An introduction to Life-HTS:

Liège University finite element models for High-Temperature Superconductors

Benoît VANDERHEYDEN

Dept. of Electrical Engineering and Computer Science
School of Engineering
Liège Université, Belgium

`B.Vanderheyden@uliege.be`
`people.montefiore.ulg.ac.be/vdh`

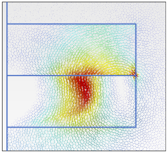
HTS Motors School

LIÈGE université

Outline

Outline

Life-HTS, scope and framework



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 non-linear constitutive laws
- Includes a coupled A-H formulation for combining ferromagnetic and superconducting materials

Liège University

campus

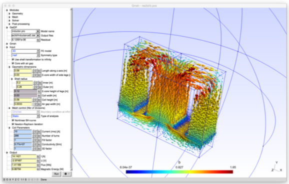
Montefiore Institute

The city of Liège

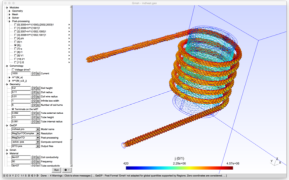
General framework

Under the hood: ONELAB

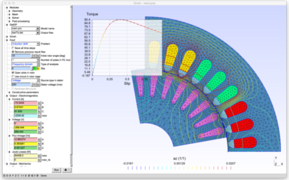
- Open Numerical Engineering LABoratory, see `onelab.info`. ONELAB is the main interface and contains
  - Gmsh, a mesh generator,
  - GetDP; a finite element solver.
- Developed at ULiège by the research group of C. Geuzaine (in collaboration with J.-F. Remacle, UCLouvain, for Gmsh).
- Open-source, available for Windows, macOS, and Linux.



transformer



induction heating



rotating machine

Outline

A simple 1D boundary value problem

- Solve

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

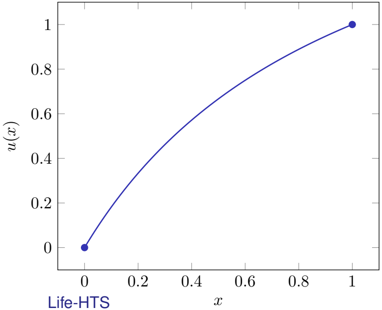
with

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

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

- Solution

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



FEM method, step 1

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

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

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

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

Functions $\phi_\ell(x)$ with $\ell > 0$:

- linearly independent
- satisfy essential[1] boundary conditions
- their superposition spans an approximation space, $\mathcal{H}_m^0$, of dimension m.

[1] as opposed to natural conditions, arising from an integral term in the weak form.

FEM 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 $\mathcal{H}_m^0$:

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

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

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

with $\gamma_0 = 1$.

FEM method, steps 3 and 4

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

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

  where

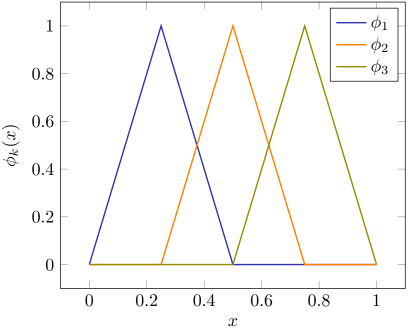
$$a_{k,\ell} = \left(a\, \frac{d\phi_\ell}{dx}, \frac{d\phi_k}{dx} \right) + (b\, \phi_\ell, \phi_k)\,, \quad k = 1, \ldots, m, \quad \ell = 0, \ldots, m.$$

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

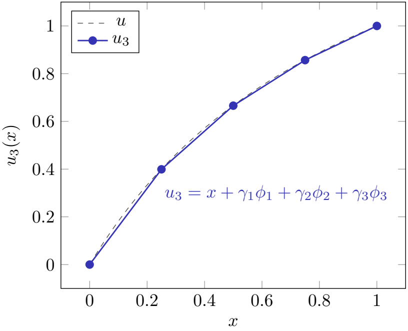
  A sparse system is obtained, which saves computational cost.

Numerical example

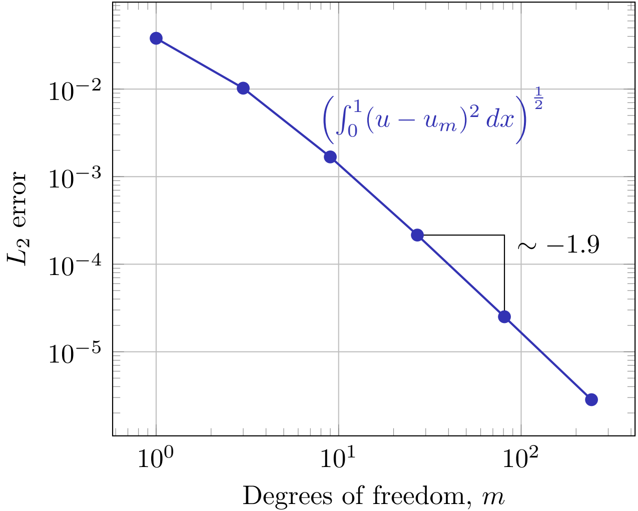
Function space: use nodal functions (here, $m = 3$),



Approximate solution:



Quality of the solution



FEM method: summary

- Need a function space for the approximations u_m,

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

- Solve $(r, \phi_k) = 0, \ldots$
- ...in the weak form, to get the linear system

$$A \cdot x = b,$$

with

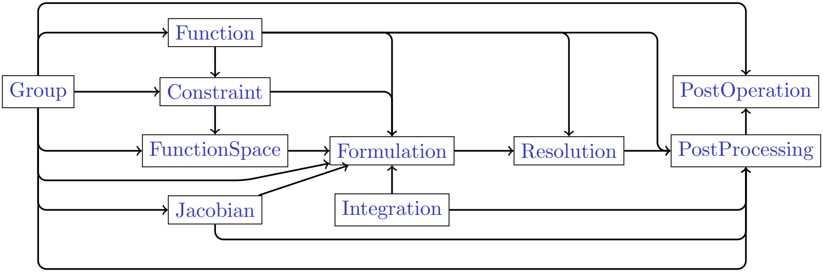
$$A_{k,\ell} = \left(a \frac{d\phi_\ell}{dx}, \frac{d\phi_k}{dx} \right) + (b\phi_\ell, \phi_k), \quad x_\ell = \gamma_\ell, \quad \text{and} \quad b_k = (f, \phi_k).$$

In GetDP, a problem is described by specifying the function space and the weak form equations!

