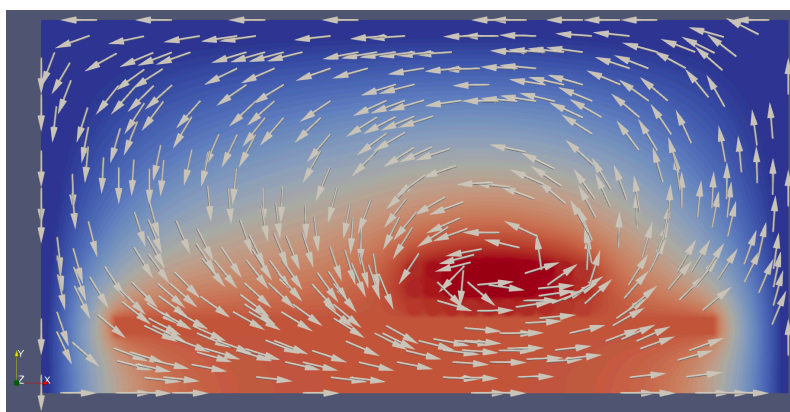


HANA Numerik SEP Projekt

Simon Stingelin (stiw@zhaw.ch)

13. April 2025



Gruppe: Samuel Dobler, Alexander Grunder, Pascal Heiniger

Thema: Induktive Erwärmung

Lerninhalt:

- Modellieren der induktiven Erwärmung.
- Kennenlernen der Maxwell-Gleichung für das Wirbelstrom Problem.
- Berechnen der schwachen Gleichungen.
- Anwenden der Methode der finiten Elemente auf ein gekoppeltes Problem.

Abgabe:

- Die Abgabe erfolgt als Kurzbericht in Form eines **PDF**-Dokuments über die moodle Abgabe bis zum in der moodle Abgabe definierten spätesten Abgabepunkt.
- Die numerischen Resultate sind im Bericht dokumentiert und können mit lauffähigen Skripts nachvollzogen werden.

Bewertung:

Note 1	Note 3	Note 4	Note 4.5	Note 5	Note 5.5	Note 6
Keine Abgabe	Nicht bewertbar: die Aufgaben wurden nicht ernsthaft bearbeitet.	Die Aufgaben wurden knapp bearbeitet und sind nur im Ansatz dokumentiert.	Die Aufgaben wurden halbwegs bearbeitet und sind sehr knapp Ansatz dokumentiert.	Die Aufgaben wurden bearbeitet und dokumentiert.	Die Aufgaben wurden detailliert bearbeitet und dokumentiert.	Die Aufgaben wurden detailliert bearbeitet und dokumentiert. Es wurden noch zusätzliche Aspekte bearbeitet.

1 Aufgabestellung

Das elektromagnetische Feld kann mit Hilfe der Maxwell Gleichungen beschrieben werden. Für die niederfrequente Anwendung wird die harmonische Wirbelstromgleichung

$$\sigma j\omega \mathbf{A} + \text{curl} \mu^{-1} \text{curl} \mathbf{A} + \sigma \nabla \varphi = \mathbf{j}_{\text{ext}} \quad \text{für } x \in \Omega_{\sigma>0} \quad (1a)$$

$$\text{div}(\sigma j\omega \mathbf{A} + \sigma \nabla \varphi) = \text{div}(\mathbf{j}_{\text{ext}}) \quad \text{für } x \in \Omega_{\sigma>0} \quad (1b)$$

$$\text{curl} \mu^{-1} \text{curl} \mathbf{A} = \mathbf{j}_{\text{ext}} \quad \text{für } x \in \Omega_{\sigma=0} \quad (1c)$$

benutzt, wobei mit \mathbf{A} das Vektorpotential und mit φ das skalare Potential bezeichnet werden. Es gilt

$$\mathbf{B} = \text{curl} \mathbf{A}$$

und

$$\mathbf{E} = -\sigma j\omega \mathbf{A} - \nabla \varphi.$$

Die zeitabhängige Lösung ist gegeben durch

$$\mathbf{A}(t, x) = \mathbf{A}(x) \cdot e^{j\omega t}$$

mit $\omega = 2\pi f$ und $j = \sqrt{-1}$.

