\dot{\Phi} = R(|I|)\, I + L\dot{I} \quad \rightarrow \quad \dot{\Phi}_j \;=\; V_c \, \frac{|I_j|^{n-1}\, I_j}{I_c^n} + L\, \frac{I_j - I_{j-1}}{\Delta t}.$$

Ring, 1st way: solve for the current I

- Discretize in time: $t_j = j\Delta t, j = 0, 1, 2, \ldots,$
- Consider the implicit Euler method with $\dot{I} \approx (I_j - I_{j-1})/\Delta t$,

$$\dot{\Phi} = R(|I|)\, I + L\dot{I} \quad \rightarrow \quad \dot{\Phi}_j \;=\; V_c \frac{|I_j|^{n-1}\, I_j}{I_c^n} + L\,\frac{I_j - I_{j-1}}{\Delta t}.$$

- Make this adimensional by introducing $x = aI_j/I_c$, to obtain

$$b = |x|^{n-1}\, x + x, \quad (I\text{-form}),$$

where

$$a = \left(\frac{V_c \Delta t}{L I_c}\right)^{1/(n-1)} \quad \text{and} \quad b = \frac{\dot{\Phi}_j + L I_{j-1}/\Delta t}{a L I_c/\Delta t}.$$

Ring, 2nd way: solve for the voltage drop across R

- Solve now in terms of $V_j = RI_j$,

$$\dot{\Phi} = R(|I|)\, I + L\dot{I} \quad \rightarrow \quad \dot{\Phi}_j = V_j + L\, \frac{I_c\, |V_j/V_c|^{1/n-1}\, V_j/V_c - I_{j-1}}{\Delta t}.$$

Ring, 2nd way: solve for the voltage drop across R

- Solve now in terms of $V_j = RI_j$,

$$\dot{\Phi} = R(|I|)\, I + L\dot{I} \quad \rightarrow \quad \dot{\Phi}_j = V_j + L\,\frac{I_c\,|V_j/V_c|^{1/n-1}\,V_j/V_c - I_{j-1}}{\Delta t}.$$

- Make this adimensional with $x = cV_j/V_c$, to get

$$d = |x|^{1/n-1}\,x + x, \quad (V\text{-form}),$$

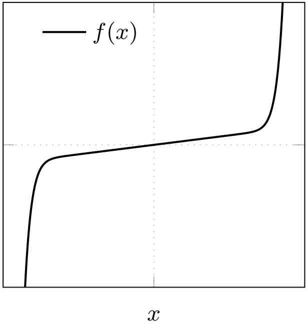
where

$$c = \left(\frac{\Delta t}{LI_c}\right)^{n/(n-1)} \quad \text{and} \quad d = \frac{\dot{\Phi}_j}{c} + \frac{LI_{j-1}}{c\Delta t}.$$

Ring example, summary

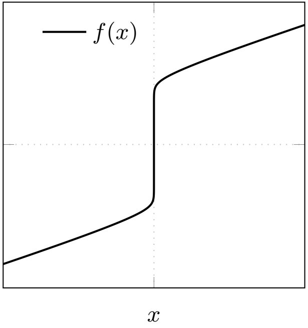
In each case, need to solve an equation of the form $f(x) =$ constant:

$$f(x) = |x|^{n-1}\, x + x$$



I-form
$\sim$ h-conform (Ampère)

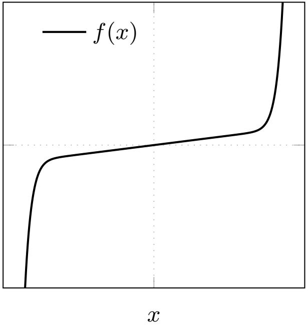
$$f(x) = |x|^{1/n-1}\, x + x$$



V-form
$\sim$ b-conform (Faraday)

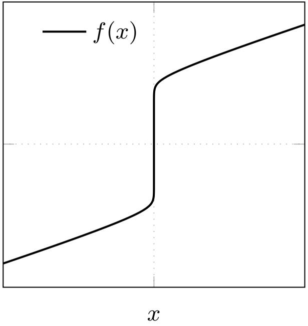
Nonlinearity in HTS for complementary formulations

$$f(x) = |x|^{n-1}x + x$$



h-ϕ-formulation $(\boldsymbol{e} = \rho\,\boldsymbol{j})$

$$f(x) = |x|^{1/n-1}x + x$$



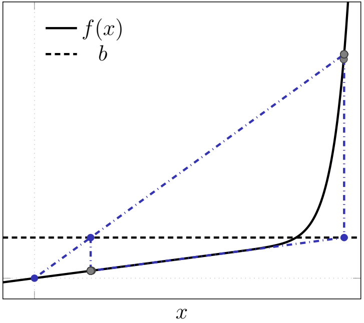
a-v-formulation $(\boldsymbol{j} = \sigma \boldsymbol{e})$

Different nonlinearities $\Rightarrow$ different numerical behaviors

Warning!

Beware of cycles

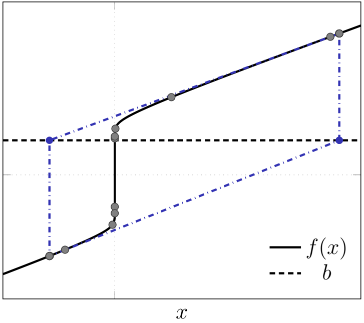
Cycles can occur in each method, depending on the shape of the function $f(x)$:



Picard iteration on

h-ϕ-formulation

Prefer Newton-Raphson!



Newton-Raphson iteration on

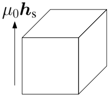
a-v-formulation

Prefer Picard!