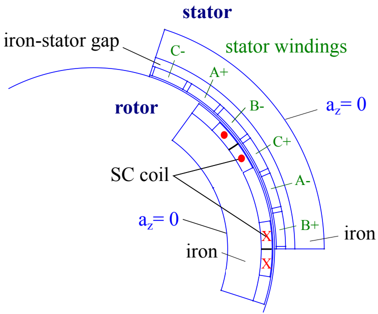
Outline

GetDP: main structure

- GetDP: General environment for the treatment of Discrete Problems
- In practice, a text script describing the problem definition structure.
- Needs an input mesh (e.g., defined with Gmsh)
- Definition structure based on different objects:



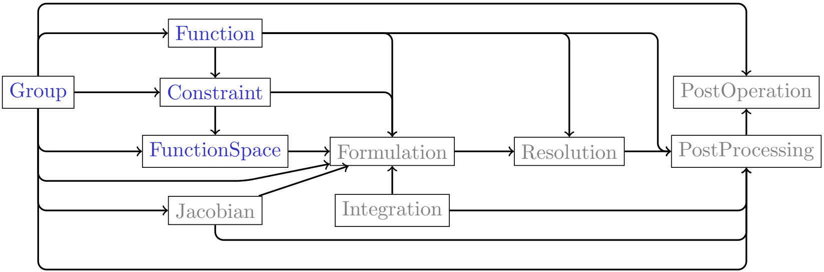
A magnetostatic example: rotating machine



- Rotating machine, 3 MW of power
- Rotor:
  - 5 pairs of poles, each pole is a coil of fixed current density $j_c = 100$ A/mm^2
  - a rotating iron cylinder ($\mu_r = 1000$)
- Stator:
  - distributed copper coils of fixed, three-phase, currents $= 10$ A/mm^2.
  - a fixed iron cylinder ($\mu_r = 1000$)

Goal: compute magnetostatic field for different stator-rotor relative angles

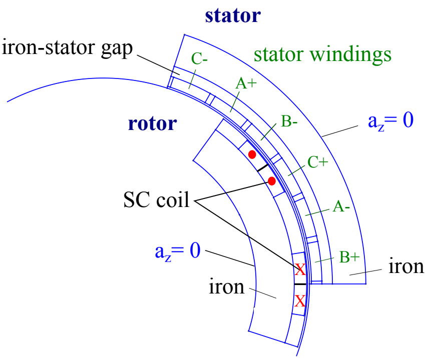
GetDP, objects, 1$^{\text{st}}$ part



Let's walk you through the objects step by step ...

Magnetostatic example: Group

File `motor.pro`



Definition of the domains:

```
Group {
    ...
    Rotor_Iron += Region[ ROTOR_IRON ];
    Stator_Iron += Region[ STATOR_IRON ];
    ...
    B_p = Region[ STATOR_INDUCTOR ];
    A_m = Region[ (STATOR_INDUCTOR+1) ];
    C_p = Region[ (STATOR_INDUCTOR+2) ];
    B_m = Region[ (STATOR_INDUCTOR+3) ];
    A_p = Region[ (STATOR_INDUCTOR+4) ];
    C_m = Region[ (STATOR_INDUCTOR+5) ];
    ...
    SurfOut = Region[ ROTOR_BND_IN ];
    SurfOut += Region[ STATOR_BND_OUT ];
    ...
    OmegaC_stranded += Region[ { A_p, A_m, ...
}
```

Here, uppercase parameters are integers used in the geometry definition to represent subdomains.

Magnetostatic example: Function

File motor.pro

Define constants, simulation parameters, geometry parameters, ...

```
Function{
    ...
    DefineConstant [ec = 1e-4]; //  Critical  electric  field  [V/m]
    DefineConstant [jc = {1e8, Name "Input/4 Material  Properties/2 jc  (A/m2)"}];
      //  Critical  current  density  [A/m2]
    ...
    DefineConstant [convergenceCriterion = 0];
    DefineConstant [tol_energy = 1e-6]; //  Relative  tolerance  on  the  energy  estimates
    DefineConstant [tol_abs = 1e-12]; //Absolute  tolerance  on  nonlinear  residual
    DefineConstant [tol_rel = 1e-6]; //  Relative  tolerance  on  nonlinear  residual
    DefineConstant [tol_incr = 5e-3]; //  Relative  tolerance  on  the  solution  increment
    ...
    //  Rotation  parameters/constants
    rpm = 60*f/p; //  Turn  per  minute
    omega = 2*Pi*rpm/60; //  Rotation  speed  (rad/s)
    ...
}
```

FunctionSpace and Constraint

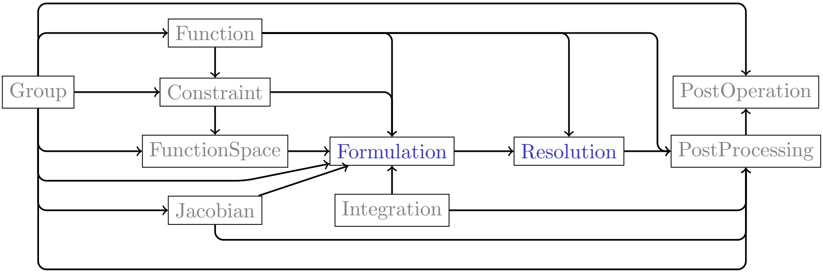
- In `motor.pro`, define a Dirichlet condition on the exterior boundary `SurfOut`:

```
Constraint {
    { Name a ;
        Case {
            {Region SurfOut ; Value 0 ;}
    ...
}
```

- In `formulations.pro`, define the approximation space for the vector potential $\mathbf{a} = (0, 0, a_z(x, y))$ (node-based):

```
FunctionSpace {
     ...
    //   1: In 2D with in-plane b
    //       a = sum a_n * psi_n        (nodes in Omega_a)
     { Name a_space_2D; Type Form1P;
        BasisFunction {
            { Name psin; NameOfCoef an; Function BF_PerpendicularEdge;
                Support Omega_a; Entity NodesOf[All]; }
        }
     Constraint {
            { NameOfCoef an; EntityType NodesOf; NameOfConstraint a; }
}}}
```

GetDP, objects, 2^{nd} part



Magnetostatic example: equations

- Start with $\mathbf{b} = \mathbf{curl}\ \mathbf{a}$ and $\mathbf{curl}\ \mathbf{h} = \mathbf{j}$ with $\mathbf{h} = (1/\mu)\,\mathbf{b}$.
- The permeability μ is defined piece-wise: $\mu = \mu_0$ in SC, $\mu > \mu_0$ in FM, ...
- Eliminating **h** and **b**, we have

$$\mathbf{curl}\ (\nu\ \mathbf{curl}\ \mathbf{a}) = \mathbf{j}, \quad \text{with} \quad \nu = 1/\mu.$$