Die Stromdichte \mathbf{J} kann in zwei Komponenten aufgeteilt werden. Es gilt

$$\mathbf{J} = \mathbf{J}_{\text{int}} + \mathbf{J}_{\text{ext}}.$$

mit der induzierten Stromdichte \mathbf{J}_{int} (Wirbelströme) und der externen Stromdichte \mathbf{J}_{ext} . Die externe Stromdichte ist zum Beispiel gegeben durch eine Stromquelle, mit welcher die Spulen getrieben werden.

Für dreidimensionale Geometrien ist es in der Regel nicht möglich sämtliche Windungen im Mesh aufzulösen. Unter Vernachlässigung von Proximity- und Skineffekt in den Windungen folgt mit $0 < \kappa < 1$ das regularisierte Wirbelstromproblem

$$\sigma j\omega \mathbf{A} + \text{curl} \mu^{-1} \text{curl} \mathbf{A} + \kappa \mathbf{A} = \mathbf{j}_{\text{ext}} \quad \text{für } x \in \Omega. \quad (2)$$

Für den Fall, dass eine

Stromquelle gegeben ist gilt

$$\mathbf{J}_{\text{ext}} = \frac{N_c i_c}{A_c} \mathbf{w}(\mathbf{x}),$$

mit Anzahl Windungen N_c , Strom in den Windungen i_c und der Querschnittsfläche A_c .

Spannungsquelle gegeben ist, folgt aus dem Induktionsgesetz die Kopplung zwischen Strom und Spannung in den Windungen. Es gilt

$$j\omega \Psi + R_c i_c = u_c.$$

Damit folgt das gekoppelte Problem

$$\begin{aligned} \sigma j\omega \mathbf{A} + \text{curl} \mu^{-1} \text{curl} \mathbf{A} + \kappa \mathbf{A} - \frac{N_c}{A_c} \mathbf{w}(\mathbf{x}) i_c &= 0 \quad \text{für } x \in \Omega. \\ j\omega \Psi + R_c i_c &= u_c. \end{aligned} \quad (3)$$

1.1 Rotationssymmetrische Gleichung

Für rotationssymmetrische Geometrien können in der Regel die Windungen im Mesh aufgelöst werden (vgl. Abbildung 1). Wir betrachten daher das Problem (1) in Zylinderkoordinaten (z, r, ϕ) . In diesem Fall gilt

$$\mathbf{A} = \begin{pmatrix} A_z \\ A_r \\ A_\phi \end{pmatrix}$$

wobei $A_z = A_r = 0$. Das Vektorpotential ist daher gegeben durch die skalare Komponente A_ϕ . Damit folgt für (1)

$$\sigma j \omega A_\phi - \left(\partial_r \left(\frac{\mu^{-1}}{r} \partial_r (r A_\phi) \right) + \partial_z \left(\mu^{-1} \partial_z A_\phi \right) \right) = j_\phi. \quad (4)$$

Für die magnetische Flussdichte gilt in dem Fall

$$\begin{aligned} B_r &= -\frac{\partial A_\phi}{\partial z} \\ B_z &= \frac{1}{r} \frac{\partial (r A_\phi)}{\partial r}. \end{aligned} \quad (5)$$

Das elektrische Feld ist orthogonal zum Magnetfeld und zeigt in ϕ Richtung. Damit folgt

$$E_z = -j \omega A_\phi. \quad (6)$$

Die natürliche Symmetriebedingung auf der Rotationsachse ist gegeben durch

$$A_\phi = 0.$$

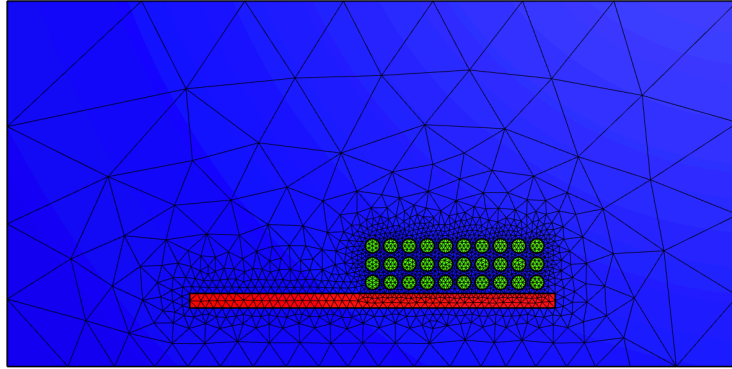


Abbildung 1: Geometrie

Im rotationssymmetrischen Modell, mit den einzelnen Windungen modelliert, folgt für die Stromdichte j_ϕ