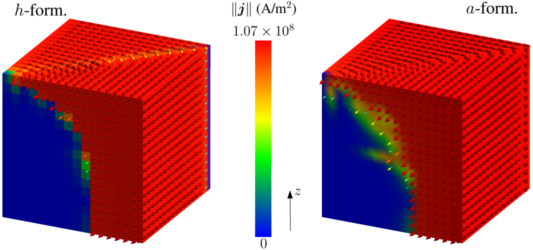
Relaxation factors can help, but no efficient solution (that we know of)

Illustration for a superconducting cube

System

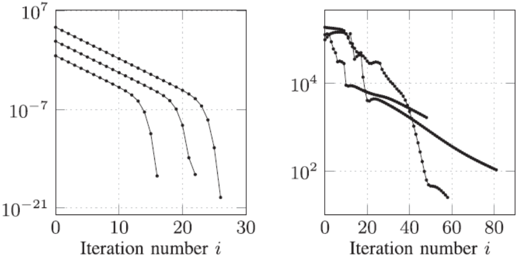


Side $a = 10$ mm.
$\mu_0 \boldsymbol{h}_s = \hat{z}\ B_0\ \sin(2\pi f t)$,
with $B_0 = 200$ mT,
$f = 50\,$Hz,
$j_c = 10^8$ A/m^2 and
$n = 100$.



Residual

- L_2 norm of $\boldsymbol{r} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$
- Left: h-ϕ-formulation
- Right: a-v-formulation



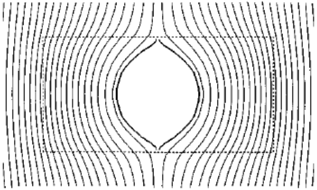
(a) Newton-Raphson technique. (b) Picard technique.

$\Rightarrow$ Much more efficient with Newton-Raphson (as is expected!)

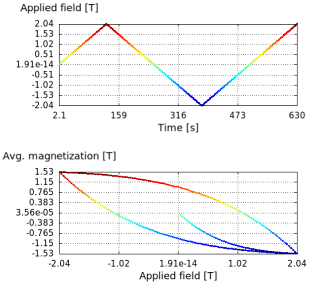
Hands-on: h-ϕ- and a-v-formulation

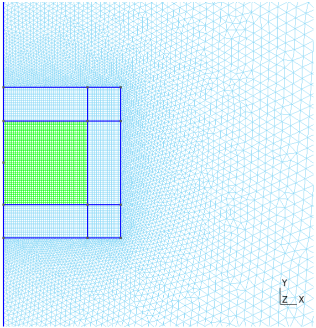
Magnetization of a superconducting pellet: phenomenology

Magnetize a cylindrical pellet of aspect ratio 0.5 (height/diameter) in an axial field of maximum $0.6 \times$ the penetration field:



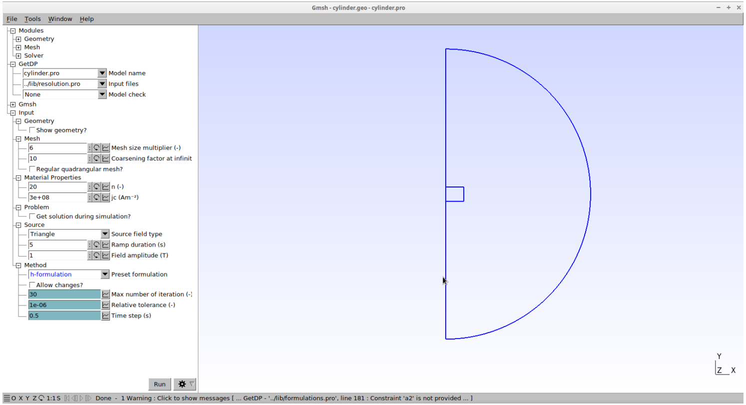
E. H. Brandt, PRB 58 (1998) 6506





Hands-on: h-ϕ- and a-v-formulation

Magnetization of a superconducting pellet: h-ϕ-formulation and a-v-formulation



models/Life-HTS/cylinder/cylinder.pro

Conclusion for HTS

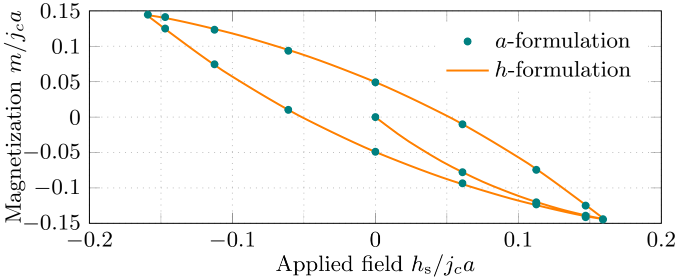
The diverging slope associated with $\boldsymbol{j} = \sigma \boldsymbol{e}$ for $\boldsymbol{j} \to 0$ is really difficult to handle.

$\Rightarrow$ Among the two simple formulations, the h-ϕ-formulation is much more efficient for systems with HTS:

- with an adaptive time-stepping algorithm,
- solved with a Newton-Raphson method,
- with a first estimate obtained by 1^{st}-order extrapolation.

One particular case: "single time step"

- For large values of n, nearly a critical state model.
- Robustness of Picard on the $\boldsymbol{j} = \sigma \boldsymbol{e}$ law can help to reduce the number of time steps.



- Here, for a magnetization cycle (3D cube problem)
  - lines: h-ϕ-formulation with 300 time steps,
  - dots: a-v-formulation with 20 time steps $\Rightarrow$ much faster!
- In practice, accurate for $\boldsymbol{j}$ and $\boldsymbol{b}$, but $\boldsymbol{e}$ is underestimated

Simple finite element formulations
- The a-v-formulation
- The h-ϕ-formulation

Resolution techniques
- Time integration
- Linearization methods
- Comparison of the formulations

Mixed finite element formulations
- The $h(-\phi)$-a-formulation
- The t-a-formulation

Illustrations

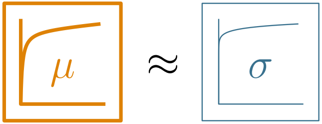
Summary

References

Ferromagnetic materials

The nonlinearity is in the magnetic constitutive law.

