

Ausarbeitung Übung 10

Studienarbeit von Dominik Schiller, Constanze Kramer, Simon Arnold & Tobias Lingenberg
Datum: 10. Februar 2021

Darmstadt



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Ausarbeitung Übung 10

Studienarbeit von Dominik Schiller, Constanze Kramer, Simon Arnold & Tobias Lingenberg

Datum: 10. Februar 2021

Darmstadt



Inhaltsverzeichnis

1	Ausarbeitung der Aufgaben	2
1.1	Impliziter Euler, Octave	2
1.2	Nichtlineare Permeabilität	4
1.3	Anhang	6

1 Ausarbeitung der Aufgaben

1.1 Impliziter Euler, Octave

Betrachtet wird ein Transformator, bestehend aus zwei Spulen die über einen Eisenkern miteinander verbunden sind. Der anregende Strom $\mathbf{i}(t) = 10[\sin(2\pi ft), 0]^T$ A hat eine Frequenz von $f = 50$ Hz. Das semidiskrete magnetoquasistatische Problem

$$\mathbf{M}_\sigma \dot{\mathbf{a}} + \tilde{\mathbf{C}} \mathbf{M}_\nu \mathbf{C} \mathbf{a} = \mathbf{X} \mathbf{i}(t) \quad (1.1)$$

soll im Zeitbereich $T = [0, 0.02]$ s gelöst werden. Die Ortsdiskretisierung wird hierbei als schon durchgeführt vorausgesetzt, deshalb wird im Folgenden nicht darauf eingegangen.

Die Octave-Routine `trafo_linear` (siehe Anhang 1.3) erstellt zu Beginn die Materialmatrix \mathbf{M}_σ und entfernt alle überflüssigen Einträge. Mit dem Aufruf der Funktion `fit_operator` (siehe Anhang 1.3) wird die Rotationsmatrix \mathbf{C} konstruiert. Mit Hilfe dieser Matrix kann nun die Rotations-Rotations-Matrix $\mathbf{K} = \tilde{\mathbf{C}} \mathbf{M}_\nu \mathbf{C}$ berechnet werden, wobei \mathbf{M}_ν die Reluktivitätsmatrix in Diagonalform darstellt. In diesem Fall wird noch ein linearer Verlauf der Reluktivität angenommen, das heißt die Reluktivität ist an jedem Punkt im Eisenkern zu jeder Zeit gleich.

Die Gleichung (1.1) wird anschließend mit dem impliziten Eulerverfahren für jeden Zeitschritt im Zeitbereich gelöst.

Im Postprocessing wird ein Linienplot der Anregung erstellt (siehe Abbildung 1.1). Außerdem wird für jeden Zeitschritt eine VTK-Datei erstellt zur weiteren Visualisierung des entstehenden Magnetfelds in Paraview.