- Project LHS-RHS on the test functions $\mathbf{a}_k'$:

$$\left(\mathbf{curl}\ (\nu\ \mathbf{curl}\ \mathbf{a}), \mathbf{a}_k'\right)_{\Omega} - \left(\mathbf{j}, \mathbf{a}_k'\right)_{\Omega_c} = 0$$

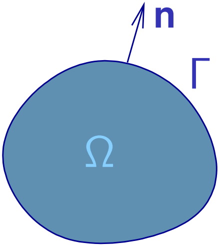
Magnetostatic example, weak formulation

- Integrate by parts, using the equality

$$(\mathbf{curl}\,\mathbf{u},\mathbf{v})_\Omega = \langle \mathbf{n}\times\mathbf{u},\mathbf{v}\rangle_\Gamma + (\mathbf{u},\mathbf{curl}\,\mathbf{v})_\Omega\,,$$

where

$$(\mathbf{u},\mathbf{v})_\Omega = \int_\Omega \mathbf{u}\cdot\mathbf{v}\,d\Omega,\quad \langle\mathbf{u},\mathbf{v}\rangle_\Gamma = \int_\Gamma \mathbf{u}\cdot\mathbf{v}\,d\Gamma,$$



to get

$$\left(\nu\,\mathbf{curl}\,\mathbf{a},\mathbf{curl}\,\mathbf{a}_k'\right)_\Omega + \left\langle \mathbf{n}\times(\nu\,\mathbf{curl}\,\mathbf{a}),\mathbf{a}_k'\right\rangle_\Gamma = \left(\mathbf{j},\mathbf{a}_k'\right)_{\Omega_c}.$$

Here the surface term $\left\langle \mathbf{n}\times(\nu\,\mathbf{curl}\,\mathbf{a}),\mathbf{a}_k'\right\rangle_\Gamma$ vanishes due to boundary conditions.

Magnetostatic example: Formulation

File `formulation.pro`

The formulation equations are a direct transcription of the weak formulation. Here, the actual formulation, simplified for a linear ferromagnetic law and a static problem:

```
Formulation {
  { Name MagDyn_avtot; Type FemEquation;
    Quantity {
      { Name a; Type Local; NameOfSpace a_space_2D; }
    }
    Equation {
    Galerkin { [ nu[] * Dof{d a} , {d a} ];
               In MagnLinDomain; Integration Int; Jacobian Vol; }
    Galerkin { [ -js[] , {a} ];
               In OmegaC_stranded; Integration Int; Jacobian Vol;  }
}}}
```

Notes:

- *d* is an exterior derivative, here the curl operator.
- `Dof{a}` denotes an unknown (it goes in $\mathbf{x}$, in $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$).
- Each Galerkin term must either be linear w.r.t. `Dof{}` (bilinear term, LHS $\mathbf{A} \cdot \mathbf{x}$) or not involve `Dof{}` (linear term, RHS $\mathbf{b}$). Galerkin terms are summed, with a sum implicitly set to 0.

Magnetostatic example: Resolution

File `resolution.pro`

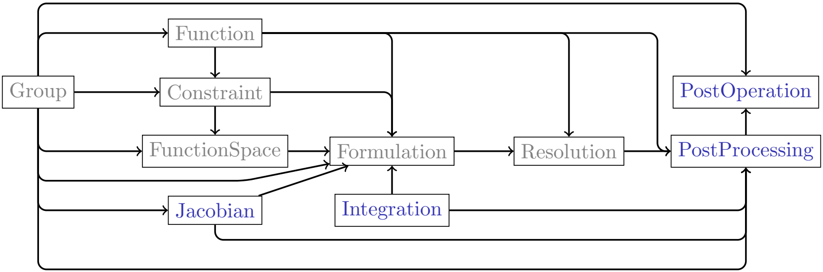
The place where you write the operations to be performed.
The simplest one:

```
Resolution {
  { Name MagSta_a;
    System {
      { Name A; NameOfFormulation MagDyn_avtot; }
    }
    Operation {
      Generate[A]; Solve[A]; SaveSolution[A];
    }
  }
}
```

Otherwise, specify

- the non-linear iteration scheme
- the time-stepping strategy

GetDP, objects, 3rd part



Magnetostatic example: Jacobian

File: `formulation.pro`

Jacobian: associated with the geometry (2D, axisymmetric, . . .)

```
Jacobian {
    // For volume integration (Dim N)
    { Name Vol ;
        Case {
            If(Axisymmetry == 0)
                // Classical transformation Jacobian
                {Region All ; Jacobian Vol ;}
            Else
                // Axisymmetric problems
                ...
                //  Second-type, better suited to PerpendicularEdge basis functions
                {Region Omega_a ; Jacobian VolAxiSqu ;}
            EndIf
        }
    }
}
```

Magnetostatic example: Integration

File: `formulation.pro`

Integration: specifies the type of integration (here, Gauss quadrature) and the number of points

```
Integration {
    { Name Int ;
        Case {
            { Type Gauss ;
                Case {
                { GeoElement Point ; NumberOfPoints 1 ; }
                { GeoElement Line ; NumberOfPoints 3 ; }
                { GeoElement Triangle ; NumberOfPoints 3 ; }
                ...
                }
            }
        }
    }
}
```

Magnetostatic example: PostProcessing / PostOperation

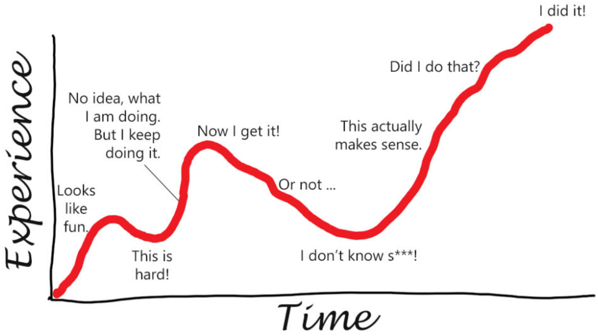
Files `formulations.pro` and `motors.pro`

```
PostProcessing {
   { Name MagDyn_avtot; NameOfFormulation MagDyn_avtot;
        Quantity {
            { Name a; Value{ Local{ [ {a} ] ;
                In Omega; Jacobian Vol; } } }
            { Name az; Value{ Local{ [ CompZ[{a}] ] ;
                In Omega; Jacobian Vol; } } }
            { Name b; Value{ Local{ [ {d a} ] ;
                In Omega; Jacobian Vol; } } }
            ...
}}}
```

```
PostOperation {
 { Name MagDyn;
            NameOfPostProcessing MagDyn_avtot;
 Operation {
    Print[ az, OnElementsOf Omega , File "res/a.pos", Name "a [Tm]", LastTimeStepOnly
   onelabInterface];
     ...
}}}
```

Postoperations can be performed after the resolution (to analyse results), or during the resolution (when auxiliary quantities are needed).