- h-ϕ-formulation the involved law is $\boldsymbol{b} = \mu \boldsymbol{h}$.

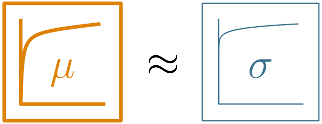


$\Rightarrow$ Easily enters cycles with Newton-Raphson.
OK with Picard, or N-R with relaxation factors but slow.

Ferromagnetic materials

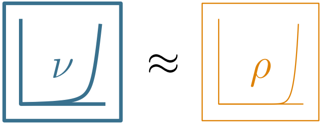
The nonlinearity is in the magnetic constitutive law.

- h-ϕ-formulation the involved law is $\boldsymbol{b} = \mu \boldsymbol{h}$.



  $\Rightarrow$ Easily enters cycles with Newton-Raphson.
  OK with Picard, or N-R with relaxation factors but slow.

- a-v-formulation the involved law is $\boldsymbol{h} = \nu \boldsymbol{b}$.



  $\Rightarrow$ Efficiently solved with Newton-Raphson.

The a-v-formulation is more appropriate for dealing with the nonlinearity, whereas for HTS, the complementary formulation was best.

Simple finite element formulations
- The a-v-formulation
- The h-ϕ-formulation

Resolution techniques
- Time integration
- Linearization methods
- Comparison of the formulations

Mixed finite element formulations
- The h(-ϕ)-a-formulation
- The t-a-formulation

Illustrations

Summary

References

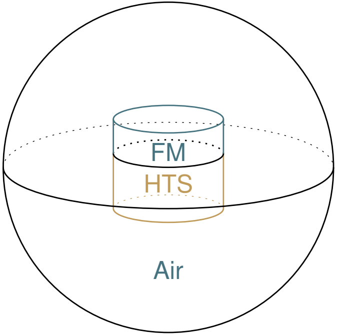
Coupled materials – $h(\text{-}\phi)\text{-}a$-formulation

Use the best formulation in each material

Decompose the domain Ω, for example into:

- $\Omega^h = \{\text{HTS}\}$
- $\Omega^a = \{\text{Ferromagnet}, \text{Air}\}$

and couple via $\Gamma_\mathrm{m} = \partial(\text{HTS})$:



$$\Big(\partial_t(\mu \boldsymbol{h})\ ,\boldsymbol{h}'\Big)_{\Omega^h} + \Big(\rho\,\textbf{curl}\ \boldsymbol{h}\ ,\textbf{curl}\ \boldsymbol{h}'\Big)_{\Omega^h_c} + (\partial_t \boldsymbol{a} \times \boldsymbol{n}_{\Omega^h}\ ,\boldsymbol{h}')_{\Gamma_\mathrm{m}} = 0,$$

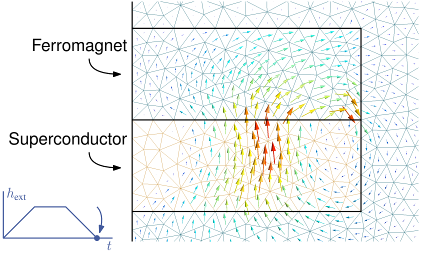
$$(\nu\,\textbf{curl}\ \boldsymbol{a}\ ,\textbf{curl}\ \boldsymbol{a}')_{\Omega^a} - (\boldsymbol{h} \times \boldsymbol{n}_{\Omega^a}\ ,\boldsymbol{a}')_{\Gamma_\mathrm{m}} = 0.$$

(For homogeneous Neumann BC)

$h(\text{-}\phi)\text{-}a$-formulation results

Example:

- Stacked cylinders
- 2D axisymmetric
- External applied field



Number of iterations for three discretization levels:

	h-ϕ-formulation	a-v-formulation	$h(\text{-}\phi)\text{-}a$-formulation
Coarse	1878	4381	**1071**
Medium	3366	7539	**1931**
Fine	4422	14594	**3753**

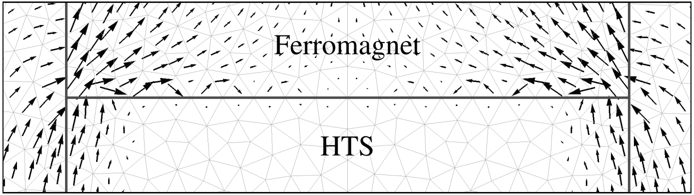
In general, a speed-up from 1.2 to 3 is obtained.

$h(\text{-}\phi)\text{-}a$-**formulation** stability

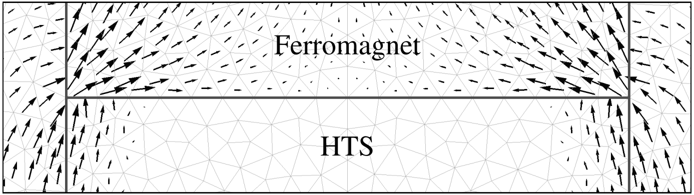
The formulation is mixed (two unknown fields on Γ_{m})

$\Rightarrow$ Shape functions must satisfy an inf-sup condition.