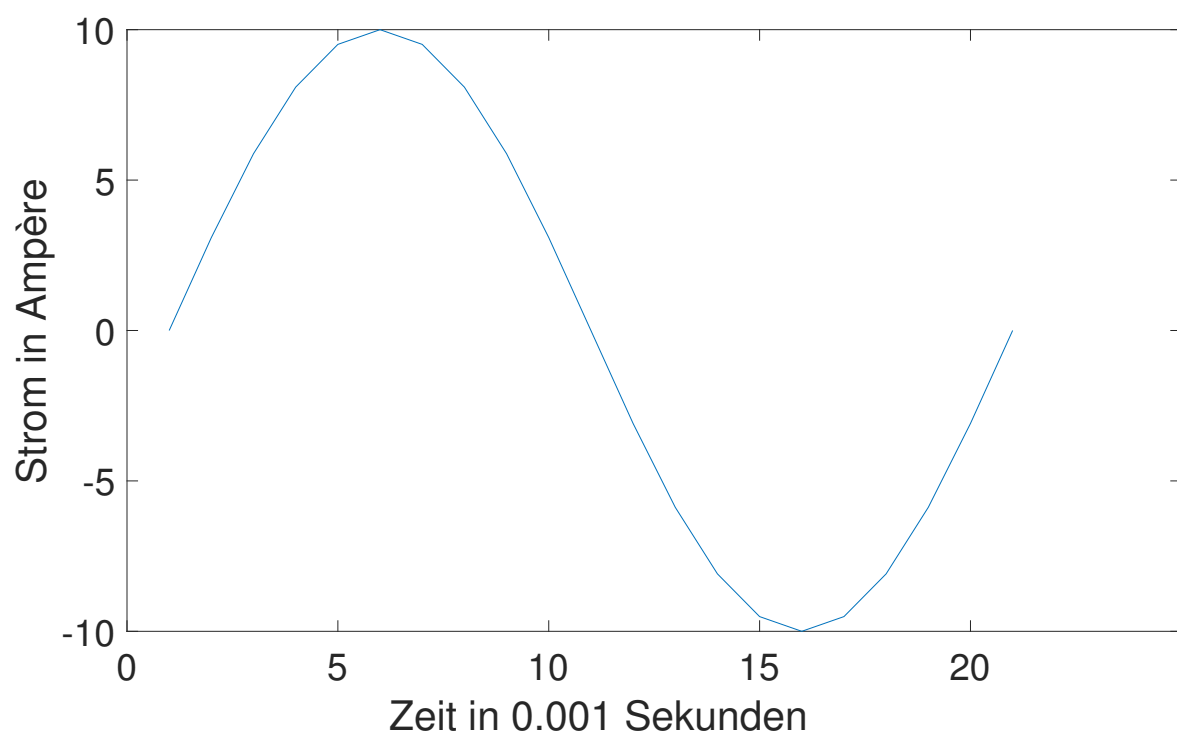


Abbildung 1.1: Anregungsstrom $i(t) = 10 \sin(2\pi ft)$

1.2 Nichtlineare Permeabilität

Im vorherigen Abschnitt wurde beschrieben, wie man die magnetische Flussdichte innerhalb eines Transformators mit linearem Material simulieren kann. Nun soll im folgenden der gleiche Prozess durchgeführt werden, dieses mal mit einer nicht linearen Kennlinie der Permeabilität. Die Permeabilität hängt hierbei von der magnetischen Flussdichte B ab, da diese zu Beginn der Berechnung nicht bekannt ist muss ein iteratives Verfahren verwendet werden. Die Kennlinie wird durch

$$\nu(B) = k_1^{k_2|B|^2} + k_3 \text{ mit } |B| = \sqrt{B_x^2 + B_y^2 + B_z^2} \quad (1.2)$$

beschrieben, wobei $k_1 = 0,3374$, $k_2 = 2,970$ und $k_3 = 388.33$ gilt.

Um das entstehende Gleichungssystem der Form $\mathbf{A}(\mathbf{x})\mathbf{x} = \mathbf{b}$ zu lösen wird die Fixpunkt-Iteration verwendet. Man nimmt den Ansatz

$$\mathbf{A}(\mathbf{x}^{(i)})\mathbf{x}^{(i+1)} = \mathbf{b}, \quad (1.3)$$

wobei die $\mathbf{x}^{(i)}$ die Lösung im Schritt i und $\mathbf{x}^{(i+1)}$ die Lösung im Schritt $i + 1$ beschreibt. Die neue Lösung wird unter Zuhilfenahme der alten Lösung berechnet. Dieses Verfahren ist in dem Skript `trafo_nichtlinear`, welches im Anhang zu finden ist, mit Hilfe einer `while`-Schleife implementiert. Die Iteration bricht ab, sobald

$$\frac{\|\mathbf{x}^{(i+1)} - \mathbf{x}^{(i)}\|}{\|\mathbf{x}^{(i+1)}\|} < 10^{-6} \quad (1.4)$$

erfüllt ist. Zunächst wird $\mathbf{x} = \mathbf{0}$ gewählt, um sicherzustellen, die erste Bedingung der Schleife $i < 2$ garantiert, dass mindestens ein „altes“ und ein „neues“ berechnet werden. \mathbf{A} ist in dem zu untersuchenden magnetostatischen Problem durch $\mathbf{K} = \mathbf{C}'\mathbf{M}_\nu\mathbf{C}$ gegeben. Die linke Seite \mathbf{b} des Gleichungssystems ist der Anregungsstromvektor \mathbf{j} . Die Unbekannte \mathbf{x} stellt das zu berechnende magnetische Vektorpotenzial \mathbf{a} dar. Zusätzlich dazu sind Randbedingungen festgelegt.

Bei sehr kleinem Anregungsstrom lässt sich kein magnetisches Feld berechnen, je größer der Anregungsstrom wird, desto mehr Iterationsschritte werden benötigt und dementsprechend dauert die Berechnung länger. Exemplarisch ist das magnetische Feld innerhalb eines Transformators mit einem Anregungsstrom der Spulen von 10 A in der Abbildung 1.2 zu sehen.