Learning curve



Check the tutorials and the numerous examples on `onelab.info`!

Magnetostatic example: demo!

Magnetostatic example: tips and tricks

A few tips on the syntax for fields:

- `Dof`$\{a\}$: indicates that the field $\{a\}$ is an unknown (i.e. is the vector x in $A \cdot x = b$)
- $\{a\}$: the last computed value of $\{a\}$
- $\{d\ a\}$: the exterior derivative of $\{a\}$,
  - a 0-form field $\{a\}$, or scalar field ("continuous across nodes"), gives $\{d\ a\} = \{\texttt{Grad}\ a\}$, a 1-form vector field ("continuous across edges");
  - a 1-form field $\{a\}$ gives $\{d\ a\} = \{\texttt{Curl}\ a\}$, a 2-form vector field ("continuous across facets")

Outline

Outline

Back to Life

- Life-HTS was developed by J. Dular, C. Geuzaine, and B. Vanderheyden
- Life-HTS is about solving Maxwell's equations in the magnetodynamic approximation,

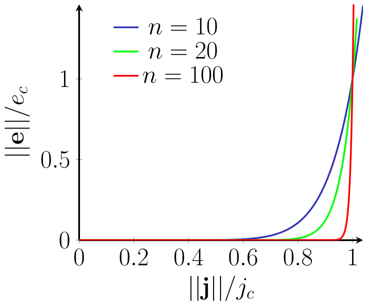
$$\mathbf{div}\,\mathbf{b} = 0 \quad \mathbf{curl}\,\mathbf{h} = \mathbf{j} \quad \mathbf{curl}\,\mathbf{e} = -\partial_t \mathbf{b},$$

  with
  b, the magnetic flux density (T),
  h, the magnetic field (A/m),
  j, the current density (A/m^2)
  e, the electric field,
  while the displacement current $\partial \mathbf{d}/\partial t$ is ignored.
- Need constitutive relationships relating **b** to **h** and **e** to **j**.

Constitutive laws

1. High-temperature superconductors (SC):

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



where the electrical resistivity is given as

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

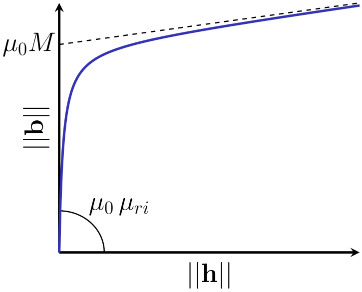
with $e_c = 10^{-4}$ V/m,
j_c, the critical current density,
n, the flux creep exponent,
$n \in [10, 1000]$.

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, cont'd

2. Ferromagnetic materials (FM): a non-linear, but anhysteretic law:

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

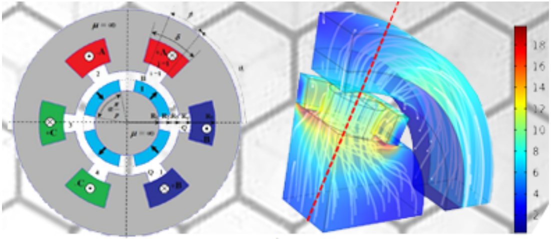


Typical values (supra50):

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

Eddy currents are neglected.

Constitutive laws, extensions



One can also consider

- conductors and coils,
- permanent magnets,
- hysteretic ferromagnetic materials,
- type-I superconductors (need a London length).

Formulations

Two classes:

- *a*-formulation, which is b-conform,
  - enforces the continuity of the normal component of **b**,
  - much used in electric rotating machine design
- *h*-formulation, which is h-conform,
  - enforces the continuity of the tangential component of **h**,
  - much used for superconducting materials.

These formulations involve the constitutive laws in opposite ways,
$\Longrightarrow$ very different numerical behaviors!

a-formulation (or A-v formulation)

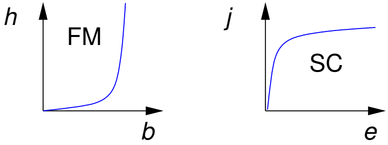
- Introduce the vector potential **a** and electric potential *v*:

$$\mathbf{b} = \mathbf{curl}\,\mathbf{a} \quad \text{and} \quad \mathbf{e} = -\partial_t\,\mathbf{a} - \mathbf{grad}\,v.$$

  This guarantees $\mathbf{div}\,\mathbf{b} = 0$ and $\mathbf{curl}\,\mathbf{e} = -\partial_t\,\mathbf{b}$.
- There remains to solve $\mathbf{curl}\,\mathbf{h} = \mathbf{j} = \sigma\,\mathbf{e}$,

$$\Rightarrow \quad \mathbf{curl}\,(\nu\,\mathbf{curl}\,\mathbf{a}) = -\sigma\,(\partial_t\,\mathbf{a} + \mathbf{grad}\,v),$$

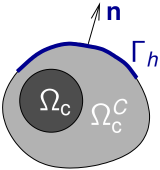
  where $\nu = 1/\mu$ and $\sigma = 1/\rho$ are defined region-wise.



a-formulation in practice

- Function space

$$\mathbf{a} = \sum_{e \in \Omega} a_e\, \psi_e \quad \text{and} \quad v = \sum_{i \in C} V_i v_i,$$



Ω_c: conductors
Γ_h: where $\mathbf{h} \times \mathbf{n}$ is fixed

Here, ψ_e are edge functions and v_i are source potential functions, while $\mathbf{a}$ is gauged in Ω_c^C.

- Weak form

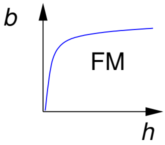
$$\begin{aligned} \left(\nu\, \mathbf{curl}\, \mathbf{a}, \mathbf{curl}\, \mathbf{a}'\right)_{\Omega} - \left\langle \mathbf{h} \times \mathbf{n}, \mathbf{a}' \right\rangle_{\Gamma_h} + \left(\sigma\, \partial_t \mathbf{a}, \mathbf{a}'\right)_{\Omega_c} & \\ + \left(\sigma\, \mathbf{grad}\, v, \mathbf{a}'\right)_{\Omega_c} &= 0, \\ \left(\sigma\, \partial_t\, \mathbf{a}, \mathbf{grad}\, v'\right)_{\Omega_c} + \left(\sigma\, \mathbf{grad}\, v, \mathbf{grad}\, v'\right)_{\Omega_c} - \sum_{i \in C} I_i \mathcal{V}_i(v') &= 0. \end{aligned}$$

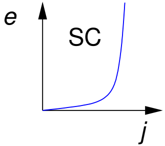
h-formulation

- In the non-conducting domain, we have $\mathbf{curl\ h} = \mathbf{0}$ (no current!). Thus, introduce the scalar magnetic potential ϕ such that $\mathbf{h} = -\mathbf{grad}\ \phi$.
- Need to solve $\mathbf{curl\ e} = -\partial_t\ \mathbf{b}$, together with $\mathbf{curl\ h} = \mathbf{j}$:

$$\mathbf{curl}\ \left(\rho\ \mathbf{curl\ h}\right) = -\partial_t\ \left(\mu \mathbf{h}\right),$$

  where μ and ρ are defined regionwise.
- Side note: $\mathbf{div\ b} = 0, \forall t$, if it does for $t = 0$, as $\mathbf{curl\ e} = -\partial_t\ \mathbf{b}$.