- First-order functions for $\boldsymbol{h}$ and $\boldsymbol{a}$ (inf-sup KO)



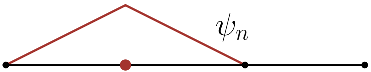
- Second-order for $\boldsymbol{a}$, first-order for $\boldsymbol{h}$ (inf-sup OK)



Life-HTS Hierarchical functions

Example for 2nd-order shape functions for $\boldsymbol{a}$ (in 2D) on Γ_{m}:

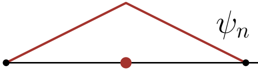
```
FunctionSpace{
  { Name a_space_2D; Type Form1P;
    BasisFunction {
      // Usual first-order functions
      { Name psin; NameOfCoef an; Function BF_PerpendicularEdge;
        Support Omega_a_AndBnd; Entity NodesOf[All]; }
      // Second-order functions on BndOmega_ha only
      { Name psin2; NameOfCoef an2; Function BF_PerpendicularEdge_2E;
        Support Omega_a_AndBnd; Entity EdgesOf[BndOmega_ha]; }
    }
    Constraint {
      { NameOfCoef an; EntityType NodesOf; NameOfConstraint a; }
      { NameOfCoef an2; EntityType EdgesOf; NameOfConstraint a2; }
    }
  }
}
```

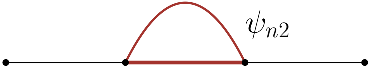


Life-HTS Hierarchical functions

Example for 2nd-order shape functions for $\boldsymbol{a}$ (in 2D) on Γ_m:

```
FunctionSpace{
  { Name a_space_2D; Type Form1P;
    BasisFunction {
      // Usual first-order functions
      { Name psin; NameOfCoef an; Function BF_PerpendicularEdge;
        Support Omega_a_AndBnd; Entity NodesOf[All]; }
      // Second-order functions on BndOmega_ha only
      { Name psin2; NameOfCoef an2; Function BF_PerpendicularEdge_2E;
        Support Omega_a_AndBnd; Entity EdgesOf[BndOmega_ha]; }
    }
    Constraint {
      { NameOfCoef an; EntityType NodesOf; NameOfConstraint a; }
      { NameOfCoef an2; EntityType EdgesOf; NameOfConstraint a2; }
    }
  }
}
```





NB: This is for a locally enriched function space. Using 2nd-order elements on the whole domain can be done directly at the meshing step (using e.g. `gmsh -order 2`).

Simple finite element formulations
The a-v-formulation
The h-ϕ-formulation

Resolution techniques
Time integration
Linearization methods
Comparison of the formulations

Mixed finite element formulations
The h(-ϕ)-a-formulation
The t-a-formulation

Illustrations

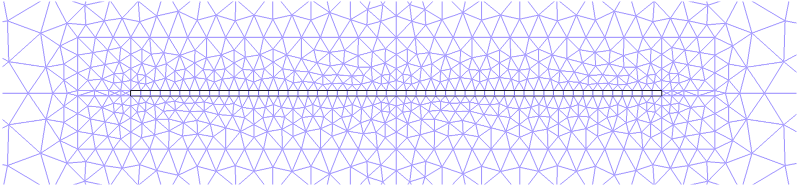
Summary

References

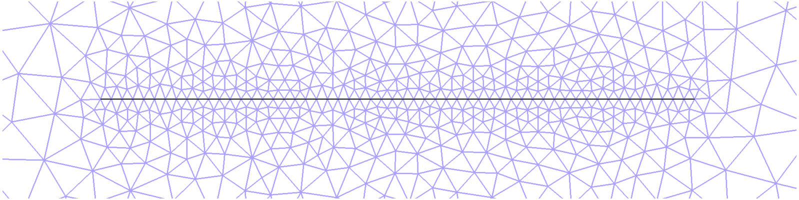
HTS tapes - t-a-formulation

To model thin superconducting tapes, two main possibilities:

1. Use the true geometry and the h-ϕ-formulation with one-element across the thickness (quadrangle):



2. Perform the slab approximation and model the tape as a line $\Rightarrow$ t-a-formulation:

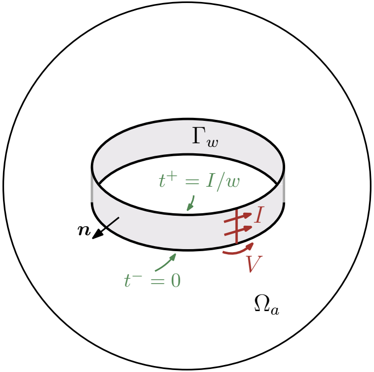


t-a-formulation

Consider a tape Γ_w of thickness w.

The current density is described by a current potential $\boldsymbol{t}$:

- such that $\boldsymbol{j} = \textbf{curl}\ \boldsymbol{t}$,
- gauged by being defined along the normal of the tape, $\boldsymbol{t} = t\boldsymbol{n}$,
- with BC related to the total current I $(t^+ - t^- = I/w)$.



In Ω_a, write the a-v-formulation and express the surface integral $(\boldsymbol{h} \times \boldsymbol{n}\ , \boldsymbol{a}')_{\Gamma_w}$ in terms of the surface current density $w\,\textbf{curl}\ \boldsymbol{t}$.

t-a-formulation

Find $\boldsymbol{a}$ and $\boldsymbol{t}$ in the chosen function spaces such that, $\forall \boldsymbol{a}', \boldsymbol{t}'$:

$$(\nu\,\textbf{curl}\ \boldsymbol{a}\ ,\textbf{curl}\ \boldsymbol{a}')_{\Omega_a} - \left(\bar{\boldsymbol{h}} \times \boldsymbol{n}_\Omega\ ,\boldsymbol{a}'\right)_{\Gamma_h} - (w\,\textbf{curl}\ \boldsymbol{t}\ ,\boldsymbol{a}')_{\Gamma_w} = 0,$$

$$(w\,\partial_t \boldsymbol{a}\ ,\textbf{curl}\ \boldsymbol{t}')_{\Gamma_w} + (w\,\rho\,\textbf{curl}\ \boldsymbol{t}\ ,\textbf{curl}\ \boldsymbol{t}')_{\Gamma_w} = -\sum_{i \in C} V_i \mathcal{I}_i(\boldsymbol{t}'),$$

with $V_i = \bar{V}_i$ for $i \in C_V$, and $\mathcal{I}_i(\boldsymbol{t}') = I_i'$ (i.e. the DoF associated with the BC $w(t^+ - t^-)$).