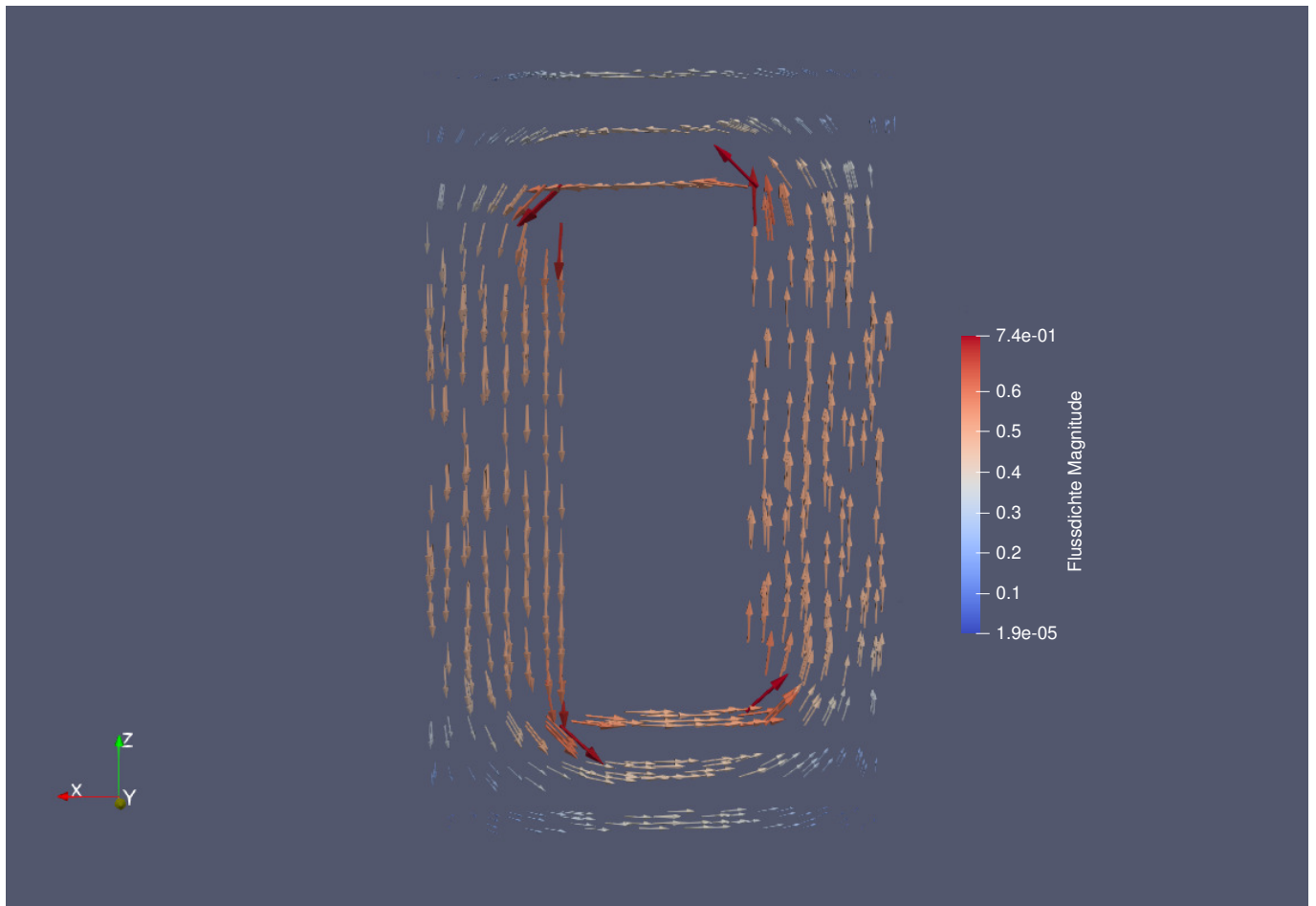


Abbildung 1.2: Das im Transformator entstehende Magnetfeld mit einem Anregungsstrom von 10 A