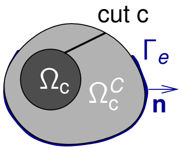




h-formulation in practice

- Function space

$$\mathbf{h} = \sum_{e \in \Omega_c} \mathbf{h}_e \, \psi_e + \sum_{n \in \Omega_c^C} \phi_n \, \mathbf{grad} \psi_n + \sum_{i \in C} I_i \, \mathbf{c}_i.$$



Ω_c: conductors
Γ_e: where $\mathbf{e} \times \mathbf{n}$ is fixed

where ψ_e, ψ_n, and $\mathbf{c}_i$ are edge, nodal, and cut functions.

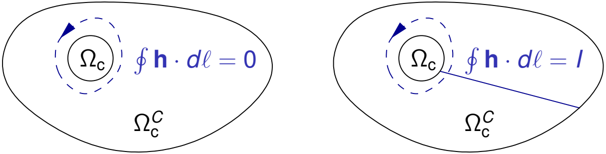
- Weak form

$$\left(\partial_t(\mu(\mathbf{h})\, \mathbf{h}), \mathbf{h}'\right)_{\Omega} + \left(\rho(\mathbf{curl}\, \mathbf{h})\, \mathbf{curl}\, \mathbf{h}, \mathbf{curl}\, \mathbf{h}'\right)_{\Omega_c} - \left\langle \mathbf{e} \times \mathbf{n}, \mathbf{h}' \right\rangle_{\Gamma_e} + \sum_{i \in C} V_i \mathcal{I}_i(\mathbf{h}') = 0$$

Why cuts?

- In Ω_c^C, we have $\mathbf{h} = -\textbf{grad}\ \phi$.
- Hence, along a closed contour around a conductor Ω_c carrying a current I,

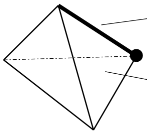
$$\oint_{\mathcal{C}} \mathbf{h} \cdot d\ell = 0 \neq I \quad !?$$



- This is an example of a multiply connected non-conductive domains. Cuts are introduced to obtain simply connected domains [1]; also, ϕ is made discontinuous across cuts.
- Essential conditions: currents are introduced through cut functions.

A. Bossavit, PIEEAPS 135 (1998) 179

A side note on elements



Nodal functions

$$\phi = \sum_{j=1}^{N} p_j \, \phi_j .$$

- p_j is the value of ϕ on node j.
- ϕ is continuous accross elements.

Edge functions

$$\mathbf{a} = \sum_{i=1}^{M} a_i \, \mathbf{a}_i .$$

- a_i is the line integral of $\mathbf{A}$ along the edge i, i.e. $a_i = \int_{\text{edge i}} d\ell \cdot \mathbf{a}$.
- The tangential component of $\mathbf{a}$ is continuous accross elements.

The structure of Life

- Based on a time-varying and non-linear weak formulations,

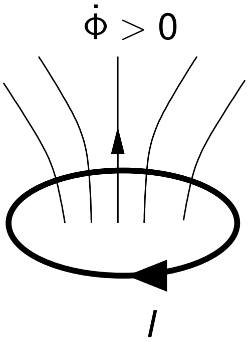
$$\mathbf{A}(\mathbf{x}, t) \cdot \mathbf{x} = \mathbf{b}(t),$$

where $\mathbf{x} = (\mathbf{a}, v)$ or $x = (\mathbf{h}, \phi)$.

- Structure: two imbricated loops,
  1. time-stepping, with adaptative time steps,
  2. iterative solution of the non-linear weak formulation.

Outline

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.

Circuit equation:

$$\dot{\Phi} = R(|I|)\, I + L\dot{I}, \tag{1}$$

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 $\dot{I} \approx (I_j - I_{j-1})/\Delta t$,

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

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

$$b = |x|^{n-1}\, x + x, \quad (I\text{-form}), \tag{2}$$

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 $V_j = RI_j$,

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

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

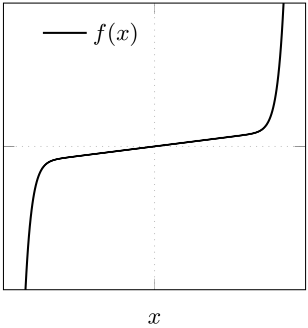
$$d = |x|^{1/n-1}\,x + x, \quad (V\text{-form}), \tag{3}$$

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

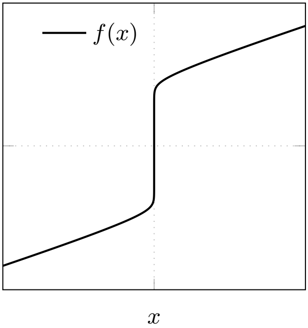
In each case, need to solve an equation of the form $f(x) = \text{Constant}$:



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

I-form

($\sim$ h-conform: Ampere)



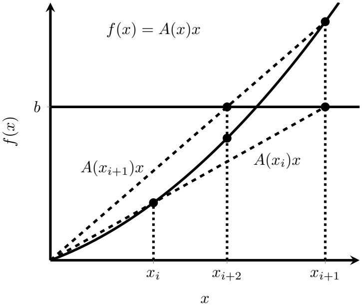
$f(x) = |x|^{1/n-1}\,x + x$

V-form

($\sim$ b-conform: Faraday-Lenz)

Solving a non-linear equation:

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

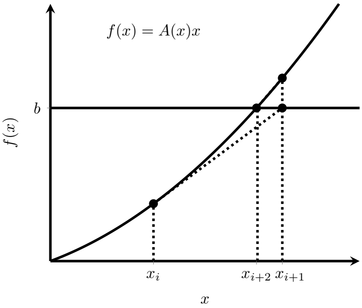


- Write $f(x)$ as $f(x) = A(x)x$.
- Get a first estimate x_0.
- At each iteration i:
  - solve $A(x_i)x = b$;
  - $x_{i+1} := x$,
  - $i := i + 1$ and loop.