It is basically an h(-ϕ)-a-formulation with a slab approximation

$\Rightarrow$ More information and applications in F. Grilli's lecture tomorrow

See also [Bortot, L., et al., IEEE TAS 30(5), 1-11 (2020)]

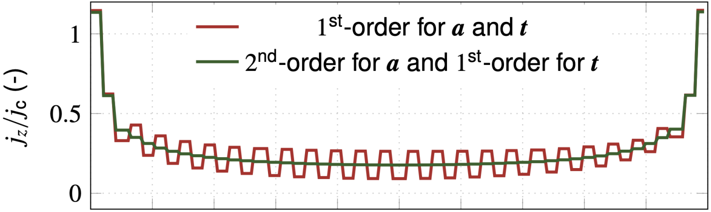
t-a-formulation - Stability

The t-a-formulation is mixed (two unknown fields on Γ_w)

$\Rightarrow$ Shape functions must satisfy an inf-sup condition

Similar conclusions than with the h(-ϕ)-a-formulation

Example for a 2D case, current density along the tape:



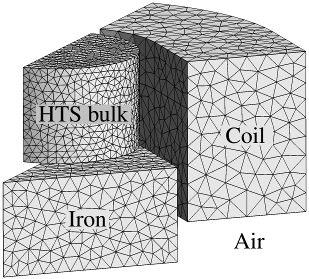
Life-HTS function space for t

Defined as a scalar quantity in the `FunctionSpace`, the normal $\boldsymbol{n}$ is introduced in the formulation:

$$t = \sum_{n \in \Gamma_w \setminus \partial\Gamma_w} t_n \psi_n + \sum_{i=1}^{N} T_i \ell_i, \quad \text{with} \quad \boldsymbol{t} = t\boldsymbol{n}.$$

```
FunctionSpace{
  { Name t_space; Type Form0;
    BasisFunction {
      // Node functions except on the lateral edges of the tapes
      { Name psin; NameOfCoef tn; Function BF_Node;
        Support Gamma_w; Entity NodesOf[All, Not LateralEdges]; }
      // Global shape function for representing a net current intensity
      { Name elli; NameOfCoef Ti; Function BF_GroupOfNodes;
        Support Gamma_w_AndBnd; Entity GroupsOfNodesOf[PositiveEdges]; }
    }
    GlobalQuantity {
      // Global quantities to be used in the formulation
      { Name T ; Type AliasOf        ; NameOfCoef Ti ; }
      { Name V ; Type AssociatedWith ; NameOfCoef Ti ; }
    }
    Constraint {
      { NameOfCoef V; EntityType GroupsOfNodesOf; NameOfConstraint Voltage; }
      { NameOfCoef T; EntityType GroupsOfNodesOf; NameOfConstraint Current_w; }
    }
  }
}
```

Hands-on: 3D HTS magnet motor pole

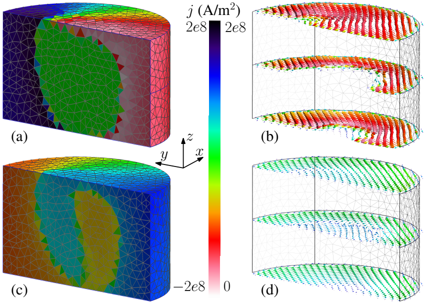


One eight of the geometry (air domain not shown)

models/Life-HTS/magnet/magnet.pro

	NL laws	Function space	Number of DOFs	$\sigma \neq 0$ in Ω_c^C?
h	ρ, μ	$\boldsymbol{h} \in \mathcal{H}(\Omega) = \{\boldsymbol{h} \in H(\Omega)\}$	Edges in Ω	Yes
h-ϕ	ρ, μ	$\boldsymbol{h} \in \mathcal{H}_\phi(\Omega) = \{\boldsymbol{h} \in H(\Omega) \mid \textbf{curl}\, \boldsymbol{h} = \boldsymbol{0} \text{ in } \Omega_c^C\}$	Edges in Ω_c + Nodes in Ω_c^C	No
$\bar{a}$	σ, ν	$\boldsymbol{a} \in \bar{\mathcal{A}}(\Omega) = \{\boldsymbol{a} \in H(\Omega)\}$	Edges in Ω	(Yes)*
a	σ, ν	$\boldsymbol{a} \in \mathcal{A}(\Omega) = \{\boldsymbol{a} \in H(\Omega) \mid \text{co-tree gauge in } \Omega_c^C\}$	Edges in Ω_c + Facets in Ω_c^C	No
h-a	ρ, ν	$\boldsymbol{h} \in \mathcal{H}_\phi(\Omega_c)$, $\boldsymbol{a} \in \mathcal{A}(\Omega_c^C)$	Edges in Ω_c + Facets† in Ω_c^C	No
h-ϕ-a	ρ, ν	$\boldsymbol{h} \in \mathcal{H}_\phi(\Omega_m^C)$, $\boldsymbol{a} \in \mathcal{A}(\Omega_m)$	Edges in $\Omega_{h,c}$ + Nodes† in $\Omega_{h,c}^C$ + Facets in Ω_m	No
h-ϕ-b	ρ, ν	$\boldsymbol{h} \in \mathcal{H}_\phi(\Omega)$, $\boldsymbol{b} \in (H_3(\Omega_m))^3$	Edges in Ω_c + Nodes in Ω_c^C + Volumes ($\times 3$) in Ω_m	No
a-j	ρ, ν	$\boldsymbol{a} \in \mathcal{A}(\Omega)$, $\boldsymbol{j} \in \mathcal{A}(\Omega_c)$	Edges ($\times 2$) in Ω_c + Facets in Ω_c^C	No

Hands-on: 3D HTS magnet motor pole



Current density in the bulk during magnetizing pulse and relaxation

[J. Dular et al. IEEE Trans. Mag. (2022)]