$$j_\phi = \frac{i_c}{r^2 \pi}.$$

Wir setzen $r = 0.001\text{m}$ und $i_c = 10\text{A}$. Die magnetische Permeabilität μ ist gegeben durch $\mu = \mu_r \cdot \mu_0$. Die relative Permeabilität ist ein Mass für die Feldverstärkung

im Material (vgl. Tabelle 1). Vom Gebiet abhängige Konstanten können mit Hilfe von `CoefficientFunction` definiert werden (vgl. Listing 7).

	Luft	Windungen	Kern
Material	Luft	Kupfer	Kupfer
el. Leitfähigkeit σ	0 S/m	$56 \cdot 10^6$ S/m	$56 \cdot 10^6$ S/m
relative mag. Permeabilität μ_r	1	1	1
Wärmeleitfähigkeit λ	0.0262 W/(m K)	400 W/(m K)	400 W/(m K)

Tabelle 1: Materialparameter.

Frequenz f	50 Hz
magnetische Feldkonstante μ_0	$4 \pi 10^{-7}$

Tabelle 2: Konstanten

1.2 Induktive Wärmequelle

Die induktive Wärmequelle ist gegeben durch die mittleren Wärmeverluste des elektromagnetischen Feldes

$$q = \frac{1}{2} \cdot |\mathbf{J} \cdot \mathbf{E}| = \frac{\sigma}{2} \cdot |\mathbf{E}|^2, \quad (7)$$

wobei mit $\mathbf{J} = \mathbf{J}_{\text{int}} + \mathbf{J}_{\text{ext}}$ die totale Stromdichte bezeichnet sei.

1.3 Thermisches Feld

Die parabolische Wärmeleitungsgleichung ist gegeben durch

$$C_p \rho \dot{T} - \text{div}(\lambda \nabla T) + \nabla T \cdot \mathbf{v} = q, \quad (8)$$

wobei in der Form auch der konvektive Wärmetransport gegeben durch ein Strömungsfeld \mathbf{v} berücksichtigt wird. Im Folgenden wird dieser Term für das erste vernachlässigt.

Die stationäre Lösung der Wärmeleitungsgleichung (8) genügt der elliptischen Differentialgleichung

$$-\text{div}(\lambda \nabla T) = q. \quad (9)$$

Randbedingungen:

1. konstante Temperatur am Rand (Dirichlet Randbedingung)

$$T(x) = 0 \quad x \in \partial\Omega$$

2. perfekte Temperatur Isolation am Rand (Neumann Randbedingung)

$$\nabla T \cdot \mathbf{n} = 0 \quad x \in \partial\Omega$$

1.4 Input zur Numerik

Für die skalare Komponente des Vektorpotentials benötigen wir für die harmonische Lösung ein komplexwertiger H^1 -FE Raum. Entsprechend setzen wir das Flag `complex=True` (vgl. Listing 1).

```
1 V = H1(mesh, order=3, complex = True, dirichlet='rotsym')
2 u, v = V.TnT()
3 gfu = GridFunction(V)
```

Listing 1: Definition FE-Raum für Wirbelstromproblem

2 Aufgaben

Aufgabe 1

1. Wie lautet die schwache Gleichung für die partielle Differentialgleichung (4)? Verifizieren Sie diese am Listing 2.

```
1 r = y
2 z = x
3
4 uz, ur = grad(u) # partielle Ableitung nach z, r
5 vz, vr = grad(v)
6
7 a = BilinearForm(V)
8 a += (nu*(1/r*u+ur)*vr + nu*uz*vz)*dx
9 a += 1j*omega*sigma*u*v*
10      dx(definedon=mesh.Materials('copper|core'))
11
12 Jimp = CoefficientFunction([0, ic/(r_wire**2*pi), 0])
13
14 f = LinearForm(V)
15 f += Jimp*v*dx(definedon=mesh.Materials('copper'))
16
17 a.Assemble()
18 f.Assemble()
```