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

Solving a non-linear equation: 2) Newton-Raphson method

2. Newton-Raphson iterative method:

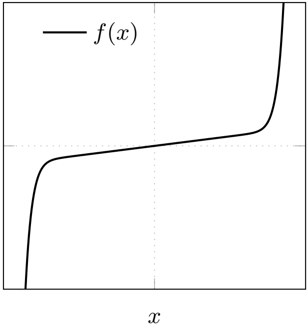


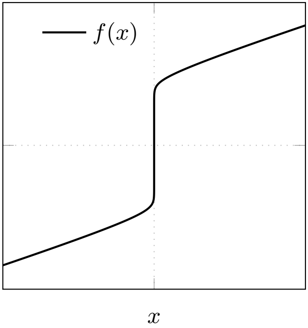
- Get a first estimate x_0.
- At each iteration i,

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

- Quadratic convergence, if the initial estimate x_0 is close enough.

A second look at the functions f





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

V-form (b-conform)

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

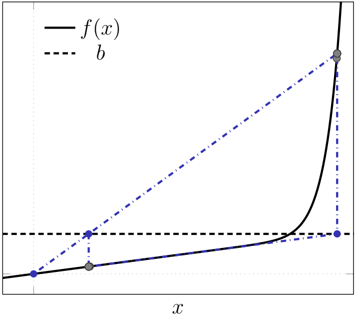
I-form (h-conform)

Which of Picard or Newton-Raphson should one use in each case?

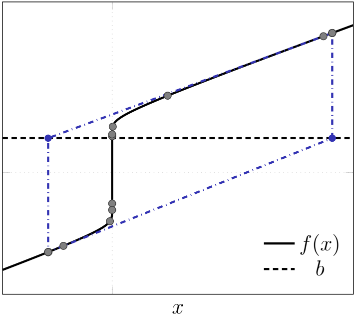
Warning!

Beware of cycles!

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



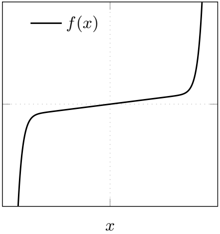
Picard iteration on an I-form.
Prefer Newton-Raphson!



N-R iteration on a V-form.
Prefer Picard!

Superconducting ring, conclusions

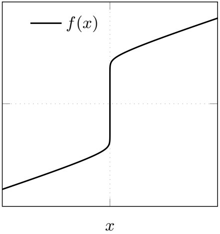
- At each time step, need to solve for a non-linear equation of the form $f(x) = b$.



$$f(x) = x^n + x$$

I-form (h-conform)

- Solve with Newton-Raphson
- Up to a quadratic convergence

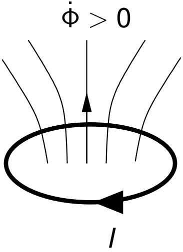


$$f(x) = x^{1/n} + x$$

V-form (b-conform)

- Solve with Picard
- Slow convergence

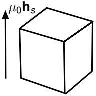
Superconducting ring: more comments



- Conclusions can be generalized to 1D, 2D, and 3D geometries:
  - I-form $\mapsto$ h-conform formulations, use Newton-Raphson;
  - V-form $\mapsto$ b-conform formulations, use Picard.
- When cycles occur, use relaxation methods?
  Maybe, but we found no systematic stable scheme.

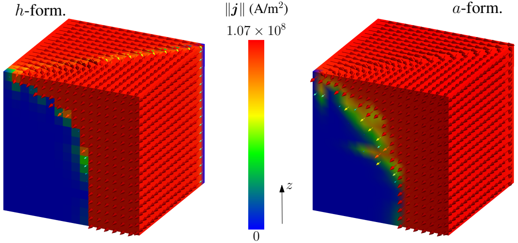
Illustration for a superconducting cube

System



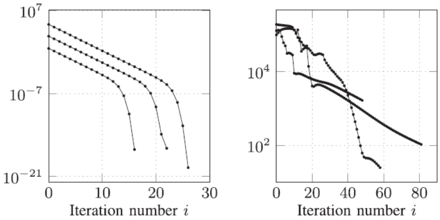
Side $a = 10$ mm.
$\mu_0 \mathbf{h}_s = \hat{z}\, B_0 \sin(2\pi f t)$,
with $B_0 = 200$ mT,
$f = 50$ Hz,
$j_c = 10^8$ A/m^2 and
$n = 100$.

Current density distribution



Residual

- L_2 norm of $\mathbf{r} = \mathbf{A} \cdot \mathbf{x} - \mathbf{b}$
- Left: *h*-formulation
- Right: *a*-formulation

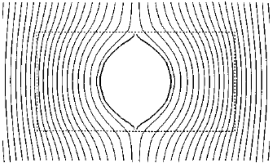


(a) Newton-Raphson technique. (b) Picard technique.

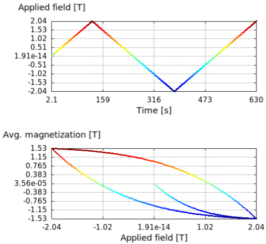
Demonstration

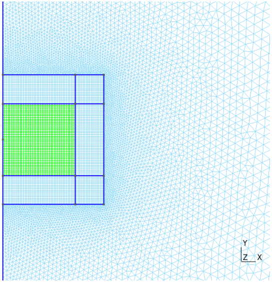
Magnetization of a superconducting pellet: phenomenology

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

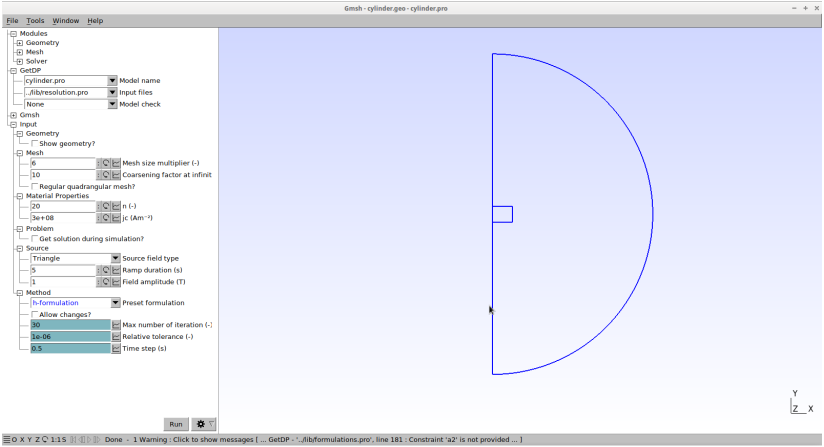




$\rightarrow$ movie case2.mpg

Demonstration

Magnetization of a superconducting pellet: *h*- and *a*- formulations

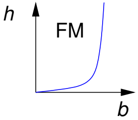


How about magnetic laws?