	HTS loss (J)	# DOFs	# iterations	Time/it.	Total time
h	6.35	35,532	4,057	3.3s	3h42
h-ϕ	6.36	12,172	3,937	1.4s	1h33
$\bar{a}$	6.38	29,010	2,955	3.1s	2h33
a	6.39	26,964	3,147	2.1s	1h48
h-a	6.31	32,045	1,124	2.7s	0h50
h-ϕ-a	6.33	15,776	1,108	2.1s	0h39
h-ϕ-b	6.37	20,821	1,104	3.2s	0h58
a-j	6.34	36,019	2,225	3.6s	2h15

Simple finite element formulations
The a-v-formulation
The h-ϕ-formulation

Resolution techniques
Time integration
Linearization methods
Comparison of the formulations

Mixed finite element formulations
The h(-ϕ)-a-formulation
The t-a-formulation

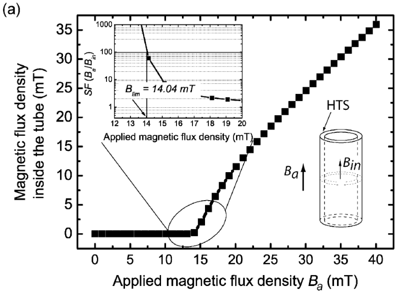
Illustrations

Summary

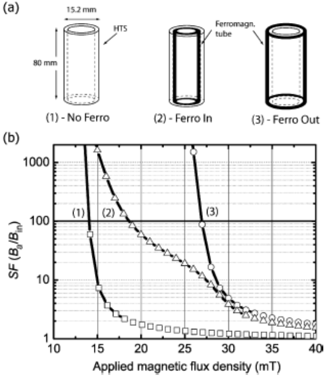
References

Improving HTS magnetic shields with a soft ferromagnetic material

Shielding an axial field with a HTS tube



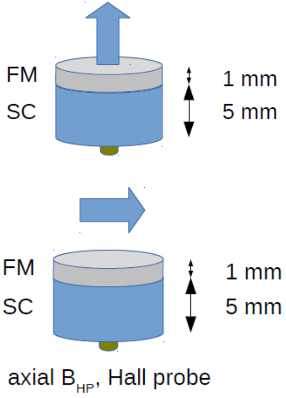
Shielding with an additional ferromagnetic tube



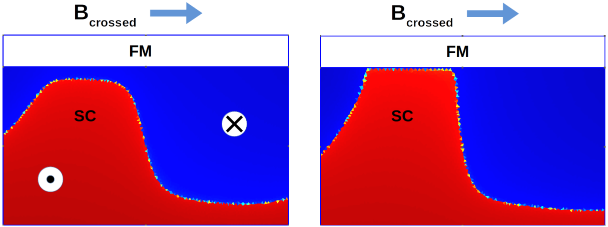
[Lousberg et al., TAS (2010)]

Protecting a bulk HTS against crossed-field demagnetisation with a ferromagnetic layer

Sequence of applied fields

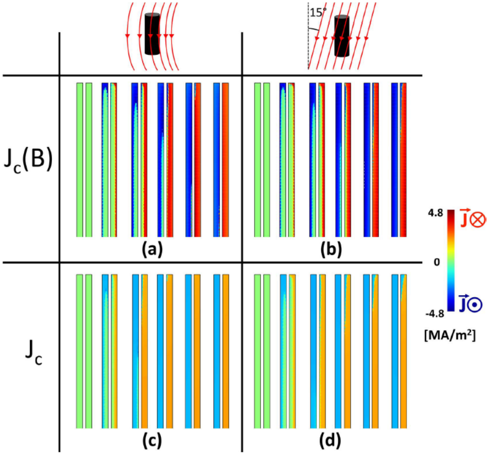


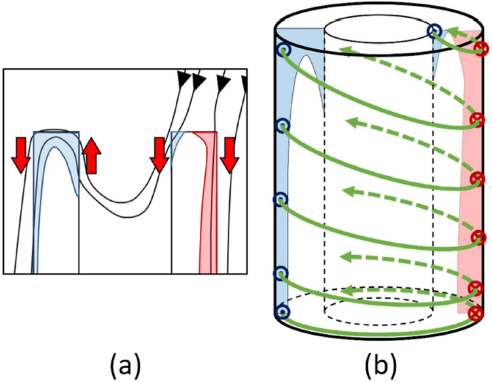
Current distribution in the bulk with a ferromagnetic top layer ($\mu_r = 10, 100$)



[Fagnard et al., SUST (2016)]

Magnetic shielding in inhomogeneous fields

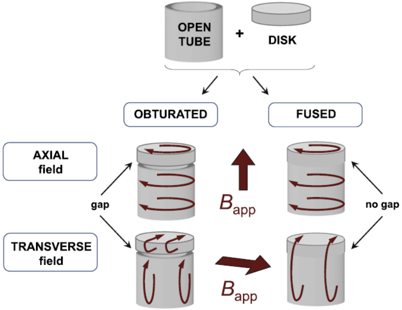




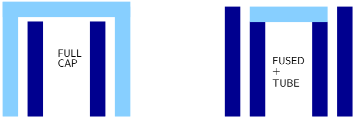
[Hogan et al., SUST (2018)]

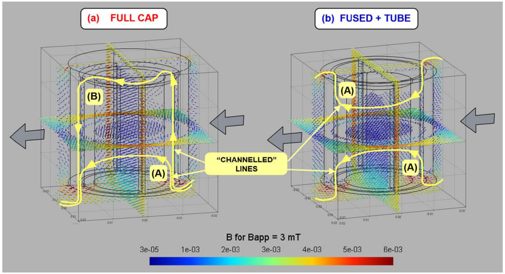
Magnetic shielding, bulk superconducting cylinders and caps