1.3 Anhang

```
1 function [C,S,Ss] = fit_operator(Nx,Ny,Nz)
2 Mx = 1;
3 My = Nx;
4 Mz = Ny * Nx;
5 Np = Nx * Ny * Nz;
6
7 Px = sparse(Np,Np);
8 Py = sparse(Np,Np);
9 Pz = sparse(Np,Np);
10
11 for p = 1 : Np
12     for q = 1 : Np
13         if (q==p)
14             Px(p,q) = -1;
15             Py(p,q) = -1;
16             Pz(p,q) = -1;
17         end
18         if (q==p+Mx)
19             Px(p,q) = 1;
20         end
21         if (q==p+My)
22             Py(p,q) = 1;
23         end
24         if (q==p+Mz)
25             Pz(p,q) = 1;
26         end
27     end
28 end
29
30 C = [sparse(Np,Np) -Pz Py; Pz sparse(Np,Np) -Px; -Py Px sparse(Np,Np)];
31 C = sparse(C);
32 S = [Px Py Pz];
33 S = sparse(S);
34 Ss = [-Px' -Py' -Pz'];
35 Ss = sparse(Ss);
```

data/fit_operator.m

```
1 % get data
2 load trafo_linear.mat;
3
4 % get mesh data
5 nx = length(prb.xmesh);
6 ny = length(prb.ymesh);
7 nz = length(prb.zmesh);
8 np = nx*ny*nz;
9
10 % get and shrink conductivity matrix
11 Msigma = prb.Msigma;
12 Msigma = Msigma(prb.idxdof,prb.idxdof);
13 M = Msigma;
14
15 % —— do something here ——
16 % construct curl-matrix
17 C = fit_operator(nx,ny,nz);
```



```

18 % —— do something here ——
19
20 % create and shrink curl-curl matrix
21 K = C'*spdiags(prb.Nu,0,3*np,3*np)*C;
22 K = K(prb.idx dof,prb.idx dof);
23
24 % get and shrink excitation matrix
25 X = prb.Qstr;
26 X = X(prb.idx dof,:);
27
28 % time parameters
29 T0 = 0.00;
30 Tend = 0.02;
31 dt = 1e-3; % try dt=1e-4 if your computer is fast....
32
33 % discrete time vector
34 T = T0:dt:Tend;
35
36 % initialize solution
37 f = 50;
38 a = zeros(size(M,1),length(T));
39 i = @(t) (10*[sin(2*pi*f*t);0*t]);
40
41 % construct preconditioner for pcg
42 % use "pcg(A,b,1e-6,1000,Z,Z)" to solve
43 % the linear problem Ax=b for x
44 Z = sparse(1:size(a,1),1:size(a,1),sqrt(diag(M/dt+K)));
45
46 % time loop with equidistant time steps
47 for jj=2:length(T)
48
49     % —— do something here ——
50     % solve M*a'+K*a=X*i with the implicit Euler method
51     h = (1/0.001);
52     A = h*M+K;
53     b = X*i(T(jj)) + h*M*a(:,jj-1);
54     %a = a(:,jj);
55     a(:,jj) = pcg(A,b,1e-6,1000,Z,Z);
56     %a(:,jj)=a(:,jj);
57     % —— do something here ——
58
59 end
60
61
62 % Postprocessing 1: plot the current excitation i(t)
63 % —— do something here ——
64 plot(i(T(:)));
65 % —— do something here ——
66
67 % Postprocessing 2: return the fluxes B!
68 A = zeros(length(prb.ds),1);
69 B = zeros(length(prb.ds),1);
70 for jj=1:length(T)
71     A(prb.idx dof) = a(:,jj);
72
73     % —— do something here ——
74     % compute the facet integrated fluxes b
75     B = C*A;

```

```

76 % —— do something here ——
77
78 % scaling and write to file
79 A(prb.idxs) = A(prb.idxs)./prb.ds(prb.idxs);
80 B(prb.idxA) = B(prb.idxA)./prb.dA(prb.idxA);
81 fit_write_vtk (prb.xmesh,prb.ymesh,prb.zmesh,['solution_' num2str(jj , '%03d') '.vtr' ],...
82               {'MVP',A; 'Flux',B},{ 'Region',prb.Elem(prb.idxV)});
83 end
84 % visualize the result in Paraview, e.g. onprb a cutplane