Conclusions on the non-linearities also apply to the magnetic constitutive laws:

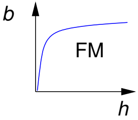
- For an *a*-formulation:
  - the term $(\nu\, \mathbf{curl}\, \mathbf{a}, \mathbf{curl}\, \mathbf{a}')_\Omega$ involves $\nu\, \mathbf{curl}\, \mathbf{a}$, or **h** as a function of **b**

  $\Rightarrow$ use Newton-Raphson

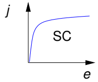
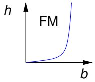


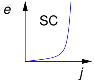
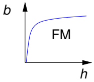
- For an *h*-formulation:
  - the term $(\partial_t(\mu\, \mathbf{h}, \mathbf{h}'))_\Omega$ involves $\mu\, \mathbf{h}$, or **b** as a function of **h**

  $\Rightarrow$ use Picard



Non-linearities: take-home message

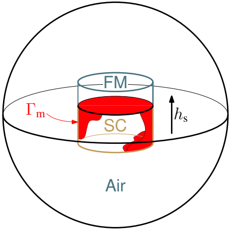
a	SC	FM
Laws		
Method	Picard	Newton

h	SC	FM
Laws		
Method	Newton	Picard

Coupled formulation

Combine the most efficient formulations, i.e.,

- for SC: *h*-formulation with Newton-Raphson,
- similarly, for FM: *a*-formulation with Newton-Raphson



Need a coupling through the boundary Γ_m:

$$\left(\partial_t(\mu\,\mathbf{h}), \mathbf{h}'\right)_{\Omega_{SC}} + \left(\rho\,\mathbf{curl}\,\mathbf{h}, \mathbf{curl}\,\mathbf{h}'\right)_{\Omega_{SC}} + \left\langle \partial_t\,\mathbf{a} \times \mathbf{n}_{\Omega_{SC}}, \mathbf{h}' \right\rangle_{\Gamma_m} = 0,$$

$$\left(\nu\,\mathbf{curl}\,\mathbf{a}, \mathbf{curl}\,\mathbf{a}'\right)_{\Omega_{FM+air}} - \left\langle \mathbf{h} \times \mathbf{n}_{\Omega_{FM+air}}, \mathbf{a}' \right\rangle_{\Gamma_m} = 0.$$

Coupled formulation: efficiency

From TAS 30 (2020) 8200113

Example: SC cylinder of 12.5 mm, 5 mm height, $j_c = 3 \times 10^8$ A/m^2, $n = 20$, FM: supra50, cylinder of same size, 5 T applied in ZFC and reduced to 0 at 25 mT/s.

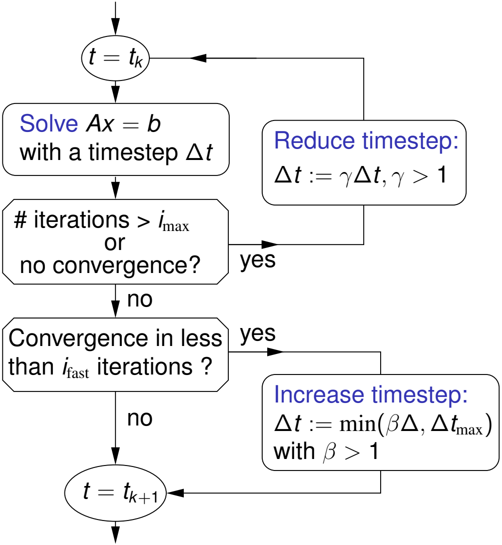
Formulation	Total h	Total a	Coupled
Linearization SC	NR	Pi	NR
Linearization FM	NR	NR	NR
Extrapolation	1$^{\text{st}}$	2$^{\text{nd}}$	1$^{\text{st}}$
Coarse	3268	4381	1071
Medium	4083	7539	1931
Fine	4422	14594	3753

Mesh	Coarse	Medium	Fine
Dofs	840	2800	10800
Δt (sec)	4	2	1

$\rightarrow$ For the fine mesh, a speedup factor of $\sim$ 1.2!

Outline

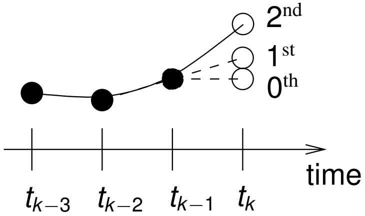
Adaptative time-stepping



Parameters:

- $\gamma = 1/2$
- $\beta = 2$
- $i_{\text{fast}} = i_{\max}/4$
- h-formulation: $i_{\max} = 500$
- a-formulation: $i_{\max} = 60$

Choosing the first iterate

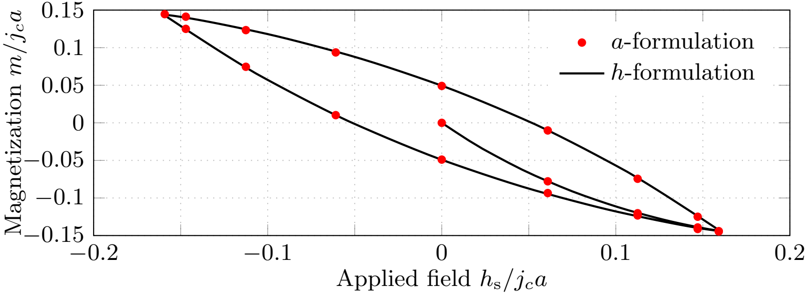


Best results:

- 1st order for the *h*-formulation,
- 2nd order for the *a*-formulation.

Outline

Single time step



- For large values of n, can reduce the # of time steps in the a-formulation.
- Here, for a magnetization cycle
  - lines: h-formulation with 300 time steps
  - dots: a-formulation with 20 time steps
- In practice, accurate for **j** and **b**, but **e** is underestimated!

Outline

Convergence criterion

- The residual is sometimes misleading (specifically with coupled formulation).
- Instead, we opted for monitoring the electromagnetic power, P:
  - *h*-formulation:

$$P = (\partial_t(\mu\, \mathbf{h}), \mathbf{h})_\Omega + (\rho\, \mathbf{j}, \mathbf{j})_{\Omega_c}\,,$$

    with $\mathbf{j} = \mathbf{curl}\ \mathbf{h}$;
  - *a*-formulation:

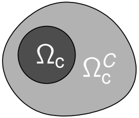
$$P = (\partial_t\, \mathbf{b}, \nu\, \mathbf{b})_\Omega + (\sigma\, \mathbf{e}, \mathbf{e})_{\Omega_c}\,,$$

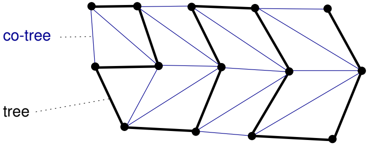
    with $\mathbf{b} = \mathbf{curl}\ \mathbf{a}$ and $\mathbf{e} = -\partial_t\, \mathbf{a} - \mathbf{grad}\ v$.
- In practice, stop when $|\Delta P/P|$ is small enough.