Induced currents vs. geometries



Tracking stray fields in composite shields

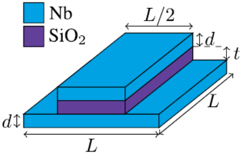




[Fagnard et al., SUST (2019)]

Critical states in stacked Niobium films

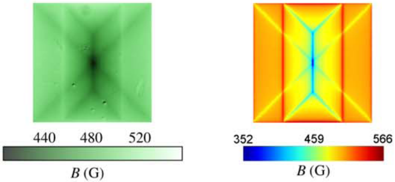
Peculiar patterns of discontinuity lines in stacks of Nb films



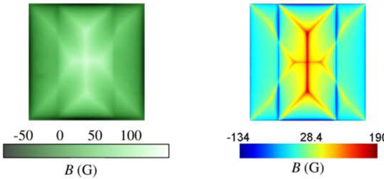
$L = 200\ \mu$m, $d = t = 300$ nm

Needs to include a genuine $J_c(B)$-dependence

Raising field stage



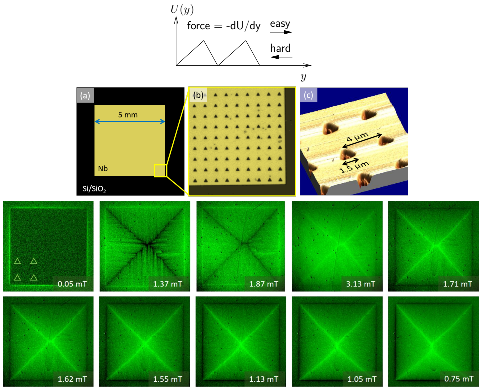
Decreasing field stage



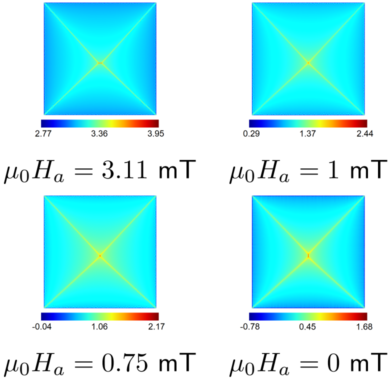
[Burger et al., SUST (2019)]

Critical states in the presence of a ratchet pinning potential

Experiment: rotation of the central discontinuity line in the decreasing field stage, after magnetization

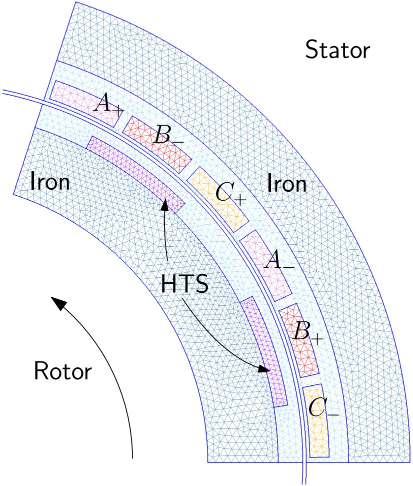


Model: an anisotropic pinning force reproduces the result



[Motta et al., Phys. Rev. B (2022)]

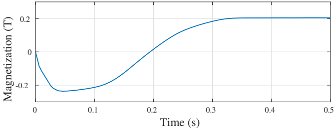
Rotating HTS motor



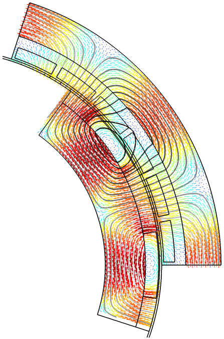
[HTS motors School (2020)]

Pulse magnetization (h-a-formulation)

$$I_{B_\pm}(t) = -I_{C_\pm}(t) = \pm\, I_{\mathsf{max}}\, \tfrac{t}{\tau}\, \exp\,(1 - t/\tau),\, I_{A_\pm}(t) = 0$$



3-phase (A-B-C) motor mode (a-formulation)



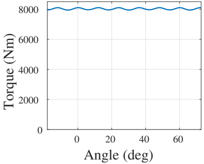
$$I_{A_\pm}(t) = \pm\, I_{\mathsf{max}}\, \sin(\omega t)$$

$$I_{B_\pm}(t) =$$

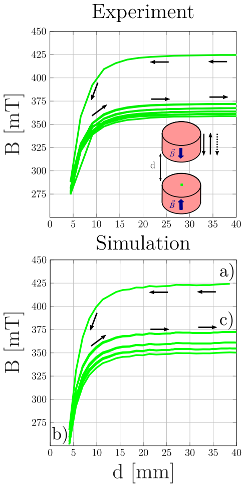
$$\pm\, I_{\mathsf{max}}\, \sin(\omega t + 2\pi/3)$$

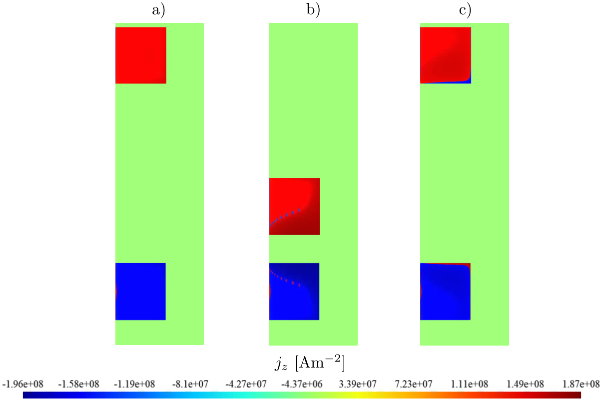
$$I_{C_\pm}(t) =$$

$$\pm\, I_{\mathsf{max}}\, \sin(\omega t - 2\pi/3)$$



2D axisymmetric model of moving bulk superconductors

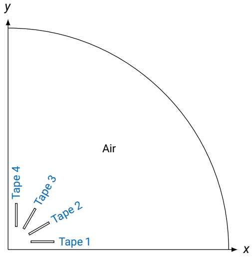


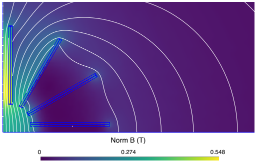


[M. Houbart et al., SUST (in press)]