Listing 2: Definition Wirbelstrom Problem

2. Berechnen Sie die Lösung und stellen Sie diese graphisch dar. Nutze Sie für die Berechnung die Formel (5).
3. Berechnen Sie die induktive Wärmequelle q mit Hilfe von Formel (7).

```
1 Ez = -1j*omega*gfu
2 Jz = sigma*Ez
3 Jtot = Jz + Jimp
4 Qe = 1/2*Norm(InnerProduct(Jtot, Conj(Ez)))
```

Listing 3: Berechnung der induktiven Wärmequelle

Aufgabe 2

Im nächsten Schritt berechnen wir die stationäre Wärmeverteilung mit Hilfe der Gleichung (9).

1. Wie lautet die schwache Gleichung für (9)?
2. Nutzen Sie einen weiteren FE-Raum (nicht komplexwertig.) für die Berechnung der induzierten Wärmeverteilung.

```
1 V2 = H1(mesh, order = 4, dirichlet = 'outer')
2 uT, vT = V2.TnT()
3 gfT = GridFunction(V2)
```

Listing 4: FE-Raum für die Temperaturverteilung

3. Verifizieren Sie die schwache Gleichung aus 1. mit der Definition der Bilinear- und Linearform:

```
1 aT = BilinearForm(V2)
2 aT += lam*grad(uT)*grad(vT)*dx
3
4 fT = LinearForm(V2)
5 fT += Qe*vT*dx
6
7 aT.Assemble()
8 fT.Assemble()
```

Listing 5: Bilinearform Temperaturfeld

4. Berechnen Sie die Temperaturverteilung und visualisieren Sie diese.

3 Bemerkungen

Die Geometrie kann mit Hilfe der `ngsolve.geom2d.SplineGeometry` Klasse erstellt werden (vgl. Listing 6). Beachten Sie, dass insbesondere für die Löcher gekrümmte Elemente für eine präzisere Geometrie Approximation sehr praktisch sind. **Netgen** bietet frei wählbare Polynom Ordnung für die Elemente.

```
1 from netgen.geom2d import SplineGeometry
2 l_air = 0.1
3 r_wire = 0.001
4 nx_wire = 10
5 ny_wire = 3
6 dxdy_wire = 0.5*r_wire
7
8 rA_core = 0.01
9 rI_core = 0.008
10 l_core = 0.08
11 epsZero = 1e-8
12
```

```

13 geo = SplineGeometry()
14 geo.AddRectangle(p1=(-l_air/2,epsZero),
15                 p2=(l_air/2,l_air/2),
16                 bcs=["rotsym","outer","outer","outer"],
17                 leftdomain=1,
18                 rightdomain=0)
19 pts = []
20 for i in range(nx_wire):
21     for j in range(ny_wire):
22         pts.append([
23             i*(2*r_wire+dx_dy_wire),
24             rA_core+dx_dy_wire+r_wire+j*(2*r_wire+dx_dy_wire)
25         ])
26         geo.AddCircle(
27             c=(i*(2*r_wire+dx_dy_wire),
28               rA_core+dx_dy_wire+r_wire+j*(2*r_wire+dx_dy_wire)),
29             r=r_wire, bc="inner",
30             leftdomain=2,
31             rightdomain=1)
32 pts = np.array(pts)
33 geo.AddRectangle(p1=(-l_core/2,rI_core),
34                 p2=(l_core/2,rA_core),
35                 bc="inner",
36                 leftdomain=3,
37                 rightdomain=1)
38 geo.SetMaterial (1, "air")
39 geo.SetMaterial (2, "copper")
40 geo.SetMaterial (3, "core")
41 geo.SetDomainMaxH(3,(rA_core-rI_core)/2)
42
43 mesh = Mesh(geo.GenerateMesh(maxh=0.0025))

```

Listing 6: Geometrie Beispiel

```

1 omega = 2*pi*50
2 mu0 = 4*pi*1e-7
3
4 sigmaCu = 56e6
5 sigmaCore = 56e6 #Kupfer
6 sigma = CoefficientFunction([0, sigmaCu, sigmaCore])
7 mur = CoefficientFunction([1,1,1])
8 nu = CoefficientFunction(1/mu0*1/mur)

```

Listing 7: Definition der vom Gebiet abhängigen Materialparameter.