Gauging the vector potential (spanning tree technique)

Required in 3D, or 2D with in-plane currents

$$\mathbf{a} = \sum_i a_i \, \psi_i$$





For simplicity, assume no potential source v:

- in the conducting region Ω_c, $\mathbf{a}$ is unique as $\mathbf{e} = -\partial_t \, \mathbf{a}$, with $\mathbf{e} = \mathbf{j}/\sigma$;
- in the non-conducting region Ω_c^C, $\mathbf{a}$ is not unique, $\rightarrow$ gauge:
  - Number of variables a_i = number of edges, too many!
  - Number of independent degrees of freedom = number of facets (one element of magnetic flux per facet).

Hence,

1. construct a suitable tree in the mesh,
2. impose $a_i = 0$ on the edges of this tree.

Newton-Raphson method for isotropic constitutive laws relating vector quantities

- Consider a constitutive law of the form

$$\mathbf{a}(\mathbf{x}) \quad = \quad g(||\mathbf{x}||)\,\mathbf{x}.$$

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

- To iterate with the Newton-Raphson method, linearize:

$$a_i(\mathbf{x}^{j+1}) \approx a_i(\mathbf{x}^j) + \sum_{k=1,3} \left(x_k^{j+1} - x_k^j\right) \frac{\partial a_i}{\partial x_k^j},$$

  where j is the iteration index.

- This expansion can be cast in the form

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

  where $\mathbf{J}$ is the 3×3 Jacobian matrix.

Generic form of the Jacobian matrix

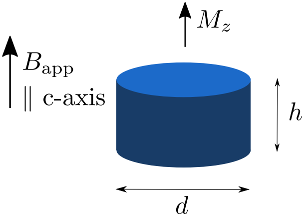
- Carrying out the partial derivatives, one easily gets

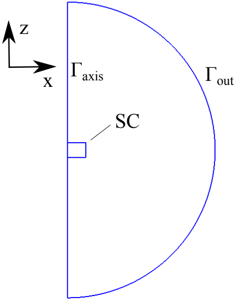
$$\mathbf{J}_{ik} = \frac{\partial a_i}{\partial x_k} = \delta_{ik}\, g(||\mathbf{x}^j||) + x_i x_k\, \frac{\partial g(||\mathbf{x}^j||)/\partial ||\mathbf{x}^j||}{||\mathbf{x}^j||}.$$

- A few examples can be found in the appendix of TAS 30 (2020) 8200113,
  - $\mathbf{e} = \rho\, \mathbf{j}$ and $\mathbf{j} = \sigma\, \mathbf{e}$ with a power law,
  - $\mathbf{b} = \mu\, \mathbf{h}$ and $\mathbf{h} = \nu\, \mathbf{b}$ with a non-linear magnetic law.
- Useful trick: sometimes, it is better to consider g as a function of $||\mathbf{x}||^2$, as this provides an additional power of $||\mathbf{x}||$ and avoids a divergence as $||\mathbf{x}|| \to 0$.

Outline

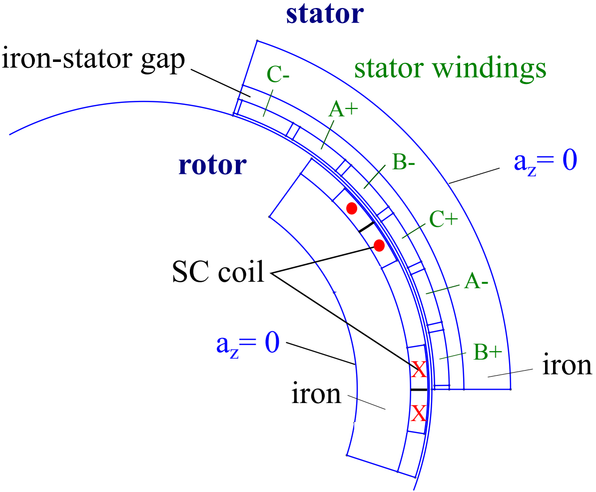
Magnetization of a superconducting pellet





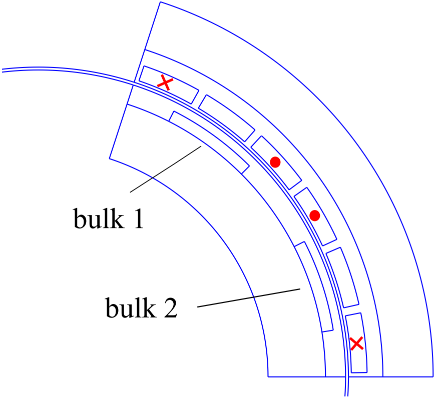
A demonstration of the *h*- and *a*-formulations in a problem with induced currents. Possibility to explore the method of large time steps in the *a*-formulation.

Magnetostatics in an electrical rotating machine



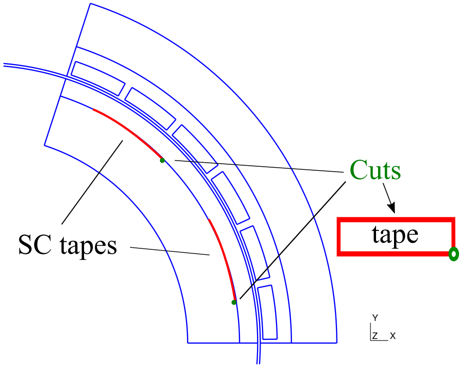
Magnetostatic resolution of the combined field for a rotor with an SC coil (imposed currents) and stator copper windings.

Rotor with trapped field magnets, magnetization with the stator winding



Example of a coupled formulation for an induced current problem.

Coupled formulation with transport currents



Example of a coupled formulation for a transport current problem.

Outline

References

- Onelab website, with codes, examples, and tutorials: `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.
- *Modélisation du champ magnétique et des courants induits dans des systèmes tridimensionnels non linéaires*,
  P. Dular, thesis (1996) U. Liège.
- *High order hybrid finite element schemes for Maxwell's equations taking thin structures and global quantities into account*,
  C. Geuzaine, thesis (2001) U. Liège.
- *The FEM method for electromagnetic modeling*,
  G. Meunier ed., Wiley, 2008.

That's Life!

Contacts: {b.vanderheyden,julien.dular,cgeuzaine}@uliege.be