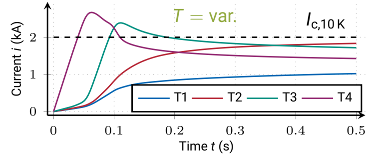
Coil of HTS Tapes

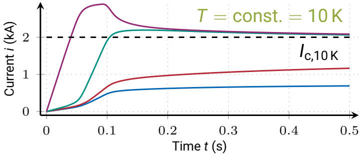
h-a formulation with thermal coupling; tapes in parallel, series or end-coupled

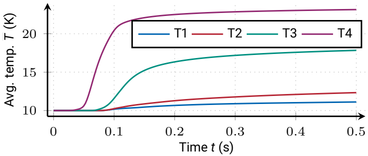


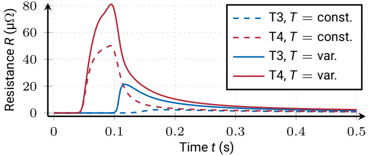


Current redistribution phenomena for current-driven tapes connected in parallel









Good agreement with reference results from COMSOL

[E. Schnaubelt et al. (2021)]

Simple finite element formulations
- The a-v-formulation
- The h-ϕ-formulation

Resolution techniques
- Time integration
- Linearization methods
- Comparison of the formulations

Mixed finite element formulations
- The h(-ϕ)-a-formulation
- The t-a-formulation

Illustrations

Summary

References

Summary

- Overview of finite element formulations for high-temperature superconductors
  - "Simple" formulations: h-ϕ-formulation , a-v-formulation
    - Different numerical behaviors (Newton-Raphson vs. Picard) due to shape of nonlinear constitutive law
    - For pure HTS problems, use h-ϕ-formulation with Newton-Raphson
    - Adaptive time-stepping a must

Summary

- Overview of finite element formulations for high-temperature superconductors
  - “Simple” formulations: h-ϕ-formulation , a-v-formulation
    - Different numerical behaviors (Newton-Raphson vs. Picard) due to shape of nonlinear constitutive law
    - For pure HTS problems, use h-ϕ-formulation with Newton-Raphson
    - Adaptive time-stepping a must
  - Mixed formulations: h(-ϕ)-a-formulation , t-a-formulation
    - Useful for hybrid systems with ferromagnetic materials
    - Should be discretized with caution to ensure well-posedness

Summary

- Overview of finite element formulations for high-temperature superconductors
  - "Simple" formulations: h-ϕ-formulation , a-v-formulation
    - Different numerical behaviors (Newton-Raphson vs. Picard) due to shape of nonlinear constitutive law
    - For pure HTS problems, use h-ϕ-formulation with Newton-Raphson
    - Adaptive time-stepping a must
  - Mixed formulations: $h(\text{-}\phi)$-a-formulation , t-a-formulation
    - Useful for hybrid systems with ferromagnetic materials
    - Should be discretized with caution to ensure well-posedness
- All formulations available in open source Life-HTS toolkit, based on ONELAB, Gmsh and GetDP

Summary

- Overview of finite element formulations for high-temperature superconductors
  - "Simple" formulations: h-ϕ-formulation , a-v-formulation
    - Different numerical behaviors (Newton-Raphson vs. Picard) due to shape of nonlinear constitutive law
    - For pure HTS problems, use h-ϕ-formulation with Newton-Raphson
    - Adaptive time-stepping a must
  - Mixed formulations: $h(\text{-}\phi)$-a-formulation , t-a-formulation
    - Useful for hybrid systems with ferromagnetic materials
    - Should be discretized with caution to ensure well-posedness
- All formulations available in open source Life-HTS toolkit, based on ONELAB, Gmsh and GetDP
  - Several available or finding their way into commercial tools (e.g. COMSOL)

Simple finite element formulations
The a-v-formulation
The h-ϕ-formulation

Resolution techniques
Time integration
Linearization methods
Comparison of the formulations

Mixed finite element formulations
The h(-ϕ)-a-formulation
The t-a-formulation

Illustrations

Summary

References

Main references

- ONELAB website, with codes, examples, and tutorials: https://onelab.info
- Life-HTS website: http://www.life-hts.uliege.be

- *Finite Element Formulations for Systems with High-Temperature Superconductors*,
  J. Dular, C. Geuzaine, and B. Vanderheyden, TAS 30 (2020) 8200113.
- *On the Stability of Mixed Finite-Element Formulations for High-Temperature Superconductors*,
  J. Dular, M. Harutyunyan, L. Bortot, S. Schöps, B. Vanderheyden, and C. Geuzaine, TAS 31 (2021) 8200412
- *What Formulation Should One Choose for Modeling a 3D HTS Motor Pole with Ferromagnetic Materials?*,
  J. Dular, K. Berger, C. Geuzaine, and B. Vanderheyden, IEEE Trans. Mag. (in press)

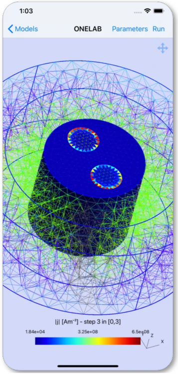
Post-Scriptum

For fun, go to the

- Google Play Store (if you are on Android)
- Apple AppStore (if you are on iOS)

and download the **ONELAB app**: it contains a full-featured version of Gmsh & GetDP

... so you can impress your friends by solving finite element problems with HTS on your smartphone!



Thanks for your attention

✉ cgeuzaine@uliege.be