```

data/trafo_linear.m

```

1 % lese Eingabedaten
2 load trafo_nichtlinear.mat;
3
4 % Gittertopologie
5 xmesh = prb.xmesh;
6 ymesh = prb.ymesh;
7 zmesh = prb.zmesh;
8 nx = length(xmesh);
9 ny = length(ymesh);
10 nz = length(zmesh);
11 Np = nx*ny*nz;
12 material = prb.Elem;
13
14 % Gitterabmessungen und Indizes
15 ddA = prb.ddA; % duale Flaecheninhalte
16 dA = prb.dA; % primale Flaecheninhalte
17 ds = prb.ds; % primale Kantenlaengen
18 idxV = prb.idxV; % Indizes der primalen Volumina im Rechengebiet
19 idxA = prb.idxA; % Indizes der primalen Flaechen im Rechengebiet
20 idxs = prb.idxs; % Indizes der primalen Kanten im Rechengebiet
21 idxdof = prb.idxdof; % Indizes der Freiheitsgrade
22 idxiron = find(material==6); % Indizes fuer Eisen
23
24 % Operatoren
25 C = prb.C; % Curl-Matrix
26
27 % Anregungsvektor mit 10A Strom pro Spule
28 jsrc = prb.Qstr*[10;10];
29 jdof = jsrc(idxdof);
30
31 % Reluktivitaet
32 mu0 = 4*pi*10^-7;
33 murE = 1000;
34 nu = ones(Np,1)/mu0;
35
36 % Linear
37
38 % Loese lineares Problem
39
40 nu(idxiron) = 1/(mu0*murE);
41 Mnu = createMny(prb.xmesh',prb.ymesh',prb.zmesh',nu);
42 K = C'*Mnu*C;
43 Kdof = K(idxdof(:),idxdof(:));
44 Kdof = sparse(Kdof);
45 Zdof = sparse(1:size(Kdof,1),1:size(Kdof,2),sqrt(diag(Kdof)));

```

```

47 a = zeros(3*Np,1);
48 a(idxdof) = pcg(Kdof,jdof,1e-7,1000,Zdof,Zdof);
49
50 % berechne magnetische Flusse im Volumen
51 b = C*a;
52 B = zeros(3*Np,1);
53 B(idxA) = b(idxA)./dA(idxA);
54 Bc = fit_pf2pc(xmesh,ymesh,zmesh,B,idxV);
55
56 % Ausgabe
57 fit_write_vtk(xmesh,ymesh,zmesh,['trafo_iter' num2str(0) '.vtr'],{},{ 'Flussdichte',Bc(idxV,:);
    'Material',material(idxV)})
58
59 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
60 % Nichtlinear
61 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
62
63 % Parameter fuer Brauer's Kurve
64 k1 = 0.3774;
65 k2 = 2.970;
66 k3 = 388.33;
67
68 % Startwerte fuer die Fixpunktiteration
69 aold = zeros(3*Np,1);
70 a = zeros(3*Np,1);
71 i = 0;
72
73 % Fixpunkt iteration
74 while (i<2) || norm(a-aold)/norm(a) > 1e-6
75
76 % update der Variablen
77 aold=a;
78 i=i+1;
79 fprintf('Iteration %d\n',i);
80
81 % berechne magnetische Flusse im Volumen
82 b = C*a;
83 B = zeros(3*Np,1);
84 B(idxA) = b(idxA)./dA(idxA);
85 Bc = fit_pf2pc(xmesh,ymesh,zmesh,B,idxV);
86
87 % Auswertung der Materialkurve
88 Biron = sqrt(sum( Bc(idxiron,:).^2, 2 ));
89 nu(idxiron) = fit_calc_nu(Biron,k1,k2,k3);
90
91 % Update der Materialmatrix
92 Mnu = createMny(xmesh',ymesh',zmesh',nu);
93
94 % create and shrink curl-curl matrix
95 K = C'*Mnu*C;
96 Kdof = K(idxdof(:),idxdof(:));
97
98 % loese GLS
99 Zdof = sparse(1:size(Kdof,1),1:size(Kdof,2),sqrt(diag(Kdof)));
100 a(idxdof) = pcg(Kdof,jdof,1e-7,1000,Zdof,Zdof,a(idxdof));
101
102 % Ausgabe
103 fit_write_vtk(xmesh,ymesh,zmesh,['trafo_ite2rrrr' num2str(i) '.vtr'],{},{ 'Flussdichte',Bc(

```

```
104         idxV,:); 'Material', material(idxV))  
105     end
```

data/trafo_nichtlinear.m



Abbildungsverzeichnis

1.1	Anregungsstrom $i(t) = 10 \sin(2\pi ft)$	3
1.2	Das im Transformator entstehende Magnetfeld mit einem Anregungsstrom von 10 A	5